https://hjfy.top/arxiv/2510.21890

扩散模型的基本原理

From Origins to Advances

Chieh-Hsin Lai
Sony AI

Yang Song
OpenAI

Dongjun Kim
Stanford University

Yuki Mitsufuji
Sony Corporation, Sony AI

Stefano Ermon
Stanford University

目录

Acknowledgements 3

A 深度生成建模导论 13

1 深度生成建模 14

1.1 什么是深度生成建模？ 15

1.2 著名的深度生成模型 21

1.3 模型的分类 24

1.4 闭幕词 25

B 扩散模型的起源与基础 26

2 变分视角：从变分自编码器到扩散模型 28

2.1 变分自编码器 29

2.2 变分视角：DDPM 39

2.3 闭幕词 51

3 基于得分的视角：从能量基模型到 NCSN 52

3.1 基于能量的模型 53

3.2 From Energy-Based to Score-Based Generative Models 60

3.3 Denoising Score Matching 64

3.4 Multi-Noise Levels of Denoising Score Matching (NCSN) 76
3.5 摘要：NCSN 与 DDPM 的比较视角 81
3.6 闭幕词 82

4 扩散模型的现状：得分 SDE 框架 83
4.1 得分 SDE：其原理 84
4.2 得分 SDE：其训练与采样 102
4.3 随机微分方程的实例 107
4.4 (Optional) Rethinking Forward Kernels in Score-Based and Variational Diffusion Models 112
4.5 （可选）通过边缘化与贝叶斯法则的福克-普朗克方程与反向时间随机微分方程 118
4.6 闭幕词 123

5 基于流的视角：从归一化流到流匹配 124
5.1 基于流的模型：归一化流与神经微分方程 126
5.2 Flow Matching Framework 133
5.3 构建分布之间的概率路径与速度 145
5.4 (Optional) Properties of the Canonical Affine Flow 156
5.5 闭幕词 162

6 扩散模型的统一与系统性视角 163
6.1 条件技巧：扩散模型的秘诀 165
6.2 阐明扩散模型训练损失的路线图 167
6.3 扩散模型中的等价性 176
6.4 其下之源：福克-普朗克方程 188
6.5 闭幕词 192

7 （可选）扩散模型与最优传输 193
7.1 分布到分布翻译导言 194
7.2 问题设置的分类 196
7.3 变体最优传输公式的关联 208
7.4 扩散模型的 SDE 是否为 SB 问题的最优解？ 214
7.5 扩散模型的 ODE 是否为最优传输问题的最优映射？ 218

C 扩散模型的采样 226

8 指导与可控生成 228
8.1 序言 . . . 229
8.2 分类器指导 . . . 234
8.3 无分类器指导 . . . 237
8.4 （可选）无需训练的引导 . . . 240
8.5 从强化学习到直接偏好优化的模型对齐 . . . 245
8.6 闭幕词 . . . 255

9 高效的采样求解器 256
9.1 序言 . . . 257
9.2 DDIM . . . 264
9.3 DEIS . . . 276
9.4 DPM-Solver . . . 283
9.5 DPM-Solver++ . . . 296
9.6 PF-ODE 求解器族及其数值类比 . . . 303
9.7 (Optional) DPM-Solver-v3 . . . 306
9.8 (Optional) ParaDiGMs . . . 317
9.9 闭幕词 . . . 323

D 面向快速扩散生成模型的学习 324

10 基于蒸馏的快速采样方法 325
10.1 序言 . . . 326
10.2 Distribution-Based Distillation . . . 331
10.3 Progressive Distillation . . . 336
10.4 闭幕词 . . . 343

11 从零开始学习快速生成器 344
11.1 序言 . . . 345
11.2 特殊流图：离散时间的一致性模型 . . . 350
11.3 特殊流图：连续时间中的一致性模型 . . . 358
11.4 General Flow Map: Consistency Trajectory Model . . . 366
11.5 General Flow Map: Mean Flow . . . 377

11.6 闭幕词 382

Appendices **383**

A 微分方程速成课 **384**

A.1 常微分方程基础 385

A.2 随机微分方程基础 395

B 密度演化：从变量变换到福克-普朗克方程 **399**

B.1 变量变换公式：
从确定性映射到随机流 400

B.2 连续性方程的直观理解 410

**C 扩散模型背后的原理：
伊藤微积分与吉尔萨诺夫定理** **413**

C.1 伊藤公式：随机过程的链式法则 414

C.2 测度的变量变换：扩散模型中的吉尔萨诺夫定理 423

D 补充材料与证明 **427**

D.1 变分视角 427

D.2 基于得分的视角 431

D.3 基于流的视角 444

D.4 理论补充：对扩散模型的统一与系统性视角 448

D.5 理论补充：快速扩散基础生成器的学习 450

D.6 （可选）扩散模型阐明（EDM） 455

扩散模型的基本原理

Chieh-Hsin Lai[1], Yang Song[2], Dongjun Kim[3], Yuki Mitsufuji[4] and Stefano Ermon[5]

[1] *Sony AI; chieh-hsin.lai@sony.com / chiehhsinlai@gmail.com*

[2] *OpenAI*; thusongyang@gmail.com*

[3] *Stanford University; dongjun@stanford.edu*

[4] *Sony Corporation, Sony AI; yuhki.mitsufuji@sony.com*

[5] *Stanford University; ermon@cs.stanford.edu*

ABSTRACT

本专著聚焦于塑造扩散模型发展的基本原理，追溯其起源，并展示不同表述如何源于共同的数学思想。

扩散模型首先通过指定一个前向破坏过程，逐步将数据转化为噪声。该前向过程通过定义一系列中间分布，将数据分布与简单的噪声分布联系起来。扩散模型的核心目标是构建另一个反向过程，该过程将噪声转换为数据，同时恢复前向破坏过程中定义的相同中间分布。

我们描述了三种互补的方式来形式化这一思想。变分视角受变分自编码器启发，将扩散视为逐步学习去除噪声的过程，通过解决一系列小规模的降噪目标，使模型学会将噪声还原为数据。基于得分的视角源于基于能量的模型，学习不断演化的数据分布的梯度，该梯度指示如何调整样本以移向更可能的区域。基于流的视角与归一化流相关，将生成过程视为在学成的速度场作用下，沿着一条平滑路径将样本从噪声移动到数据。

这些视角共享一个共同的基础：一个学成的、随时间变化的速度场，其流动将简单的先验分布转换为数据分布。在此基础上，采样等价于求解一个微分方程，该方程沿连续生成轨迹将噪声演化为数据。基于此基础，本书讨论了用于可控生成的引导方法、用于高效采样的高级数值求解器，以及受扩散启发的流映射模型，这些模型学习在该轨迹上任意时间点之间的直接映射关系。

---

*Affiliation reflects the institution at the time of the work.

本书面向具备基础深度学习背景的读者，旨在帮助他们清晰、概念性且数学严谨地理解扩散模型。书中阐明了理论基础，解释了其多种表述形式背后的推理逻辑，并为该快速发展的领域中的进一步学习与研究提供了稳固的基础。本书既可作为研究人员的系统参考，也可作为学习器的入门读物。

Acknowledgements

作者衷心感谢首尔大学及韩国基础科学研究所的**金道贤教授**在百忙之中抽出宝贵时间参与本研究工作。他仔细审阅了 Chapter 7的部分内容，帮助确保了陈述与证明的正确性，并在多个重要讨论中提供了宝贵的见解，使表述更加清晰。除了技术性建议外，他深思熟虑的反馈以及乐于分享观点的态度，始终激励着本书的撰写过程。我们真诚感谢他的支持与友好合作精神，这些都极大地提升了最终版本的质量。

前言与路线图

扩散模型已迅速成为生成式建模的核心范式，相关研究涵盖了机器学习、计算机视觉、自然语言处理等多个领域。该领域的文献分散在不同社区中，体现了多个方面的进展：包括与建模原则、训练目标、采样器设计及其数学原理相关的理论基础；涵盖工程实践和架构选择的实现进展；将模型适配到特定领域或任务的实际应用；以及提升计算、内存和部署效率的系统级最优化。

本专著旨在为扩散模型提供一个*严谨的基础*，重点关注以下几个核心主题：

- 我们介绍了支撑扩散模型研究的基本概念和公式，使读者能够掌握理解更广泛文献所需的核心知识。我们并未涵盖所有变体或特定领域的应用；相反，我们建立了一个稳定的概念基础，以便于理解这些后续发展。

- 与学习从噪声到数据的直接映射的经典生成式模型不同，扩散模型将生成过程视为随时间逐步变换的过程，逐步将粗略结构细化为精细细节。这一核心思想已通过三种主要视角发展而来，即*变分*、*得分*和*流形*方法，它们提供了互补的方式来理解和实现扩散建模。我们聚焦于这些形式化方法的核心原理和基础，旨在追溯其关键思想的起源，厘清不同形式化之间的关系，并建立一种将直观洞察与严谨数学表述相连接的连贯理解。

- 在这些基础之上，我们探讨如何进一步发展扩散模型，以更高效地生成样本，对生成过程提供更强的控制能力，并激发基于扩散原理的独立生成式建模形式。

本书旨在为具备深度学习基础知识（例如，了解神经网络的含义及训练机制）或更具体地说，具备深度生成建模基础的研究人员、研究生和实践者而撰写，

帮助他们超越对扩散模型的表面认知，深入理解其原理。阅读完本书后，读者将能够系统地掌握扩散建模的基本原理，能够在统一的框架下解释不同形式的模型，并具备应用现有模型的信心以及开展新研究方向所需的背景知识。

本书纲要

本专著系统地介绍了扩散模型的基础，追溯至其核心潜在原理。

推荐阅读路径。 我们建议按照本文献中给出的顺序阅读以建立全面的理解。标记为 可选的章节，对于已经熟悉基础内容的读者可以跳过。例如，对深度生成模型（DGM）已熟悉的读者可跳过 Chapter 1 中的概述部分。同样，若已具备变分自编码器 (Section 2.1)、基于能量的模型 (Section 3.1) 或归一化流 (Section 5.1) 的先验知识，则可跳过这些介绍性章节。其他可选部分提供了对高级或专门主题的深入见解，可根据需要查阅。

该专著分为四个主要部分。

A & B 部分：扩散模型的基础。 本节通过回顾三种奠定基础的视角，追溯了扩散模型的起源。Figure 2 为此部分提供了概览。

部分 A：深度生成建模（DGM）简介。 我们从 Chapter 1开始，回顾深度生成建模的基本目标。从一组数据样本出发，其目标是构建一个模型，能够生成新的样本，这些样本看起来来自同一潜在的、通常未知的数据分布。许多方法通过学习数据的分布来实现这一目标，要么显式地通过概率模型，要么隐式地通过学成的变换。随后，我们解释这些模型如何使用神经网络表示数据分布，如何从样本中学习，以及如何生成新样本。本章最后对主要的生成框架进行了分类，突出它们的核心思想和关键区别。

第二部分：扩散模型的核心观点。 在概述了深度生成建模的一般目标和机制之后，我们现在转向扩散模型，这是一类将生成过程实现为从噪声到数据的逐步变换的方法。我们考察三个相互关联的框架，每个框架都具有一个逐步添加噪声的前向过程，以及一个通过一系列模型执行逐步降噪来近似的时间反向过程：

- **变分视角** (Chapter 2)：起源于变分自编码器（VAEs）(**kingma2013auto**)，将扩散过程视为通过变分目标学习一个降噪过程，从而产生了降噪扩散概率模型（DDPMs）(**sohl2015deep**; **ho2020denoising**)。

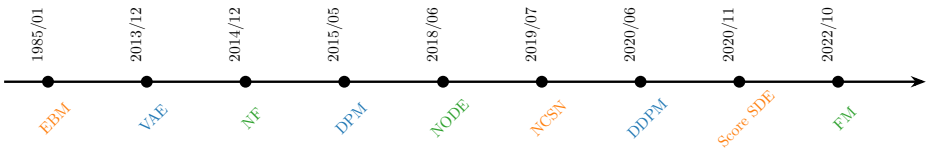


图 1: 扩散模型视角的时间线。每个组使用相同的颜色。
▪ 在 Chapter 2，变分自编码器（VAE）（**kingma2013auto**）→ 扩散概率模型（DPM）（**sohl2015deep**）→ DDPM（**ho2020denoising**）。
▪ 在 Chapters 3 and 4，基于能量的模型（EBM）（**ackley1985learning**）→ 噪声条件得分网络（NCSN）（**song2019generative**）→ 得分 SDE（**song2020score**）。
▪ 在 Chapter 5，归一化流（NF）（**rezende2015variational**）→ 神经 ODE（NODE）（**chen2018neural**）→ 流匹配（FM）（**lipman2022flow**）。

- **基于得分的视角** (Chapter 3)：源于基于能量的模型（EBMs）(**ackley1985learning**) 并发展为噪声条件评分网络(NCSN) (**song2019generative**)。它学习评分函数，即数据对数密度的梯度，该函数指导如何逐步从样本中去除噪声。在连续时间下，Chapter 4 引入了 *评分 SDE 框架*，将这一降噪过程描述为随机微分方程（SDE），其确定性对应形式为常微分方程（ODE）。这一视角将扩散建模与经典的微分方程理论相联系，为分析和算法设计提供了清晰的数学基础。

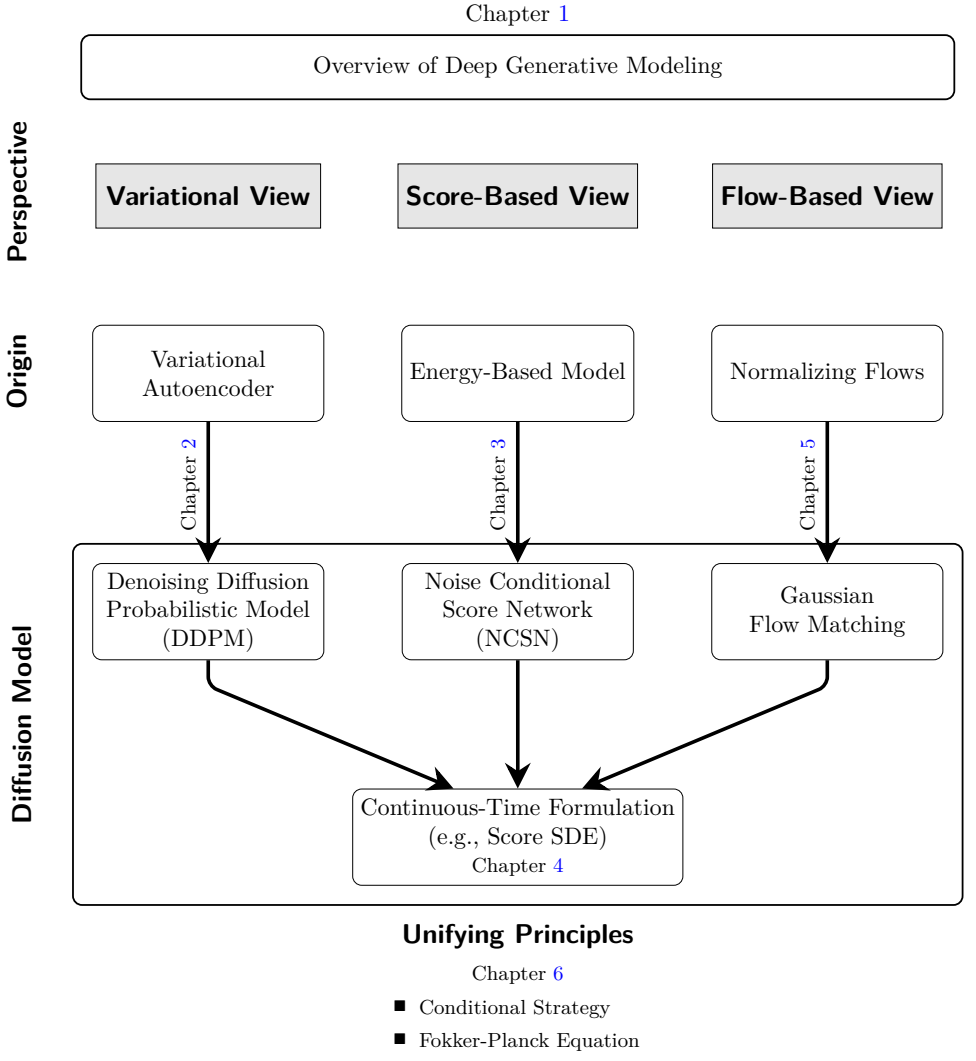
- **基于流的视角** (Chapter 5)：在归一化流 (**rezende2015variational**) 的基础上发展而来，并由流匹配 (**lipman2022flow**) 进一步推广，该视角将生成过程建模为一种连续变换，将样本从简单的先验分布传输至数据分布。这一演化过程由一个速度场通过常微分方程（ODE）控制，明确地定义了概率质量随时间的移动方式。这种基于流的表述自然地扩展了从先验到数据的生成，适用于更一般的*分布到分布的转换*问题，其中目标是学习连接任意一对源分布与目标分布的流。

尽管这些观点起初看起来有所不同，Chapter 6 表明它们之间有着深刻的联系。每种方法都采用一种 *条件化策略*，将学习目标转化为易处理的回归问题。在更深层次上，它们都描述了从先验到数据的概率分布的相同时间演化过程。这一演化由 *福克-普朗克方程*控制，该方程可被视为密度的连续时间变量变换，确保了随机与确定性表述之间的一致性。

由于扩散模型可以被看作是将一个分布传输到另一个分布的方法，Chapter 7 探讨了它们与经典最优传输以及薛定谔桥之间的联系，后者可被解释为带有熵正则化的最优传输。我们回顾了静态和动态两种表述，并解释了它们与连续性方程及福克-普朗克视角的关系。对于关注实际应用的读者而言，本章为可选内容，但对于希望深入研究这些联系的读者，它提供了严格的数学背景和对经典

文献的指引。

图 2: 第二部分：扩散模型的统一与严谨视角。该图直观地将经典的生成式建模方法——变分自编码器、基于能量的模型和归一化流——与其对应的扩散模型表述联系起来。每条垂直路径展示了概念上的传承关系，最终汇聚到连续时间框架。三种视角（变分、得分、流）提供了不同但数学上等价的解释。



第 C 部分和第 D 部分：控制与加速扩散采样。 在统一了基础原理之后，我们现在转向利用扩散模型进行高效生成的实际方面。从扩散模型中采样对应于求

解微分方程，但这一过程通常计算成本较高。C 和 D 部分通过改进采样和学成加速技术，专注于提升生成质量、可控性和效率。

C 部分：从扩散模型中采样。 扩散模型的生成过程展现出一种独特的由粗到细的精细化特征：噪声被逐步去除，生成的样本结构和细节逐渐变得更加连贯。这一特性伴随着权衡。在积极方面，它提供了细粒度的控制能力；通过向学成的、与时间相关的速度场添加引导项，我们可以引导微分方程流以反映用户意图，使采样过程可控制。在消极方面，所需的迭代积分使得采样速度相较于单次生成器较慢。本部分专注于在推理阶段改进生成过程，而无需重新训练。

- **方向生成** (Chapter 8)**:** 命名实体识别（NER）和分类器自由引导等技术使得可以根据用户定义的目标或属性来控制生成过程。在此基础上，我们接下来讨论如何利用偏好数据集进一步使扩散模型与这些偏好对齐。

- **使用数值求解器实现快速生成** (Chapter 9)**:** 通过使用先进的数值求解器，可以显著加速采样过程，这些求解器以更少的步骤近似反向过程，在降低计算成本的同时保持生成质量。

D 部分：学习快速生成式模型。 除了改进现有的采样算法外，我们还研究如何直接学习能够近似扩散过程的快速生成器。

- **基于蒸馏的方法** (Chapter 10)：该方法专注于训练一个学生模型，使其模仿预训练的、运行较慢的扩散模型（教师模型）的行为。与减小教师模型大小不同，其目标是用更少的积分步骤（通常只有几步甚至一步）来复现教师模型的采样轨迹或输出分布。

- **从零开始学习** (Chapter 11)**:** 由于扩散模型中的采样可以看作求解一个常微分方程，该方法直接从零开始学习解映射（即流映射），而无需依赖教师模型。学成的映射可以直接将噪声转换为数据，或更一般地，在解轨迹上实现任意时间到任意时间的跳跃。

附录 为了确保我们的旅程对所有人开放，附录提供了基础概念的背景知识。Chapter A 为扩散模型所采用的微分方程提供了一门速成课程。

扩散模型的核心洞察，尽管其视角和起源各异，都建立在变量变换公式之上。这一基础自然延伸至更深层次的概念，如福克-普朗克方程和连续性方程，它们描述了概率密度在由函数（离散时间）或微分方程（连续时间）定义的映射下

如何变换与演化。Chapter B提供了一个温和的入门，将这些基础概念与更高级的主题联系起来。在 Chapter C中，我们介绍了两种强大但常被忽视的支撑扩散模型的工具：*伊藤公式*和*吉尔萨诺夫定理*，它们为福克-普朗克方程和反向时间采样过程提供了严格的理论支持。最后，Chapter D汇集了主章节中讨论的部分命题与定理的证明。

本专著涵盖的内容及其不涵盖的内容。 我们追求模型的持久性。从自顶向下的视角出发，本书从一个基本原理开始：构建连续时间动态过程，将简单的先验分布传输至数据分布，同时确保每个时刻的边缘分布与由数据到噪声的预设前向过程所诱导的边缘分布相匹配。基于这一原理，我们推导出随机与确定性流，实现采样功能，展示如何控制轨迹（引导机制），并解释如何加速该过程（数值求解器）。随后，我们研究了受扩散启发的快速生成方法，包括蒸馏技术和流映射模型。借助这些工具，读者能够将新论文置于统一框架中，理解方法有效的原因，并设计出更优的模型。

我们不会对扩散模型文献进行全面的综述，也不会列举架构、训练方法、超参数，比较不同方法的实证结果，涵盖数据集和排行榜，描述特定领域或模态的应用，讨论系统级部署，提供大规模训练的指南，或探讨硬件工程。这些主题发展迅速，更适合由专门的综述、开放仓库和实现指南来覆盖。

符号记法

Numbers and Arrays

a	A scalar.
$\mathbf{a}$	A column vector (e.g., $\mathbf{a} \in \mathbb{R}^D$).
$\mathbf{A}$	A matrix (e.g., $\mathbf{A} \in \mathbb{R}^{m \times n}$).
$\mathbf{A}^\top$	Transpose of $\mathbf{A}$.
$\mathrm{Tr}(\mathbf{A})$	Trace of $\mathbf{A}$.
$\mathbf{I}_D$	Identity matrix of size $D \times D$.
$\mathbf{I}$	Identity matrix; dimension implied by context.
$\mathrm{diag}(\mathbf{a})$	Diagonal matrix with diagonal entries given by $\mathbf{a}$.
$\boldsymbol{\phi}$, $\boldsymbol{\theta}$	Learnable neural network parameters.
$\boldsymbol{\phi}^\times$, $\boldsymbol{\theta}^\times$	Parameters after training (fixed during inference).
$\boldsymbol{\phi}^*$, $\boldsymbol{\theta}^*$	Optimal parameters of an optimization problem.

Calculus

$\dfrac{\partial \mathbf{y}}{\partial \mathbf{x}}$	Partial derivatives of $\mathbf{y}$ w.r.t. $\mathbf{x}$ (componentwise).
$\dfrac{\mathrm{d}\mathbf{y}}{\mathrm{d}\mathbf{x}}$ or $\mathrm{D}\mathbf{y}(\mathbf{x})$	Total (Fréchet) derivative of $\mathbf{y}$ w.r.t. $\mathbf{x}$.
$\nabla_{\mathbf{x}} y$	Gradient of scalar $y : \mathbb{R}^D \to \mathbb{R}$; a column in $\mathbb{R}^D$.
$\dfrac{\partial \mathbf{F}}{\partial \mathbf{x}}$ or $\nabla_{\mathbf{x}} \mathbf{F}$	Jacobian of $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^m$; shape $m \times n$.
$\nabla \cdot \mathbf{y}$	Divergence of a vector field $\mathbf{y} : \mathbb{R}^D \to \mathbb{R}^D$; a scalar.
$\nabla_{\mathbf{x}}^2 f(\mathbf{x})$ or $\mathbf{H}(f)(\mathbf{x})$	Hessian of $f : \mathbb{R}^D \to \mathbb{R}$; shape $D \times D$.
$\displaystyle\int f(\mathbf{x})\,\mathrm{d}\mathbf{x}$	Integral of f over the domain of $\mathbf{x}$.

Probability and Information Theory

$p(\mathbf{a})$	Density/distribution over a continuous vector $\mathbf{a}$.
p_{data}	Data distribution.
p_{prior}	Prior distribution (e.g., standard normal).
p_{src}	Source distribution.
p_{tgt}	Target distribution.
$\mathbf{a} \sim p$	Random vector $\mathbf{a}$ is distributed as p.
$\mathbb{E}_{\mathbf{x}\sim p}\big[\mathbf{f}(\mathbf{x})\big]$	Expectation of $\mathbf{f}(\mathbf{x})$ under $p(\mathbf{x})$.
$\mathbb{E}\big[\mathbf{f}(\mathbf{x})\vert\mathbf{z}\big]$, or $\mathbb{E}_{\mathbf{x}\sim p(\cdot\vert\mathbf{z})}\big[\mathbf{f}(\mathbf{x})\big]$	Conditional expectation of $\mathbf{f}(\mathbf{x})$ given $\mathbf{z}$, with $\mathbf{x}$ distributed as $p(\cdot\vert\mathbf{z})$.
$\mathrm{Var}\big(\mathbf{f}(\mathbf{x})\big)$	Variance under $p(\mathbf{x})$.
$\mathrm{Cov}\big(\mathbf{f}(\mathbf{x}), \mathbf{g}(\mathbf{x})\big)$	Covariance under $p(\mathbf{x})$.
$\mathcal{D}_{\mathrm{KL}}\left(p\Vert q\right)$	Kullback–Leibler divergence from q to p.
$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$	Standard normal sample.
$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian over $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

澄清。 我们使用相同的符号表示随机向量及其取值。这种约定在深度学习和生成式建模中很常见，可以使符号更加简洁明了。具体含义由上下文决定。

例如，在类似 $p(\mathbf{x})$ 的表达式中，符号 $\mathbf{x}$ 作为虚变量，该表达式表示为输入

的函数的分布或密度。因此，$p(\mathbf{x})$ 指的是函数形式，而非在特定样本处的取值。当需要在某一点进行求值时，我们会明确指出（例如，“在给定点 $\mathbf{x}$ 处求 p 的值”）。

条件表达式由上下文决定。对于 $p(\mathbf{x}|\mathbf{y})$ ，固定 $\mathbf{y}$ 使其成为 $\mathbf{x}$ 中的密度；固定 $\mathbf{x}$ 使其成为 $\mathbf{y}$ 的函数。

对于条件期望，$\mathbb{E}[\mathbf{f}(\mathbf{x})|\mathbf{z}]$ 表示 $\mathbf{z}$ 的函数，给出在给定 $\mathbf{z}$ 条件下 $\mathbf{f}(\mathbf{x})$ 的期望值。当对特定实现的取值进行条件化时，我们记作 $\mathbb{E}[\mathbf{f}(\mathbf{x})|\mathbf{Z} = \mathbf{z}]$ 。等价地，这可以写成相对于条件分布的积分形式，

$$\mathbb{E}_{\mathbf{x}\sim p(\cdot|\mathbf{z})}[\mathbf{f}(\mathbf{x})] = \int \mathbf{f}(\mathbf{x})\, p(\mathbf{x}|\mathbf{z})\, \mathrm{d}\mathbf{x}.$$

这一区别明确了 $\mathbf{z}$ 是被视为定义函数的变量 $\mathbf{z} \mapsto \mathbb{E}[\mathbf{f}(\mathbf{x})|\mathbf{z}]$ ，还是被视为该函数求值的固定值。

Part A

深度生成建模导论

1

深度生成建模

> 我无法创造的，我就无法理解。
>
> 理查德 · P · 费曼

深度生成模型（DGMs）是神经网络，它们学习高维数据（如图像、文本、音频）上的概率分布，从而能够生成与数据集相似的新样本。我们用 $p_{\boldsymbol{\phi}}$ 表示模型分布，用 p_{data} 表示数据分布。给定一个有限数据集，我们通过最小化衡量 $p_{\boldsymbol{\phi}}$ 与 p_{data} 之间距离的损失来拟合 $\boldsymbol{\phi}$ 。训练完成后，生成过程即运行模型的采样程序以抽取 $\mathbf{x} \sim p_{\boldsymbol{\phi}}$ （密度 $p_{\boldsymbol{\phi}}(\mathbf{x})$ 可能可直接计算，也可能不可直接计算，具体取决于模型类别）。模型质量通过生成样本及其统计量与 p_{data} 的匹配程度进行评估，同时结合特定任务或感知指标。

本章构建了这些思想背后的数学和概念基础。我们在 Section 1.1 中形式化了问题，在 Section 1.2 中介绍了具有代表性的模型类别，并在 Section 1.3 中总结了一个实用的分类法。

1.1 什么是深度生成建模？

DGMs 将大量来自未知且复杂的分布 p_{data} 的真实世界样本（例如，图像、文本）作为输入，并输出一个训练好的神经网络，该网络参数化了一个近似的分布 $p_{\boldsymbol{\phi}}$ 。其目标有两个方面：

1. **真实生成**：生成与真实数据无法区分的新颖、真实样本。
2. **可控生成**：以实现对生成过程的细粒度和可解释性控制。

本节介绍了深度生成模型（DGMs）的基本概念和动机，为深入探索其数学框架和实际应用做好准备。

1.1.1 数学设定

我们假设可以访问一个从潜在的、复杂的数据分布 $p_{\text{data}}(\mathbf{x})$ 中独立同分布（i.i.d.）抽取的有限集样本 [1].

DGM 的目标。 DGM 的主要目标是从有限数据集中学得一个易处理的概率分布。这些数据点被视为从未知且复杂的真分布 $p_{\text{data}}(\mathbf{x})$ 中采样的观测值。由于 $p_{\text{data}}(\mathbf{x})$ 的形式未知，我们无法直接从中抽取新的样本。因此，核心挑战在于构建一个模型，使其能够足够好地近似该分布，从而实现生成新的、真实的样本。

为此，一种 DGM 使用深度神经网络来参数化模型分布 $p_{\boldsymbol{\phi}}(\mathbf{x})$ ，其中 $\boldsymbol{\phi}$ 表示网络的可训练参数。训练目标是找到最优参数 $\boldsymbol{\phi}^*$ ，以最小化模型分布 $p_{\boldsymbol{\phi}}(\mathbf{x})$ 与真实数据分布 $p_{\text{data}}(\mathbf{x})$ 之间的散度。从概念上讲，

$$p_{\boldsymbol{\phi}^*}(\mathbf{x}) \approx p_{\text{data}}(\mathbf{x}).$$

当统计模型 $p_{\boldsymbol{\phi}^*}(\mathbf{x})$ 与数据分布 $p_{\text{data}}(\mathbf{x})$ 接近时，它可以作为生成新样本和评估概率值的代理。该模型 $p_{\boldsymbol{\phi}}(\mathbf{x})$ 通常被称为 *生成式模型*。

DGM 的能力。 一旦获得了数据分布的代理 $p_{\boldsymbol{\phi}}(\mathbf{x})$，我们就可以使用蒙特卡罗采样等采样法从 $p_{\boldsymbol{\phi}}(\mathbf{x})$ 生成任意数量的新数据点。此外，我们可以通过评估 $p_{\boldsymbol{\phi}}(\mathbf{x}')$ 来计算任意给定数据样本 $\mathbf{x}'$ 的概率（或似然）。

[1] 这是机器学习中的一个常见假设。为简便起见，我们使用符号 p 来表示概率分布或其概率密度/概率质量函数，具体取决于上下文。

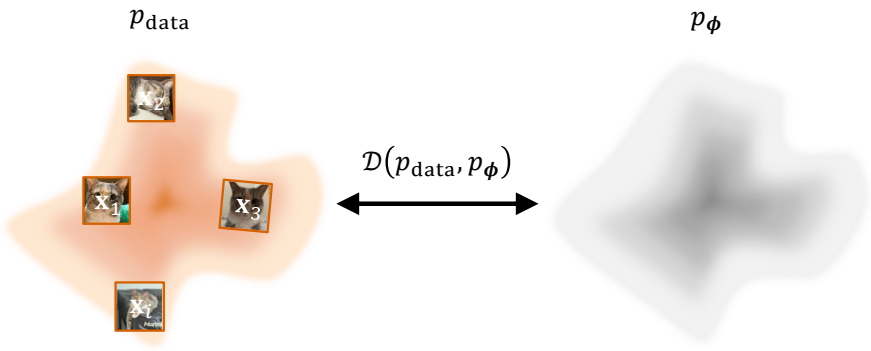


图 1.1: DGM 中目标的示意图。训练 DGM 本质上是尽量减小模型分布 p_{ϕ} 与未知数据分布 p_{data} 之间的差异。由于 p_{data} 无法直接获取，必须利用从其抽取的独立同分布（i.i.d.）样本有限集 $\mathbf{x}_i$ 有效估计这一差异。

DGM 的训练。 我们通过最小化差异 $\mathcal{D}(p_{\text{data}}, p_{\phi})$ 来学习模型族 $\{p_{\phi}\}$ 的参数 ϕ ：

$$\phi^* \in \arg\min_{\phi} \mathcal{D}(p_{\text{data}}, p_{\phi}). \tag{1.1.1}$$

由于 p_{data} 未知，$\mathcal{D}$ 的实际选择必须能够从 p_{data} 的独立同分布样本中进行高效估计。在足够容量的情况下，p_{ϕ^*} 可以很好地逼近 p_{data}。

前向 KL 散度与极大似然估计（MLE）。 一种标准的选择是（前向）Kullback–Leibler 散度[2]

$$\begin{aligned}\mathcal{D}_{\text{KL}}(p_{\text{data}} \| p_{\phi}) &:= \int p_{\text{data}}(\mathbf{x}) \log \frac{p_{\text{data}}(\mathbf{x})}{p_{\phi}(\mathbf{x})} \, \mathrm{d}\mathbf{x} \\ &= \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\big[\log p_{\text{data}}(\mathbf{x}) - \log p_{\phi}(\mathbf{x})\big].\end{aligned}$$

这是不对称的，即

$$\mathcal{D}_{\text{KL}}(p_{\text{data}} \| p_{\phi}) \neq \mathcal{D}_{\text{KL}}(p_{\phi} \| p_{\text{data}}).$$

重要的是，最小化 $\mathcal{D}_{\text{KL}}(p_{\text{data}} \| p_{\phi})$ 会鼓励 *模式覆盖*：如果存在一个正测度的集合 A ，使得 $p_{\text{data}}(A) > 0$ 但 $p_{\phi}(\mathbf{x}) = 0$ 对于 $\mathbf{x} \in A$ ，那么被积函数在 A 上包含 $\log\left(p_{\text{data}}(\mathbf{x})/0\right) = +\infty$ ，因此 $\mathcal{D}_{\text{KL}} = +\infty$ 。因此，最小化前向 KL 会迫使模型在数据具有支撑的地方分配概率。

[2]所有积分均按勒贝格意义定义，在计数测度下均简化为求和。

尽管数据密度 $p_{\text{data}}(\mathbf{x})$ 无法显式评估，前向 KL 散度可以分解为

$$\begin{aligned}\mathcal{D}_{\text{KL}}(p_{\text{data}}\|p_{\boldsymbol{\phi}}) &= \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\left[\log\frac{p_{\text{data}}(\mathbf{x})}{p_{\boldsymbol{\phi}}(\mathbf{x})}\right] \\ &= -\,\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\big[\log p_{\boldsymbol{\phi}}(\mathbf{x})\big] + \mathcal{H}(p_{\text{data}}),\end{aligned}$$

其中 $\mathcal{H}(p_{\text{data}}) := -\,\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\big[\log p_{\text{data}}(\mathbf{x})\big]$ 是数据分布的熵，相对于 $\boldsymbol{\phi}$ 为常数。这一观察结果意味着以下等价关系：

Lemma 1.1.1: Minimizing KL $\Leftrightarrow$ MLE

$$\min_{\boldsymbol{\phi}} \mathcal{D}_{\text{KL}}(p_{\text{data}}\,\|\,p_{\boldsymbol{\phi}}) \iff \max_{\boldsymbol{\phi}} \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\big[\log p_{\boldsymbol{\phi}}(\mathbf{x})\big]. \tag{1.1.2}$$

换言之，最小化前向 KL 散度等价于执行最大似然估计。

在实践中我们用其从独立同分布样本得到的蒙特卡罗估计替换总体期望 $\{\mathbf{x}^{(i)}\}_{i=1}^{N}\sim p_{\text{data}}$ ，从而得到经验极大似然估计目标

$$\hat{\mathcal{L}}_{\text{MLE}}(\boldsymbol{\phi}) := -\frac{1}{N}\sum_{i=1}^{N}\log p_{\boldsymbol{\phi}}(\mathbf{x}^{(i)}),$$

通过小批量随机梯度进行优化；无需对 $p_{\text{data}}(\mathbf{x})$ 进行评估。

Fisher 散度。 Fisher 散度是（基于得分的）扩散建模中的另一个重要概念（参见 Chapter 3）。对于两个分布 p 和 q ，其定义为

$$\mathcal{D}_{\text{F}}(p\|q) := \mathbb{E}_{\mathbf{x}\sim p}\left[\|\nabla_{\mathbf{x}}\log p(\mathbf{x}) - \nabla_{\mathbf{x}}\log q(\mathbf{x})\|_2^2\right]. \tag{1.1.3}$$

它衡量了评分函数 $\nabla_{\mathbf{x}}\log p(\mathbf{x})$ 与 $\nabla_{\mathbf{x}}\log q(\mathbf{x})$ 之间的差异，这两个向量场指向概率更高的区域。简而言之，$\mathcal{D}_{\text{F}}(p\|q)\geq 0$ ，当且仅当 $p=q$ 几乎处处成立时取等号。由于评分函数仅依赖于对数密度的梯度，该度量对归一化常数不变，并构成了分数匹配 (Equations (3.1.3) and (3.2.1)) 的基础：一种通过学习对数密度梯度来实现生成（基于评分的模型）的方法。在此情景中，数据分布 $p=p_{\text{data}}$ 作为目标，而模型 $q=p_{\boldsymbol{\phi}}$ 被训练以使其评分场与数据的评分场对齐。

超越 KL 散度。尽管 KL 散度是衡量概率分布之间差异最广泛使用的度量，但它并非唯一的选择。不同的散度捕捉了不同的几何或统计意义上的偏差概念，这反过来又影响了学习算法的最优化动态。一个广泛的家族是 f -散度 (**csiszar1963informationstheoretische**)：

$$\mathcal{D}_f(p\|q) = \int q(\mathbf{x}) f\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) \mathrm{d}\mathbf{x}, \qquad f(1) = 0, \tag{1.1.4}$$

其中 $f : \mathbb{R}_+ \to \mathbb{R}$ 为凸函数。通过改变 f ，我们可以得到许多著名的散度：

$$\begin{aligned}
f(u) &= u \log u && \Rightarrow \quad \mathcal{D}_f = \mathcal{D}_{\mathrm{KL}}(p\|q) \quad \text{(forward KL)}, \\
f(u) &= \tfrac{1}{2}\left[u \log u - (u+1) \log \tfrac{1+u}{2}\right] && \Rightarrow \quad \mathcal{D}_f = \mathcal{D}_{\mathrm{JS}}(p\|q) \quad \text{(Jensen–Shannon)}, \\
f(u) &= \tfrac{1}{2}|u-1| && \Rightarrow \quad \mathcal{D}_f = \mathcal{D}_{\mathrm{TV}}(p,q) \quad \text{(total variation)}.
\end{aligned}$$

为明确起见，显式形式如下：

$$\mathcal{D}_{\mathrm{JS}}(p\|q) = \tfrac{1}{2}\mathcal{D}_{\mathrm{KL}}\big(p\|\tfrac{1}{2}(p+q)\big) + \tfrac{1}{2}\mathcal{D}_{\mathrm{KL}}\big(q\|\tfrac{1}{2}(p+q)\big),$$

并且

$$\mathcal{D}_{\mathrm{TV}}(p,q) = \tfrac{1}{2}\int_{\mathbb{R}^D} |p-q| \, \mathrm{d}\mathbf{x} = \sup_{A \subset \mathbb{R}^D} |p(A) - q(A)|.$$

直观上，JS 散度提供了一种平滑且对称的度量，能够平衡两个分布，并避免了 KL 散度无界的惩罚（我们稍后会看到，这有助于理解生成对抗网络（GAN）框架），而总变差距离则捕捉了两个分布之间可能的最大概率差异。

另一种观点来自最优传输（见 Chapter 7），其代表性指标是 Wasserstein 距离（见）。它衡量将概率质量从一个分布移动到另一个分布所需的最小成本。与比较密度比的 f 散度不同，Wasserstein 距离依赖于样本空间的几何结构，在 p 和 q 的支持集不重叠时仍具有意义。

每种散度都体现了一种不同的分布间接近程度的概念，因而导致了不同的学习行为。在本书后续内容中，当这些散度在生成式建模的自然情境下出现时，我们将再次讨论它们。

1.1.2 建模分布的挑战

为了建模复合数据分布，我们可以使用参数为 $\boldsymbol{\phi}$ 的神经网络来参数化概率密度函数 p_{data} ，从而构建一个我们记为 $p_{\boldsymbol{\phi}}$ 的模型。为了使 $p_{\boldsymbol{\phi}}$ 成为一个有效

的概率密度函数，它必须满足两个基本性质：

(i) **非负性**： $p_{\boldsymbol{\phi}}(\mathbf{x}) \geq 0$ 对于定义域中的所有 $\mathbf{x}$ 成立。

(ii) **规范化**：在整个领域上的积分必须等于一，即 $\int p_{\boldsymbol{\phi}}(\mathbf{x})\,\mathrm{d}\mathbf{x} = 1$ 。

网络可以自然地为输入 $\mathbf{x}$ 生成一个实标量 $E_{\boldsymbol{\phi}}(\mathbf{x}) \in \mathbb{R}$。为了将此输出解释为有效的密度，必须对其进行变换以满足条件 (i) 和 (ii)。一种实用的替代方法是将 $E_{\boldsymbol{\phi}} \colon \mathbb{R}^D \to \mathbb{R}$ 视为定义了一个*非规范化的密度*，然后显式地施加这些性质。

第一步：确保非负性。 我们可以通过对神经网络 $E_{\boldsymbol{\phi}}(\mathbf{x})$ 的原始输出应用一个正值函数来保证模型的输出始终非负，例如 $|E_{\boldsymbol{\phi}}(\mathbf{x})|$ 、$E^2_{\boldsymbol{\phi}}(\mathbf{x})$ 。一种标准且方便的选择是指数函数。这为我们提供了一个保证为正的非规范化密度 $\tilde{p}_{\boldsymbol{\phi}}(\mathbf{x})$ ：

$$\tilde{p}_{\boldsymbol{\phi}}(\mathbf{x}) = \exp(E_{\boldsymbol{\phi}}(\mathbf{x})).$$

步骤 2：强制规范化。 函数 $\tilde{p}_{\boldsymbol{\phi}}(\mathbf{x})$ 为正，但其积分不等于一。为了构造一个有效的概率密度，我们必须将其除以在整个空间上的积分。这得到了我们模型的最终形式：

$$p_{\boldsymbol{\phi}}(\mathbf{x}) = \frac{\tilde{p}_{\boldsymbol{\phi}}(\mathbf{x})}{\int \tilde{p}_{\boldsymbol{\phi}}(\mathbf{x}')\,\mathrm{d}\mathbf{x}'} = \frac{\exp(E_{\boldsymbol{\phi}}(\mathbf{x}))}{\int \exp(E_{\boldsymbol{\phi}}(\mathbf{x}'))\,\mathrm{d}\mathbf{x}'}.$$

该表达式中的分母被称为 *归一化常数*或 *配分函数*，记为 $Z(\boldsymbol{\phi})$ ：

$$Z(\boldsymbol{\phi}) := \int \exp(E_{\boldsymbol{\phi}}(\mathbf{x}'))\,\mathrm{d}\mathbf{x}'.$$

虽然该过程为 $p_{\boldsymbol{\phi}}(\mathbf{x})$ 提供了一个有效构造，但它引入了一个主要的计算挑战。对于大多数高维问题，计算归一化常数 $Z(\boldsymbol{\phi})$ 所需的积分是难以处理的。这种难以处理性是一个核心问题，推动了多种深度生成模型族的发展。

在接下来的章节中，我们介绍几种典型的 DGM 方法。每种方法均旨在规避或降低评估该归一化常数的计算成本。

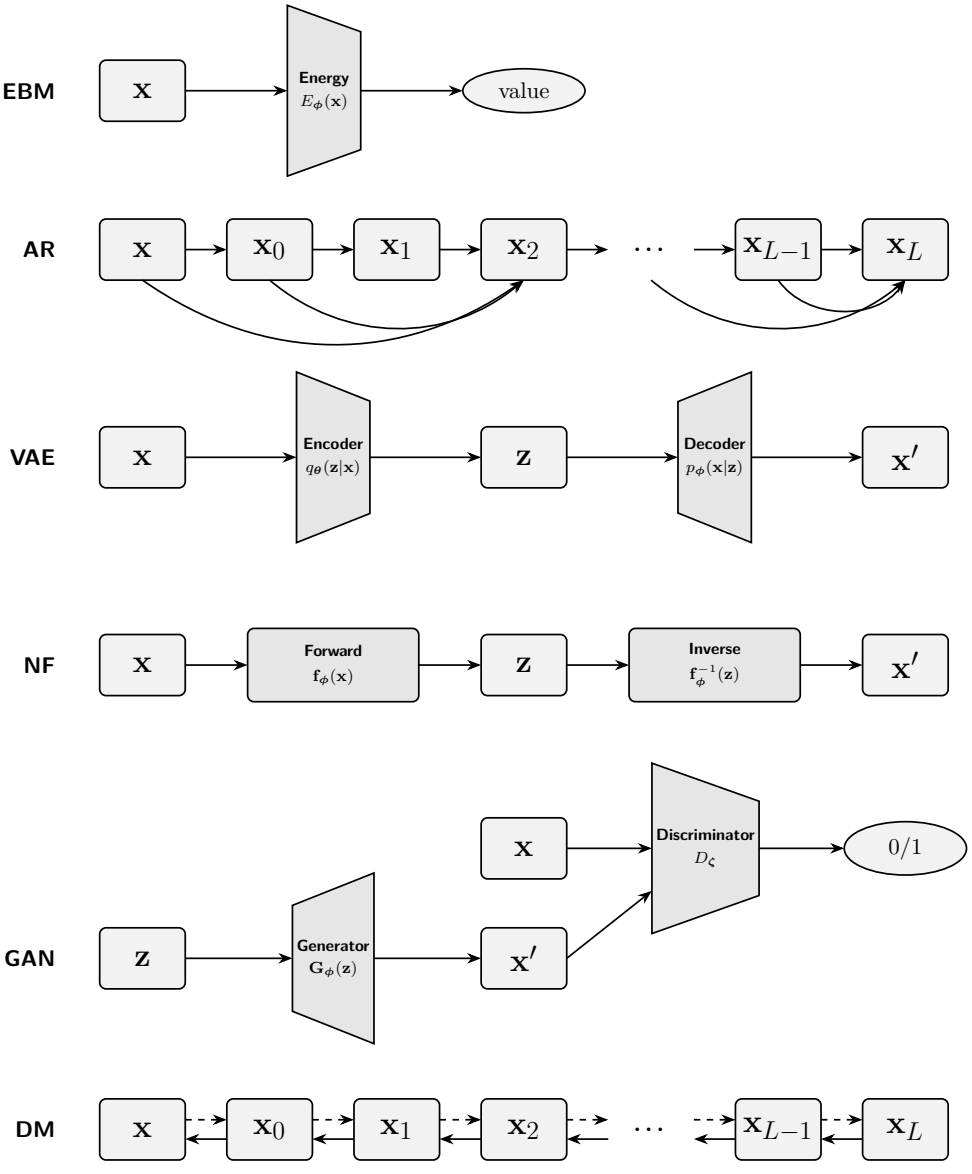


图 1.2: 主流深度生成模型的计算图。从上到下：**EBM** 将输入 $\mathbf{x}$ 映射为一个标量能量；**AR** 以因果依赖关系从左到右生成序列 $\{\mathbf{x}_\ell\}$；**VAE** 将 $\mathbf{x}$ 编码为潜在变量 $\mathbf{z}$ 并解码为重构结果 $\mathbf{x}'$；**NF** 在 $\mathbf{x}$ 和 $\mathbf{z}$ 之间应用可求逆变换 $\mathbf{f}_\phi$，并使用 $\mathbf{f}_\phi^{-1}$ 生成 $\mathbf{x}'$；**GAN** 将噪声 $\mathbf{z}$ 变换为样本 $\mathbf{x}'$，由判别器 D_ζ 对其与真实数据 $\mathbf{x}$ 进行判断；**DM** 通过多步降噪链 $\{\mathbf{x}_\ell\}$ 迭代地优化噪声样本。方框表示变量，梯形表示可学习网络，椭圆表示标量；箭头表示计算流。

1.2 著名的深度生成模型

生成式建模的一个核心挑战是学习能够捕捉高维数据丰富且复杂结构的表达性强的概率模型。多年来，人们提出了多种建模策略，每种策略在可处理性、表达能力和训练效率之间做出了不同的权衡。在本节中，我们探讨了一些最具影响力的策略，这些策略塑造了该领域的发展，并附有它们计算图的对比，如Figure 1.2所示。

基于能量的模型 (EBMs)。 EBMs (**ackley1985learning**; **lecun2006tutorial**) 通过能量函数 $E_{\boldsymbol{\phi}}(\mathbf{x})$ 定义概率分布，该函数为更可能的数据点分配较低的能量。数据点的概率定义为：

$$p_{\boldsymbol{\phi}}(\mathbf{x}) := \frac{1}{Z(\boldsymbol{\phi})} \exp(-E_{\boldsymbol{\phi}}(\mathbf{x})),$$

在哪里

$$Z(\boldsymbol{\phi}) = \int \exp(-E_{\boldsymbol{\phi}}(\mathbf{x}))\, \mathrm{d}\mathbf{x}$$

是配分函数。训练能量基模型（EBMs）通常涉及最大化数据的对数似然。然而，这需要采用技术手段来应对由于配分函数不可计算性带来的计算挑战。在下一章中，我们将探讨扩散模型如何通过从对数密度的梯度生成数据，提供一种替代方案，该方法不依赖于归一化常数，从而避免了配分函数的计算需求。

自回归模型 深度自回归（AR）模型 (**frey1995does**; **larochelle2011neural**; **uria2016neural**) 将联合数据分布 p_{data} 分解为一系列条件概率的乘积，使用概率的 链式法则：

$$p_{\text{data}}(\mathbf{x}) = \prod_{i=1}^{D} p_{\boldsymbol{\phi}}(x_i|\mathbf{x}_{<i}),$$

其中 $\mathbf{x} = (x_1, \ldots, x_D)$ 和 $\mathbf{x}_{<i} = (x_1, \ldots, x_{i-1})$。

每个条件 $p_{\boldsymbol{\phi}}(x_i|\mathbf{x}_{<i})$ 均由神经网络参数化，例如 Transformer，从而能够灵活建模复杂的依赖关系。由于每一项在设计上都经过规范化（例如，离散变量使用 Softmax，连续变量使用参数化的高斯分布），全局规范化变得非常简单。

训练过程通过最大化确切似然，或等价地最小化负对数似然来实现。

尽管自回归模型在密度估计和确切似然方面表现优异，但其顺序特性限制了采样速度，并可能因固定顺序而影响灵活性。然而，它们仍然是基于似然的生成式模型的基础类别，也是现代研究中的关键方法。

变分自编码器（VAEs） VAEs (**kingma2013auto**) 通过引入潜变量 $\mathbf{z}$ 来扩展经典自编码器，这些潜变量捕捉数据中的隐藏结构 $\mathbf{x}$ 。与直接学习 $\mathbf{x}$ 与 $\mathbf{z}$ 之间的映射不同，VAE 采用概率视角：它们同时学习一个 *编码器* $q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ ，用于近似给定数据时未知的潜变量分布，以及一个 *解码器* $p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{z})$ ，用于从这些潜变量中重建数据。为了使训练可行，VAE 最大化一个易处理的对数似然替代量，称为证据下界（Evidence Lower Bound，ELBO）：

$$\mathcal{L}_{\text{ELBO}}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})} \left[\log p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{z})\right] - \mathcal{D}_{\text{KL}} \left(q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) \,\|\, p_{\text{prior}}(\mathbf{z})\right).$$

此处，第一项鼓励数据的准确重构，而第二项通过将潜变量保持在接近简单先验分布 $p_{\text{prior}}(\mathbf{z})$（通常为高斯分布）来对潜变量进行正则化。

变分自编码器（VAEs）为将神经网络与潜变量模型相结合提供了一种合理的方法，至今仍是使用最广泛的基于似然的方法之一。然而，它们也面临一些实际挑战，例如生成样本的清晰度有限以及训练过程中的病态问题（例如编码器倾向于忽略潜变量）。尽管存在这些局限性，VAEs 为后续的发展奠定了重要基础，包括扩散模型。

归一化流。 经典的基于流的模型，如归一化流（NFs）(**rezende2015variational**) 和神经微分方程（NODEs）（**chen2018neural**），旨在通过可逆算子学习一个双射映射 $\mathbf{f}_{\boldsymbol{\phi}}$，该映射将简单的潜在分布 $\mathbf{z}$ 与复杂的数据分布 $\mathbf{x}$ 相关联。这一目标可通过一系列双射变换（在 NFs 中）或通过将变换建模为常微分方程（在 NODEs 中）来实现。这些模型利用 “密度的变量变换公式”，从而支持最大似然估计（MLE）训练：

$$\log p_{\boldsymbol{\phi}}(\mathbf{x}) = \log p(\mathbf{z}) + \log \left| \det \frac{\partial \mathbf{f}_{\boldsymbol{\phi}}^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right|,$$

其中 $\mathbf{f}_{\boldsymbol{\phi}}$ 表示将 $\mathbf{z}$ 映射到 $\mathbf{x}$ 的求逆变换。归一化流（NFs）使用具有易处理雅克比行列式的求逆变换显式建模规范化的密度。通过变量变换公式，规范化常数被解析吸收，使得似然计算确切且易处理。

尽管概念上优雅，经典流模型通常面临实际限制。例如，NFs 通常施加严格的架构约束以确保双射性，而 NODEs 可能因求解微分方程的计算开销而导致训练效率低下。这两种方法在处理高维数据时均存在扩展困难。在后续章节中，我们将探讨扩散模型如何与这些经典流模型相关联并在此基础上进一步发展。

生成对抗网络（GANs）。 GANs (**goodfellow2014generative**) 由两个神经网络组成，分别是生成器 $\mathbf{G}_{\boldsymbol{\phi}}$ 和判别器 $D_{\boldsymbol{\zeta}}$ ，它们相互竞争。生成器旨在从随机噪声 $\mathbf{z} \sim p_{\text{prior}}$ 生成逼真的样本 $\mathbf{G}_{\boldsymbol{\phi}}(\mathbf{z})$ ，而判别器则试图区分真实样本 $\mathbf{x}$ 与生成样本 $\mathbf{G}_{\boldsymbol{\phi}}(\mathbf{z})$ 。GANs 的目标函数可表述为：

$$\min_{\mathbf{G}_{\boldsymbol{\phi}}} \max_{D_{\boldsymbol{\zeta}}} \underbrace{\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D_{\boldsymbol{\zeta}}(\mathbf{x})]}_{\text{real}} + \underbrace{\mathbb{E}_{\mathbf{z}\sim p_{\text{prior}}(\mathbf{z})}\left[\log(1 - D_{\boldsymbol{\zeta}}\left(\mathbf{G}_{\boldsymbol{\phi}}(\mathbf{z})\right))\right]}_{\text{fake}}.$$

GANs 不定义显式的密度函数，因此完全跳过了似然估计。它们不计算规范化常数，而是专注于生成与数据分布密切匹配的样本。

从散度的角度来看，判别器隐式地衡量了真实数据分布 p_{data} 与生成器分布 $p_{\mathbf{G}_{\boldsymbol{\phi}}}$ 之间的差异，其中 $p_{\mathbf{G}_{\boldsymbol{\phi}}}$ 表示由噪声 $\mathbf{z} \sim p_{\text{prior}}$ 生成的样本 $\mathbf{G}_{\boldsymbol{\phi}}(\mathbf{z})$ 所服从的分布。对于固定的生成器 $\mathbf{G}_{\boldsymbol{\phi}}$ ，最优判别器的计算如

$$\frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_{\mathbf{G}_{\boldsymbol{\phi}}}(\mathbf{x})},$$

生成器的最小化简化为

$$\min_{\mathbf{G}_{\boldsymbol{\phi}}} \; 2\,\mathcal{D}_{\text{JS}}\left(p_{\text{data}} \,\|\, p_{\mathbf{G}_{\boldsymbol{\phi}}}\right) - \log 4.$$

此处，$\mathcal{D}_{\text{JS}}$ 表示 Jensen–Shannon 散度，定义为

$$\mathcal{D}_{\text{JS}}(p \,\|\, q) := \tfrac{1}{2}\mathcal{D}_{\text{KL}}\left(p \,\|\, \tfrac{p+q}{2}\right) + \tfrac{1}{2}\mathcal{D}_{\text{KL}}\left(q \,\|\, \tfrac{p+q}{2}\right).$$

这表明 GANs 隐式地最小化 $\mathcal{D}_{\text{JS}}(p_{\text{data}} \,\|\, p_{\mathbf{G}_{\boldsymbol{\phi}}})$。更广泛地说，诸如 f -GANs (**nowozin2016f**) 这类扩展通过证明对抗训练可以最小化一类 f -散度，将这一观点进行了泛化，使得 GANs 与其他生成式模型一样，处于相同的散度最小化框架之中。

尽管生成对抗网络能够生成高质量的数据，但其极小极大训练过程众所周知地不稳定，通常需要精心设计的架构和工程技巧才能达到满意的效果。然而，生成对抗网络后来被重新用作辅助组件，以提升其他生成式模型的性能，尤其是扩散模型。

1.3 模型的分类

正如我们所见，深度生成模型（DGMs）涵盖了广泛的建模策略。一个根本的区别在于这些模型如何对潜在数据分布进行参数化，也就是说，它们是显式地指定 $p_{\boldsymbol{\phi}}(\mathbf{x})$ 还是仅隐式地指定，而与训练目标无关。

- **显式模型**：这些模型通过易处理的或近似易处理的概率密度函数或概率质量函数直接参数化概率分布 $p_{\boldsymbol{\phi}}(\mathbf{x})$。例如，自回归模型（ARs）、归一化流（NFs）、变分自编码器（VAEs）和扩散模型（DMs），它们均能确切或通过易处理的界来定义 $p_{\boldsymbol{\phi}}(\mathbf{x})$。
- **隐式模型**：这些模型仅通过采样过程来指定分布，通常形式为 $\mathbf{x} = \mathbf{G}_{\boldsymbol{\phi}}(\mathbf{z})$，其中 $\mathbf{z} \sim p_{\text{prior}}$ 为某个噪声变量。在这种情况下，$p_{\boldsymbol{\phi}}(\mathbf{x})$ 无法以闭式表达，甚至可能根本未定义。

Table 1.1 中的表格对这些对比方法进行了简明的总结。

表 1.1: 显式生成模型与隐式生成模型的比较

	Explicit		**Implicit**
	Exact Likelihood	**Approx. Likelihood**	
Likelihood	Tractable	Bound/Approx.	Not Directly Modeled/ Intractable
Objective	MLE	ELBO	Adversarial
Examples	NFs, ARs	VAEs, DMs	GANs

与扩散模型的连接。 综上所述，这些经典的生成模型家族展示了建模复杂分布的互补策略。除了各自独立的重要性外，它们还为理解扩散模型提供了指导原则。扩散方法借鉴了其中多个视角的思想：通过变分训练目标与变分自编码器（VAE）相联系，通过得分匹配方法学习对数密度梯度（与能量函数密切相关）而与能量基模型（EBM）相联系，并通过连续时间变换与归一化流（NF）相联系。

为了为后续章节中讨论的扩散方法奠定基础，我们将重点介绍三个核心范式：变分自编码器（VAEs）(Section 2.1)、能量模型（EBMs）(Section 3.1) 和归一化流（NFs）(Section 5.1)。这一探索为现代基于扩散的生成式建模的核心原理提供了基础，这些原理将在接下来的章节中进一步展开。

1.4 闭幕词

本章建立了深度生成建模的基础概念。我们首先定义主要目标：学习一个易处理的模型分布 p_{model}（由 ϕ 参数化），以近似未知的复合数据分布 p_{data}。一个核心挑战是归一化常数，即配分函数 $Z(\phi)$ 的计算不可行性，而该常数是定义有效概率密度所必需的。

为解决这一问题，人们开发了多种深度生成模型家族，每种模型采用不同的策略。我们综述了几种重要的方法，包括基于能量的模型（EBMs）、自回归模型（ARs）、变分自编码器（VAEs）、归一化流（NFs）以及生成对抗网络（GANs）。这些模型可大致分为显式模型和隐式模型：显式模型定义了一个易处理的概率密度，而隐式模型则仅通过采样过程来定义分布。

尽管这些经典框架各自具有重要意义，但其中三个特别构成了本书所聚焦的扩散模型的概念起源：变分自编码器（VAEs）、能量基模型（EBMs）和归一化流（NFs）。在接下来的章节中，我们将追溯扩散模型从这三种基础范式中的演化历程：

1. 第二部分将首先探讨变分视角 (Chapter 2)，展示（变分自编码器的分层潜变量结构）如何自然地导出降噪扩散概率模型（DDPMs）的表述。

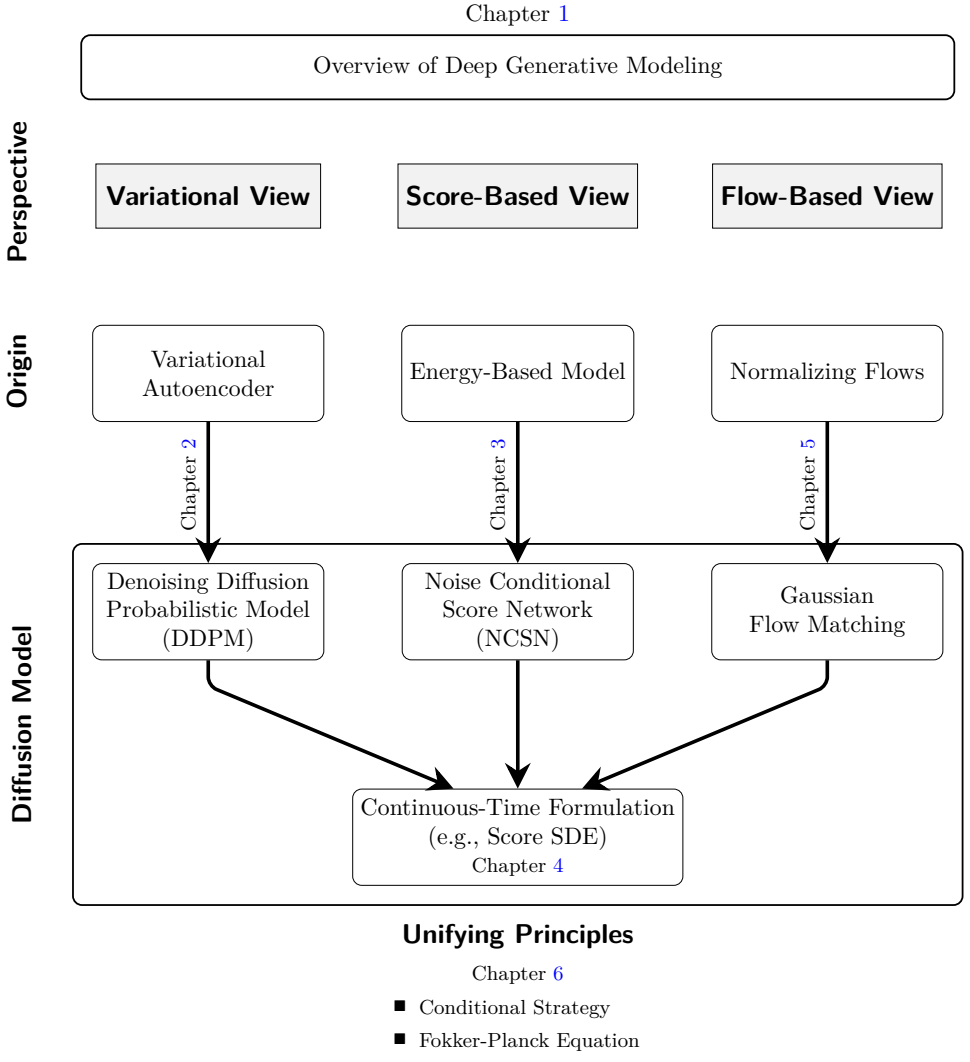
2. 接下来，我们将考察基于分数的视角 (Chapter 3)，它起源于能量基模型（EBMs）和分数匹配，进而发展为噪声条件分数网络（NCSN）以及更通用的分数随机微分方程（Score SDE）框架 (Chapter 4)。

3. 最后，我们将研究基于流的视角 (Chapter 5)，该视角基于归一化流的原理，将生成建模为一种连续变换，并通过流匹配的概念进行泛化。

通过理解这些原点，我们将构建一个连贯的框架，用于解释扩散模型的各种表述，并揭示其背后统一的深层原理。

Part B

扩散模型的起源与基础


Chapter 1
Overview of Deep Generative Modeling
Perspective
Variational View
Score-Based View
Flow-Based View
Origin
Variational Autoencoder
Energy-Based Model
Normalizing Flows
Chapter 2
Chapter 3
Chapter 5
Diffusion Model
Denoising Diffusion Probabilistic Model (DDPM)
Noise Conditional Score Network (NCSN)
Gaussian Flow Matching
Continuous-Time Formulation (e.g., Score SDE)
Chapter 4
Unifying Principles
Chapter 6
Conditional Strategy
Fokker-Planck Equation

2

变分视角：从变分自编码器到扩散模型

在本章中，我们从变分的角度来审视扩散模型。我们首先从变分自编码器（VAE）入手，它通过潜变量表示数据，并通过最大化对数似然函数的易处理下界来进行训练。在这种情景下，学成的编码器将观测值映射到潜变量，而学成的解码器则将潜变量重新映射回观测值，从而完成建模环路。

基于这一模式，分层变体（分层变分自编码器）通过堆叠多个潜层来捕捉多尺度结构。在此框架下，降噪扩散概率模型（DDPM）遵循相同的范式：不联合训练编码器和解码器，而是将编码器固定为前向加噪过程，逐步将数据映射到噪声，而训练则学习一个解码器，通过一系列降噪步骤逆向恢复该路径。从这一视角看，变分自编码器、分层变分自编码器以及扩散模型均通过变分界定义的似然近似进行优化，为本文介绍的方法提供了共同的基础。

2.1 变分自编码器

神经网络如何学习生成真实数据？一个自然的起点是*自编码器*，它由两个网络组成：一个确定性的*编码器*将输入压缩为低维度的潜在代码，以及一个确定性的*解码器*从该代码重构输入。训练过程旨在最小化原始输入与其重构之间的重构误差。尽管这种结构能够实现精确的重构，但潜在空间是无结构的：随机采样潜在代码通常会产生无意义的输出，从而限制了模型在生成任务中的应用。

变分自编码器（*Variational Autoencoder, VAE*）（**kingma2013auto**）通过在潜在空间上施加概率结构来解决这一问题。这使得模型从一个简单的重构工具转变为真正的生成式模型，能够生成新颖且逼真的数据。

2.1.1 概率编码器和解码器

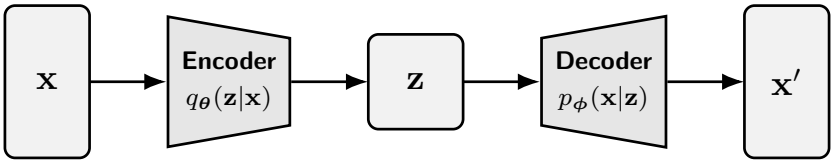


图 2.1: 变分自编码器示意图。它由一个随机编码器 $q_{\theta}(\mathbf{z}|\mathbf{x})$ 组成，该编码器将数据 $\mathbf{x}$ 映射到潜变量 $\mathbf{z}$，以及一个解码器 $p_{\phi}(\mathbf{x}|\mathbf{z})$，用于从潜变量重构数据。

解码器（生成器）的构建。在变分自编码器（VAE）中，我们区分两种类型的变量：观测变量 $\mathbf{x}$，对应于我们所看到的数据（例如，一张图像），以及潜变量 $\mathbf{z}$，用于捕捉隐藏的变差因素（例如，物体形状、颜色或风格）。该模型假设每个观测值 $\mathbf{x}$ 由从一个简单的*先验分布*中采样的潜变量生成，该分布通常为标准高斯分布，$\mathbf{z} \sim p_{\text{prior}} := \mathcal{N}(\mathbf{0}, \mathbf{I})$。

为了将 $\mathbf{z}$ 映射回数据空间，我们定义一个*解码器（生成器）*分布 $p_{\phi}(\mathbf{x}|\mathbf{z})$。在实际应用中，该解码器通常保持简单，常为分解的高斯分布（见 Section 2.1.3）或类似分布，从而使学习过程聚焦于提取有用的潜变量特征，而非记忆数据。直观上，直接逐个生成像素极其困难；相反，潜变量提供了一种紧凑的表示，从该表示中解码出确切的像素排列变得容易得多。新样本通过先采样 $\mathbf{z} \sim p_{\text{prior}}$，再通过 $\mathbf{x} \sim p_{\phi}(\mathbf{x}|\mathbf{z})$ 解码得到。

变分自编码器通过边缘似然函数定义了一种潜在变量生成模型：

$$p_{\phi}(\mathbf{x}) = \int p_{\phi}(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) \, \mathrm{d}\mathbf{z}.$$

理想情况下，解码器参数 $\boldsymbol{\phi}$ 通过最大化此边缘似然来学习，如同极大似然估计（见 Equation (1.1.2)）。然而，由于对 $\mathbf{z}$ 的积分对于表达能力强的非线性解码器来说是难处理的，直接的极大似然估计在计算上不可行，这促使了变分方法在变分自编码器（VAE）中的使用。

编码器（推理网络）的构建。 为了将我们的不可处理生成器与真实数据联系起来，考虑逆向问题：给定一个观测值 $\mathbf{x}$ ，哪些潜在编码 $\mathbf{z}$ 可能产生了它？根据贝叶斯法则，后验分布为

$$p_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \;=\; \frac{p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p_{\boldsymbol{\phi}}(\mathbf{x})}.$$

难点在于分母涉及边缘似然函数 $p_{\boldsymbol{\phi}}(\mathbf{x})$ ，需要对所有潜变量进行积分，在非线性解码器情况下难以处理。因此，从 $\mathbf{x}$ 精确推断 $\mathbf{z}$ 在计算上是不可行的。

VAE 中的“变分”步骤通过用易处理的近似代替难以处理的后验分布来解决这一问题。我们引入一个*编码器*（或推断网络）$q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$，由神经网络参数化，其作用是作为可学习的代理：

$$q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) \;\approx\; p_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}).$$

在实际应用中，编码器将每个观测到的数据点 $\mathbf{x}$ 映射到潜在码的分布，提供了一条可行且可训练的路径，从 $\mathbf{x}$ 回到 $\mathbf{z}$，从而实现学习。

2.1.2 通过证据下界（ELBO）进行训练

我们现在定义一个可计算的训练目标。虽然我们无法直接优化 $\log p_{\boldsymbol{\phi}}(\mathbf{x})$ ，但我们可以最大化其下界——*证据下界 (ELBO)*：

Theorem 2.1.1: 证据下界 (ELBO)

对于任意数据点 $\mathbf{x}$ ，对数似然满足：

$$\log p_{\boldsymbol{\phi}}(\mathbf{x}) \geq \mathcal{L}_{\mathrm{ELBO}}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}),$$

其中证据下界由下式给出：

$$\mathcal{L}_{\mathrm{ELBO}} = \underbrace{\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{z})\right]}_{\text{Reconstruction Term}} - \underbrace{\mathcal{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\right)}_{\text{Latent Regularization}}. \tag{2.1.1}$$

Proof for Theorem.

证据下界源于詹森不等式：

$$\begin{aligned}\log p_{\boldsymbol{\phi}}(\mathbf{x}) &= \log \int p_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z}) \mathrm{d}\mathbf{z} = \log \int q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) \frac{p_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})} \mathrm{d}\mathbf{z} \\ &= \log \mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\left[\frac{p_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right] \geq \mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right].\end{aligned}$$

ELBO 目标自然地分解为两部分：

- **重构**：鼓励从其潜在编码 $\mathbf{z}$ 准确恢复 $\mathbf{x}$。在假设编码器和解码器为高斯分布的情况下，该项恰好简化为自编码器中熟悉的重构损失（参见 Section 2.1.3）。然而，如同自编码器一样，仅优化该项可能导致对训练数据的记忆，从而促使引入额外的正则化。

- **潜在 KL**：鼓励编码器分布 $q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ 保持接近简单的高斯先验 $p_{\mathrm{prior}}(\mathbf{z})$ 。这种正则化将潜在空间塑造成平滑且连续的结构，从而确保从先验中采样的样本能够被可靠地解码，实现有意义的生成。

这种权衡确保了忠实的重构和一致的采样。

信息论视角：ELBO 作为散度的上界。 ELBO 目标具有自然的信息论解释。回想一下，极大似然训练等价于最小化 KL 散度。

$$\mathcal{D}_{\mathrm{KL}}(p_{\mathrm{data}}(\mathbf{x})\|p_{\boldsymbol{\phi}}(\mathbf{x})),$$

这衡量了模型分布对数据分布的逼近程度。由于该术语通常难以处理，变分框架引入了联合比较。

具体而言，考虑两个联合分布：

- **生成联合**，$p_{\boldsymbol{\phi}}(\mathbf{x},\mathbf{z}) = p(\mathbf{z})p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{z})$ ，用于描述模型如何生成数据；
- **推理联合**，$q_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}) = p_{\text{data}}(\mathbf{x})q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ ，它将真实数据与其推断的潜在变量耦合。

比较这些分布可得不等式

$$\mathcal{D}_{\text{KL}}(p_{\text{data}}(\mathbf{x})\|p_{\boldsymbol{\phi}}(\mathbf{x})) \leq \mathcal{D}_{\text{KL}}(q_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})\|p_{\boldsymbol{\phi}}(\mathbf{x},\mathbf{z})), \tag{2.1.2}$$

有时称为 KL 散度的链式法则。直观上，仅比较边缘分布 ($\mathbf{x}$) 可能会隐藏在考虑完整的潜在变量-数据联合分布时才会显现的不匹配。

正式地，可以将联合 KL 散度展开为

$$\begin{aligned}
&\underbrace{\mathcal{D}_{\text{KL}}(q_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})\|p_{\boldsymbol{\phi}}(\mathbf{x},\mathbf{z}))}_{\text{Total Error Bound}} \\
=&\mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}\left[\log\frac{p_{\text{data}}(\mathbf{x})q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\phi}}(\mathbf{x})p_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\right] \\
=&\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\log\frac{p_{\text{data}}(\mathbf{x})}{p_{\boldsymbol{\phi}}(\mathbf{x})} + \mathcal{D}_{\text{KL}}\left(q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\|p_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\right)\right] \\
=&\underbrace{\mathcal{D}_{\text{KL}}(p_{\text{data}}\|p_{\boldsymbol{\phi}})}_{\text{True Modeling Error}} + \underbrace{\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\mathcal{D}_{\text{KL}}(q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\|p_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}))\right]}_{\text{Inference Error}},
\end{aligned}$$

其中第一项是真实的建模误差，第二项是推断误差，即近似后验与真实后验之间的差距。后者始终非负，这解释了 Equation (2.1.2)。

最后，请注意

$$\log p_{\boldsymbol{\phi}}(\mathbf{x}) - \mathcal{L}_{\text{ELBO}}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}) = \mathcal{D}_{\text{KL}}\left(q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\|p_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\right).$$

因此，推断误差恰好是对数似然与变分下界 (ELBO) 之间的差距。最大化 ELBO 因此直接减少了推断误差，确保训练过程最小化了整体界限中有意义的部分。

2.1.3 高斯变分自编码器

变分自编码器的标准形式在编码器和解码器中均采用高斯分布。

编码器部件 编码器 $q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ 通常被建模为高斯分布，形式如下：

$$q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) := \mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}^2_{\boldsymbol{\theta}}(\mathbf{x}))\right),$$

其中 $\boldsymbol{\mu}_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}^d$ 和 $\boldsymbol{\sigma}_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}^d_+$ 是编码器网络的确定性输出。

解码器部分 解码器通常被建模为方差固定的高斯分布：

$$p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{z}) := \mathcal{N}\big(\mathbf{x}; \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{z}), \sigma^2\mathbf{I}\big),$$

其中 $\boldsymbol{\mu}_{\boldsymbol{\phi}} : \mathbb{R}^d \to \mathbb{R}^D$ 为神经网络，$\sigma > 0$ 为控制方差的小常数。

在该假设下，ELBO 中的重构项简化为

$$\mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{z})\right] = -\frac{1}{2\sigma^2}\mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\left[\|\mathbf{x} - \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{z})\|^2\right] + C,$$

其中 C 是与 $\boldsymbol{\theta}$ 和 $\boldsymbol{\phi}$ 无关的常数。因此 ELBO 目标简化为：

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\left[\frac{1}{2\sigma^2}\|\mathbf{x} - \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{z})\|^2\right] + \mathcal{D}_{\mathrm{KL}}\big(q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\|p_{\mathrm{prior}}(\mathbf{z})\big),$$

由于高斯假设，KL 项具有闭式解。因此，训练变分自编码器（VAE）可简化为最小化正则化的重构损失。

2.1.4 标准变分自编码器的缺点

尽管变分自编码器框架在理论上具有吸引力，但它存在一个关键缺陷：生成的输出往往模糊不清。

VAE 中的模糊生成。 为了理解这一现象，考虑一个固定的高斯编码器 $q_{\mathrm{enc}}(\mathbf{z}|\mathbf{x})$，以及一种形式如下的解码器

$$p_{\mathrm{dec}}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}(\mathbf{z}), \sigma^2\mathbf{I}),$$

其中 $\boldsymbol{\mu}(\mathbf{z})$ 表示解码器网络。对于任意编码器，优化 ELBO 等价于（忽略一个加性常数）最小化期望重构误差：

$$\arg\min_{\boldsymbol{\mu}} \mathbb{E}_{p_{\text{data}}(\mathbf{x})q_{\text{enc}}(\mathbf{z}|\mathbf{x})}\left[\|\mathbf{x}-\boldsymbol{\mu}(\mathbf{z})\|^2\right].$$

这是一个标准最小二乘问题 in $\boldsymbol{\mu}(\mathbf{z})$，其解以闭式解由条件均值给出：

$$\boldsymbol{\mu}^*(\mathbf{z}) = \mathbb{E}_{q_{\text{enc}}(\mathbf{x}|\mathbf{z})}[\mathbf{x}],$$

其中 $q_{\text{enc}}(\mathbf{x}|\mathbf{z})$ 是由编码器诱导的在潜在变量给定输入下的后验分布，通过贝叶斯法则定义：

$$q_{\text{enc}}(\mathbf{x}|\mathbf{z}) = \frac{q_{\text{enc}}(\mathbf{z}|\mathbf{x})p_{\text{data}}(\mathbf{x})}{p_{\text{prior}}(\mathbf{z})}.$$

通过贝叶斯规则得到的最优生成器的等价形式为：

$$\boldsymbol{\mu}^*(\mathbf{z}) = \frac{\mathbb{E}_{p_{\text{data}}(\mathbf{x})}[q_{\text{enc}}(\mathbf{z}|\mathbf{x})\cdot\mathbf{x}]}{\mathbb{E}_{p_{\text{data}}(\mathbf{x})}[q_{\text{enc}}(\mathbf{z}|\mathbf{x})]}.$$

现在假设两个不同的输入 $\mathbf{x}\neq\mathbf{x}'$ 被映射到潜在空间中重叠的区域，即 $q_{\text{enc}}(\cdot|\mathbf{x})$ 和 $q_{\text{enc}}(\cdot|\mathbf{x}')$ 的支撑集相交。这意味着 $\boldsymbol{\mu}^*(\mathbf{z})$ 对多个可能无关的输入进行平均，导致输出模糊且不明确。这种在冲突模式上的平均效应是变分自编码器生成样本出现模糊特征的根本原因。

2.1.5 （可选）从标准 VAE 到分层 VAE

为了建模复合数据，分层变分自编码器（HVAEs）(**vahdat2020nvae**) 通过引入潜变量的层次结构来增强变分自编码器（VAEs）。这种深度的分层结构使模型能够捕捉数据在多个抽象层次上的特征，显著提升表达能力，并模拟现实世界数据的组合性质。

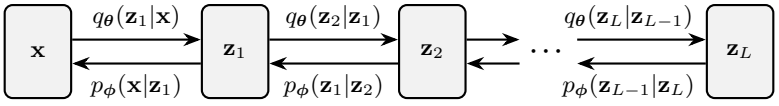


图 2.2: HVAE 的计算图。它具有分层结构，包含多个潜层上堆叠的可训练编码器和解码器。

HVAE 的建模。 与使用单一潜变量 $\mathbf{z}$ 的标准变分自编码器（VAE）不同，分层变分自编码器（HVAEs）引入了多个按自顶向下层级排列的潜变量层。每一潜层均条件化其下方的层，形成一系列条件先验链，从而捕捉逐步细化的抽象层次结构。这导致联合分布的如下自顶向下因子分解：

$$p_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z}_{1:L}) = p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{z}_1) \prod_{i=2}^{L} p_{\boldsymbol{\phi}}(\mathbf{z}_{i-1}|\mathbf{z}_i) p(\mathbf{z}_L).$$

该结构定义了边缘数据分布，

$$p_{\text{HVAE}}(\mathbf{x}) := \int p_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z}_{1:L}) \, \mathrm{d}\mathbf{z}_{1:L}.$$

生成过程逐步进行：从顶层潜变量 $\mathbf{z}_L$ 开始，逐个解码至 $\mathbf{z}_1$，随后生成最终观测 $\mathbf{x}$。

在编码部分，HVAEs 采用一种结构化、可学习的变分编码器 $q_{\boldsymbol{\theta}}(\mathbf{z}_{1:L}|\mathbf{x})$，其结构与生成层次相对应。一个常见的选择是自下而上的马尔可夫因子分解：

$$q_{\boldsymbol{\theta}}(\mathbf{z}_{1:L}|\mathbf{x}) = q_{\boldsymbol{\theta}}(\mathbf{z}_1|\mathbf{x}) \prod_{i=2}^{L} q_{\boldsymbol{\theta}}(\mathbf{z}_i|\mathbf{z}_{i-1}).$$

HVAE 的证据下界（ELBO）。 与 Equation (2.1.1) 类似，ELBO 通过 Jensen 不等式推导得出：

$$\begin{aligned}\log p_{\text{HVAE}}(\mathbf{x}) &= \log \int p_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z}_{1:L})\, \mathrm{d}\mathbf{z}_{1:L} \\ &= \log \int \frac{p_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z}_{1:L})}{q_{\boldsymbol{\theta}}(\mathbf{z}_{1:L}|\mathbf{x})} q_{\boldsymbol{\theta}}(\mathbf{z}_{1:L}|\mathbf{x})\, \mathrm{d}\mathbf{z}_{1:L} \\ &= \log \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{z}_{1:L}|\mathbf{x})}\left[\frac{p_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z}_{1:L})}{q_{\boldsymbol{\theta}}(\mathbf{z}_{1:L}|\mathbf{x})}\right] \\ &\geq \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{z}_{1:L}|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z}_{1:L})}{q_{\boldsymbol{\theta}}(\mathbf{z}_{1:L}|\mathbf{x})}\right] \\ &=: \mathcal{L}_{\text{ELBO}}(\boldsymbol{\phi}).\end{aligned} \tag{2.1.3}$$

代入分解的表达式得到：

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{z}_{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{z}_L)\prod_{i=2}^{L} p_{\boldsymbol{\phi}}(\mathbf{z}_{i-1}|\mathbf{z}_i) p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{z}_1)}{q_{\boldsymbol{\theta}}(\mathbf{z}_1|\mathbf{x})\prod_{i=2}^{L} q_{\boldsymbol{\theta}}(\mathbf{z}_i|\mathbf{z}_{i-1})}\right].$$

这种分层的 ELBO 可分解为可解释的项，包括重构项以及每个生成条件与其对应的变分近似之间的 KL 散度。

从浅层网络到深度网络的跨越彻底改变了机器学习，类似的思想也重塑了生成式模型。HVAE 展示了利用深度堆叠层构建数据的强大能力。这种分层层次结构是现代生成式建模的基石，在基于得分的方法 (Section 3.4) 和归一化流 (Section 5.1) 中再次出现。其核心洞察简单而强大：

观察 2.1.1:

堆叠多层使得模型能够逐步生成数据，从粗略细节开始，每一步添加更精细的特征。这一过程大大简化了对高维数据复杂结构的捕捉。

平坦变分自编码器中更深的网络并不足够。 标准平面变分自编码器存在两个基本局限性，仅通过加深编码器和解码器无法解决。

第一个限制是变分族。在标准的变分自编码器中，

$$q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}\big(\mathbf{z};\, \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}), \operatorname{diag}(\sigma_{\theta}^2(\mathbf{x}))\big),$$

因此，对于每个固定的 $\mathbf{x}$，编码器后验是一个对角协方差的单峰值高斯分布。网

络深度的增加能够提高 $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ 与 $\sigma_{\boldsymbol{\theta}}$ 的准确率，但并不会扩展该分布族；即使采用完整协方差，其形式仍为一个单峰值椭球体。当 $p_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ 为多峰值时，此分布族无法与其匹配，这会导致 ELBO 放松并削弱推断效果。解决这一问题需要更丰富的后验分布类，而不仅仅是加深网络。

其次，如果解码器过于强大，模型可能会出现后验崩溃问题。为了理解原因，让我们回顾一下变分自编码器的目标是

$$\begin{aligned}
&\mathbb{E}_{p_{\text{data}}(\mathbf{x})}[\mathcal{L}_{\text{ELBO}}(\mathbf{x})] \\
&\quad = \mathbb{E}_{p_{\text{data}}(\mathbf{x})q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{z})] - \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\big[\mathcal{D}_{\text{KL}}\big(q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\big)\big] \\
&\quad = \mathbb{E}_{p_{\text{data}}(\mathbf{x})q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{z})] - \mathcal{I}_q(\mathbf{x};\mathbf{z}) - \mathcal{D}_{\text{KL}}(q_{\boldsymbol{\theta}}(\mathbf{z})\|p(\mathbf{z})),
\end{aligned}$$

其中 $\mathcal{I}_q(\mathbf{x};\mathbf{z})$ 为由下式定义的互信息

$$\mathcal{I}_q(\mathbf{x};\mathbf{z}) = \mathbb{E}_{q(\mathbf{x},\mathbf{z})}\Big[\log \tfrac{q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})}\Big] = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\big[\mathcal{D}_{\text{KL}}(q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\|q(\mathbf{z}))\big],$$

并且聚合后验为 $q_{\boldsymbol{\theta}}(\mathbf{z}) = \int p_{\text{data}}(\mathbf{x})q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\,\mathrm{d}\mathbf{x}$ 。

如果解码器类在不使用 $\mathbf{z}$ 的情况下能够很好地建模数据（即其包含一些接近 p_{data} 的 $p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{z}) = r(\mathbf{x})$ ），那么 ELBO 的最大化器将设定 $q_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = p(\mathbf{z})$ ，从而导致 $\mathcal{I}_q(\mathbf{x};\mathbf{z}) = 0$ 与 $q_{\boldsymbol{\theta}}(\mathbf{z}) = p(\mathbf{z})$ 。这种“忽略 $\mathbf{z}$ ”的解决方案并不会因网络加深而消失：(1) 学成的编码与 $\mathbf{x}$ 无关（因此不携带对下游任务有用的依赖于数据的结构），且 (2) 在 $\mathbf{z}$ 中进行条件设定或移动对生成样本无影响，导致可控生成失败。

层级发生了什么变化？ HVAE 引入了多个潜在层次，

$$p_{\boldsymbol{\phi}}(\mathbf{x},\mathbf{z}_{1:L}) = p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{z}_1)\prod_{i=2}^{L} p_{\boldsymbol{\phi}}(\mathbf{z}_{i-1}|\mathbf{z}_i)p(\mathbf{z}_L),$$

变分下界（ELBO）

$$\begin{aligned}
\mathcal{L}_{\text{ELBO}}(\mathbf{x}) = \mathbb{E}_q[\log p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{z}_1)] - \mathbb{E}_q\Big[\mathcal{D}_{\text{KL}}(q_{\boldsymbol{\theta}}(\mathbf{z}_1|\mathbf{x})\|p_{\boldsymbol{\phi}}(\mathbf{z}_1|\mathbf{z}_2))\Big] \\
- \sum_{i=2}^{L-1}\mathbb{E}_q\Big[\mathcal{D}_{\text{KL}}(q_{\boldsymbol{\theta}}(\mathbf{z}_i|\mathbf{z}_{i-1})\|p_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{z}_{i+1}))\Big] \\
- \mathbb{E}_q\Big[\mathcal{D}_{\text{KL}}(q_{\boldsymbol{\theta}}(\mathbf{z}_L|\mathbf{z}_{L-1})\|p(\mathbf{z}_L))\Big].
\end{aligned}$$

此处，我们记 $\mathbb{E}_q := \mathbb{E}_{p_{\text{data}}(\mathbf{x})q_{\boldsymbol{\theta}}(\mathbf{z}_{1:L}|\mathbf{x})}$ 。每次推理条件均与其自顶向下的生成对应项对齐：$q_{\boldsymbol{\theta}}(\mathbf{z}_1|\mathbf{x})$ 与 $p_{\boldsymbol{\phi}}(\mathbf{z}_1|\mathbf{z}_2)$ ，中间层与 $p_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{z}_{i+1})$ ，顶层与先验 $p(\mathbf{z}_L)$ 。这使得信息惩罚分布在各层级之间，并通过这些相邻的 KL 项定位学习信号。这些特性源于分层潜在图结构，而非简单地在平坦的 VAE 中加深网络。

未来将如何发展？ 虽然 HVAE 通过引入多个潜层扩展了 VAE 框架以提升表达能力，但其训练过程面临独特的挑战。由于编码器和解码器必须联合优化，学习过程变得不稳定：低层和解码器已经能够重构 $\mathbf{x}$ ，导致高层潜变量接收到的有效信号极少。此外，传递到深层变量的梯度信息通常间接且微弱，使得它们难以产生有意义的贡献。另一个难点在于平衡模型容量，因为过于表达能力强的条件分布可能主导重构任务，从而抑制高层潜变量的效用。

有趣的是，深度分层结构的核心思想在变分扩散模型中得到了更强大的体现，这一主题我们将在 Section 2.2 中探讨。扩散模型继承了 HVAE 的渐进结构，但巧妙地避开了其核心缺陷。通过固定编码过程并仅专注于学习生成的逆过程，它们实现了前所未有的稳定性与建模灵活性，从而显著提升了生成结果的质量。

为便于记号简洁，我们偏离了常见的变分自编码器（VAE）惯例，该惯例使用 q 表示编码器，p 表示生成器。为避免分歧，我们用 p 表示分布，并始终通过适当的下标或上标明确其角色，在上下文中加以说明。

2.2 变分视角：DDPM

降噪扩散概率模型（DDPMs）(**sohl2015deep**; **ho2020denoising**) 是扩散建模的基石。从概念上讲，它们在变分框架内运行，类似于变分自编码器（VAEs）和分层变分自编码器（HVAEs）。然而，DDPMs 引入了一个巧妙的机制，解决了其前辈所面临的部分挑战。

从本质上讲，DDPMs 涉及两个不同的随机过程：

- **前向过程（固定编码器）**：该过程通过转移核 $p(\mathbf{x}_i|\mathbf{x}_{i-1})$ 在多个步骤中逐步向数据注入高斯噪声，使数据逐渐被破坏。数据最终演化为各向同性高斯分布，实际上变为纯噪声。这意味着编码器是固定的，未被学成。

- **反向降噪过程（可学习的解码器）**：在这里，神经网络通过参数化分布 $p_{\phi}(\mathbf{x}_{i-1}|\mathbf{x}_i)$ 学习逆转噪声污染。从纯噪声开始，该过程迭代地进行降噪，以生成逼真的样本。关键在于，每个单独的降噪步骤都比从头开始生成完整样本更为简单，这正是变分自编码器（VAEs）通常所面临的挑战。

通过固定编码器并专注于渐进生成轨迹的学习，DDPMs 实现了显著的稳定性与表达能力。

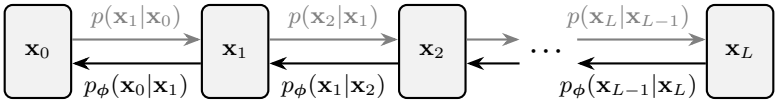


图 2.3: DDPM 的示意图。它包含一个固定的前向过程（灰色），逐步向数据中添加高斯噪声，以及一个学成的反向过程，逐步去噪以生成新的样本。

在本节中，我们专注于 DDPMs，更广泛的讨论将留到 Section 4.4中进行，在那里我们将提出一个更加通用和灵活的框架。

2.2.1 前向过程（固定编码器）

在 DDPM 中，前向过程是一个固定的、不可训练的操作，用作编码器。它通过在多个步骤中逐步添加噪声来破坏原始数据，最终将其转换为一个简单的先验分布 $p_{\text{prior}} := \mathcal{N}(\mathbf{0}, \mathbf{I})$ 。这一变换如 Figure 2.3 中的前向链所示，或如 Figure 2.4 所示的图示。

让我们逐步形式化这一退化过程：

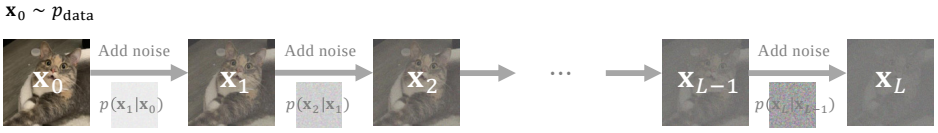


图 2.4: DDPM 前向过程示意图，其中高斯噪声被逐步添加，将数据样本逐渐破坏为纯噪声。

固定高斯转移。 前向过程中的每一步均由一个固定的高斯转移核控制[1]：

$$p(\mathbf{x}_i|\mathbf{x}_{i-1}) := \mathcal{N}(\mathbf{x}_i; \sqrt{1-\beta_i^2}\mathbf{x}_{i-1}, \beta_i^2\mathbf{I}).$$

此处，过程从 $\mathbf{x}_0$ 开始，表示从真实数据分布 p_{data} 中抽取的一个样本。序列 $\{\beta_i\}_{i=1}^L$ 表示一个预先确定的、单调递增的噪声调度，其中每个 $\beta_i \in (0,1)$ 控制在步骤 i 注入的高斯噪声的方差。为方便起见，我们定义 $\alpha_i := \sqrt{1-\beta_i^2}$ 。该数学定义与以下直观的迭代更新精确等价：

$$\mathbf{x}_i = \alpha_i \mathbf{x}_{i-1} + \beta_i \boldsymbol{\epsilon}_i,$$

其中 $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 为独立同分布的。这意味着在每一步 i ，我们将前一状态 $\mathbf{x}_{i-1}$ 按 α_i 进行缩放，并添加一个由 β_i 缩放的受控高斯噪声。

扰动核与先验分布。 通过递归应用转移核，我们得到了在步骤 i 时给定原始数据 $\mathbf{x}_0$ 的噪声样本分布的闭式表达式：

$$p_i(\mathbf{x}_i|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_i; \bar{\alpha}_i \mathbf{x}_0, (1-\bar{\alpha}_i^2)\mathbf{I}\right),$$

在哪里

$$\bar{\alpha}_i := \prod_{k=1}^{i} \sqrt{1-\beta_k^2} = \prod_{k=1}^{i} \alpha_k.$$

[1]这种表述方式虽然看起来可能有所不同，但在数学上等价于原始的 DDPM 转移核。

这意味着我们可以直接从 $\mathbf{x}$ 采样 $\mathbf{x}_i$[2]

$$\mathbf{x}_i = \bar{\alpha}_i \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_i^2}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{2.2.1}$$

设噪声调度 $\{\beta_i\}_{i=1}^L$ 为一个递增序列，则前向过程的边缘分布收敛为

$$p_L(\mathbf{x}_L|\mathbf{x}_0) \longrightarrow \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \text{as } L \to \infty,$$

这促使我们选择先验分布为

$$p_{\text{prior}} := \mathcal{N}(\mathbf{0}, \mathbf{I})$$

且不依赖于数据 $\mathbf{x}_0$ 。

2.2.2 逆向降噪过程（可学习的解码器）

其核心本质在于，DDPMs 能够 *逆转* 由前向扩散过程施加的受控退化。从纯的、无结构的噪声 $\mathbf{x}_L \sim p_{\text{prior}}$ 出发，目标是逐步 *去噪* 这种随机性，一步步地，直到产生一个连贯且有意义的数据样本。这一逆向生成过程通过马尔可夫链进行，如 Figure 2.5 所示。

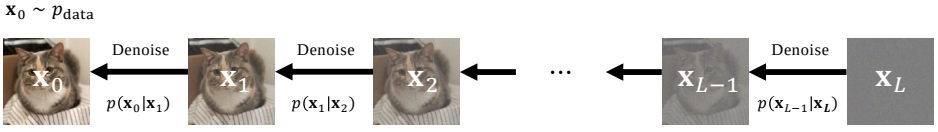


图 2.5: DDPM 反向（降噪）过程示意图。从噪声 $\mathbf{x}_L \sim p_{\text{prior}}$ 开始，模型依次采样 $\mathbf{x}_{i-1} \sim p(\mathbf{x}_{i-1}|\mathbf{x}_i)$ 以进行 $i = L, \ldots, 1$ ，从而生成新的数据 $\mathbf{x}$ 。由于真实转移 $p(\mathbf{x}_{i-1}|\mathbf{x}_i)$ 未知，因此我们旨在对其进行近似。

其根本挑战，以及指导 DDPM 发展的核心问题，随之变为：

[2]对于固定的索引 t ，我们在此及后续内容中常使用高斯扰动形式

$$p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\big(\mathbf{x}_t; \alpha_t \mathbf{x}_0, \sigma_t^2 \mathbf{I}\big),$$

我们等价地将其写作分布意义上的恒等式

$$\mathbf{x}_t \overset{d}{=} \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}, \qquad \text{即} \qquad \text{Law}(\mathbf{x}_t) = \text{Law}(\alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}),$$

其中 $\mathbf{x}_0 \sim p_{\text{data}}$ ，$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 与 $\mathbf{x}_0$ 独立，且 α_t, σ_t 为确定性标量。等式 “$\overset{d}{=}$ ” 表示两个随机变量具有相同的概率密度（即 分布），因此对任意测试函数 ϕ 具有相同的期望：

$$\mathbb{E}[\phi(\mathbf{x}_t)] = \mathbb{E}[\phi(\alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon})].$$

为简洁起见，我们将 $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}$ 记作该等式，其含义可理解为分布相等，或根据上下文理解为一个样本实现；此简写将在全文中持续使用。

Question 2.2.1

我们能否精确计算这些反向转移核 $p(\mathbf{x}_{i-1}|\mathbf{x}_i)$，或者至少有效地近似它们，特别是在考虑 $\mathbf{x}_i \sim p_i(\mathbf{x}_i)$ 的复杂分布时？

与其像原始的 DDPM 论文那样立即深入证据下界（ELBO）的数学复杂推导（对此的详细讨论将在 Section 2.2.5中进行），我们将改从更直观的角度来理解训练目标：通过利用条件概率来实现易处理的公式化。

概述：建模与训练目标。 为了实现生成过程，我们的目标是近似未知的真实反向转移核，$p(\mathbf{x}_{i-1}|\mathbf{x}_i)$。我们通过引入一个可学习的参数化模型，$p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i)$，并训练该模型以最小化期望的 KL 散度：

$$\mathbb{E}_{p_i(\mathbf{x}_i)}\left[\mathcal{D}_{\mathrm{KL}}(p(\mathbf{x}_{i-1}|\mathbf{x}_i)\|p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i))\right]. \tag{2.2.2}$$

然而，直接计算目标分布 $p(\mathbf{x}_{i-1}|\mathbf{x}_i)$ 是具有挑战性的。根据贝叶斯定理，我们需要评估：

$$p(\mathbf{x}_{i-1}|\mathbf{x}_i) = p(\mathbf{x}_i|\mathbf{x}_{i-1}) \underbrace{\frac{p_{i-1}(\mathbf{x}_{i-1})}{p_i(\mathbf{x}_i)}}_{\text{intractable}}.$$

边际分布 $p_i(\mathbf{x}_i)$ 和 $p_{i-1}(\mathbf{x}_{i-1})$ 是在未知数据分布 p_{data} 下的期望，表示为：

$$p_i(\mathbf{x}_i) = \int p_i(\mathbf{x}_i|\mathbf{x}_0)p_{\text{data}}(\mathbf{x}_0)\mathrm{d}\mathbf{x}_0,$$

并且类似地适用于 $p_{i-1}(\mathbf{x}_{i-1})$。由于 p_{data} 未知，这些积分无法进行闭式求解；最多只能从样本中进行近似，因此在实际中无法获得确切的密度。

通过条件化克服顽固性问题。 DDPM 中的一个核心见解解决了这一不可处理性问题：我们以一个干净的数据样本 $\mathbf{x}$ 条件化反向转移。这一微妙而强大的步骤将不可处理的核转化为数学上易处理的形式：

$$p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}) = p(\mathbf{x}_i|\mathbf{x}_{i-1})\frac{p(\mathbf{x}_{i-1}|\mathbf{x})}{p(\mathbf{x}_i|\mathbf{x})}.$$

这种易处理性源于前向过程的两个关键性质：其马尔可夫性质，即 $p(\mathbf{x}_i|\mathbf{x}_{i-1}, \mathbf{x}) = p(\mathbf{x}_i|\mathbf{x}_{i-1})$，以及所有相关分布的高斯性质。因此，$p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x})$ 本身是高斯分布，并具有闭式表达式（我们在 Equation (2.2.4) 中已经看到）。关键在于，这种优美的条件化策略使我们能够推导出一个易处理的目标，该目标在函数上等价

于 Equation (2.2.2) 中看似不可处理的边缘 KL 散度。

Theorem 2.2.1: 边际与条件 KL 最小化之间的等价性

以下等式成立：

$$\begin{aligned}&\mathbb{E}_{p_i(\mathbf{x}_i)}\left[\mathcal{D}_{\mathrm{KL}}\big(p(\mathbf{x}_{i-1}|\mathbf{x}_i)\|p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i)\big)\right]\\&=\mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})}\mathbb{E}_{p(\mathbf{x}_i|\mathbf{x})}\left[\mathcal{D}_{\mathrm{KL}}\big(p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x})\|p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i)\big)\right]+C,\end{aligned}\tag{2.2.3}$$

其中 C 是与 $\boldsymbol{\phi}$ 无关的常数。此外，Equation (2.2.3) 的最小化解满足

$$p^*(\mathbf{x}_{i-1}|\mathbf{x}_i)=\mathbb{E}_{p(\mathbf{x}|\mathbf{x}_i)}\big[p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x})\big]=p(\mathbf{x}_{i-1}|\mathbf{x}_i),\quad \mathbf{x}_i\sim p_i.$$

Proof for Theorem.

> 该证明通过展开定义、应用概率链式法则，并利用对数恒等式将 KL 散度期望重写，将其分解为条件 KL 散度期望与边缘 KL 散度之和。完整推导过程见 Section D.1.1。

这种替代视角：通过条件化以获得易处理的目标，构成了 DDPM 的基础，并揭示了其与其他有影响力的扩散模型之间深刻的共性，我们将在 Chapter 3和 Chapter 5中进一步探讨。

它揭示了一个强大的等价关系：最小化边缘分布之间的 KL 散度在数学上等价于最小化特定条件分布之间的 KL 散度。后一种表述极为有用，因为关键的条件分布 $p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x})$ 具有方便的闭式表达式：

Lemma 2.2.2: Reverse Conditional Transition Kernel

$p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x})$ is Gaussian with the closed-form expression:

$$p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x})=\mathcal{N}\left(\mathbf{x}_{i-1};\boldsymbol{\mu}\left(\mathbf{x}_i,\mathbf{x},i\right),\sigma^2(i)\mathbf{I}\right),$$

where

$$\boldsymbol{\mu}\left(\mathbf{x}_i,\mathbf{x},i\right):=\frac{\bar{\alpha}_{i-1}\beta_i^2}{1-\bar{\alpha}_i^2}\mathbf{x}+\frac{(1-\bar{\alpha}_{i-1}^2)\alpha_i}{1-\bar{\alpha}_i^2}\mathbf{x}_i,\quad \sigma^2(i):=\frac{1-\bar{\alpha}_{i-1}^2}{1-\bar{\alpha}_i^2}\beta_i^2.\tag{2.2.4}$$

在引理 4.4.2 中，我们给出了一个更通用的公式，该公式超越了 Equation (2.2.1) 中描述的 DDPM 加噪过程。

2.2.3 逆转移核的建模 $p_\phi(\mathbf{x}_{i-1}|\mathbf{x}_i)$

利用定理 2.2.1 中的梯度级等价性以及引理 2.2.2 中的反向条件的高斯形式 $p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x})$，DDPM 假设每个反向转移 $p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i)$ 是高斯分布，其参数化为

$$p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i) := \mathcal{N}\big(\mathbf{x}_{i-1};\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_i,i),\sigma^2(i)\mathbf{I}\big), \tag{2.2.5}$$

其中 $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\cdot,i)\colon \mathbb{R}^D \to \mathbb{R}^D$ 为可学习的均值函数，$\sigma^2(i)>0$ 为如 Equation (2.2.4) 所定义的固定函数。

我们记在数据 $\mathbf{x}_0 \sim p_{\text{data}}$ 条件下，对时间步 i 取平均的 KL 散度，以匹配所有层的分布为：

$$\mathcal{L}_{\text{diffusion}}(\mathbf{x}_0;\boldsymbol{\phi}) := \sum_{i=1}^{L} \mathbb{E}_{p(\mathbf{x}_i|\mathbf{x}_0)}\left[\mathcal{D}_{\text{KL}}\big(p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x}_0)\|p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i)\big)\right]. \tag{2.2.6}$$

得益于两个分布的高斯形式以及式 Equation (2.2.5) 中定义的参数化，目标函数具有闭式表达式，可简化为：

$$\mathcal{L}_{\text{diffusion}}(\mathbf{x}_0;\boldsymbol{\phi}) = \sum_{i=1}^{L} \frac{1}{2\sigma^2(i)} \left\|\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_i,i) - \boldsymbol{\mu}(\mathbf{x}_i,\mathbf{x}_0,i)\right\|_2^2 + C, \tag{2.2.7}$$

其中 C 是与 $\boldsymbol{\phi}$ 无关的常数。对数据分布取平均并省略不影響最優化的常數 C ，最終的 DDPM 訓練目標[F]

$$\mathcal{L}_{\text{DDPM}}(\boldsymbol{\phi}) := \sum_{i=1}^{L} \frac{1}{2\sigma^2(i)} \mathbb{E}_{\mathbf{x}_0}\mathbb{E}_{p(\mathbf{x}_i|\mathbf{x}_0)}\left[\left\|\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_i,i) - \boldsymbol{\mu}(\mathbf{x}_i,\mathbf{x}_0,i)\right\|_2^2\right], \tag{2.2.8}$$

其中 $\mathbf{x}_0 \sim p_{\text{data}}$ 。

2.2.4 预测与损失的实际选择

$\boldsymbol{\epsilon}$ -预测。 在典型的 DDPM 实现中，训练并非直接使用基于均值预测参数化的原始损失函数 Equation (2.2.8)。相反，通常采用一种等价的重参数化方法，即$\boldsymbol{\epsilon}$-预测（噪声预测）公式。

回想一下，在 DDPM 的前向过程中，通过噪声水平 i 生成一个带噪声的样本 $\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x})$ 。

$$\mathbf{x}_i = \bar{\alpha}_i\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_i^2}\boldsymbol{\epsilon}, \quad \mathbf{x}_0 \sim p_{\text{data}}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I}). \tag{2.2.9}$$

使用该表达式，反向平均值 $\boldsymbol{\mu}(\mathbf{x}_i, \mathbf{x}_0, i)$ 从 Equation (2.2.4) 可以重写为：

$$\boldsymbol{\mu}(\mathbf{x}_i, \mathbf{x}_0, i) = \frac{1}{\alpha_i}\left(\mathbf{x}_i - \frac{1-\alpha_i^2}{\sqrt{1-\bar{\alpha}_i^2}}\boldsymbol{\epsilon}\right).$$

这促使我们采用神经网络 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_i, i)$ 对模型均值 $\boldsymbol{\mu}_{\boldsymbol{\phi}}$ 进行参数化，该神经网络可直接预测噪声：

$$\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_i, i) = \frac{1}{\alpha_i}\left(\mathbf{x}_i - \frac{1-\alpha_i^2}{\sqrt{1-\bar{\alpha}_i^2}}\underbrace{\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_i, i)}_{\boldsymbol{\epsilon}\text{-prediction}}\right).$$

将其代入原始损失后，得到预测噪声与真实噪声之间的平方 ℓ_2 误差：

$$\|\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_i, i) - \boldsymbol{\mu}(\mathbf{x}_i, \mathbf{x}_0, i)\|_2^2 \propto \|\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_i, i) - \boldsymbol{\epsilon}\|_2^2,$$

i 依赖的权重因子。直观上，该模型充当一个 “噪声侦探”，估计前向过程中每一步添加的随机噪声。从损坏的样本中减去这一估计值，使其更接近干净的原始数据，重复此逐步过程即可从纯噪声中重建数据。

简化损失与 $\boldsymbol{\epsilon}$ -预测。 在实际应用中，通过省略权重项进一步简化了该表达式，得到广泛使用的 DDPM 训练损失：

$$\mathcal{L}_{\text{simple}}(\boldsymbol{\phi}) := \mathbb{E}_i \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\|\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_i, i) - \boldsymbol{\epsilon}\|_2^2\right], \tag{2.2.10}$$

其中 $\mathbf{x}_i = \bar{\alpha}_i \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_i^2}\boldsymbol{\epsilon}$ 与 $\mathbf{x}_0 \sim p_{\text{data}}$ 。由于目标噪声在每个时间步 t 均具有单位方差，因此 ℓ_2 损失在 Equation (2.2.10) 中保持了跨所有 t 的一致尺度。这避免了目标消失或爆炸，并消除了显式损失加权的需要。

重要的是，$\mathcal{L}_{\text{DDPM}}$ 与 $\mathcal{L}_{\text{simple}}$ 具有相同的最优解 $\boldsymbol{\epsilon}^*$，这是因为 Equation (2.2.10) 本质上可简化为一个最小二乘问题（如命题 4.2.1 或命题 6.3.1 所示）：

$$\boldsymbol{\epsilon}^*(\mathbf{x}_i, i) = \mathbb{E}\left[\boldsymbol{\epsilon}|\mathbf{x}_i\right], \quad \mathbf{x}_i \sim p_i.$$

直观上，$\boldsymbol{\epsilon}$-预测网络 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_i, i)$ 估计的是前向过程中添加的噪声以生成 $\mathbf{x}_i$。在最优情况下，该估计与真实噪声的条件期望一致，尽管 $\mathbf{x}_i$ 并不能唯一确定原始的干净样本。

另一种等价的参数化：$\mathbf{x}$ -预测。 Equation (2.2.4) 提出了一种替代但等价的参数化方法，称为 $\mathbf{x}$ -预测（干净预测），其中神经网络 $\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_i, i)$ 被训练以从给定的噪声输入 $\mathbf{x}_i \sim p_i(\mathbf{x}_i)$ 在噪声水平 i 下预测一个干净（去噪）样本。用 $\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_i, i)$ 替换反向均值表达式中的真实干净样本 $\mathbf{x}$，得到如下模型：

$$\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_i, i) = \frac{\bar{\alpha}_{i-1}\beta_i^2}{1-\bar{\alpha}_i^2}\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_i, i) + \frac{(1-\bar{\alpha}_{i-1}^2)\alpha_i}{1-\bar{\alpha}_i^2}\mathbf{x}_i.$$

类似于 $\boldsymbol{\epsilon}$ -预测的公式，训练目标可以表示为

$$\|\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_i, i) - \boldsymbol{\mu}(\mathbf{x}_i, \mathbf{x}_0, i)\|_2^2 \propto \|\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_i, i) - \mathbf{x}_0\|_2^2, \quad \mathbf{x}_0 \sim p_{\text{data}},$$

其中，模型被训练以从其噪声版本 $\mathbf{x}_i$ 预测原始数据样本 $\mathbf{x}$。这种等价性将 Equation (2.2.8) 中的均值匹配损失简化为

$$\mathbb{E}_i \mathbb{E}_{\mathbf{x}_0 \sim p_{\text{data}}} \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\omega_i \|\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_i, i) - \mathbf{x}_0\|_2^2 \right],$$

对于某个权重函数 ω_i ，由于这是一个最小二乘问题，最优解由（参见命题 4.2.1 或命题 6.3.1）给出

$$\mathbf{x}^*(\mathbf{x}_i, i) = \mathbb{E}\left[\mathbf{x}_0 | \mathbf{x}_i\right], \quad \mathbf{x}_i \sim p_i, \tag{2.2.11}$$

也就是说，模型应在给定时间步 i 的噪声观测 $\mathbf{x}_i$ 的情况下，预测期望的干净数据。

$\mathbf{x}$ 预测和 $\boldsymbol{\epsilon}$ 预测的参数化在数学上是等价的，并通过前向过程相互关联：

$$\mathbf{x}_i = \bar{\alpha}_i \mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_i, i) + \sqrt{1-\bar{\alpha}_i^2}\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_i, i). \tag{2.2.12}$$

也就是说，可以预测干净样本 $\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_i, i)$ 或噪声 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_i, i)$ ，使得它们的组合在前向加噪过程中重现 $\mathbf{x}_i$ 。

2.2.5 DDPM 的证据下界（ELBO）

将反向转移定义为如 Equation (2.2.5) 所示，这导致了 DDPM 中联合生成分布的定义：

$$p_{\boldsymbol{\phi}}(\mathbf{x}_0, \mathbf{x}_{1:L}) := p_{\boldsymbol{\phi}}(\mathbf{x}_0|\mathbf{x}_1) p_{\boldsymbol{\phi}}(\mathbf{x}_1|\mathbf{x}_2) \cdots p_{\boldsymbol{\phi}}(\mathbf{x}_{L-1}|\mathbf{x}_L) p_{\text{prior}}(\mathbf{x}_L),$$

数据的边缘生成模型由下式给出：

$$p_{\boldsymbol{\phi}}(\mathbf{x}_0) := \int p_{\boldsymbol{\phi}}(\mathbf{x}_0, \mathbf{x}_{1:L})\, \mathrm{d}\mathbf{x}_{1:L}.$$

事实上，通过 Equation (2.2.6) 进行的 DDPM 训练可以在极大似然估计 (Equation (1.1.2)) 的严格框架下进行。具体而言，其目标构成了一个 ELBO，类似于 Section 2.1 中的 VAEs 和 HVAEs，该 ELBO 作为对数密度的下界：

Theorem 2.2.3: DDPM 的 ELBO

$$\begin{aligned} -\log p_{\boldsymbol{\phi}}(\mathbf{x}_0) &\leq -\mathcal{L}_{\text{ELBO}}(\mathbf{x}_0; \boldsymbol{\phi}) \\ &:= \mathcal{L}_{\text{prior}}(\mathbf{x}_0) + \mathcal{L}_{\text{recon.}}(\mathbf{x}_0; \boldsymbol{\phi}) + \mathcal{L}_{\text{diffusion}}(\mathbf{x}_0; \boldsymbol{\phi}) \end{aligned} \tag{2.2.13}$$

此处，各损失分量定义如下：

$$\begin{aligned} \mathcal{L}_{\text{prior}}(\mathbf{x}_0) &:= \mathcal{D}_{\text{KL}}\Big(p(\mathbf{x}_L|\mathbf{x}_0) \| p_{\text{prior}}(\mathbf{x}_L)\Big) \\ \mathcal{L}_{\text{recon.}}(\mathbf{x}_0; \boldsymbol{\phi}) &:= \mathbb{E}_{p(\mathbf{x}_1|\mathbf{x}_0)}\left[-\log p_{\boldsymbol{\phi}}(\mathbf{x}_0|\mathbf{x}_1)\right] \\ \mathcal{L}_{\text{diffusion}}(\mathbf{x}_0; \boldsymbol{\phi}) &= \sum_{i=1}^{L} \mathbb{E}_{p(\mathbf{x}_i|\mathbf{x}_0)}\left[\mathcal{D}_{\text{KL}}\Big(p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}_0) \| p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i)\Big)\right]. \end{aligned}$$

Proof for Theorem.

该推导应用了 Jensen 不等式，类似于 HVAE/VAE 的 ELBO(Equation (2.1.3))，并进行了进一步简化。详细证明见 Section D.1.2。

ELBO $\mathcal{L}_{\text{ELBO}}$ 由三项组成：

- $\mathcal{L}_{\text{prior}}$ 可以通过选择噪声调度 $\{\beta_i\}$ 使得 $p(\cdot|\mathbf{x}_0) \approx p_{\text{prior}}(\cdot)$ 而变得可忽略。
- 对于 $\mathcal{L}_{\text{recon.}}$，可以使用蒙特卡罗估计进行近似和优化；参见 (**ho2020denoising**; **kingma2021variational**) 了解实际实现。
- $\mathcal{L}_{\text{diffusion}}$ （参见 Equation (2.2.6)）在所有步骤 i 中均与反向条件语句 $p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i)$ 到 $p(\mathbf{x}_{i-1}|\mathbf{x}_i)$ 匹配。

ELBO 目标 $\mathcal{L}_{\text{ELBO}}$ 也可以通过潜在变量 $\mathbf{z} = \mathbf{x}_{1:L}$ 的数据处理不等式视角来

解释，如 Equation (2.1.2) 所示：

$$\mathcal{D}_{\mathrm{KL}}(p_{\mathrm{data}}(\mathbf{x}_0)\|p_{\boldsymbol{\phi}}(\mathbf{x}_0)) \leq \mathcal{D}_{\mathrm{KL}}\left(p(\mathbf{x}_0,\mathbf{x}_{1:L})\|p_{\boldsymbol{\phi}}(\mathbf{x}_0,\mathbf{x}_{1:L})\right),$$

其中 $p(\mathbf{x}_0,\mathbf{x}_{1:L}) := p_{\mathrm{data}}(\mathbf{x}_0)p(\mathbf{x}_1|\mathbf{x}_0)p(\mathbf{x}_2|\mathbf{x}_1)\cdots p(\mathbf{x}_L|\mathbf{x}_{L-1})$ 表示前向过程中的联合分布。

Remark.

> 扩散模型的变分视角符合 HVAE 框架：固定前向加噪链充当" 编码器"，且潜在空间 $\mathbf{x}_{1:T}$ 与数据维度相同。训练过程最大化相同 ELBO。该框架无需学成的编码器与逐层 KL 项，而是将目标分解为从大到小噪声（由粗到细）的良态降噪子问题，从而实现稳定最优化，在保持时序/噪声维度上由粗到细层级结构的同时获得高质量样本。

2.2.6 采样

训练完 $\boldsymbol{\epsilon}$ -预测模型后，$\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_i,i)$ [3]，采样按 Figure 2.5 所示顺序进行，使用参数化转移 $p_{\boldsymbol{\phi}^\times}(\mathbf{x}_{i-1}|\mathbf{x}_i)$ 代替。

更具体地说，从一个随机种子 $\mathbf{x}_L \sim p_{\mathrm{prior}} = \mathcal{N}(\mathbf{0},\mathbf{I})$ 开始，我们根据下面的更新规则递归地从 $p_{\boldsymbol{\phi}^\times}(\mathbf{x}_{i-1}|\mathbf{x}_i)$ 中采样 $i = L, L-1, \ldots, 1$ ：

$$\mathbf{x}_{i-1} \leftarrow \underbrace{\frac{1}{\alpha_i}\left(\mathbf{x}_i - \frac{1-\alpha_i^2}{\sqrt{1-\bar{\alpha}_i^2}}\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_i,i)\right)}_{\boldsymbol{\mu}_{\boldsymbol{\phi}^\times}(\mathbf{x}_i,i)} + \sigma(i)\boldsymbol{\epsilon}_i, \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0},\mathbf{I}). \tag{2.2.14}$$

这个“降噪”过程会持续进行，直到获得最终的干净生成样本 $\mathbf{x}_0$。

DDPM 采样过程的另一种理解。 由 Equation (2.2.12)，与噪声估计 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_i,i)$ 对应的干净样本预测可表示为

$$\mathbf{x}_{\boldsymbol{\phi}^\times}(\mathbf{x}_i,i) = \frac{\mathbf{x}_i - \sqrt{1-\bar{\alpha}_i^2}\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_i,i)}{\bar{\alpha}_i}.$$

将其代入 Equation (2.2.14) 中的 DDPM 采样规则，得到等价的更新：

$$\mathbf{x}_{i-1} \leftarrow (\text{interpolation between } \mathbf{x}_i \text{ and clean prediction } \mathbf{x}_{\boldsymbol{\phi}^\times}) + \sigma(i)\boldsymbol{\epsilon}_i$$

[3]我们使用符号“× ”表示模型已经训练完成并被冻结。

表明每一步都围绕着预测的干净样本进行，同时添加了按 $\sigma(i)$ 缩放的高斯噪声。

这表明，DDPM 采样可以被看作是一个迭代降噪过程，该过程在以下步骤之间交替进行：

1. 从当前的噪声输入 $\mathbf{x}_i$ 估计干净数据 $\mathbf{x}_{\phi^\times}(\mathbf{x}_i, i)$ ，

2. 通过使用此干净估计的更新规则，对噪声更小的潜在变量 $\mathbf{x}_{i-1}$ 进行采样。

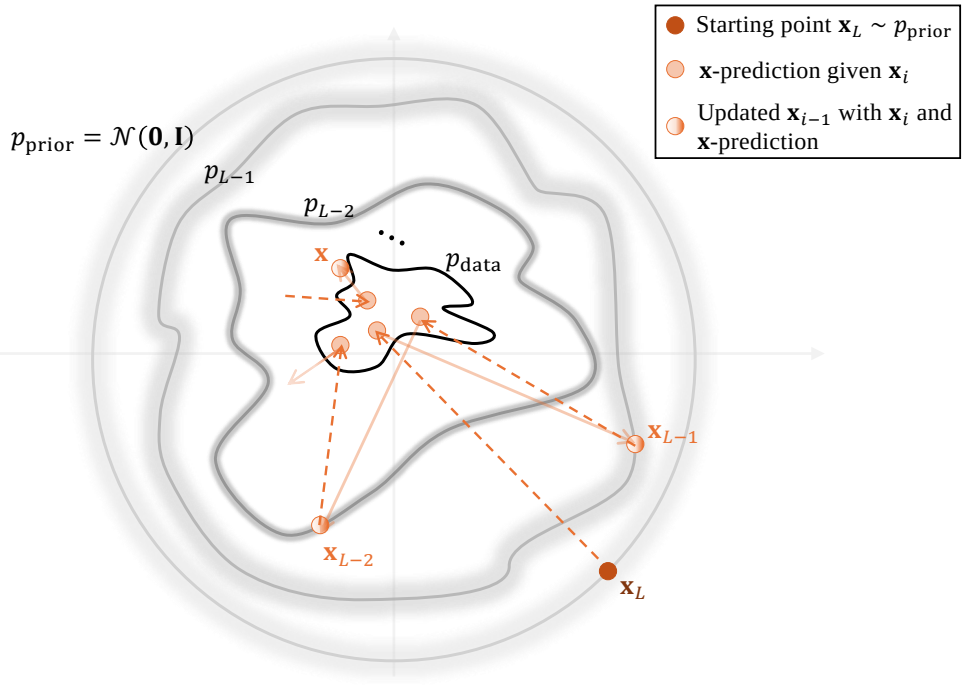


图 2.6: DDPM 采样使用干净预测的示意图：从 $\mathbf{x}_{\phi^\times}(\mathbf{x}_i, i)$ 估计 $\mathbf{x}_i$ ，然后更新至 $\mathbf{x}_{i-1}$ 。

然而，即使 $\mathbf{x}_{\phi^\times}$ 被训练为最优去噪器（即条件期望最小化器；见 Equation (2.2.11)），它也只能预测给定 $\mathbf{x}_i$ 的平均干净样本。这一限制导致预测结果模糊，尤其是在噪声水平较高时，从严重损坏的输入中恢复细节结构变得困难。

从这一视角来看，扩散采样通常是从高噪声向低噪声逐步进行，并不断优化对干净信号的估计。早期步骤确定全局结构，后期步骤添加精细细节，随着噪声的去除，样本变得越来越真实。

DDPM 的采样速度较慢。 DDPM（又称扩散模型）采样本质上速度较慢[4] 由于其逆过程具有顺序性，受以下因子制约。

定理 2.2.1 表明，一个表达能力强的 $p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i)$ 从理论上可以匹配真实的反向分布 $p(\mathbf{x}_{i-1}|\mathbf{x}_i)$。然而，在实际应用中，通常将 $p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i)$ 建模为高斯分布以近似 $p(\mathbf{x}_{i-1}|\mathbf{x}_i)$，这限制了其表达能力。

对于较小的前向噪声尺度 β_i，真实的逆向分布近似为高斯分布，从而可以实现准确的近似。相反，较大的 β_i 会引发多模态或强烈的非高斯性，单个高斯分布无法捕捉这些特性。为了保持准确性，DDPM 采用许多小的 β_i 步骤，形成一个序列链，其中每一步都依赖于前一步，并且需要进行一次神经网络评估 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_i, i)$。这导致了 $\mathcal{O}(L)$ 次序列传递，无法并行化，从而减慢了生成速度。

在 Chapter 4 中，我们展示了这种固有的采样瓶颈的更严谨的解释，将其视为一个微分方程问题，这启发了用于加速生成的连续时间数值策略。

[4] DDPM 通常需要 1000 步降噪。

2.3 闭幕词

在本章中，我们通过变分视角追溯了扩散模型的起源。我们从变分自编码器（VAE）开始，这是一种基础的生成式模型，它通过证据下界（ELBO）学习数据与结构化潜空间之间的概率映射。我们看到，分层变分自编码器（HVAE）通过堆叠潜层扩展了这一思想，引入了逐步的、由粗到细生成的强大概念。然而，这些模型在训练稳定性和样本质量方面仍面临挑战。

随后，我们将降噪扩散概率模型（DDPMs）视为这一变分框架中的关键演化。通过将编码器固定为渐进的加噪过程，仅学习反向去噪步骤，DDPMs 巧妙地规避了 HVAEs 的训练不稳定性问题。至关重要的是，我们证明了 DDPMs 也是通过最大化对数似然的变分界来进行训练的，其训练目标可分解为一系列简单的去噪任务。这种易处理性得益于一种强大的条件策略，该策略将难以处理的边缘目标转化为易处理的条件目标，这在扩散模型中是一个反复出现的主题。

尽管这种变分框架为 DDPMs 提供了完整而强大的理论基础，但这并非理解它们的唯一方式。另一种同样根本的视角源自基于能量的建模原则。在下一章中，我们将探讨这种基于得分的视角：

1. 我们将把关注点从学习去噪转移概率 $p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i)$ 转向直接学习数据的对数密度梯度，即评分函数。

2. 我们将看到，这种起源于能量基准模型（EBMs）的方法如何催生出噪声条件评分网络（NCSN），并揭示了在 DDPMs 中学习到的噪声预测（$\boldsymbol{\epsilon}$ -预测）与评分函数之间深刻的数学等价性。

这一替代视角不仅将提供新的洞见，还将作为后续将要构建的统一连续时间扩散模型框架的另一基石。

3

基于得分的视角：从能量基模型到 NCSN

在前几章中，我们将扩散模型追溯到其变分根源，并展示了它们如何在变分自编码器（VAE）的框架内产生。我们现在转向第二个同样基础的视角：基于能量的模型（EBMs）（**ackley1985learning**; **lecun2006tutorial**）。EBM 通过一个能量景观来表示分布，数据所在区域的能量较低，而其他区域的能量较高。采样通常依赖于朗之万动力学，该方法通过遵循此景观的梯度，将样本移动到高密度区域。这一梯度场被称为 *得分*，指向概率更高的方向。

核心观察是，仅知道得分就足以进行生成：它能够将样本移向高概率区域，而无需计算难以处理的归一化常数。基于得分的扩散模型直接建立在这一思想之上。它们不仅关注干净数据分布，还考虑一系列被高斯噪声扰动的分布，这些分布的得分更容易近似。学习这些得分可得到一组向量场，逐步引导含噪样本回归到数据，从而将生成过程转化为渐进式降噪。

3.1 基于能量的模型

对于已经熟悉 EBMs 的读者，本节旨在简明地回顾并搭建通往扩散模型得分函数视角的桥梁。

3.1.1 使用能量函数建模概率分布

令 $\mathbf{x} \in \mathbb{R}^D$ 表示一个数据点。能量模型通过一个能量函数 $E_\phi(\mathbf{x})$ 来定义概率密度，该函数由参数 ϕ 参数化，为更可能的配置分配更低的能量。由此得到的分布为

$$p_\phi(\mathbf{x}) := \frac{\exp(-E_\phi(\mathbf{x}))}{Z_\phi}, \quad Z_\phi := \int_{\mathbb{R}^D} \exp(-E_\phi(\mathbf{x}))\,\mathrm{d}\mathbf{x},$$

其中 Z_ϕ 称为 *配分函数*，用于确保规范化：

$$\int_{\mathbb{R}^D} p_\phi(\mathbf{x})\,\mathrm{d}\mathbf{x} = 1.$$

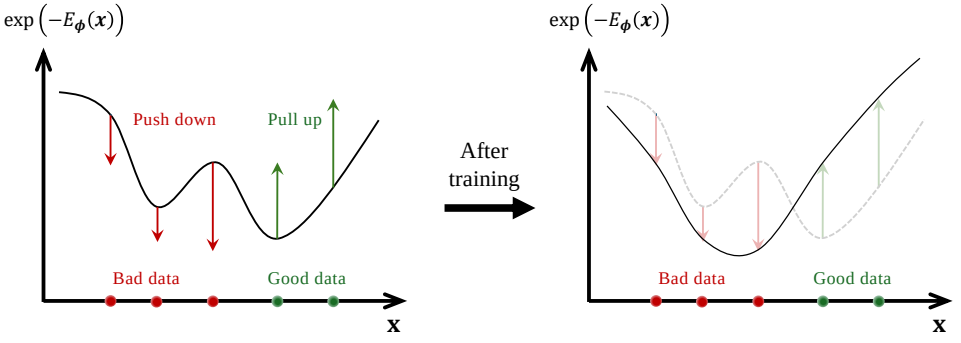


图 3.1: EBM 训练的示意图。模型在“差”的数据点（红色箭头）处降低密度（提高能量），在“好”的数据点（绿色箭头）处提高密度（降低能量）。

在这种视角下，能量较低的点对应较高的概率，类似于小球滚入山谷的过程。配分函数 Z_ϕ 确保所有概率之和为一，因此只有能量的相对值才重要。例如，向所有能量添加一个常数会使分子和分母同时乘以相同的因子，从而保持分布不变。

此外，由于配分函数 Z_ϕ 强制概率之和为一，从数学上可以得出结论：降低某个区域内的能量会增加该区域的概率，同时其补集的概率相应减少。因此，能

量基模型（EBMs）遵循严格的全局权衡：使一个山谷更深必然会使其他山谷变浅，概率质量在整个空间中重新分配，而不是独立地分配给每个区域。

极大似然训练在能量模型中的挑战。 原则上，能量模型可以通过极大似然进行训练，这自然地在拟合数据与全局正则化之间取得了平衡（见 Equation (1.1.2)）：

$$\begin{aligned}\mathcal{L}_{\mathrm{MLE}}(\boldsymbol{\phi}) &= \mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})}\left[\log \frac{\exp(-E_{\boldsymbol{\phi}}(\mathbf{x}))}{Z_{\boldsymbol{\phi}}}\right] \\ &= -\underbrace{\mathbb{E}_{p_{\mathrm{data}}}[E_{\boldsymbol{\phi}}(\mathbf{x})]}_{\text{lowers energy of data}} \underbrace{-\log \int \exp(-E_{\boldsymbol{\phi}}(\mathbf{x}))\,\mathrm{d}\mathbf{x}}_{\text{global regularization}},\end{aligned} \tag{3.1.1}$$

其中 $Z_{\boldsymbol{\phi}} = \int \exp(-E_{\boldsymbol{\phi}}(\mathbf{x}))\,\mathrm{d}\mathbf{x}$。第一项降低真实数据的能量，第二项通过配分函数强制规范化。

然而，在高维度下计算 $\log Z_{\boldsymbol{\phi}}$ 及其梯度是不可行的，因为这需要在模型分布下的期望。这促使了采用替代目标函数，这些目标函数要么近似该项，例如对比散度 (**hinton2002training**)，要么通过 分数匹配完全避免该项。

接下来，我们首先介绍 Section 3.1.2 中得分函数的概念，并提出分数匹配作为一种易处理的训练目标，可绕过 Section 3.1.3 中的配分函数，然后讨论在 Section 3.1.4 中使用得分函数的拉普拉斯动力学作为一种实用的采样法。

3.1.2 动机：得分是多少？

对于密度 $p(\mathbf{x})$ 在 $\mathbb{R}^D$ 上，得分函数是对数密度的梯度：

$$\mathbf{s}(\mathbf{x}) := \nabla_{\mathbf{x}} \log p(\mathbf{x}), \qquad \mathbf{s}\colon \mathbb{R}^D \to \mathbb{R}^D.$$

直观上，得分形成了一个指向高概率区域的向量场，为数据最可能出现的位置提供了局部指引（见 Figure 3.2）。

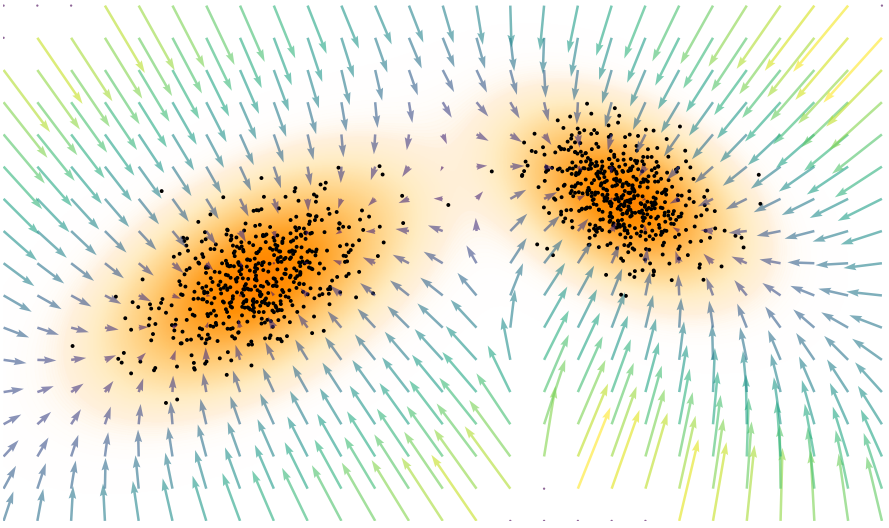


图 3.2: 得分向量场的示意图。得分向量场 $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ 表示密度增加的方向。

为何使用模型得分而非密度？ 建模得分既具有理论优势，也具有实际应用价值：

1. 与归一化常数无关。 许多分布仅以未归一化的密度 $\tilde{p}(\mathbf{x})$ 定义，例如 EBMs 中的 $\exp(-E_{\boldsymbol{\phi}}(\mathbf{x}))$：

$$p(\mathbf{x}) = \frac{\tilde{p}(\mathbf{x})}{Z}, \qquad Z = \int \tilde{p}(\mathbf{x})\, \mathrm{d}\mathbf{x}.$$

虽然计算 Z 是难以处理的，但得分仅依赖于 $\tilde{p}$：

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}) = \nabla_{\mathbf{x}} \log \tilde{p}(\mathbf{x}) - \underbrace{\nabla_{\mathbf{x}} \log Z}_{=0} = \nabla_{\mathbf{x}} \log \tilde{p}(\mathbf{x}), \tag{3.1.2}$$

由于 Z 在 $\mathbf{x}$ 中为常数。这完全绕过了配分函数。

2. 完整的表示。 评分函数完全刻画了潜在分布。由于它是对数密度的梯度，因此可以通过以下方式恢复密度（至多一个常数）

$$\log p(\mathbf{x}) = \log p(\mathbf{x}_0) + \int_0^1 \mathbf{s}(\mathbf{x}_0 + t(\mathbf{x} - \mathbf{x}_0))^\top (\mathbf{x} - \mathbf{x}_0)\, \mathrm{d}t,$$

其中 $\mathbf{x}_0$ 为参考点，$\log p(\mathbf{x}_0)$ 由规范化固定。因此，建模得分与直接建模 $p(\mathbf{x})$ 本身具有相同的表达能力，而在生成式建模中通常更易处理。

3.1.3 通过分数匹配训练 EBM

在能量模型中，密度定义为 $p_{\boldsymbol{\phi}}(\mathbf{x}) = \frac{\exp(-E_{\boldsymbol{\phi}}(\mathbf{x}))}{Z_{\boldsymbol{\phi}}}$。极大似然训练需要计算 $Z_{\boldsymbol{\phi}}$，这通常是不易处理的。一个关键观察是，模型得分 $p_{\boldsymbol{\phi}}$ 简化为：$-\nabla_{\mathbf{x}} E_{\boldsymbol{\phi}}(\mathbf{x})$，与 $Z_{\boldsymbol{\phi}}$ 无关（见 Equation (3.1.2)）。

分数匹配 (**hyvarinen2005estimation**) 利用了得分仅依赖于能量函数这一事实。与其拟合规范化的概率，不如通过将模型得分与（未知的）数据得分对齐来训练能量基模型。

$$\mathcal{L}_{\mathrm{SM}}(\boldsymbol{\phi}) = \tfrac{1}{2}\mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})} \left\| \nabla_{\mathbf{x}} \log p_{\boldsymbol{\phi}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\mathrm{data}}(\mathbf{x}) \right\|_2^2. \tag{3.1.3}$$

尽管数据得分不可访问，分部积分可得到仅涉及能量及其导数的等价表达式（详见命题 3.2.1 的更多细节）：

$$\mathcal{L}_{\mathrm{SM}}(\boldsymbol{\phi}) = \mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})} \left[\mathrm{Tr}\left(\nabla_{\mathbf{x}}^2 E_{\boldsymbol{\phi}}(\mathbf{x}) \right) + \tfrac{1}{2} \| \nabla_{\mathbf{x}} E_{\boldsymbol{\phi}}(\mathbf{x}) \|_2^2 \right] + C,$$

其中 $\nabla_{\mathbf{x}}^2 E_{\boldsymbol{\phi}}(\mathbf{x})$ 是 $E_{\boldsymbol{\phi}}$ 的海森矩阵，C 是与 $\boldsymbol{\phi}$ 无关的常数。

这种表述具有吸引力，因为它消除了配分函数，并在训练过程中避免了从模型中采样。其主要缺点是需要计算二阶导数，在高维度下计算成本可能过高。我们将在本章稍后重新讨论解决这一局限性的方法。

3.1.4 基于评分函数的 Langevin 采样

通过由能量函数 $E_{\boldsymbol{\phi}}(\mathbf{x})$ 定义的基于能量的模型（EBMs）进行采样，可以使用 *Langevin 动力学*实现。我们首先介绍离散时间的 Langevin 更新，然后将其

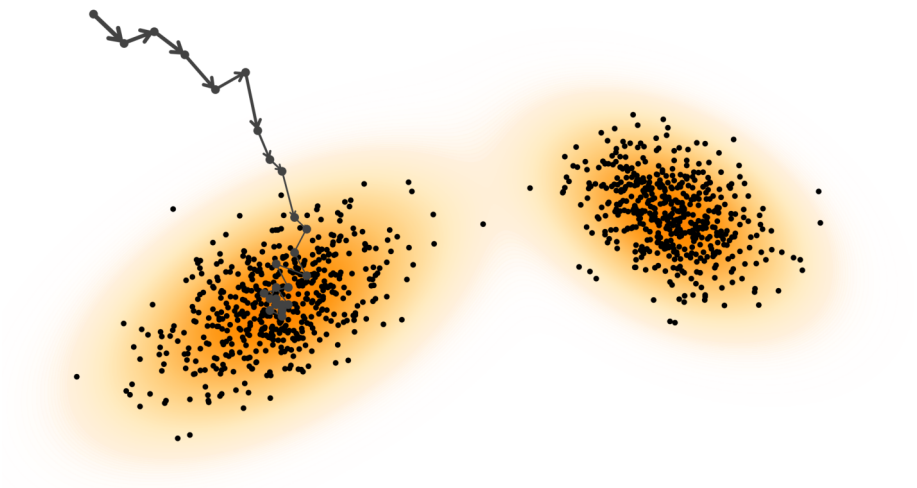


图 3.3: Langevin 采样示意图。使用评分函数 $\nabla_{\mathbf{x}} \log p_{\boldsymbol{\phi}}(\mathbf{x})$ 指导轨迹向高密度区域移动，更新方式如 Equation (3.1.5)(indicating by arrows) 所示。

连续时间极限表示为随机微分方程（SDE）。最后，我们将讨论 Langevin 动力学如何实现对复杂能量景观的高效探索的物理直观。

离散时间朗之万动力学 离散时间朗之万更新为

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \eta \nabla_{\mathbf{x}} E_{\boldsymbol{\phi}}(\mathbf{x}_n) + \sqrt{2\eta}\boldsymbol{\epsilon}_n, \quad n = 0, 1, 2, \ldots, \tag{3.1.4}$$

其中 $\mathbf{x}_0$ 从某个分布（通常是高斯分布）初始化，$\eta > 0$ 为步长，$\boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 为高斯噪声。噪声通过引入随机性，使得探索能够超越局部极小。

由于评分函数可以计算为

$$\nabla_{\mathbf{x}} \log p_{\boldsymbol{\phi}}(\mathbf{x}) = -\nabla_{\mathbf{x}} E_{\boldsymbol{\phi}}(\mathbf{x}).$$

更新可等价地写成

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \eta \nabla_{\mathbf{x}} \log p_{\boldsymbol{\phi}}(\mathbf{x}_n) + \sqrt{2\eta}\boldsymbol{\epsilon}_n, \tag{3.1.5}$$

其中，评分函数引导样本向高密度区域移动。这一公式是扩散模型的核心，稍后将详细说明。

连续时间朗之万动力学。 当步长 η 趋近于零时，离散的 Langevin 更新自然收敛到由 *Langevin 随机微分方程 (SDE)* 描述的连续时间过程[1]：

$$\mathrm{d}\mathbf{x}(t) = \nabla_{\mathbf{x}} \log p_{\boldsymbol{\phi}}(\mathbf{x}(t))\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{w}(t), \tag{3.1.6}$$

其中 $\mathbf{w}(t)$ 表示标准布朗运动（也称为维纳过程）[2]. 理解这一点非常重要：Equation (3.1.4) 中的离散更新规则是该连续随机微分方程的欧拉-马鲁亚玛离散化。

在标准正则性假设下（例如，$p_{\boldsymbol{\phi}} \propto e^{-E_{\boldsymbol{\phi}}}$ 服从一个约束的、足够光滑的 $E_{\boldsymbol{\phi}}$），当 $t \to \infty$ 时，$\mathbf{x}(t)$ 的分布会（指数快速地）收敛到 $p_{\boldsymbol{\phi}}$；因此我们可以通过模拟（求解）SDE Equation (3.1.6) 来进行采样。

为什么要使用朗之万采样？ 理解 Langevin 采样的一个自然方式是通过物理学的视角，其中能量函数 $E_{\boldsymbol{\phi}}(\mathbf{x})$ 定义了一个势能景观，该景观决定了粒子的行为。根据牛顿动力学，处于该能量所导出的力场中的粒子运动由常微分方程（ODE）描述。

$$\mathrm{d}\mathbf{x}(t) = -\nabla_{\mathbf{x}} E_{\boldsymbol{\phi}}\big(\mathbf{x}(t)\big)\,\mathrm{d}t,$$

这会确定性地驱动粒子向能量函数的局部极小值方向移动。然而，这种确定性动态可能会陷入局部极小值，从而阻碍对完整数据分布的探索。

为克服这一限制，朗之万动力学引入了随机扰动，从而得到随机微分方程（SDE）。

$$\mathrm{d}\mathbf{x}(t) = -\nabla_{\mathbf{x}} E_{\boldsymbol{\phi}}\big(\mathbf{x}(t)\big)\,\mathrm{d}t + \underbrace{\sqrt{2}\,\mathrm{d}\mathbf{w}(t)}_{\text{injected noise}},$$

其中 $\mathbf{w}(t)$ 为标准布朗运动。噪声项使得粒子能够通过跨越能垒来逃逸局部极小，从而使轨迹成为一个随机过程，其稳态分布收敛到玻尔兹曼分布。

$$p_{\boldsymbol{\phi}}(\mathbf{x}) \propto e^{-E_{\boldsymbol{\phi}}(\mathbf{x})}.$$

从这一视角来看，EBMs 可以被看作是学习一个力场，该力场将样本推向

---

[1] 在因子 $\sqrt{2}$ 作用下，朗之万动力学保持 $p_{\boldsymbol{\phi}}$ 随时间不变。即 $p_{\boldsymbol{\phi}}$ 是平稳的：若 $\mathbf{x}(0) \sim p_{\boldsymbol{\phi}}$，则对所有 $t \geq 0$ 有 $\mathbf{x}(t) \sim p_{\boldsymbol{\phi}}$。等价地，$p_{\boldsymbol{\phi}}$ 是福克-普朗克方程的平稳解（见 Chapter B）：

$$\partial_t \rho = -\nabla \cdot (\rho \nabla \log p_{\boldsymbol{\phi}}) + \tfrac{\sigma^2}{2} \Delta \rho.$$

在情景 $\rho = p_{\boldsymbol{\phi}}$ 下可得 $(\frac{\sigma^2}{2} - 1)\Delta p_{\boldsymbol{\phi}} = 0$，这仅当 $\sigma = \sqrt{2}$ 时成立。

[2] 布朗运动增量满足 $\mathbf{w}(t+\eta) - \mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \eta\mathbf{I})$。因此，欧拉-马鲁亚玛方法使用步长噪声 $\sqrt{2}[\mathbf{w}(t+\eta) - \mathbf{w}(t)] = \sqrt{2\eta}\boldsymbol{\epsilon}_n$，其中 $\boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$，这解释了 $\sqrt{\eta}$ 因子；这也是方根缩放的来源。有关布朗运动和随机微分方程的详细介绍，请参见 Chapter A。

高概率区域。Langevin 采样对 EBMs 尤为有用，因为它提供了一种实用的方法，能够在不显式计算配分函数的情况下从模型分布 $p_{\boldsymbol{\phi}}(\mathbf{x})$ 生成样本。通过迭代应用 Langevin 更新，可以获得近似于目标分布的样本。

Langevin 采样固有的挑战。 Langevin dynamics，一种广泛使用的基于 MCMC 的采样器，在高维空间中面临严重局限性。其效率对步长 η 、噪声尺度以及逼近目标分布所需的迭代次数选择极为敏感。

这种低效率的核心问题在于混合时间较差：在具有多个孤立模式的复杂数据分布中，Langevin 采样通常需要极长时间才能在高概率区域之间转移。随着维度增加，这一问题变得更加严重，导致收敛速度变得极其缓慢。

可以将采样视为探索一个广阔而崎岖的地形，其中许多遥远的山谷对应着不同的数据模式。基于局部随机更新的 Langevin 动力学难以高效地在这些山谷之间穿越，因此往往无法充分捕捉分布的多样性。

这种低效性暗示了需要采用更加结构化和有引导性的采样法，以比纯粹的随机探索更有效地导航复合数据流形。

3.2 From Energy-Based to Score-Based Generative Models

能量基模型（EBMs）表明，生成过程仅依赖于得分，该得分指向概率较高的区域，而非完整的规范化密度。虽然分数匹配避免了配分函数的计算，但通过能量进行训练仍需要昂贵的二阶导数。关键思想是，由于使用 Langevin 动力学采样仅需得分，因此我们可以直接用神经网络来学习得分。这种从建模能量到建模得分的转变，构成了基于得分的生成式模型的基础。

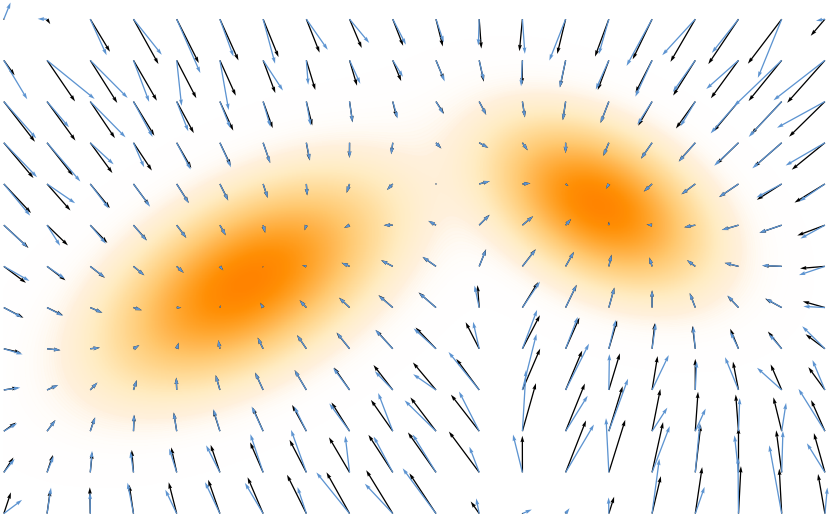


图 3.4: 分数匹配示意图。神经网络得分 $\mathbf{s}_{\phi}(\mathbf{x})$ 通过均方误差损失（MSE loss）训练以匹配真实值得分 $\mathbf{s}(\mathbf{x})$。两者均表示为向量场。

3.2.1 训练分数匹配

分数匹配 为了从 p_{data} 的样本近似得分函数 $\mathbf{s}(\mathbf{x}) = \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$，我们直接将其近似为由神经网络 $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})$ 参数化的向量场（见 Figure 3.4）：

$$\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}) \approx \mathbf{s}(\mathbf{x}).$$

分数匹配通过最小化真实分数与估计分数之间的均方误差（MSE）来拟合该向量场：

$$\mathcal{L}_{\mathrm{SM}}(\boldsymbol{\phi}) := \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}}\left[\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}) - \mathbf{s}(\mathbf{x})\|_2^2\right]. \tag{3.2.1}$$

易处理的分数匹配 乍看之下，这个目标似乎不可行，因为作为回归目标的真得分 $\mathbf{s}(\mathbf{x})$ 是未知的。幸运的是，**hyvarinen2005estimation** 表明，分部积分法可得出一个等价的目标，该目标仅依赖于模型 $\mathbf{s}_{\boldsymbol{\phi}}$ 和数据样本，而无需访问真得分。我们将在以下命题中陈述这一关键结果：

Proposition 3.2.1: Hyvärinen 的可处理形式的分数匹配

我们可以将以下方程表示为：

$$\mathcal{L}_{\mathrm{SM}}(\boldsymbol{\phi}) = \widetilde{\mathcal{L}}_{\mathrm{SM}}(\boldsymbol{\phi}) + C.$$

其中

$$\widetilde{\mathcal{L}}_{\mathrm{SM}}(\boldsymbol{\phi}) := \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}(\mathbf{x})}\left[\mathrm{Tr}\left(\nabla_{\mathbf{x}}\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})\right) + \frac{1}{2}\left\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})\right\|_2^2\right], \tag{3.2.2}$$

且 C 是一个与 $\boldsymbol{\phi}$ 无关的常数。最小化器 $\mathbf{s}^*$ 的表达式为：$\mathbf{s}^*(\cdot) = \nabla_{\mathbf{x}} \log p(\cdot)$ 。

Proof for Proposition.

> 该结果通过对 $\mathcal{L}_{\mathrm{SM}}$ 中的均方误差进行展开并应用分部积分得到。证明详见 Section D.2.1。

使用公式 Equation (3.2.2) 中的等价目标，我们仅从 p_{data} 的观测样本中训练评分模型 $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})$，从而无需真实评分函数。

对 Equation (3.2.2) 的直觉理解。 另一种分数匹配目标 $\widetilde{\mathcal{L}}_{\mathrm{SM}}(\boldsymbol{\phi})$ 可以直接从其两个项来理解。范数项 $\frac{1}{2}\|s_{\boldsymbol{\phi}}(\mathbf{x})\|^2$ 抑制了在 p_{data} 较大的区域中的分数，使其变为平稳的。散度项 $\mathrm{Tr}(\nabla_{\mathbf{x}} s_{\boldsymbol{\phi}}(\mathbf{x}))$ 倾向于负值，因此这些平稳点表现为吸引的汇点。两者共同作用，使高密度区域转化为分数场中的稳定且收缩的点。我们将在下面详细解释这一点。

来自幅度项的平稳性。 由于 $\widetilde{\mathcal{L}}_{\mathrm{SM}}(\boldsymbol{\phi})$ 中的期望是在 p_{data} 下取的，因此 $p_{\mathrm{data}}(\mathbf{x})$ 较大（数据密度较高）的区域对损失的贡献最大。因此，幅度项 $\frac{1}{2}\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})\|^2$ 会精确地将 $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}) \to 0$ 拉向这些高概率区域，即这些位置变为 *平稳的*。

当场近似为梯度时的凹性。 散度项 $\mathrm{Tr}(\nabla_{\mathbf{x}}\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}))$ 促使向量场在数据密度较高的区域具有负散度。负散度意味着附近的向量相互汇聚而非发散，因此该区域内的平稳点表现为一个 *汇点*：附近的轨迹被拉向内部。为了使这一点更加精确，假设 $\mathbf{s}_{\boldsymbol{\phi}} = \nabla_{\mathbf{x}} u$ 对于某个标量函数 $u : \mathbb{R}^D \to \mathbb{R}$，这在匹配对数密度时是自然的。于是有 $\nabla_{\mathbf{x}}\mathbf{s}_{\boldsymbol{\phi}} = \nabla^2_{\mathbf{x}} u$ （海森矩阵）和 $\nabla \cdot \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}) = \mathrm{Tr}(\nabla^2_{\mathbf{x}} u(\mathbf{x}))$ （散度）。

在驻点 $\mathbf{x}_\star$ 处，当 $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_\star) = \nabla_{\mathbf{x}} u(\mathbf{x}_\star) = \mathbf{0}$ 时，二阶泰勒展开给出

$$u(\mathbf{x}) = u(\mathbf{x}_\star) + \tfrac{1}{2}(\mathbf{x} - \mathbf{x}_\star)^\top \nabla^2_{\mathbf{x}} u(\mathbf{x}_\star)(\mathbf{x} - \mathbf{x}_\star) + o(\|\mathbf{x} - \mathbf{x}_\star\|^2).$$

如果海森矩阵 $\nabla^2_{\mathbf{x}} u(\mathbf{x}_\star)$ 为负定，则 u 在 $\mathbf{x}_\star$ 处局部凹且对数密度在该点取得严格局部极大点[3] there. 由于海森矩阵的所有特征值均为负数，迹也为负数：$\mathrm{Tr}(\nabla^2_{\mathbf{x}} u(\mathbf{x}_\star)) < 0$. 因此，学成的向量场具有负散度，平稳点为一个 *汇*：微小扰动将被收缩回 $\mathbf{x}_\star$ 。

3.2.2 基于朗之万动力学的采样

训练完成后，通过最小化 Equation (3.2.2)，得分模型 $\mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x})$ 可以在采样过程中替代兰杰文动力学中的真实得分：

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \eta \mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}_n) + \sqrt{2\eta}\boldsymbol{\epsilon}_n, \quad \boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{3.2.3}$$

对于 $n = 0, 1, 2, \ldots$，初始化为 $\mathbf{x}_0$。与 EBM 情况 Equation (3.1.6) 类似，此递归恰好是连续时间 Langevin SDE 的 Euler–Maruyama 离散化：

$$\mathrm{d}\mathbf{x}(t) = \mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}(t))\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{w}(t),$$

通过初始化 $\mathbf{x}(0)$。因此，在步长较小时，离散形式和连续形式趋于一致。在实际应用中，可以运行离散采样器，或直接模拟 SDE。

[3] 我们注意到严格凹性（因此对数密度的严格局部极大点）要求整个 Hessian $\nabla^2_{\mathbf{x}} u$ 是负定的，而不仅仅是迹为负。迹为负保证特征值之和为负，但某些特征值仍可能为正，从而导致鞍点而非极大点。

3.2.3 前言：基于得分的生成式模型

在本章的剩余部分，我们探讨评分函数在现代扩散模型中的基础性作用。最初引入评分函数是为了实现能量基模型（EBM）的高效训练，而如今评分函数已演变为新一代生成式模型的核心组成部分。在此基础上，我们研究评分函数如何指导“基于评分的扩散模型”的理论表述与实际实现，为通过随机过程进行数据生成提供一个严谨的框架。

3.3 Denoising Score Matching

3.3.1 动机

尽管在 Equation (3.2.2) 中存在其他目标

$$\widetilde{\mathcal{L}}_{\mathrm{SM}}(\boldsymbol{\phi}) = \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}}\left[\mathrm{Tr}\big(\nabla_{\mathbf{x}}\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})\big) + \frac{1}{2}\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})\|_2^2\right]$$

更易处理，但仍需要计算雅克比的迹，$\mathrm{Tr}(\nabla_{\mathbf{x}}\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}))$，其最坏情况下的复杂度为 $\mathcal{O}(D^2)$。这种复杂度限制了在高维数据上的可扩展性。

为解决此问题，切片分数匹配 (**song2020sliced**) 用基于随机投影的随机估计替代了迹项。我们简要概述其思路如下。

切片分数匹配与哈钦森估计量。 切片分数匹配用随机“切片”方向上的方向导数平均值替代了分数匹配中的迹。设 $\mathbf{u} \in \mathbb{R}^D$ 为一个 各向同性随机向量（例如 Rademacher 或标准高斯分布），满足 $\mathbb{E}[\mathbf{u}] = 0$ 和 $\mathbb{E}[\mathbf{u}\mathbf{u}^\top] = \mathbf{I}$ 。根据 Hutchinson 恒等式

$$\mathrm{Tr}(\mathbf{A}) = \mathbb{E}_{\mathbf{u}}[\mathbf{u}^\top\mathbf{A}\mathbf{u}], \quad \text{and} \quad \mathbb{E}_{\mathbf{u}}[(\mathbf{u}^\top\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}))^2] = \|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})\|_2^2,$$

我们得到了确切的形式

$$\widetilde{\mathcal{L}}_{\mathrm{SM}}(\boldsymbol{\phi}) = \mathbb{E}_{\mathbf{x},\mathbf{u}}\Big[\mathbf{u}^\top\big(\nabla_{\mathbf{x}}\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})\big)\mathbf{u} + \tfrac{1}{2}(\mathbf{u}^\top\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}))^2\Big].$$

该目标可通过自动微分高效评估，使用雅克比和向量-雅克比积运算（JVP/VJP）代替显式计算大型雅克比或海森矩阵。对 K 个随机探测点取平均可得到一个无偏估计量，其方差为 $\mathcal{O}(1/K)$，方向项 $\mathbf{u}^\top(\nabla_{\mathbf{x}}\mathbf{s}_{\boldsymbol{\phi}})\mathbf{u}$ 可通过 JVP/VJP 运算高效计算，无需显式雅克比矩阵。直观上，这意味着我们仅检查模型在随机方向上的行为：投影得分被引导以与数据密度较高的区域对齐，从而使数据点在期望下处于平稳状态。

从切片到降噪分数匹配。 切片分数匹配避开了雅克比矩阵，但仍依赖于原始数据分布。这使其变得脆弱：对于位于低维流形上的图像数据，分数 $\nabla_{\mathbf{x}}\log p_{\mathrm{data}}(\mathbf{x})$ 可能未定义或不稳定，且该方法仅在观测点上约束向量场，对其邻域的控制较弱。此外，它还受到探测引起的方差影响，并存在重复的 JVP/VJP 计算开销。

一种更稳健的替代方法，我们在此重点关注的是 降噪分数匹配 (DSM) (**vincent2011con**
它提供了一种合理且可扩展的解决方案。

3.3.2 训练

让我们重新审视 Equation (3.2.1) 中的 SM 损失：

$$\mathcal{L}_{\mathrm{SM}}(\boldsymbol{\phi}) = \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}(\mathbf{x})}\left[\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}) - \nabla_{\mathbf{x}}\log p_{\mathrm{data}}(\mathbf{x})\|_2^2\right],$$

其中问题源于难以处理的项 $\nabla_{\mathbf{x}}\log p_{\mathrm{data}}(\mathbf{x})$。

vincent2011connection 通过条件法的解。 为克服 $\nabla_{\mathbf{x}}\log p_{\mathrm{data}}(\mathbf{x})$ 的不可行性，**vincent2011connection** 提出通过已知的条件分布 $p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})$ 以尺度 σ 向数据 $\mathbf{x}\sim p_{\mathrm{data}}$ 注入噪声。神经网络 $\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma)$ 被训练以逼近边缘扰动分布的得分

$$p_\sigma(\tilde{\mathbf{x}}) = \int p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})p_{\mathrm{data}}(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

通过最小化损失

$$\mathcal{L}_{\mathrm{SM}}(\boldsymbol{\phi};\sigma) := \frac{1}{2}\mathbb{E}_{\tilde{\mathbf{x}}\sim p_\sigma}\left[\|\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma) - \nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}})\|_2^2\right]. \tag{3.3.1}$$

尽管 $\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}})$ 通常难以处理，**vincent2011connection** 表明对 $\mathbf{x}\sim p_{\mathrm{data}}$ 进行条件化可得到一个等价的易处理目标——降噪分数匹配（DSM）损失：

$$\mathcal{L}_{\mathrm{DSM}}(\boldsymbol{\phi};\sigma) := \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}},\tilde{\mathbf{x}}\sim p_\sigma(\cdot|\mathbf{x})}\left[\|\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma) - \nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})\|_2^2\right]. \tag{3.3.2}$$

最优极小化器 $\mathbf{s}^*$ 满足 Equation (3.3.2)

$$\mathbf{s}^*(\tilde{\mathbf{x}};\sigma) = \nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}}),$$

这也对 Equation (3.3.1) 也是最优的。

例如，当 $p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})$ 为方差为 σ^2 的高斯噪声时，

$$p_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}};\mathbf{x},\sigma^2\mathbf{I}),$$

梯度 $\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})$ 具有闭式解（见 Equation (3.3.4)），使得回归目标显式且计算上易处理。

此外，当 $\sigma \approx 0$ 、$p_\sigma(\tilde{\mathbf{x}}) \approx p_{\text{data}}(\mathbf{x})$ 和

$$\mathbf{s}^*(\tilde{\mathbf{x}};\sigma) = \nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}}) \approx \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}),$$

表明学成得分近似于原始数据得分，使其可用于生成。

我们将关于 $\mathcal{L}_{\text{SM}}$ 与 $\mathcal{L}_{\text{DSM}}$ 之间梯度等价性的讨论形式化为以下定理：

Theorem 3.3.1: $\mathcal{L}_{\text{SM}}$ 与 $\mathcal{L}_{\text{DSM}}$ 的等价性

对于任意固定的噪声尺度 $\sigma > 0$，以下关系成立：

$$\mathcal{L}_{\text{SM}}(\boldsymbol{\phi};\sigma) = \mathcal{L}_{\text{DSM}}(\boldsymbol{\phi};\sigma) + C, \tag{3.3.3}$$

其中 C 是与参数 $\boldsymbol{\phi}$ 无关的常数。此外，两个损失的最小化器 $\mathbf{s}^*(\cdot;\sigma)$ 满足

$$\mathbf{s}^*(\tilde{\mathbf{x}};\sigma) = \nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}}), \quad \text{对几乎所有的} \tilde{\mathbf{x}}$$

Proof for Theorem.

该等价性可通过直接计算得出：通过展开 $\mathcal{L}_{\text{SM}}$ 和 $\mathcal{L}_{\text{DSM}}$ 中的均方误差，所有与 $\boldsymbol{\phi}$ 相关的项相互抵消，仅留下与 $\boldsymbol{\phi}$ 无关的常数差。
最小化器的推导过程与命题 4.2.1 的论证方法一致。

该定理如同 DDPM 中的定理 2.2.1所示，揭示了一个关键的共通原理：

洞察 3.3.1: 条件化技术

条件化技术同样出现在 DDPM 中扩散模型的变分视角（见定理 2.2.1），其中对数据点 $\mathbf{x}$ 进行条件化可将一个难以处理的损失转化为可用于蒙特卡罗估计的易处理形式。类似的思想也出现在基于流的视角中（例如，流匹配 (**lipman2022flow**)），我们将在 Section 5.2 中看到这一点。

特殊情况：加性高斯噪声。 我们现在考虑一种常见情况，即对每个数据点 $\mathbf{x} \sim p_{\text{data}}$ 添加方差为 σ^2 的高斯噪声 $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ ：

$$\tilde{\mathbf{x}} = \mathbf{x} + \sigma\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

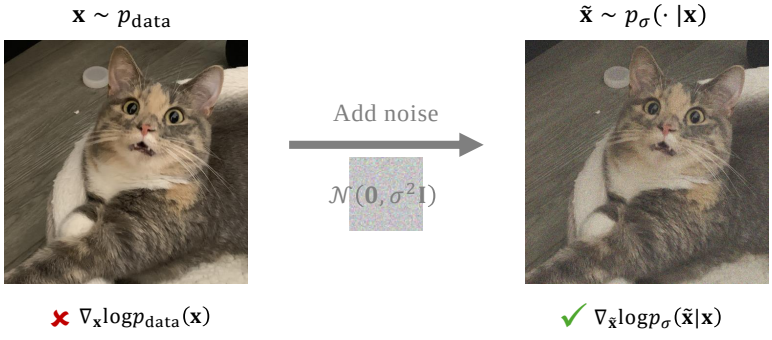


图 3.5: 通过条件化技术展示 DSM。通过对数据分布 p_{data} 添加小的加性高斯噪声 $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$，得到的条件分布 $p_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}}; \mathbf{x}, \sigma^2\mathbf{I})$ 具有闭式评分函数。

使得损坏的数据 $\tilde{\mathbf{x}}$ 保持不变

$$p_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}}; \mathbf{x}, \sigma^2\mathbf{I}).$$

在此情景下，条件得分在数学上可表示为

$$\nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = \frac{\mathbf{x} - \tilde{\mathbf{x}}}{\sigma^2}.$$

因此，DSM 损失简化为：

$$\begin{aligned} \mathcal{L}_{\text{DSM}}(\boldsymbol{\phi}; \sigma) &= \frac{1}{2}\mathbb{E}_{\mathbf{x},\tilde{\mathbf{x}}|\mathbf{x}}\left[\left\|\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}}; \sigma) - \frac{\mathbf{x} - \tilde{\mathbf{x}}}{\sigma^2}\right\|_2^2\right] \\ &= \frac{1}{2}\mathbb{E}_{\mathbf{x},\boldsymbol{\epsilon}}\left[\left\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x} + \sigma\boldsymbol{\epsilon}; \sigma) + \frac{\boldsymbol{\epsilon}}{\sigma}\right\|_2^2\right], \end{aligned} \tag{3.3.4}$$

其中 $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$。此目标构成了（基于得分的）扩散模型的核心。

当噪声水平 σ 较小时，高斯平滑后的边缘分布 $p_\sigma = p_{\text{data}} * \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ 与原始分布非常接近，因此它们的高密度区域和得分几乎重合：$\nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}}) \approx \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$。由此可知，沿着噪声得分方向 $\nabla_{\tilde{\mathbf{x}}} \log p_\sigma$ 做一个小步移动，可将噪声样本推向与干净分布几乎相同的高似然区域，这与分数匹配背后的直觉（如 Section 3.2.1 所总结）相似。相比之下，当 σ 较大时，平滑过程“过度简化”了分布结构：p_σ 会抹去局部模式，其得分主要将样本拉向全局质量中心（可类比向均值收缩），导致降噪效果粗糙且可能过度平滑。然而在实际应用中，DSM 通常假设注入的噪声较小且温和。

为了更清楚地理解目标为何自然对应于一个“降噪”过程，我们将进一步探讨 Sections 3.3.4 and 3.3.5 中的讨论。

3.3.3 采样

一旦我们获得了在噪声水平 σ 下训练好的得分模型 $\mathbf{s}_{\phi^\times}(\tilde{\mathbf{x}};\sigma)$，便可以通过用学成的模型替换真实得分，利用 Langevin 动态生成样本。更新规则为：

$$\tilde{\mathbf{x}}_{n+1} = \tilde{\mathbf{x}}_n + \eta \underbrace{\mathbf{s}_{\phi^\times}(\tilde{\mathbf{x}}_n;\sigma)}_{\approx\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}}_n)} + \sqrt{2\eta}\boldsymbol{\epsilon}_n, \quad \boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0},\mathbf{I}), \tag{3.3.5}$$

对于 $n = 0, 1, 2, \ldots$，从初始值 $\tilde{\mathbf{x}}_0$ 开始。如果 σ 足够小，则经过足够多的迭代后，$\tilde{\mathbf{x}}_n$ 将近似于从 p_{data} 生成的样本。

噪声注入的优势。 我们还注意到，与 Equation (3.2.1) 中的原始分数匹配相比，注入高斯噪声以形成 p_σ（例如，Equation (3.3.4)）提供了两个关键优势 (**song2019generative**)

- **梯度定义良好**。噪声将数据扰动远离其低维流形，导致分布 p_σ 在 $\mathbb{R}^D$ 上具有完整的支撑集。因此，评分函数 $\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}})$ 在所有地方都有定义。

- **改进的覆盖**。噪声平滑了模式之间的稀疏区域，提升了训练信号的质量，并促进了 Langevin 动力学更有效地穿越低密度区域。

3.3.4 为什么 DSM 是降噪：特威迪公式

我们从 *Tweedie* 公式 (**efron2011tweedie**) 开始，该公式为仅从噪声观测中进行降噪提供了合理的理论基础。具体而言，它指出：给定一个来自未知 $\mathbf{x} \sim p_{\text{data}}$ 的单个高斯噪声污染观测 $\tilde{\mathbf{x}} \sim \mathcal{N}(\cdot;\alpha\mathbf{x},\sigma^2\mathrm{I})$，通过将 $\tilde{\mathbf{x}}$ 沿其噪声边缘的得分 $\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}})$ 方向移动大小为 σ^2 的一步，即可获得去噪估计（即在给定 $\tilde{\mathbf{x}}$ 的所有可能干净信号上的平均值）：

$$p_\sigma(\tilde{\mathbf{x}}) := \int \mathcal{N}(\tilde{\mathbf{x}};\alpha\mathbf{x}_0,\sigma^2\mathrm{I})p_{\text{data}}(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

我们下面正式提出这一命题。

Lemma 3.3.2: Tweedie's Formula

Assume $\mathbf{x} \sim p_{\text{data}}$ and, conditionally on $\mathbf{x}$, $\tilde{\mathbf{x}} \sim \mathcal{N}(\,\cdot\,; \alpha\mathbf{x}, \sigma^2\mathrm{I})$ with $\alpha \neq 0$. Then Tweedie's formula states

$$\alpha\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}|\tilde{\mathbf{x}})}\left[\mathbf{x}|\tilde{\mathbf{x}}\right] = \tilde{\mathbf{x}} + \sigma^2\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}}), \tag{3.3.6}$$

where the expectation is taken over the posterior distribution $p(\mathbf{x}|\tilde{\mathbf{x}})$ of $\mathbf{x}$ given $\tilde{\mathbf{x}}$.

Proof for Lemma.

The proof proceeds by computing the score of the marginal $p(\tilde{\mathbf{x}}) = \int p(\tilde{\mathbf{x}}|\mathbf{x})p_{\text{data}}(\mathbf{x})\,\mathrm{d}\mathbf{x}$. Differentiating under the integral and using the Gaussian form of the conditional density leads directly to an expression that rearranges into the desired identity linking the score with the posterior mean. See Section D.2.3 for details.

Tweedie 公式在扩散模型中起着核心作用，其中如 DDPM 所示引入了多层噪声。它通过评分函数实现了从带噪声观测中估计干净样本，从而建立了评分预测与去噪器之间的基本联系：

$$\underbrace{\mathbb{E}\left[\mathbf{x}|\tilde{\mathbf{x}}\right]}_{\substack{\text{denoiser}\\ \text{estimated from } \tilde{\mathbf{x}}}} = \frac{1}{\alpha}\left(\tilde{\mathbf{x}} + \sigma^2\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}})\right).$$

特别地，对噪声对数似然进行一次特定步长 σ^2 的梯度上升步骤即为降噪估计（条件平均干净信号）。这使得 DSM 训练与降噪紧密相关：若 $\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}}) \approx \nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}})$ 由 DSM 训练得到，则

$$\frac{1}{\alpha}\left(\tilde{\mathbf{x}} + \sigma^2\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}})\right)$$

是去噪器。

（可选）高阶 Tweedie 公式。 经典的 Tweedie 公式将后验均值 $\mathbb{E}[\mathbf{x}_0|\tilde{\mathbf{x}}]$ 表示为梯度 $\nabla_{\tilde{\mathbf{x}}}\log p(\tilde{\mathbf{x}})$ 的函数。高阶推广 (**meng2021estimating**) 将后验协方差和高阶累积量表示为 $\log p(\tilde{\mathbf{x}})$ 的高阶导数。

指数族设定与对数归一化因子 $\lambda(\tilde{\mathbf{x}})$ 。 假设给定潜在自然参数 $\boldsymbol{\eta} \in \mathbb{R}^D$ 时，$\tilde{\mathbf{x}}$ 的条件分布属于如下形式的自然指数族

$$q_\sigma(\tilde{\mathbf{x}}|\boldsymbol{\eta}) = \exp\big(\boldsymbol{\eta}^\top \tilde{\mathbf{x}} - \psi(\boldsymbol{\eta})\big)\, q_0(\tilde{\mathbf{x}}).$$

此处 $q_0(\tilde{\mathbf{x}})$ 为 基测度，即不依赖于 $\boldsymbol{\eta}$ 的部分；对于方差为 $\sigma^2\mathbf{I}$ 的加性高斯噪声，其值等于 $(2\pi\sigma^2)^{-D/2}\exp(-\|\tilde{\mathbf{x}}\|^2/2\sigma^2)$ 。令 $p(\boldsymbol{\eta})$ 为潜在自然参数的预定义分布，可视为重参数化的干净数据分布（对于高斯位置，$\boldsymbol{\eta} = \mathbf{x}/\sigma^2$ ）。观测到的噪声边际分布为

$$p_\sigma(\tilde{\mathbf{x}}) = \int q_\sigma(\tilde{\mathbf{x}}|\boldsymbol{\eta})\, p(\boldsymbol{\eta})\, \mathrm{d}\boldsymbol{\eta}.$$

通过 $\tilde{\mathbf{x}}$ 定义 对数划分（对数归一化因子）为

$$\lambda(\tilde{\mathbf{x}}) := \log p_\sigma(\tilde{\mathbf{x}}) - \log q_0(\tilde{\mathbf{x}}).$$

然后，在给定 $\tilde{\mathbf{x}}$ 的条件下，$\boldsymbol{\eta}$ 的后验为

$$p(\boldsymbol{\eta}|\tilde{\mathbf{x}}) \;\propto\; \exp\big(\boldsymbol{\eta}^\top \tilde{\mathbf{x}} - \psi(\boldsymbol{\eta}) - \lambda(\tilde{\mathbf{x}})\big)\, p(\boldsymbol{\eta}),$$

这表明，作为 $\tilde{\mathbf{x}}$ 的函数，后验分布具有指数族形式，其自然参数为 $\tilde{\mathbf{x}}$ ，充分统计量为 $\boldsymbol{\eta}$ ，对数划分函数为 $\lambda(\tilde{\mathbf{x}})$ 。

导数为 λ 产生后验累积量。 有两个简单的规则在起作用。首先，规范化：对于每个 $\tilde{\mathbf{x}}$ ，

$$\int \exp\big(\boldsymbol{\eta}^\top \tilde{\mathbf{x}} - \psi(\boldsymbol{\eta}) - \lambda(\tilde{\mathbf{x}})\big)\, p(\boldsymbol{\eta})\, \mathrm{d}\boldsymbol{\eta} = 1.$$

对该恒等式关于 $\tilde{\mathbf{x}}$ 求导，会从指数函数中提取出 $\boldsymbol{\eta}$ 的幂次以及 $\lambda(\tilde{\mathbf{x}})$ 的导数；令结果为零，可得到 λ 的导数与 $\boldsymbol{\eta}$ 的后验矩之间的等式关系。其次，指数族的一个标准性质：对数划分函数是充分统计量的累积量生成函数。因此

$$\nabla_{\tilde{\mathbf{x}}}\lambda(\tilde{\mathbf{x}}) = \mathbb{E}[\boldsymbol{\eta}|\tilde{\mathbf{x}}], \quad \nabla^2_{\tilde{\mathbf{x}}}\lambda(\tilde{\mathbf{x}}) = \mathrm{Cov}[\boldsymbol{\eta}|\tilde{\mathbf{x}}], \quad \nabla^{(k)}_{\tilde{\mathbf{x}}}\lambda(\tilde{\mathbf{x}}) = \kappa_k(\boldsymbol{\eta}|\tilde{\mathbf{x}}) \quad (k \geq 3),$$

其中 κ_k 是随机向量 $\boldsymbol{\eta}$ 在给定 $\tilde{\mathbf{x}}$ 条件下的阶数为 k 的条件累积量，通过标准的矩–累积量关系得到。

这是高阶 Tweedie 公式。在高斯位置模型中令 $\boldsymbol{\eta} = \mathbf{x}/\sigma^2$，可得到关于

$\log p_\sigma(\tilde{\mathbf{x}})$ 导数的熟悉形式：

$$\mathbb{E}[\mathbf{x}|\tilde{\mathbf{x}}] = \tilde{\mathbf{x}} + \sigma^2 \nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}}), \qquad \mathrm{Cov}[\mathbf{x}|\tilde{\mathbf{x}}] = \sigma^2 \mathrm{I} + \sigma^4 \nabla^2_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}}),$$

且更高阶累积量与 $\log p_\sigma(\tilde{\mathbf{x}})$ 的更高阶导数成比例。

已有若干研究探讨了训练神经网络以估计高阶得分 (**meng2021estimating**; **lu2022maximum**; **lai2023fp**)。相比之下，我们的目标是阐明它们与统计量之间的关系，方法学细节请参阅这些文献。

3.3.5 （可选）为什么 DSM 是降噪：SURE

SURE（Stein 无偏风险估计量） 在高层次上，Stein 的无偏风险估计量（SURE）是一种技术，能够估计去噪器 $\mathbf{D}$ *在未知干净信号的情况下的均方误差*（MSE）。换句话说，当仅有噪声数据可用时，SURE 提供了一种选择或训练去噪器的方法。

为清晰起见，考虑加性高斯噪声情景：

$$\tilde{\mathbf{x}} = \mathbf{x} + \sigma \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

其中 $\mathbf{x} \in \mathbb{R}^D$ 是未知的干净信号，$\tilde{\mathbf{x}}$ 是观测到的噪声版本。去噪器是指任意（弱可微）映射 $\mathbf{D} : \mathbb{R}^D \to \mathbb{R}^D$ ，其生成 $\mathbf{x}$ 的估计值 $\mathbf{D}(\tilde{\mathbf{x}})$ 。

自然的质量度量是条件均方误差

$$R(\mathbf{D}; \mathbf{x}) := \mathbb{E}_{\tilde{\mathbf{x}}|\mathbf{x}} \left[\|\mathbf{D}(\tilde{\mathbf{x}}) - \mathbf{x}\|_2^2 \,\middle|\, \mathbf{x} \right].$$

该量依赖于未知的真实值 $\mathbf{x}$，因此无法直接计算。然而，Stein 的恒等式（见 Section D.2.4）给出了如下可观测的替代量：

$$\mathrm{SURE}(\mathbf{D}; \tilde{\mathbf{x}}) = \|\mathbf{D}(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}\|_2^2 + 2\sigma^2 \, \nabla_{\tilde{\mathbf{x}}} \cdot \mathbf{D}(\tilde{\mathbf{x}}) - D\sigma^2, \tag{3.3.7}$$

其中 $\nabla_{\tilde{\mathbf{x}}} \cdot \mathbf{D}(\tilde{\mathbf{x}})$ 表示 $\mathbf{D}$ 的散度。我们强调，$\mathrm{SURE}(\mathbf{D}; \tilde{\mathbf{x}})$ 仅需要噪声观测 $\tilde{\mathbf{x}}$ ，而不需要干净的 $\mathbf{x}$ 。

直观上，SURE 由两个互补的部分组成。项 $\|\mathbf{D}(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}\|^2$ 衡量去噪器输出与含噪声输入之间的偏离程度；仅凭此项会低估真实误差，因为 $\tilde{\mathbf{x}}$ 已经受到噪声污染。散度项起到修正作用：它捕捉了去噪器对其输入微小扰动的敏感程度，有效考虑了噪声引入的方差。

重要的是，对于任何固定的但未知的 $\mathbf{x}$ ，

$$\mathbb{E}_{\tilde{\mathbf{x}}|\mathbf{x}}\left[\mathrm{SURE}(\mathbf{D};\mathbf{x}+\sigma\boldsymbol{\epsilon})\,\middle|\,\mathbf{x}\right]=R(\mathbf{D};\mathbf{x}),$$

其中期望是关于高斯噪声 $\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$ 的。因此，最小化 SURE（在期望意义上或经验上）等价于仅依赖于噪声数据来最小化真实的 MSE，在实践中，对 SURE 在 $\mathbf{x}\sim p_{\mathrm{data}}$ 和扰动噪声 $\boldsymbol{\epsilon}$ 上进行平均，可以得到全局 MSE 风险的无偏估计。

Tweedie 公式与贝叶斯最优性的联系。 令 $p_\sigma(\tilde{\mathbf{x}})=\big(p_{\mathrm{data}}*\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})\big)(\tilde{\mathbf{x}})$ 表示本节中考虑的噪声边缘分布。

SURE 是关于噪声的均方误差的无偏估计量，条件于 $\mathbf{x}$ ：

$$\mathbb{E}_{\tilde{\mathbf{x}}|\mathbf{x}}\big[\mathrm{SURE}(\mathbf{D};\tilde{\mathbf{x}})\big]=\mathbb{E}_{\tilde{\mathbf{x}}|\mathbf{x}}\big[\|\mathbf{D}(\tilde{\mathbf{x}})-\mathbf{x}\|^2\big].$$

因此，最小化期望 SURE 等价于最小化贝叶斯风险 $\mathbb{E}_{(\mathbf{x},\tilde{\mathbf{x}})}\big[\|\mathbf{D}(\tilde{\mathbf{x}})-\mathbf{x}\|^2\big]=\mathbb{E}_{\tilde{\mathbf{x}}}\big[\mathbb{E}_{\mathbf{x}|\tilde{\mathbf{x}}}\big[\|\mathbf{D}(\tilde{\mathbf{x}})-\mathbf{x}\|^2\big]\big]$ 由全期望定律（塔性质）得出。该分解给出了一个逐点最优化：对几乎每个 $\tilde{\mathbf{x}}$ ，

$$\mathbf{D}^*(\tilde{\mathbf{x}})=\arg\min_{\mathbf{z}}\ \mathbb{E}_{\mathbf{x}|\tilde{\mathbf{x}}}\big[\|\mathbf{z}-\mathbf{x}\|^2\big]=\mathbb{E}[\mathbf{x}|\tilde{\mathbf{x}}].$$

因此，SURE 最优去噪器与 Section 3.3.4中的贝叶斯估计量一致，根据 Tweedie 恒等式：

$$\mathbf{D}^*(\tilde{\mathbf{x}})=\mathbb{E}[\mathbf{x}|\tilde{\mathbf{x}}]=\tilde{\mathbf{x}}+\sigma^2\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}}).\tag{3.3.8}$$

SURE 与分数匹配的关系。 Equation (3.3.8) 中的恒等式启发我们通过得分场对去噪器 $\mathbf{D}$ 进行参数化：

$$\mathbf{D}(\tilde{\mathbf{x}})=\tilde{\mathbf{x}}+\sigma^2\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma),$$

其中 $\mathbf{s}_{\boldsymbol{\phi}}(\cdot;\sigma)$ 用于近似噪声得分 $\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\cdot)$ 。将 $\mathbf{D}(\tilde{\mathbf{x}})=\tilde{\mathbf{x}}+\sigma^2\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma)$ 代入 Equation (3.3.7) 得

$$\frac{1}{2\sigma^4}\,\mathrm{SURE}(\mathbf{D};\tilde{\mathbf{x}})=\mathrm{Tr}\big(\nabla_{\tilde{\mathbf{x}}}s_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma)\big)+\tfrac{1}{2}\|\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma)\|_2^2+\mathrm{const}(\sigma).$$

因此，对 $\tilde{\mathbf{x}} \sim p_\sigma$ 取期望后，最小化 SURE 等价于（相差一个常数项）在噪声水平 σ 下最小化 Hyvärinen 评分匹配的目标函数，其中期望是在 p_σ 下取的（见 Equation (3.2.2)）。因此，这两个目标函数具有相同的极小化器，即 Equation (3.3.8) 中的去噪器。

3.3.6 （可选）广义得分匹配

动机。 经典分数匹配、降噪分数匹配以及高阶变体均以目标

$$\frac{\mathcal{L}p(\mathbf{x})}{p(\mathbf{x})}, \quad \text{for some density } p$$

一个线性算符 $\mathcal{L}$ 作用于密度上。在经典情形 $\mathcal{L} = \nabla_{\mathbf{x}}$ 中，这给出

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}) = \frac{\nabla_{\mathbf{x}} p(\mathbf{x})}{p(\mathbf{x})}$$

$\frac{\mathcal{L}p}{p}$ 结构允许通过分部积分消除归一化常数，从而得到仅依赖于 p 的样本和学成场 $\mathbf{s}_{\boldsymbol{\phi}}$ 的易处理目标。这一观点推动了广义得分匹配框架的发展。

广义 Fisher 散度 设 p 为数据分布，q 为任意模型分布。对于定义在 $\mathbf{x}$ 上的标量函数上的线性算子 $\mathcal{L}$ ，定义广义费舍尔散度

$$\mathcal{D}_{\mathcal{L}}(p \,\|\, q) := \int p(\mathbf{x}) \left\| \frac{\mathcal{L}p(\mathbf{x})}{p(\mathbf{x})} - \frac{\mathcal{L}q(\mathbf{x})}{q(\mathbf{x})} \right\|_2^2 \mathrm{d}\tilde{\mathbf{x}}.$$

如果 $\mathcal{L}$ 是 完备的，即

$$\frac{\mathcal{L}p_1}{p_1} = \frac{\mathcal{L}p_2}{p_2} \text{ a.e.} \quad \text{implies} \quad p_1 = p_2 \text{ a.e.},$$

然后 $\mathcal{D}_{\mathcal{L}}(p \,\|\, q) = 0$ 识别 $q = p$ 。对于 $\mathcal{L} = \nabla_{\tilde{\mathbf{x}}}$ ，这恢复了经典的 Fisher 散度（参见 Equation (1.1.3)）。

得分参数化。 在实际应用中，我们不建模规范化的密度 q 。相反，我们直接参数化一个向量场 $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})$ 来近似泛化得分 $\frac{\mathcal{L}p(\mathbf{x})}{p(\mathbf{x})}$ 。考虑

$$\mathcal{D}_{\mathcal{L}}(p \,\|\, \mathbf{s}_{\boldsymbol{\phi}}) := \mathbb{E}_{\mathbf{x}\sim p}\left[\left\| \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}) - \frac{\mathcal{L}p(\mathbf{x})}{p(\mathbf{x})} \right\|_2^2 \right].$$

尽管 $\frac{\mathcal{L}p(\mathbf{x})}{p(\mathbf{x})}$ 未知，”分部积分法”使得损失仅依赖于 $\mathbf{s}_{\boldsymbol{\phi}}$。设 $\mathcal{L}^\dagger$ 为 $\mathcal{L}$ 的伴随算子，其定义为

$$\int (\mathcal{L}f)^\top g = \int f\,(\mathcal{L}^\dagger g) \quad \text{for all test functions } f, g,$$

当边界项消失时，该式正式“移动”了 $\mathcal{L}$ 跨越积分。展开方阵并应用此恒等式可得到易处理的目标

$$\mathcal{L}_{\mathrm{GSM}}(\boldsymbol{\phi}) = \mathbb{E}_{\mathbf{x}\sim p}\left[\frac{1}{2}\left\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})\right\|_2^2 - \left(\mathcal{L}^\dagger \mathbf{s}_{\boldsymbol{\phi}}\right)(\mathbf{x})\right] + \text{const},$$

其中常数不依赖于 $\boldsymbol{\phi}$。我们仅通过期望使用 p，因此广义得分匹配损失可以从训练数据中得到一个经验估计量，正如经典得分匹配一样确切地实现。

对于 $\mathcal{L} = \nabla$，我们有 $\mathcal{L}^\dagger = -\nabla\cdot$，这在 Equation (3.2.2) 中恢复了 Hyvärinen 的分数匹配目标 $\mathbb{E}_p\big[\frac{1}{2}\|\mathbf{s}_{\boldsymbol{\phi}}\|_2^2 + \nabla\cdot\mathbf{s}_{\boldsymbol{\phi}}\big]$。

运算符示例。

- **经典分数匹配**。考虑 $\mathcal{L} = \nabla_{\mathbf{x}}$。则泛化得分退化为经典评分函数

$$\frac{\mathcal{L}p(\mathbf{x})}{p(\mathbf{x})} = \nabla_{\mathbf{x}} \log p(\mathbf{x}).$$

- **降噪分数匹配**。对于加性高斯噪声，定义算子

$$(\mathcal{L}f)(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}\, f(\tilde{\mathbf{x}}) + \sigma^2 \nabla_{\tilde{\mathbf{x}}} f(\tilde{\mathbf{x}}).$$

  然后

$$\frac{\mathcal{L}p_\sigma(\tilde{\mathbf{x}})}{p_\sigma(\tilde{\mathbf{x}})} = \tilde{\mathbf{x}} + \sigma^2 \nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}}) = \mathbb{E}[\mathbf{x}_0|\tilde{\mathbf{x}}],$$

  其中包含 $p_\sigma(\tilde{\mathbf{x}}) := \int \mathcal{N}(\tilde{\mathbf{x}}; \alpha\mathbf{x}_0, \sigma^2\mathrm{I}) p_{\mathrm{data}}(\mathbf{x})\,\mathrm{d}\mathbf{x}$ 和 $\tilde{\mathbf{x}} = \mathbf{x} + \sigma\boldsymbol{\epsilon}$。这正是 Tweedie 的恒等式。通过该算子最小化 $\mathcal{L}_{\mathrm{GSM}}$ 可以训练 $\mathbf{s}_{\boldsymbol{\phi}}$ 以逼近去噪器，从而恢复降噪分数匹配的目标。

- **高阶目标**。将导数嵌套在 $\mathcal{L}$ 内，可揭示 $\nabla^2 \log p$ 及更高阶导数，这些与后验协方差及高阶累积量对齐。

扩展功能与应用场景。 广义得分匹配将连续变量的范围扩展到离散情形，包括语言模型化 (**meng2022concrete**; **lou2024discrete**)。它还激发了基于得分的

训练方法，从而产生类似降噪的目标。这种算子视角统一了多种目标，能够从数据中进行经验估计，并通过适当选择 $\mathcal{L}$ 为设计损失函数提供了一般性原则。

3.4 Multi-Noise Levels of Denoising Score Matching (NCSN)

3.4.1 动机

向数据分布添加具有单一固定方差的高斯噪声在一定程度上可以平滑分布，但在单一噪声水平下训练得分模型会引入关键限制。在注入噪声水平较低时，由于低密度区域的梯度消失，Langevin 动力学难以在多模态分布中跨越不同模式。相反，在高噪声水平下，采样变得更容易，但模型仅能捕捉粗略结构，导致生成的样本模糊且缺乏细节。此外，Langevin 动力学在高维空间中可能收敛缓慢甚至失败。由于其依赖对数密度的梯度进行引导，初始值不佳，特别是在平坦区域或鞍点附近，会阻碍探索过程，或导致采样器陷入单一模式中。

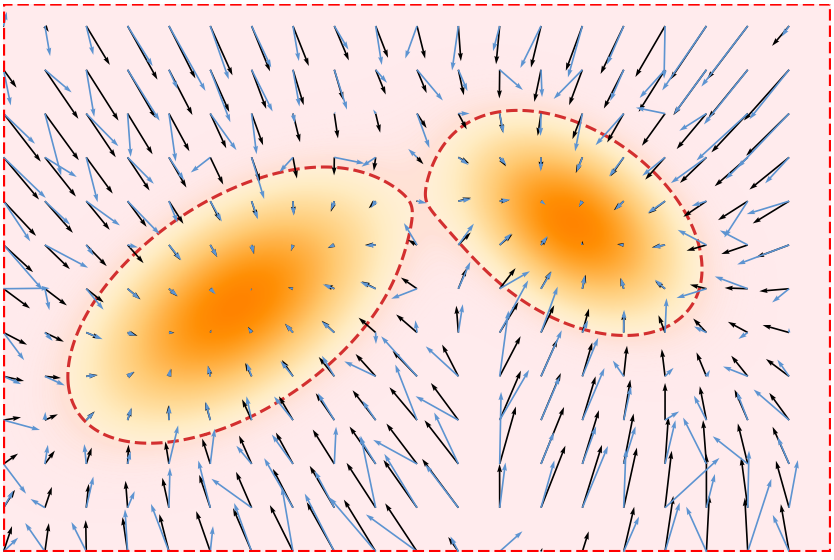


图 3.6: SM 不准确性的示意图（重新审视 Figure 3.4）。红色区域表示低密度区域，由于样本覆盖有限，可能导致得分估计不准确，而高密度区域则往往能产生更准确的估计。

为应对这些挑战，**song2019generative** 提出在数据分布的多个层次注入高斯噪声，并联合训练一个噪声条件下的评分网络（NCSN），以估计不同噪声尺度下的评分函数。在生成过程中，采用噪声退火的方式应用朗之万动力学：从高噪声水平开始以实现粗粒度探索，然后逐步降低噪声水平，以恢复精细细节。

3.4.2 训练

为了克服在单一噪声水平下训练的得分模型的局限性，**song2019generative** 提出在数据分布上添加多级高斯噪声。具体而言，选择一个序列的 L 噪声水平

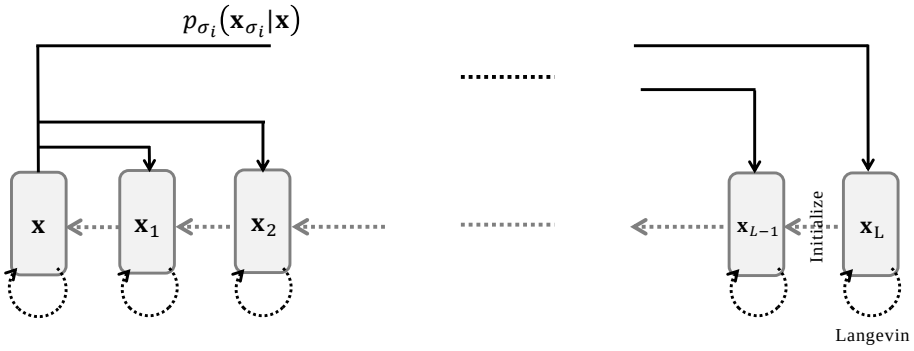


图 3.7: NCSN 的示意图。前向过程通过多级加性高斯噪声 $p_\sigma(\mathbf{x}_\sigma|\mathbf{x})$ 对数据进行扰动。生成过程在每个噪声级别上通过 Langevin 采样进行，利用当前级别的结果来初始化下一更低方差级别的采样。

$\{\sigma_i\}_{i=1}^L$ ，使得

$$0 < \sigma_1 < \sigma_2 < \cdots < \sigma_L,$$

其中 σ_1 足够小以保留数据的大部分细节，而 σ_L 足够大以充分平滑分布，从而便于训练。

每个噪声样本是通过扰动一个干净的数据点 $\mathbf{x} \sim p_{\text{data}}$ 得到的，即 $\mathbf{x}_\sigma = \mathbf{x} + \sigma\boldsymbol{\epsilon}$ 加上 $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 。这定义了

扰动核：

$$p_\sigma(\mathbf{x}_\sigma|\mathbf{x}) := \mathcal{N}(\mathbf{x}_\sigma; \mathbf{x}, \sigma^2\mathbf{I}),$$

，从而诱导出

边缘分布：

$$p_\sigma(\mathbf{x}_\sigma) = \int p_\sigma(\mathbf{x}_\sigma|\mathbf{x}) p_{\text{data}}(\mathbf{x})\, \mathrm{d}\mathbf{x},$$

在每个噪声水平 σ 下。它展示了经过高斯平滑的数据分布。

NCSN 的训练目标。 目标是训练一个噪声条件下的评分网络 $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}, \sigma)$，以估计所有 $\sigma \in \{\sigma_i\}_{i=1}^L$ 的评分函数 $\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x})$。这是通过在所有噪声水平上最小化 DSM 目标来实现的：

$$\mathcal{L}_{\text{NCSN}}(\boldsymbol{\phi}) := \sum_{i=1}^{L} \lambda(\sigma_i) \mathcal{L}_{\text{DSM}}(\boldsymbol{\phi}; \sigma_i), \tag{3.4.1}$$

在哪里

$$\mathcal{L}_{\mathrm{DSM}}(\boldsymbol{\phi};\sigma)=\frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}(\mathbf{x}),\tilde{\mathbf{x}}\sim p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}\left[\left\|\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}},\sigma)-\left(\frac{\mathbf{x}-\tilde{\mathbf{x}}}{\sigma^2}\right)\right\|_2^2\right],$$

且 $\lambda(\sigma_i)>0$ 是每个尺度的权重函数。

最小化此目标得到得分模型 $\mathbf{s}^*(\mathbf{x},\sigma)$，该模型在每个噪声水平下都能恢复真实得分：

$$\mathbf{s}^*(\cdot,\sigma)=\nabla_{\mathbf{x}}\log p_\sigma(\cdot),\quad \text{for all } \sigma\in\{\sigma_i\}_{i=1}^L,$$

因为它本质上是 DSM 最小化（参见定理 3.3.1）。

与 DDPM 损失的关系。 设 $\mathbf{x}_\sigma=\mathbf{x}+\sigma\boldsymbol{\epsilon}$ 与 $\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$ 相关，记 p_σ 为边缘分布。根据 Tweedie 公式，

$$\nabla_{\mathbf{x}_\sigma}\log p_\sigma(\mathbf{x}_\sigma)=-\frac{1}{\sigma}\mathbb{E}\left[\boldsymbol{\epsilon}|\mathbf{x}_\sigma\right].$$

因此，NCSN 的最优解是真实的得分 $\mathbf{s}^*(\mathbf{x}_\sigma,\sigma)=\nabla_{\mathbf{x}_\sigma}\log p_\sigma(\mathbf{x}_\sigma)$，而 DDPM 损失下的贝叶斯最优噪声预测器 Equation (2.2.10) 是 $\boldsymbol{\epsilon}^*(\mathbf{x}_\sigma,\sigma)=\mathbb{E}[\boldsymbol{\epsilon}|\mathbf{x}_\sigma]$。它们通过

$$\mathbf{s}^*(\mathbf{x}_\sigma,\sigma)=-\frac{1}{\sigma}\boldsymbol{\epsilon}^*(\mathbf{x}_\sigma,\sigma),\qquad \boldsymbol{\epsilon}^*(\mathbf{x}_\sigma,\sigma)=-\sigma\mathbf{s}^*(\mathbf{x}_\sigma,\sigma).$$

在 DDPM 的扰动 Equation (2.2.9) 与离散索引 i 中，

$$\mathbf{x}_i=\bar{\alpha}_i\mathbf{x}_0+\sqrt{1-\bar{\alpha}_i^2}$$

相同的关系给出

$$\mathbf{s}^*(\mathbf{x}_i,i)=-\frac{1}{\sigma_i}\mathbb{E}\left[\boldsymbol{\epsilon}|\mathbf{x}_i\right],$$

因此最小化 Equation (2.2.10) 学习到针对 $\boldsymbol{\epsilon}$ 的条件去噪器，这是在噪声水平 i 处真实得分的缩放重参数化。

我们将系统地比较和总结 Chapter 6 中参数化之间的等价性。

3.4.3 采样

在多个噪声水平下均可获得训练好的得分网络

$$\mathbf{s}_{\boldsymbol{\phi}^\times}(\cdot,\sigma_1),\quad \mathbf{s}_{\boldsymbol{\phi}^\times}(\cdot,\sigma_2),\quad \cdots,\quad \mathbf{s}_{\boldsymbol{\phi}^\times}(\cdot,\sigma_{L-1}),\quad \mathbf{s}_{\boldsymbol{\phi}^\times}(\cdot,\sigma_L),$$

Algorithm 1 Annealed Langevin Dynamics

Input: Trained score $\mathbf{s}_{\boldsymbol{\phi}^\times}(\cdot, \sigma_\ell)$, step sizes η_ℓ, and Langevin iteration budgets N_ℓ for each noise level $\ell = L, \ldots, 2$

$\mathbf{x}^{\sigma_L} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $\ell = L, \ldots, 2$ **do**
  $\tilde{\mathbf{x}}_0 \leftarrow \mathbf{x}^{\sigma_\ell}$
    ▷ Initialize Langevin from previous noise level's output
  for $n = 0$ **to** $N_\ell - 1$ **do**
    $\boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\tilde{\mathbf{x}}_{n+1} \leftarrow \tilde{\mathbf{x}}_n + \eta_\ell \mathbf{s}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_n, \sigma_\ell) + \sqrt{2\eta_\ell}\boldsymbol{\epsilon}_n$
  end for
  $\mathbf{x}^{\sigma_{\ell-1}} \leftarrow \tilde{\mathbf{x}}_{N_\ell}$
    ▷ Output used as initialization for next noise level
end for

Output: $\mathbf{x}^{\sigma_1}$

称为退火 *Langevin* 动力学 (**song2019generative**) 的采样过程通过从高噪声水平 σ_L 逐步降噪至低噪声水平 $\sigma_1 \approx 0$ 来生成数据。

从高斯噪声 $\mathbf{x}^{\sigma_L} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 开始，该算法在每个噪声水平 σ_ℓ 上应用 Langevin 动态，以近似采样自扰动分布 $p_{\sigma_\ell}(\mathbf{x})$。在层级 σ_ℓ 处的输出用于为下一个更低的噪声层级 $\sigma_{\ell-1}$ 提供一个 更好的初始化。

在每一层，朗之万动力学迭代更新：

$$\tilde{\mathbf{x}}_{n+1} = \tilde{\mathbf{x}}_n + \eta_\ell \mathbf{s}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_n, \sigma_\ell) + \sqrt{2\eta_\ell}\boldsymbol{\epsilon}_n, \quad \boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

从 $\tilde{\mathbf{x}}_0 := \mathbf{x}^{\sigma_\ell}$ 开始。步长通常按噪声水平进行缩放：

$$\eta_\ell = \delta \cdot \frac{\sigma_\ell^2}{\sigma_1^2}, \quad \text{for some fixed } \delta > 0.$$

这种噪声退火精炼过程会持续到最低噪声水平 σ_1，此时获得最终样本 $\mathbf{x}^{\sigma_1}$。通过逐步将前一级的输出作为下一级更优的初始化，该策略能够实现更有效的探索并提升对复合数据分布的覆盖。Algorithm 1 总结了该过程。

NCSN 的采样速度较慢。 NCSN 使用退火马尔可夫链蒙特卡洛（通常为朗之万动力学）在噪声尺度 $\{\sigma_i\}_{i=1}^L$ 上生成样本。对于每个尺度 σ_i ，它执行 K 次迭代更新，形式为 “使用得分 $\mathbf{s}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_n, \sigma_i)$ 加上一个小的随机扰动来更新 $\tilde{\mathbf{x}}_n$”，每次更新都需要通过得分网络进行一次前向传播。两个因子导致需要较大的 $L \times K$ ：

(i) **局部准确率和稳定性**：学成得分仅在小扰动下可靠，需要在每个噪声水平下采用较小的步长和较多的迭代次数，以避免偏差或不稳定；

(ii) **高维度下的混合缓慢**：局部的 MCMC 移动在探索多模态、高维度的目标时效率低下，需要大量迭代才能到达典型的数据区域。

由于更新是严格顺序进行的（每次迭代都依赖于前一次），且每次更新都需要一次昂贵的网络评估，因此总成本为 $\mathcal{O}(LK)$ 次顺序网络传递，导致采样计算速度较慢。

3.5 摘要：NCSN 与 DDPM 的比较视角

比较 我们首先在 Figure 3.7 中比较 NCSN 与 DDPM 的图模型，关键差异与相似之处总结于 Table 3.1。

表 3.1: NCSN 与 DDPM 的比较

	NCSN	DDPM
$\mathbf{x}_{i+1}\|\mathbf{x}_i$	Derive as $\mathbf{x}_{i+1} = \mathbf{x}_i + \sqrt{\sigma_{i+1}^2 - \sigma_i^2}\boldsymbol{\epsilon}$	Define as $\mathbf{x}_{i+1} = \sqrt{1-\beta_i}\mathbf{x}_i + \sqrt{\beta_i}\boldsymbol{\epsilon}$
$\mathbf{x}_i\|\mathbf{x}$	Define as $\mathbf{x}_i = \mathbf{x} + \sigma_i^2\boldsymbol{\epsilon}$	Derive as $\mathbf{x}_i = \bar{\alpha}_i\mathbf{x} + \sqrt{1-\bar{\alpha}_i^2}\boldsymbol{\epsilon}$
p_{prior}	$\mathcal{N}(\mathbf{0}, \sigma_L^2\mathbf{I})$	$\mathcal{N}(\mathbf{0}, \mathbf{I})$
Loss	$\mathbb{E}_i\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\left\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_i, \sigma_i) + \frac{\boldsymbol{\epsilon}}{\sigma_i}\right\|_2^2\right]$	$\mathbb{E}_i\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\|\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_i, i) - \boldsymbol{\epsilon}\|_2^2\right]$
Sampling	Apply Langevin per layer; use output to initialize the next	Traversing the Markovian chain with $p_{\boldsymbol{\phi}^\times}(\mathbf{x}_{i-1}\|\mathbf{x}_i)$

一个共同的瓶颈。 尽管 NCSN 和 DDPM 的公式不同，但两者都依赖于稠密的时间离散化。这导致了一个关键限制：采样通常需要数百甚至上千次迭代，使得生成过程缓慢且计算开销大。

Question 3.5.1

如何加速扩散模型中的采样？

我们将将在 Chapter 9 和 Chapter 10 中重新讨论这一挑战。

3.6 闭幕词

本章描绘了扩散模型的第二条主要路径，其起点是基于能量的模型（EBMs）的得分视角。我们首先指出了 EBMs 的核心挑战——难以处理的配分函数，并引入了评分函数 $\nabla_{\mathbf{x}} \log p(\mathbf{x})$，作为一个强大的工具，完全规避了这一问题。

我们的研究历程从经典的分数匹配发展到其更可扩展且更稳健的变体——降噪分数匹配（DSM）。通过 DSM，我们看到，通过对数据添加噪声，可以构建一个易处理的训练目标，再次利用条件化策略来生成简单的回归目标。此外，我们建立起了分数估计与去噪行为之间深刻的联系，这一联系通过特威迪公式得以揭示：该公式表明，分数提供了从噪声观测中估计干净信号所需的精确方向。

该原则随后从单一噪声水平扩展到连续的噪声条件得分网络（NCSN），后者学习一个在多个噪声尺度上条件化的统一得分模型，并通过退火的 Langevin 动力学生成样本。在探索的最后，我们发现，尽管 NCSN 与变分视角下的 DDPM 起源不同，但它们具有惊人的相似结构和共同瓶颈：缓慢的顺序采样。

这种收敛并非偶然；它暗示着更深层次的统一数学结构。这些离散时间模型的局限性促使我们寻求一个更为通用的框架。在下一章中，我们将迈出这关键的一步：

1. 我们将转入连续时间视角，展示 DDPMs 和 NCSNs 可以优雅地统一为由随机微分方程（SDE）描述的单一强大过程的不同离散化形式。

2. 该得分 SDE 框架将正式连接变分观点与得分观点，将生成问题重新表述为求解微分方程的问题。

这一统一的视角不仅将提供深刻的理论洞察，还将开启一类先进的数值方法，以应对采样速度缓慢的根本性挑战。

4

扩散模型的现状：得分 SDE 框架

> 描述自然规律的唯一精确方法就是使用微分方程。它们的优点在于根本性，而且据我们所知，它们是精确的。
>
> 理查德 · P · 费曼

到目前为止，我们从两个角度研究了扩散模型：变分视角和基于得分的视角，后者自然地源自能量基模型（EBM）的表述。我们现在迈出了下一步，进入连续时间框架。其核心是得分 *SDE*，即 DDPM 与 NCSN 统一于单一公式的连续极限。这一视角之所以强大，是因为它通过微分方程（DE）提供了清晰且具有理论基础的描述，从而扩展了离散更新过程。

在此视角下，生成过程简化为求解一个随时间演化的微分方程。这使我们能够直接应用数值分析中的工具：例如，基本的欧拉方法可用于模拟动态过程，而更高级的求解器则能提升稳定性和效率。

通过在连续时间中工作，我们还获得了更丰富的数学结构，并为理解、分析和改进扩散模型奠定了统一的基础。本专著将进一步发展这一视角。

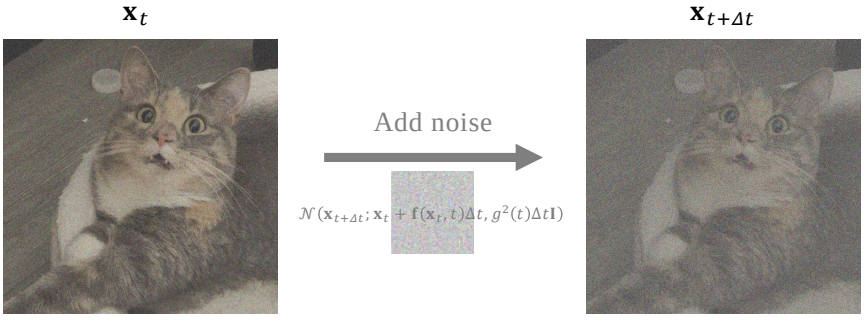


图 4.1: 离散时间加噪步骤的示意图。它从 t 到 $t+\Delta t$ 添加均值漂移为 $\mathbf{f}(\mathbf{x}_t, t)$ 、扩散系数为 $g(t)$ 的噪声。

4.1 得分 SDE：其原理

使用多种噪声尺度是 NCSN 和 DDPM 框架成功的关键因素。在本节中，我们介绍 *Score SDE* (**song2020score**) 的基础，该方法通过考虑连续的噪声水平来提升这一思想。前向和反向扩散过程的连续时间极限早已被**sohl2015deep**指出，但**song2020score**将其观点置于核心位置，将数据演化表述为随机/常微分方程，其中噪声水平随时间平滑增加。这种连续时间表述不仅统一了先前的离散时间模型，还通过将生成式建模问题转化为求解微分方程，为生成式建模提供了原理严谨且灵活的基础。

4.1.1 动机：从离散时间到连续时间过程

我们重新审视 NCSN 和 DDPM 的前向噪声注入方案。NCSN 使用一系列递增的噪声水平 $\{\sigma_i\}_{i=1}^{L}$。每个干净样本 $\mathbf{x} \sim p_{\text{data}}$ 被扰动为

$$\mathbf{x}_{\sigma_i} = \mathbf{x} + \sigma_i \boldsymbol{\epsilon}_i, \qquad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

DDPM 通过一个方差调度 $\{\beta_i\}_{i=1}^{L}$ 逐步注入噪声：

$$\mathbf{x}_i = \sqrt{1-\beta_i^2}\mathbf{x}_{i-1} + \beta_i \boldsymbol{\epsilon}_i, \qquad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

我们将它们共同视为在一个离散时间网格上，从 $\mathbf{x}_t$ 到 $\mathbf{x}_{t+\Delta t}$ 的顺序更新形

式为[1]：

$$\textbf{NCSN:}\quad \mathbf{x}_{t+\Delta t}=\mathbf{x}_t+\sqrt{\sigma_{t+\Delta t}^2-\sigma_t^2}\boldsymbol{\epsilon}_t \qquad \approx \mathbf{x}_t+\sqrt{\frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t}\Delta t}\boldsymbol{\epsilon}_t$$

$$\textbf{DDPM:}\quad \mathbf{x}_{t+\Delta t}=\sqrt{1-\beta_t}\mathbf{x}_t+\sqrt{\beta_t}\boldsymbol{\epsilon}_t \qquad \approx \mathbf{x}_t-\frac{1}{2}\beta_t\mathbf{x}_t\Delta t+\sqrt{\beta_t\Delta t}\boldsymbol{\epsilon}_t,$$

其中 $\boldsymbol{\epsilon}_t\sim\mathcal{N}(\mathbf{0},\mathbf{I})$。有趣的是，两种噪声注入过程遵循一种共同的结构模式：

$$\mathbf{x}_{t+\Delta t}\approx\mathbf{x}_t+\mathbf{f}(\mathbf{x}_t,t)\Delta t+g(t)\sqrt{\Delta t}\boldsymbol{\epsilon}_t, \tag{4.1.1}$$

其中 $\mathbf{f}:\mathbb{R}^D\times\mathbb{R}\to\mathbb{R}^D$ 和 $g:\mathbb{R}\to\mathbb{R}$ 定义为：

$$\textbf{NCSN:}\quad \mathbf{f}(\mathbf{x},t)=0, \qquad g(t)=\sqrt{\frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t}}$$

$$\textbf{DDPM:}\quad \mathbf{f}(\mathbf{x},t)=-\frac{1}{2}\beta(t)\mathbf{x}, \qquad g(t)=\sqrt{\beta(t)}.$$

该公式对应以下高斯转移：

$$p(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t):=\mathcal{N}\left(\mathbf{x}_{t+\Delta t};\mathbf{x}_t+\mathbf{f}(\mathbf{x}_t,t)\Delta t,g^2(t)\Delta t\mathbf{I}\right), \tag{4.1.2}$$

其中，为了略微滥用符号起见，我们将 $\mathbf{x}_t$ 视为一个固定样本，将 $\mathbf{x}_{t+\Delta t}$ 视为一个随机变量。

当 $\Delta t\to 0$ （可从概念上理解为准备 无限多层噪声）时，离散时间过程收敛到一个随时间正向演化的连续时间 SDE[2]：

$$\mathrm{d}\mathbf{x}(t)=\mathbf{f}(\mathbf{x}(t),t)\,\mathrm{d}t+g(t)\,\mathrm{d}\mathbf{w}(t),$$

其中 $\mathbf{w}(t)$ 为标准维纳过程（或布朗运动）。

Remark.

> 虽然这里不需要完整的形式化定义，但维纳过程是一种连续时间的随机过程 $\mathbf{w}(t)$，它从零开始，具有独立增量，并且满足对于任意 $s<t$，增量 $\mathbf{w}(t)-\mathbf{w}(s)$

[1] 为方便起见，我们互换使用 $\mathbf{x}(t)$ 和 $\mathbf{x}_t$ （以及其他时间相关变量亦如此）来表示时间 t 时的样本。

[2] Equation (4.1.2) 中的前向核收敛，如 $\Delta t\to 0$ 所示，趋于相应伊藤随机微分方程的解。一个完全严格的证明依赖于高级结果，我们将其留待文献中讨论。

服从均值为零、方差为 $t-s$ 的正态分布。它表示随时间累积的独立高斯波动，尽管几乎必然连续，但却处处不可微。在无穷小的时间间隔 $[t, t+\mathrm{d}t]$ 内，维纳过程的增量定义为

$$\mathrm{d}\mathbf{w}(t) := \mathbf{w}(t+\mathrm{d}t) - \mathbf{w}(t),$$

该增量被建模为一个均值为零、方差为 $\mathrm{d}t$ 的高斯随机变量：

$$\mathrm{d}\mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \mathrm{d}t\mathbf{I}).$$

关于随机微分方程（SDEs）基础的简要介绍见 Section A.2，更深入的讨论见 Chapter C。然而，我们可以从概念上理解离散与连续表述之间的联系如下：

- $\mathbf{x}(t+\Delta t) - \mathbf{x}(t) \approx \mathrm{d}\mathbf{x}(t)$,
- $\Delta t \approx \mathrm{d}t$,
- $\sqrt{\Delta t}\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \Delta t\mathbf{I}) \approx \mathrm{d}\mathbf{w}(t)$.

一旦指定漂移 $\mathbf{f}(\mathbf{x}, t)$ 和扩散 $g(t)$，前向时间 SDE 会自动诱导出一个反向时间 SDE，该 SDE 将终态噪声分布逆向传输至数据分布。反向动力学仅涉及一个未知项，即每个连续时间层级的 *评分函数*。这表明分数匹配是训练目标；一旦评分函数被学成，采样过程就等价于使用学成的评分函数对反向时间 SDE 进行数值积分。

虽然 Section 4.2 提供了实际实现，但我们首先考察 Section 4.1.2 和 Section 4.1.3 中前向过程和反向过程的理论基础。

4.1.2 前向时间 SDE：从数据到噪声

在此公式化下，基于离散时间的早期方法，如 NCSN (**song2019generative**) 和 DDPM (**sohl2015deep**; **ho2020denoising**)，可以通过定义在区间 $[0, T]$ 上的前向 SDE 所控制的随机过程 $\mathbf{x}(t)$ 统一到 *连续时间框架* 中：

$$\mathrm{d}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}(t), \quad \mathbf{x}(0) \sim p_{\text{data}}. \tag{4.1.3}$$

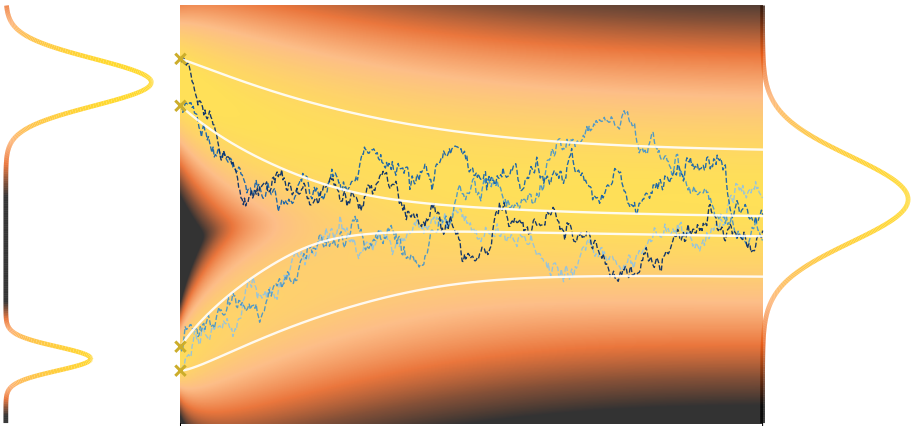


图 4.2: (1D) 扩散模型前向过程的可视化。该过程从从一个复杂的双峰数据分布（$p_0 = p_{\text{data}}$）中采样的初始点（记为“×”）开始，逐渐演化至一个简单的单峰值高斯先验（$p_T \approx p_{\text{prior}}$）。背景热力图展示了随时间演化的边缘概率密度 p_t，其逐渐变得平滑。样本轨迹从 $t = 0$ 演化到 $t = T$，对比了随机的前向 SDE 过程（蓝色路径）与其确定性对应版本——PF-ODE（白色路径）。需要注意的是，PF-ODE 是一种用于密度的确定性传输映射，并非从单一起点出发的样本路径均值。

其中，$\mathbf{f}(\cdot, t)\colon \mathbb{R}^D \to \mathbb{R}^D$ 为漂移项，$g(t) \in \mathbb{R}$ 为标量扩散系数，$\mathbf{w}(t)$ 表示标准维纳过程。我们将其称为 *前向 SDE*，它描述了干净数据如何随时间逐渐被扰动为噪声。

一旦指定了漂移项 $\mathbf{f}$ 和扩散系数 g，前向过程即被完全确定，描述了数据变量如何通过注入高斯噪声逐步被破坏。特别是，会诱导出两类随时间变化的概率密度：

扰动内核 条件律

$$p_t(\mathbf{x}_t|\mathbf{x}_0)$$

描述了干净的数据样本 $\mathbf{x}_0 \sim p_{\text{data}}$ 如何在时间 t 演化为其噪声版本 $\mathbf{x}_t$。一般来说，Equation (4.1.3) 中的漂移项 $\mathbf{f}(\mathbf{x}, t)$ 可以是 $\mathbf{x}$ 的任意函数，但一个常见且解析上方便的选择是假设其为仿射形式：

$$\mathbf{f}(\mathbf{x}, t) = f(t)\mathbf{x}, \tag{4.1.4}$$

其中 $f(t)$ 是 t 的标量函数，通常取非正值。在此结构下，过程在每个时刻仍保持高斯分布，且条件分布可通过求解相应的均值-方差常微分方程组 (**sarkka2019applied**)(see

also Section 4.3.3) 得到闭式解。特别是，

$$p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\big(\mathbf{x}_t; \mathbf{m}(t), P(t)\mathbf{I}_D\big),$$

与

$$\mathbf{m}(t) = \exp\Big(\int_0^t f(u)\,\mathrm{d}u\Big)\mathbf{x}_0, \quad P(t) = \int_0^t \exp\Big(2\int_s^t f(u)\,\mathrm{d}u\Big)g^2(s)\,\mathrm{d}s,$$

以及初值 $\mathbf{m}(0) = \mathbf{x}_0$ ， $P(0) = 0$ 。

这种显式形式允许直接从 $\mathbf{x}_t$ 采样 $\mathbf{x}_0$，而无需对 SDE 进行数值积分，因此称为 *仿真自由*。NCSN 和 DDPM 都属于这种仿射漂移情景。

在余下部分，我们将为任意漂移 $\mathbf{f}(\mathbf{x}, t)$ 建立一般理论，但当闭式分析有用时，将回到仿射漂移的情形。

边缘密度。 时间边缘密度 $p_t(\mathbf{x}_t)$ 通过对接扰动核积分得到：

$$p_t(\mathbf{x}_t) := \int p_t(\mathbf{x}_t|\mathbf{x}_0)p_{\text{data}}(\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_0, \quad \text{with } p_0 = p_{\text{data}}. \tag{4.1.5}$$

通过适当选择系数 $f(t)$ 和 $g(t)$ ，前向过程逐步添加噪声，直到初始状态的影响被有效遗忘。当 T 变得较大时，条件分布 $p_T(\mathbf{x}_T|\mathbf{x}_0)$ 不再依赖于 $\mathbf{x}_0$ ，因为其均值演化为

$$\mathbf{m}(T) = \exp\Big(\int_0^T f(u)\,\mathrm{d}u\Big)\mathbf{x}_0 \longrightarrow \mathbf{0}, \quad \text{当} T \to \infty,$$

前提是 $f(u)$ 非正，使得指数因子衰减。与此同时，方差不断增长并稳定下来，以匹配选定的先验分布。因此，边际

$$p_T(\mathbf{x}_T) = \int p_T(\mathbf{x}_T|\mathbf{x}_0)p_{\text{data}}(\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_0,$$

其最初表示数据样本上的复杂混合分布，收敛到一个简单的先验 p_{prior} ，通常为高斯分布。在此极限下，

$$p_T(\mathbf{x}_T) \approx p_{\text{prior}}(\mathbf{x}_T) \quad \text{and} \quad p_T(\mathbf{x}_T|\mathbf{x}_0) \approx p_{\text{prior}}(\mathbf{x}_T),$$

因此，前向过程将任意数据分布映射到一个易处理的先验分布，为反向过程和生成提供了一个方便的起点。

4.1.3 生成用的反向时间随机过程

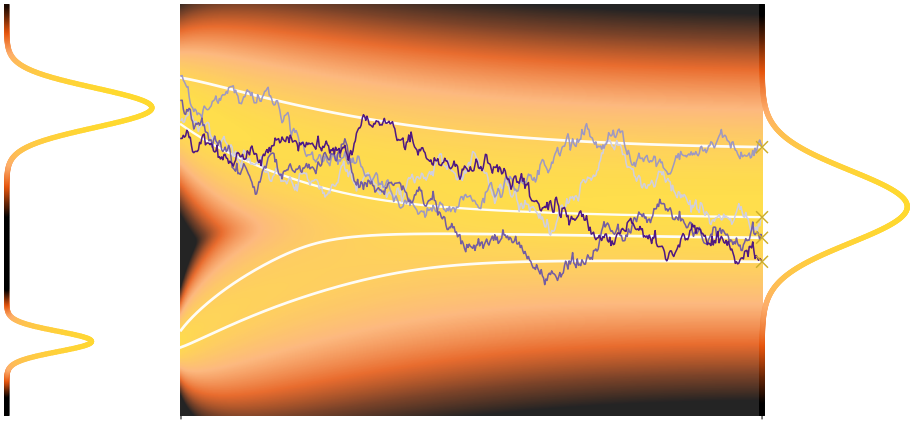


图 4.3: 数据生成的反向时间随机过程的可视化。它从在 $t = T$ 时刻从一个简单的先验分布（p_{prior}）中抽取的样本（记为“×”）开始，这些样本通过反向 SDE 沿时间反向演化。最终的轨迹在 $t = 0$ 处终止，并共同构成目标双峰数据分布（$p_0 = p_{\text{data}}$）。背景热力图展示了概率密度如何从简单的高斯分布逐步变换为目标复杂分布的过程。

直观上，从噪声生成数据可以通过“反向”前向过程来实现：从先验分布中随机采样一个点，并将其沿时间反向演化，从而得到生成的样本。对于确定性系统（即常微分方程），这一想法可以自然地实现。由于不涉及随机性，时间反向即意味着沿着与前向过程相同的路径，以相反方向追踪该点的轨迹。[3]. 相比之下，随机微分方程在每个时间步都包含随机性，这意味着一个点可能沿着许多合理的随机轨迹演化。因此，逆转此类过程更为微妙。[4].

虽然单个随机轨迹不具备可逆性，但一个重要的洞察是，这些轨迹上的分布是可以被反转的。这由**anderson1982reverse**的一个基础结果所形式化，该结果表明时间反演过程 $\{\bar{\mathbf{x}}(t)\}_{t\in[0,T]}$ [5] 前向过程 Equation (4.1.3) 本身由一个明确的随机微分方程（SDE）所支配。该逆向过程从 T 演化到 0，其动力学由以下公式给出：

[3]从技术上讲，这对应于通过时间反转置换 $t \leftrightarrow T - t$ 求解常微分方程。

[4]天真地反转时间并不能得到正确的逆过程。

[5]我们使用“bar”记号来区分反向过程 $\{\bar{\mathbf{x}}(t)\}_{t\in[0,T]}$ 与由前向时间 SDE 定义的前向过程 $\{\mathbf{x}(t)\}_{t\in[0,T]}$。

$$\mathrm{d}\bar{\mathbf{x}}(t) = \big[\mathbf{f}(\bar{\mathbf{x}}(t), t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\bar{\mathbf{x}}(t))\big]\,\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}(t), \qquad (4.1.6)$$
$$\bar{\mathbf{x}}(T) \sim p_{\text{prior}} \approx p_T.$$

此处，$\bar{\mathbf{w}}(t)$ 表示一个反向时间的标准维纳过程，定义为 $\bar{\mathbf{w}}(t) := \mathbf{w}(T-t) - \mathbf{w}(T)$ 。

为了直观理解 Equation (4.1.6)，我们在 Section 4.1.6 中给出一个具体例子，其中数据分布为高斯分布，动态过程为线性的-高斯型。该情景是易处理的：仅通过基本微积分和线性代数即可直接推导出时间反演公式，而无需引入 **anderson1982reverse** 的完整一般理论。

注意到随机性 ($g \neq 0$) 的存在引入了一个额外的修正项 $-g^2(t)\nabla_{\mathbf{x}} \log p_t(\bar{\mathbf{x}}(t))$ ，该修正项考虑了扩散效应，并确保反向演化正确地重现了前向 SDE 所诱导的边缘分布的演化（见 Section 4.1.5）。

从概念上讲，为什么逆过程有效？ Section 4.5.2 通过将反向时间 SDE 与 DDPM 变分框架联系起来，给出了一个直观的推导过程（可选但富有洞察力）。在此，我们提供另一种补充性的直觉，说明反向时间动态如何从噪声中恢复出结构化数据。

乍看之下，反向时间过程中存在布朗噪声似乎显得自相矛盾。如果前向扩散过程将数据逐渐扩散到越来越嘈杂的配置中，那么如何通过一个引入额外随机性的反向过程——特别是通过 $\bar{\mathbf{w}}(t)$ 引入的随机性——产生集中在数据流形附近的干净、结构化的样本，这一点尚不明确。关键在于，反向时间随机微分方程（SDE）并不会注入任意的随机性。扩散项 $g(t)\,\mathrm{d}\bar{\mathbf{w}}(t)$ 始终与由得分驱动的漂移项 $-g^2(t)\nabla_{\mathbf{x}} \log p_t(\bar{\mathbf{x}}(t))$ 相耦合。这两项共同作用，彼此平衡：得分引导轨迹向高密度区域移动，而噪声则引入受控的随机性，使得探索得以进行，同时又不会淹没系统动力学。

为了更清晰地理解这一点，回到 Equation (3.1.6) 中的朗之万直觉。当 $f(t) \equiv 0$ 时，Equation (4.1.6) 变为

$$\mathrm{d}\bar{\mathbf{x}}(t) = -g^2(t)\nabla_{\mathbf{x}} \log p_t\big(\bar{\mathbf{x}}(t)\big)\,\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}(t).$$

通过 $s := T - t$ 正向重参数化时间（因此 $\mathrm{d}t = -\mathrm{d}s$ ），并按分布重命名布朗运

动使得 $\mathrm{d}\bar{\mathbf{w}}(t) = -\,\mathrm{d}\mathbf{w}_s$ 。记 $\bar{\mathbf{x}}_s := \bar{\mathbf{x}}(T-s)$ 和 $\pi_s := p_{T-s}$ ，则有

$$\begin{aligned}\mathrm{d}\bar{\mathbf{x}}_s &= g^2(T-s)\nabla \log \pi_s\big(\bar{\mathbf{x}}_s\big)\,\mathrm{d}s + g(T-s)\,\mathrm{d}\mathbf{w}_s \\ &= 2\tau(s)\nabla \log \pi_s\big(\bar{\mathbf{x}}_s\big)\,\mathrm{d}s + \sqrt{2\tau(s)}\,\mathrm{d}\mathbf{w}_s, \quad \tau(s) := \tfrac{1}{2}g^2(T-s).\end{aligned}$$

这具有随时间变化的温度 $\tau(s)$ 的 Langevin 形式，目标是演化密度 π_s 。根据 Tweedie 公式 (Equation (3.3.6))，得分方向 $\nabla \log \pi_s$ 在每个时间切片上指向条件干净信号，因此漂移项持续“拉回”去噪后的结构。

关键的是，$g(t)$ 沿反向轨迹进行 退火。早期（$s \approx 0$ ，即 $t \approx T$ ），$g(T-s)$ 通常较大，因此注入的噪声较强，过程广泛探索。随着 s 增大，$g(T-s)$ 减小，随机项减弱，得分项占主导，将样本拉入 π_s 的高密度区域；到 $s = T$ （即 $t = 0$ ）时，轨迹集中于数据流形附近。

逆时序随机微分方程功能概述。 令人着迷的是，随时间变化的评分函数

$$\mathbf{s}(\mathbf{x}, t) := \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$$

自然地出现在 Equation (4.1.6) 中。一旦前向系数 $f(t)$ 和 $g(t)$ 被指定，得分便是反向动力学中唯一的未知量。这突显了其核心作用：掌握得分后，反向过程即被确定，采样等价于使用学成的得分对 Equation (4.1.6) 进行数值积分。

由于最优得分通常没有闭式表达式，我们采用 Chapter 3 的方法，并训练一个神经网络 $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}, t)$ 通过分数匹配来近似该得分；详见 Section 4.2.1。将 $\mathbf{s}(\mathbf{x}, t)$ 替换为 $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}, t)$ 后代入 Equation (4.1.6)，即可完全确定反向动态过程。

生成对应于从 $t = T$ 开始，反向求解逆时序 SDE，从 $\mathbf{x}_T \sim p_{\text{prior}}$ 开始，到 $t = 0$ 结束。重要的是，**anderson1982reverse** 证明了前向过程与逆向过程的边缘密度一致，确保当 $p_{\text{prior}} \approx p_T$ 时，$t = 0$ 处的样本近似服从 p_{data}。我们将在 Section 4.2.2 中进一步探讨这一点。

4.1.4 生成的确定性过程（概率流 ODE）

尽管 Equation (4.1.6) 中的随机微分方程引入了随机性，并可能提高生成样本的多样性，但一个问题随之产生：

Question 4.1.1

是否必须使用 *Equation* (4.1.6) 中的随机微分方程进行采样？

受 **maoutsa2020interacting** 启发，**song2020score** 也引入了一个确定性过程，即一个常微分方程（ODE），该过程演化样本时保持与前向随机微分方程（SDE）相同的边缘分布。该过程 $\{\tilde{\mathbf{x}}(t)\}_{t\in[0,T]}$ [6]，称为*概率流常微分方程*（*PF-ODE*），其表达式为：

$$\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{x}}(t) = \mathbf{f}(\tilde{\mathbf{x}}(t), t) - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}}\log p_t(\tilde{\mathbf{x}}(t)). \tag{4.1.7}$$

与 SDE 情况类似，可以用学成的近似得分代替真实得分，并从 $t = T$ 积分到 $t = 0$ 的反向时间 ODE 以生成样本。具体而言，生成的样本（PF-ODE 在时间 $t = 0$ 的解）具有如下形式

$$\tilde{\mathbf{x}}(T) + \int_T^0 \left[\mathbf{f}(\tilde{\mathbf{x}}(\tau), \tau) - \frac{1}{2}g^2(\tau)\nabla_{\mathbf{x}}\log p_\tau(\tilde{\mathbf{x}}(\tau))\right]\mathrm{d}\tau,$$

其中初始条件为 $\tilde{\mathbf{x}}(T) \sim p_{\text{prior}}$。由于该积分在闭式解下难以处理，实际生成依赖于数值求解器（例如欧拉法，见 Equation (4.2.4)）。

与反向时间 SDE 相比，PF-ODE 具有两个关键优势：

- 常微分方程可以朝任一方向积分，从 $t = 0$ 到 $t = T$ 或从 $t = T$ 到 $t = 0$，使用相同的方程形式，只需在所选端点处指定相应的初值条件即可。这种双向性与随机微分方程不同，后者通常仅允许向前时间积分。

- 它得益于为常微分方程开发的广泛且成熟的现成数值求解器。

我们强调，PF-ODE 并非通过简单地移除 Equation (4.1.6) 中的扩散项得到的；值得注意的是，其漂移项中的 $\frac{1}{2}$ 因子具有合理的来源。从高层次上看，Equation (4.1.7) 是通过选择一个常微分方程的漂移项，使得其演化过程保持与 Equation (4.1.3) 中前向随机微分方程相同的边缘分布密度而产生的。确保这种边缘分布对齐的潜在原理（即福克-普朗克方程 (**oksendal2003stochastic**)）将在下一节中详细讨论。

[6] 我们使用波浪号来区分与前向和反向时间 SDE 相关的过程。今后为简便起见，我们将省略这种记法区别。

4.1.5 前向/后向时间 SDE 与 PF-ODE 中边缘分布的匹配

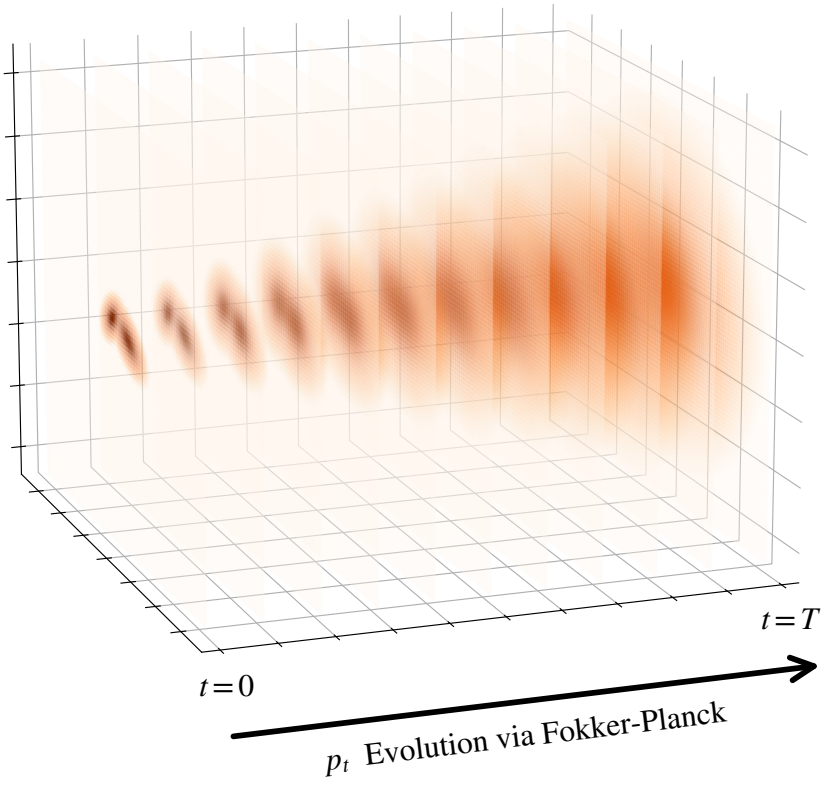


图 4.4: (2D) 边缘密度 p_t 的时间演化。前向 SDE 在 $[0,T]$ 上具有 $\mathbf{f} \equiv \mathbf{0}$ 与 $g(t) = \sqrt{2t}$ 。其起始于 $p_0 = p_{\text{data}}$ 的双模高斯混合分布，最终演变为 $p_T \approx p_{\text{prior}} := \mathcal{N}(\mathbf{0}, T^2\mathbf{I})$ 。p_t 的时空演化遵循福克-普朗克方程。

福克-普朗克方程以确保边缘概率密度的一致性。 扩散模型中的一个核心概念是，不同的过程可能导致相同的边缘分布序列（我们将在本小节后面加以说明）。目标是构建一个过程，通过在时间上对齐边缘分布，将 p_{prior} 转变为 p_{data} ，特别是确保在 $t = 0$ 处的边缘分布一致。该过程的具体形式次之，只要其易处理且支持高效采样即可。这自然引出了一个基本问题：

Question 4.1.2

如何确保不同过程产生相同的边缘分布？

回到我们的设定，一旦前向 SDE 被指定，它就定义了从 p_{data} 到 p_{prior} 的边缘分布的演化。随后构造反向时间 SDE 和 PF-ODE，使得它们的轨迹所生成的边缘分布恰好与前向过程的边缘分布相匹配。这种对应关系的关键在于福克-普朗克方程，该方程控制着扩散过程中边缘分布的演化。以下定理 (**anderson1982reverse**; **song2020score**) 建立了这一联系的基础：

Theorem 4.1.1: Fokker–Planck 方程确保边缘分布对齐

设 $\{\mathbf{x}(t)\}_{t\in[0,T]}$ 按如下前向 SDE 演化：

$$\mathrm{d}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}(t),$$

初始条件为 $\mathbf{x}(0) \sim p_0 = p_{\text{data}}$ 。则其边缘密度 p_t 满足 Fokker–Planck 方程

$$\begin{aligned}\partial_t p_t(\mathbf{x}) &= -\nabla_{\mathbf{x}} \cdot \big[\mathbf{f}(\mathbf{x}, t)p_t(\mathbf{x})\big] + \tfrac{1}{2}g^2(t)\Delta_{\mathbf{x}} p_t(\mathbf{x}) \\ &= -\nabla_{\mathbf{x}} \cdot \big[\tilde{\mathbf{f}}(\mathbf{x}, t)p_t(\mathbf{x})\big],\end{aligned} \tag{4.1.8}$$

其中 $\Delta_{\mathbf{x}}$ 表示拉普拉斯算子，且

$$\mathbf{v}(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t) - \tfrac{1}{2}g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x}).$$

此时，PF-ODE 与逆时间 SDE 均产生相同的分布族 $\{p_t\}_{t\in[0,T]}$ ，后者在逆时间中演化：

(i) *PF-ODE* $\{\tilde{\mathbf{x}}(t)\}_{t\in[0,T]}$

$$\frac{\mathrm{d}\tilde{\mathbf{x}}(t)}{\mathrm{d}t} = \mathbf{v}(\tilde{\mathbf{x}}(t), t),$$

若从 $\tilde{\mathbf{x}}(0) \sim p_0$ 开始并沿 t 正向运行，或等价地从 $\tilde{\mathbf{x}}(T) \sim p_T$ 开始并沿 t 逆向运行，则对所有 $t \in [0, T]$ 具有边缘分布 $\tilde{\mathbf{x}}(t) \sim p_t$ 。

(ii) 逆时间 *SDE* $\{\bar{\mathbf{x}}(t)\}_{t\in[0,T]}$

$$\mathrm{d}\bar{\mathbf{x}}(t) = \big[\mathbf{f}(\bar{\mathbf{x}}(t), t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\bar{\mathbf{x}}(t))\big]\,\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}(t),$$

其中 $\bar{\mathbf{x}}(0) \sim p_T$ 且 $\bar{\mathbf{w}}(t)$ 为逆时间中的标准维纳过程，具有边缘分布 $\bar{\mathbf{x}}(t) \sim p_{T-t}$ 。

Proof for Theorem.

该定理的证明见 Section D.2.5，而 Section 4.5.1 利用概率的边缘化技术对 Fokker–Planck 方程提供了进一步的直观解释。

固定边缘分布下的多重条件分布。 为了理解 PF-ODE 如何将 p_{data} 向前推进时间（或等价地，将 p_{prior} 向后推进），考虑流映射 $\boldsymbol{\Psi}_{s\to t}:\mathbb{R}^D\to\mathbb{R}^D$，其中 $\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s)$ 表示从时间 s 的初始状态 $\mathbf{x}_s$ 出发，在时间 t 处的 PF-ODE 解，对于任意时间 $s,t\in[0,T]$。换句话说，该映射将初始状态 $\mathbf{x}_s$ 直接跳转到其在 t 时的状态：

$$\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s):=\mathbf{x}_s+\int_s^t \mathbf{v}(\mathbf{x}_\tau,\tau),\mathrm{d}\tau, \tag{4.1.9}$$

速度场

$$\mathbf{v}(\mathbf{x},\tau):=\mathbf{f}(\mathbf{x},\tau)-\frac{1}{2}g^2(\tau)\nabla_{\mathbf{x}}\log p_\tau(\mathbf{x}).$$

此处，积分捕捉了沿 PF-ODE 轨迹 $\mathbf{x}_\tau$ 累积的净位移。在对 $\mathbf{v}$ 作出适度光滑性假设下，流映射 $\boldsymbol{\Psi}_{s\to t}:\mathbb{R}^D\to\mathbb{R}^D$ 为光滑双射。[7].

对于任意 $t\in[0,T]$，前推密度定义为

$$p_t^{\text{fwd}}(\mathbf{x}_t):=\int \delta(\mathbf{x}_t-\boldsymbol{\Psi}_{t\to0}(\mathbf{x}_0))p_{\text{data}}(\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_0,$$

记为 $\boldsymbol{\Psi}_{t\to0}\#p_{\text{data}}$，表示在 $\boldsymbol{\Psi}_{t\to0}$ 作用下时间 t 时的分布。定理 4.1.1 保证了 $p_t^{\text{fwd}}=p_t$，其中 p_t 为前向 SDE 的边缘密度，等价于确定性 PF-ODE 与随机核：

$$p_t(\mathbf{x}_t)=\int p_t(\mathbf{x}_t|\mathbf{x}_0)p_{\text{data}}(\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_0=\int \delta(\mathbf{x}_t-\boldsymbol{\Psi}_{t\to0}(\mathbf{x}_0))p_{\text{data}}(\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_0.$$

这意味着无穷多个条件语句 $Q_t(\mathbf{x}_t|\mathbf{x}_0)$ 会产生相同的 $p_t(\mathbf{x}_t)$，例如：

- **随机（无需仿真）**：$Q_t(\mathbf{x}_t|\mathbf{x}_0)=p_t(\mathbf{x}_t|\mathbf{x}_0)$，
- **确定性（需要求解常微分方程）**：$Q_t(\mathbf{x}_t|\mathbf{x}_0)=\delta\big(\mathbf{x}_t-\boldsymbol{\Psi}_{t\to0}(\mathbf{x}_0)\big)$，
- **混合物**：$Q_t(\mathbf{x}_t|\mathbf{x}_0)=\lambda p_t(\mathbf{x}_t|\mathbf{x}_0)+(1-\lambda)\delta(\mathbf{x}_t-\boldsymbol{\Psi}_{t\to0}(\mathbf{x}_0))$，$\lambda\in[0,1]$.

$Q_t(\mathbf{x}_t|\mathbf{x}_0)$ 的非唯一性源于边际约束无法唯一确定条件分布这一事实。该概念在 Section 5.2.2 和 Section 9.2.3 中再次出现。特别是，存在一整个族反向时间 SDE，它们与相同的边际 p_t 一致。

[7]提示：PF-ODE 流映射 $\boldsymbol{\Psi}_{s\to t}$ 恰好是将 p_s 映射到 p_t 的正规化流（NF）双射（将在 Section 5.1 中详细说明）。二者区别在于，PF-ODE 固定了由 SDE 的福克-普朗克动力学决定的唯一向量场，而 NF（或连续时间 NF）则参数化该向量场，但依赖于相同的变量变换原理。

Observation 4.1.1: Matching Prescribed Marginal Densities

Multiple processes can give rise to the same sequence of marginal densities; what truly matters is satisfying the Fokker–Planck equation. This fundamental insight affords us remarkable flexibility in designing generative processes that transition from p_{prior} to p_{data}, or vice versa.

Fokker–Planck 方程是扩散模型的核心，其基础在于概率密度的 变量变换公式（参见 Chapter B 的系统性讨论）。这一原理远非次要的技术细节，而贯穿于我们整个推导过程，尤其在 Section 5.2 中尤为显著。

4.1.6 一个可计算的例子：高斯动力学的演化

当 p_{data} 为正态分布（或高斯分布的混合）时，评分函数具有闭式表达式。这使其成为构建扩散过程直观理解的理想情景：我们仅需使用基本微积分即可显式推导反向时间 SDE 和 PF-ODE，而无需依赖高级数学工具。在本小节中，我们将展示在这种易处理的情况下这些方程的行为。

带有高斯分布的反向时间 SDE 的精确计算。 当 p_{data} 为高斯分布时，公式 Equation (4.1.6) 可以直接推导得出，无需依赖 **anderson1982reverse** 的一般理论和证明。为说明核心思想，我们考虑一维情形；推广到高维的情形亦同理。

从前向 SDE 开始

$$\mathrm{d}x(t) = f(t)x(t)\,\mathrm{d}t + g(t)\,\mathrm{d}w_t,$$

并进行一次大小为 $\Delta t > 0$ 的小欧拉步进：

$$x_{t+\Delta t} = ax_t + r\epsilon,$$

其中 $a := 1 + f(t)\Delta t$ ， $r := g(t)\sqrt{\Delta t}$ ，以及 $\epsilon \sim \mathcal{N}(0,1)$ 。等价地，前向一步转移核为高斯分布：

$$x_{t+\Delta t}|x_t \sim \mathcal{N}\big(ax_t, r^2\big).$$

由于 p_{data} 被假定为高斯分布，时间 t 时的边缘分布也是高斯分布，其形式如下：

$$x_t \sim \mathcal{N}(m_t, s_t^2),$$

对于某些标量 m_t 和 s_t。因此，条件化相当于将两个高斯分布相乘并重新归一化。这使得代数运算保持简单。

根据贝叶斯法则，条件密度（忽略常数因子）等于先验与转移核的乘积：

$$\begin{aligned} p(x_t|x_{t+\Delta t}) &\propto p(x_{t+\Delta t}|x_t)p_t(\mathbf{x}_t) \\ &\propto \exp\Big(-\frac{(x_t-m_t)^2}{2s_t^2}\Big)\exp\Big(-\frac{(x_{t+\Delta t}-ax_t)^2}{2r^2}\Big). \end{aligned}$$

指数是关于 x_t 的二次式。展开两个平方项并合并同类项后，恰好显示出哪些系数是重要的：

$$-2\log p(x_t|x_{t+\Delta t}) = Ax_t^2 - 2Bx_t + \text{const},$$

与

$$A := \frac{1}{s_t^2} + \frac{a^2}{r^2}, \quad B := \frac{m_t}{s_t^2} + \frac{ax_{t+\Delta t}}{r^2}.$$

此处 A 为准确率的和（先验准确率加上通过 a 传输的转移核准确率），而 B 为相应的准确率加权的目标和。有了这些，配方平方即可在一行内得出后验：

$$Ax_t^2 - 2Bx_t = A\Big(x_t - \frac{B}{A}\Big)^2 - \frac{B^2}{A},$$

因此，条件分布是方差为 $1/A$ 、均值为 B/A 的高斯分布：

$$\mathrm{Var}(x_t|x_{t+\Delta t}) = \frac{1}{\frac{1}{s_t^2}+\frac{a^2}{r^2}}, \qquad \mathbb{E}[x_t|x_{t+\Delta t}] = \frac{\frac{m_t}{s_t^2}+\frac{ax_{t+\Delta t}}{r^2}}{\frac{1}{s_t^2}+\frac{a^2}{r^2}}.$$

这些封闭形式已经描述了任意小 Δt 下的逆向转移。为了推导出逆时间 SDE，我们现在对小 Δt 进行展开。

使用 $a = 1 + f(t)\Delta t$ 和 $r^2 = g^2(t)\Delta t$。由于 $\Delta t \to 0$，贡献 $\frac{a^2}{r^2} \sim \frac{1}{g^2(t)\Delta t}$ 主导了准确率，因此方差变为

$$\mathrm{Var}(x_t|x_{t+\Delta t}) = \left(\frac{1}{s_t^2}+\frac{a^2}{r^2}\right)^{-1} = g^2(t)\Delta t + \mathcal{O}(\Delta t^2),$$

这告诉我们反向步骤具有与前向步骤相同的扩散尺度 $g(t)$。对于均值，将比值 B/A 展开到一阶：

$$\mathbb{E}[x_t|x_{t+\Delta t}] = x_{t+\Delta t} + \Delta t\left[-\left(f(t)+\frac{g^2(t)}{s_t^2}\right)x_{t+\Delta t} + \frac{g^2(t)}{s_t^2}m_t\right] + \mathcal{O}(\Delta t^2).$$

将均值和方差结合在一起，即可得到一步逆向转移核。

$$x_t|x_{t+\Delta t} \sim \mathcal{N}\left(x_{t+\Delta t} + \Delta t\left[-\left(f + \frac{g^2}{s_t^2}\right)x_{t+\Delta t} + \frac{g^2}{s_t^2}m_t\right], g^2\Delta t\right) + \mathcal{O}(\Delta t^2).$$

这被识别为向后从 $t+\Delta t$ 到 t 的欧拉-马鲁亚玛更新：

$$x_t - x_{t+\Delta t} = \Delta t\left[-\left(f + \frac{g^2}{s_t^2}\right)x_{t+\Delta t} + \frac{g^2}{s_t^2}m_t\right] + g\sqrt{\Delta t}\epsilon + O(\Delta t^2).$$

令 $\Delta t \to 0$ 可得原始时钟下的 SDE（时间沿路径减小）

$$\mathrm{d}x(t) = \Big[-\Big(f(t) + \frac{g^2(t)}{s_t^2}\Big)x(t) + \frac{g^2(t)}{s_t^2}m_t\Big]\,\mathrm{d}t + g(t)\,\mathrm{d}\bar{w}_t.$$

这种漂移可以用得分表示，因为对于高斯边缘分布 $p_t = \mathcal{N}(m_t, s_t^2)$ ，

$$\partial_x \log p_t(x) = -\frac{x - m_t}{s_t^2} \Longrightarrow -\Big(f + \frac{g^2}{s_t^2}\Big)x + \frac{g^2}{s_t^2}m_t = -fx + g^2\partial_x \log p_t(x).$$

为了表达传统的 *forward-in-t* 反向时间参数化，定义反向过程 $\bar{x}(t) := x(T-t)$（这样我们现在在 t 中向前演化）。时间翻转将漂移项变为

$$\mathrm{d}\bar{x}(t) = \Big[f(t)\bar{x}(t) - g^2(t)\partial_x \log p_t(\bar{x}(t))\Big]\,\mathrm{d}t + g(t)\,\mathrm{d}\bar{w}_t,$$

其中 $\bar{x}(T) \sim p_{\text{prior}} \approx p_T$。这正是传统的反向时间随机微分方程。以向量形式表示，这与一般的 Equation (4.1.6) 一致，其中用 $\nabla_{\mathbf{x}} \log p_t$ 替代了一维导数。

PF–ODE 的高斯函数精确计算 当数据分布被假设为高斯分布时，我们也可以直接推导出 PF-ODE 公式，从而避免使用诸如福克-普朗克方程等复杂的数学工具。最终我们将看到，PF-ODE 的边缘密度与前向随机微分方程和反向时间随机微分方程的边缘密度一致，这为第 Section 4.1.5节将要讨论的福克-普朗克理论提供了构造性验证。

假设在时间 t 时为 $x_t \sim \mathcal{N}(m_t, s_t^2)$。大小为 Δt 的一个小的确定性步长可以写成一个光滑映射

$$x_{t+\Delta t} = \Phi_{t,\Delta t}(x_t) = x_t + \Delta t v_t(x_t) + \mathcal{O}(\Delta t^2),$$

这仅仅是关于 Δt 的一阶泰勒展开。我们的目标是观察 v_t 必须具有何种形式，才

能使得当输入为高斯分布时，输出仍保持高斯分布。

为此，将 v_t 在当前均值 m_t 附近展开：

$$v_t(x) = v_t(m_t) + v_t'(m_t)(x - m_t) + \tfrac{1}{2}v_t''(m_t)(x - m_t)^2 + \cdots .$$

现在设定 $y := x_t - m_t$ ，使得 $y \sim \mathcal{N}(0, s_t^2)$ 。接着，通过减去其均值（对 Δt 的一阶项）来使输出居中：

$$z := x_{t+\Delta t} - \mathbb{E}[x_{t+\Delta t}] = y + \Delta t\big(v_t'(m_t)y + \tfrac{1}{2}v_t''(m_t)(y^2 - s_t^2)\big) + \mathcal{O}(\Delta t^2).$$

此时，回想一下，高斯分布的偏度为零；换句话说，其三阶中心矩为零。因此，一阶近似计算 $\mathbb{E}[z^3]$ ，并利用 $\mathbb{E}[y] = 0$ 、$\mathbb{E}[y^2] = s_t^2$ 、$\mathbb{E}[y^3] = 0$ 、$\mathbb{E}[y^4] = 3s_t^4$ ，我们得到

$$\mathbb{E}[z^3] = 3\Delta t \cdot \tfrac{1}{2}v_t''(m_t)\big(\mathbb{E}[y^4] - s_t^2\mathbb{E}[y^2]\big) + \mathcal{O}(\Delta t^2) = 3\Delta t v_t''(m_t)s_t^4 + \mathcal{O}(\Delta t^2).$$

为了使输出在所有小的 Δt 下保持高斯分布，该量在阶数 Δt 处必须为零，这迫使 $v_t''(m_t) = 0$ 。对更高阶导数重复相同的论证排除了更高次幂的存在。因此，v_t 必须是线性的加上一个平移：

$$v_t(x) = a_t x + b_t.$$

将其代回步骤中，得到

$$x_{t+\Delta t} = (1 + \alpha_t \Delta t)x_t + \beta_t \Delta t + \mathcal{O}(\Delta t^2), \qquad \alpha_t := a_t, \quad \beta_t := b_t.$$

我们现在将 $x_t \sim \mathcal{N}(m_t, s_t^2)$ 通过此映射并一阶追踪均值和方差：

$$\begin{aligned}\mathbb{E}[x_{t+\Delta t}] &= m_t + \Delta t(\alpha_t m_t + \beta_t) + \mathcal{O}(\Delta t^2),\\ \mathrm{Var}(x_{t+\Delta t}) &= s_t^2 + \Delta t(2\alpha_t s_t^2) + \mathcal{O}(\Delta t^2).\end{aligned}$$

另一方面，前向 SDE $\mathrm{d}x = f(t)x\,\mathrm{d}t + g(t)\,\mathrm{d}w_t$ 具有基本的矩公式（见 Equation (4.3.3)）：

$$m_t' = f(t)m_t, \qquad (s_t^2)' = 2f(t)s_t^2 + g^2(t).$$

将 Δt 的系数进行匹配得

$$\alpha_t = f(t) + \frac{g^2(t)}{2s_t^2}, \qquad \beta_t = -\frac{g^2(t)}{2s_t^2} m_t.$$

在这些选择之下，这一步就变成了

$$x_{t+\Delta t} = x_t + \Delta t \Big[\Big(f(t) + \frac{g^2(t)}{2s_t^2} \Big) x_t - \frac{g^2(t)}{2s_t^2} m_t \Big] + \mathcal{O}(\Delta t^2).$$

由于对于高斯分布 $p_t = \mathcal{N}(m_t, s_t^2)$ 有 $\partial_x \log p_t(x) = -(x - m_t)/s_t^2$ ，我们可以将括号内改写为 $f(t)x_t - \frac{1}{2} g^2(t) \partial_x \log p_t(x_t)$ 。因此，

$$x_{t+\Delta t} = x_t + \Delta t \Big[f(t)x_t - \tfrac{1}{2} g^2(t) \partial_x \log p_t(x_t) \Big] + \mathcal{O}(\Delta t^2).$$

最后，两边同时除以 Δt 并令 $\Delta t \to 0$ ，得到 PF-ODE

$$x'(t) = f(t)x(t) - \tfrac{1}{2} g^2(t) \partial_x \log p_t\big(x(t)\big).$$

为了理解为何该常微分方程具有与前向随机微分方程（以及反向时间随机微分方程）相同的边缘分布，注意到上述漂移项为线性项加上一个平移项。因此，$x(t)$ 与 $x(0)$ 呈仿射关系，而仿射映射将高斯分布映射为高斯分布。此外，沿该常微分方程的均值 m_t 与方差 s_t^2 满足与前向随机微分方程完全相同的两个标量常微分方程（根据我们的匹配条件），且初始值相同。因此，在任意时刻 t ，$p_t = \mathcal{N}(m_t, s_t^2)$ 在两种演化过程中完全一致。

4.2 得分 SDE：其训练与采样

4.2.1 训练

基于 Chapter 3 中的哲学思想，我们使用一个时间相关的神经网络来近似最优得分 $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$

$$\mathbf{s}_{\boldsymbol{\phi}} = \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}, t)$$

在所有 $t \in [0, T]$ 上，通过最小化如 Equation (3.2.1) 所示的得分匹配目标来实现。

$$\mathcal{L}_{\mathrm{SM}}(\boldsymbol{\phi}; \omega(\cdot)) := \frac{1}{2}\mathbb{E}_{t\sim p_{\mathrm{time}}}\mathbb{E}_{\mathbf{x}_t\sim p_t}\left[\omega(t)\left\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)\right\|_2^2\right],$$

其中 p_{time} 为某种时间分布（例如在 $[0, T]$ 上的均匀分布），$\omega(\cdot)$ 为时间权重函数。

为了避免依赖难以处理的预言机得分 $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$，采用 Equation (3.3.2) 中的 DSM 损失。在给定数据点 $\mathbf{x}_0$ 的条件下，该方法可通过 Equation (D.2.4) 利用解析上易处理的得分 $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)$，具体例子见 Section 4.3。具体而言，我们利用以下损失函数：

$$\begin{aligned}&\mathcal{L}_{\mathrm{DSM}}(\boldsymbol{\phi}; \omega(\cdot)) := \\ &\frac{1}{2}\mathbb{E}_t\mathbb{E}_{\mathbf{x}_0}\mathbb{E}_{p_t(\mathbf{x}_t|\mathbf{x}_0)}\left[\omega(t)\left\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)\right\|_2^2\right],\end{aligned} \tag{4.2.1}$$

其中 $\mathbf{x}_0 \sim p_{\mathrm{data}}$. Equation (4.2.1) 可被理解为 Equation (3.4.1) 的连续时间对应形式，离散情况下求和被积分所替代。

与定理 3.3.1中的结果类似，Equation (4.2.1) 的极小化元唯一地确定如下：

Proposition 4.2.1: DSM 最小化器

最小化器 $\mathbf{s}^*$ 满足

$$\mathbf{s}^*(\mathbf{x}_t, t) = \mathbb{E}_{\mathbf{x}_0\sim p(\mathbf{x}_0|\mathbf{x}_t)}\left[\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)\right] = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t), \tag{4.2.2}$$

对几乎每个 $\mathbf{x}_t \sim p_t$ 和 $t \in [0, T]$ 成立。

Proof for Proposition.

DSM 目标可以理解为一个最小二乘误差问题。具体而言，在每个时间点 t，最优评分函数由对数条件密度梯度的条件期望给出，根据贝叶斯规则，这等价于对数边缘密度的梯度。详细证明参见附录 D.2.6。

4.2.2 采样与推断

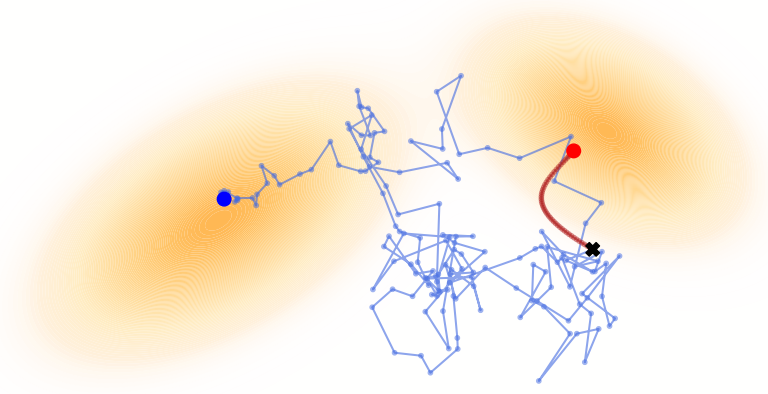


图 4.5: (2D) 从得分 SDE 中采样的示意图。采样通过求解反向时间 SDE（蓝色；通过 Equation (4.2.4)）和 PF-ODE（红色；通过 Equation (4.2.6)），针对与 Figure 4.4 中相同的前向 SDE 设置。从一个随机点 $\mathbf{x}_T \sim p_{\text{prior}}$（深“×”）出发，两条轨迹均终止于 p_{data} 支持区域附近的 $t = 0$。

学习之后

$$\mathbf{s}_{\phi^\times} := \mathbf{s}_{\phi^\times}(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log p_t(\mathbf{x}),$$

我们将反向时间 SDE (Equation (4.1.6)) 和 PF-ODE (Equation (4.1.7)) 中难以处理的预言机得分 $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ 替换为学成的代理得分 $\mathbf{s}_{\phi^\times}(\mathbf{x}, t)$。这种置换使得通过 SDE 或 ODE 实现易处理的推断成为可能。为清晰起见，我们分别将所得过程记为 $\mathbf{x}^{\text{SDE}}_{\phi^\times}(t)$ 和 $\mathbf{x}^{\text{ODE}}_{\phi^\times}(t)$，但在后续章节中将省略此区分。[8]

经验反向时间随机微分方程。 通过用训练好的得分模型 $\mathbf{s}_{\phi^\times}$ 替代 Equation (4.1.6) 中的真实得分，我们得到了用于生成的参数化反向时间 SDE：

$$\mathrm{d}\mathbf{x}^{\text{SDE}}_{\phi^\times}(t) = \left[\mathbf{f}\left(\mathbf{x}^{\text{SDE}}_{\phi^\times}(t), t\right) - g^2(t)\mathbf{s}_{\phi^\times}\left(\mathbf{x}^{\text{SDE}}_{\phi^\times}(t), t\right)\right]\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}(t). \tag{4.2.3}$$

[8] 为简化记号，自本小节起采用此记法。

为了生成一个样本，我们首先从先验分布 p_{prior} 中抽取一个初始值 $\mathbf{x}_T$ ，然后数值求解从 $t=T$ 到 $t=0$ 逆时间方向的 Equation (4.2.3) 。对此的标准数值求解器是欧拉-马鲁亚玛方法，它提供了离散更新规则：

$$\mathbf{x}_{t-\Delta t} \leftarrow \mathbf{x}_t - \left[\mathbf{f}(\mathbf{x}_t, t) - g^2(t)\mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t, t)\right]\Delta t + g(t)\sqrt{\Delta t}\cdot\boldsymbol{\epsilon}, \tag{4.2.4}$$

其中 $\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$ 和 $\Delta t>0$ 为步长。

迭代此更新规则可得到最终样本 $\mathbf{x}^{\text{SDE}}_{\boldsymbol{\phi}^\times}(0)$ 。如果得分模型准确，这些生成样本的分布，记为 $p^{\text{SDE}}_{\boldsymbol{\phi}^\times}(\cdot;0)$ ，将对真实数据分布提供一个良好的近似。[9]：

$$p^{\text{SDE}}_{\boldsymbol{\phi}^\times}(\cdot;0) \approx p_{\text{data}}(\cdot).$$

事实上，Equation (2.2.14) 中提出的 DDPM 采样方案是将此欧拉-马鲁亚玛离散化应用于特定的 $\mathbf{f}$ 和 g 选择时的一个特例（见 Section 4.3）。

经验型 PF-ODE。 PF-ODE 定义了一个连续流，连接 p_{prior} 和 p_{data} ，从而实现采样、编码和确切似然评估。下文将详细介绍这些操作。

I. 使用 PF-ODE 进行采样。 将 Equation (4.1.7) 中的 oracle 得分替换为 $\mathbf{s}_{\boldsymbol{\phi}^\times}$ ，得到经验 PF-ODE：

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}^{\text{ODE}}_{\boldsymbol{\phi}^\times}(t) = \mathbf{f}\left(\mathbf{x}^{\text{ODE}}_{\boldsymbol{\phi}^\times}(t), t\right) - \frac{1}{2}g^2(t)\mathbf{s}_{\boldsymbol{\phi}^\times}\left(\mathbf{x}^{\text{ODE}}_{\boldsymbol{\phi}^\times}(t), t\right). \tag{4.2.5}$$

为了生成样本，我们首先从先验分布 p_{prior} 中抽取一个初始样本 $\mathbf{x}_T$。然后我们数值求解从 Equation (4.2.5) 向后时间的 PF-ODE，从 $t=T$ 求解到 $t=0$。此过程等价于近似如下积分：

$$\mathbf{x}^{\text{ODE}}_{\boldsymbol{\phi}^\times}(0) = \mathbf{x}_T + \int_T^0 \left[\mathbf{f}\left(\mathbf{x}^{\text{ODE}}_{\boldsymbol{\phi}^\times}(\tau), \tau\right) - \frac{1}{2}g^2(\tau)\mathbf{s}_{\boldsymbol{\phi}^\times}\left(\mathbf{x}^{\text{ODE}}_{\boldsymbol{\phi}^\times}(\tau), \tau\right)\right]\mathrm{d}\tau.$$

通过求解该积分得到最终样本，$\mathbf{x}^{\text{ODE}}_{\boldsymbol{\phi}^\times}(0)$。通过此确定性过程生成的样本分布，记为 $p^{\text{ODE}}_{\boldsymbol{\phi}^\times}(\cdot;0)$，对数据分布提供了一个近似，使得 $p^{\text{ODE}}_{\boldsymbol{\phi}^\times}(\cdot;0)\approx p_{\text{data}}$。

令 $\Delta t>0$ 表示离散化步长。一种标准的数值积分方法是欧拉法，它估计

$$\mathbf{f}(\mathbf{x}_\tau,\tau) - \frac{1}{2}g^2(\tau)\mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\tau,\tau) \approx \mathbf{f}(\mathbf{x}_t,t) - \frac{1}{2}g^2(t)\mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t,t), \quad \tau\in[t-\Delta t, t],$$

[9]理论上，估计准确率取决于 p_T 与 p_{prior} 之间的差异（通常可忽略）、模型训练误差以及数值离散化误差 (**de2022convergence**)。此处我们不追求形式化的界。

导致如下更新规则：

$$\mathbf{x}_{t-\Delta t} \leftarrow \mathbf{x}_t - \left[\mathbf{f}(\mathbf{x}_t, t) - \frac{1}{2} g^2(t) \mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t, t)\right] \Delta t. \tag{4.2.6}$$

这种联系使我们能够以以下核心洞见重新审视生成过程：

洞察 4.2.1: 生成 ⇔ ODE/SDE 求解

从扩散模型中采样本质上等价于求解相应的概率流 ODE 或反向时间 SDE。

这种等价性为扩散模型采样速度缓慢的问题（如问题 3.5.1所述）提供了清晰的解释，因为生成过程计算量大，这是由于这些微分方程的数值求解器本质上是迭代的，通常需要许多步骤才能准确近似轨迹。[10]. 然而，PF-ODE 公式也具有优势，因为它使我们能够利用加速数值求解器领域的大量文献。探索这些技术以加快扩散模型采样是 Chapter 9的主要关注点。

II. 基于 PF-ODE 的反演。 如前所述，与 SDE 的情况不同，我们可以对相同的 Equation (4.2.5) 同时进行时间上的正向（从 0 到 T）和逆向（从 T 到 0）求解。在正向求解时，ODE 流将数据映射到所有 $t \in [0, T]$ 上的（含噪声的）潜在表示，这起到了编码器的作用。这一概念使得可控生成等强大应用成为可能，例如图像翻译与编辑等 (**mokady2023null**; **su2022dual**)。

III. 通过 PF-ODE 进行确切对数似然计算。 我们将 Equation (4.2.5) 中的动力学重新解释为一种神经微分方程 (**chen2018neural**) 变体（在 Section 5.1.2 中提出），该变体仅参数化得分函数，而非完整的速度场。这种 PF-ODE 形式使得可以通过变量变换公式实现确切对数似然计算。

将公式 Equation (5.1.9) 中的恒等式应用于 Equation (4.2.5) 中的 PF-ODE，我们定义速度场为

$$\mathbf{v}_{\boldsymbol{\phi}^\times}(\mathbf{x}, t) := \mathbf{f}(\mathbf{x}, t) - \frac{1}{2} g^2(t) \mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}, t),$$

其中学成得分 $\mathbf{s}_{\boldsymbol{\phi}^\times}$ 。沿 PF-ODE 轨迹 $\{\mathbf{x}^{\mathrm{ODE}}_{\boldsymbol{\phi}^\times}(t)\}_{t\in[0,T]}$ 的对数密度 $p^{\mathrm{ODE}}_{\boldsymbol{\phi}^\times}(\cdot; t)$ 的

[10] 例如，DDPM 和 Score SDE 通常在生成过程中使用 1,000 次函数求值。

时间演化满足

$$\frac{\mathrm{d}}{\mathrm{d}t}\log p_{\boldsymbol{\phi}^\times}^{\mathrm{ODE}}\left(\mathbf{x}_{\boldsymbol{\phi}^\times}^{\mathrm{ODE}}(t),t\right)=-\nabla\cdot\mathbf{v}_{\boldsymbol{\phi}^\times}\left(\mathbf{x}_{\boldsymbol{\phi}^\times}^{\mathrm{ODE}}(t),t\right),$$

其中 $\nabla\cdot\mathbf{v}$ 表示在 $\mathbf{x}$ 中的散度。

为了评估数据点 $\mathbf{x}_0\sim p_{\mathrm{data}}$ 的似然，我们从 $t=0$ 积分以下增广微分方程组至 $t=T$：

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\mathbf{x}(t)\\ \delta(t)\end{bmatrix}=\begin{bmatrix}\mathbf{v}_{\boldsymbol{\phi}^\times}(\mathbf{x}(t),t)\\ \nabla\cdot\mathbf{v}_{\boldsymbol{\phi}^\times}(\mathbf{x}(t),t)\end{bmatrix},\quad\begin{bmatrix}\mathbf{x}(0)\\ \delta(0)\end{bmatrix}=\begin{bmatrix}\mathbf{x}_0\\ 0\end{bmatrix},\tag{4.2.7}$$

其中 $\delta(t)$ 累积了随时间变化的对数密度变化量。在将系统求解至 $t=T$ 后，我们得到终止状态：

$$\begin{bmatrix}\mathbf{x}(T)\\ \delta(T)\end{bmatrix}.$$

在模型下，原始样本 $\mathbf{x}_0$ 的对数似然可以被评估为

$$\log p_{\boldsymbol{\phi}^\times}^{\mathrm{ODE}}(\mathbf{x}_0;0)=\log p_{\mathrm{prior}}\left(\mathbf{x}(T)\right)+\delta(T),$$

其中 $p_{\mathrm{prior}}\left(\mathbf{x}(T)\right)$ 表示在 $\mathbf{x}(T)$ 处计算的闭式先验密度。

4.3 随机微分方程的实例

song2020score 根据方差在演化过程中的行为，将前向 SDE 中的漂移项 $\mathbf{f}(\mathbf{x}, t)$ 和扩散项 $g(t)$ 分为三类。本文重点关注两种常用类型：*Variance Explosion* (VE) SDE 和 *Variance Preserving* (VP) SDE。尽管可以设计自定义噪声调度器，但其设计会显著影响实验性能。表 4.1 总结了这两种 SDE 的实例。

表 4.1: 前向随机微分方程的总结

	VE SDE	VP SDE
$\mathbf{f}(\mathbf{x}, t)$	$\mathbf{0}$	$-\frac{1}{2}\beta(t)\mathbf{x}$
$g(t)$	$\sqrt{\frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t}}$	$\sqrt{\beta(t)}$
SDE	$\mathrm{d}\mathbf{x}(t) = g(t)\,\mathrm{d}\mathbf{w}(t)$	$\mathrm{d}\mathbf{x}(t) = -\frac{1}{2}\beta(t)\mathbf{x}(t)\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{w}(t)$
$p_t(\mathbf{x}_t\|\mathbf{x}_0)$	$\mathcal{N}\left(\mathbf{x}_t; \mathbf{x}_0, \left(\sigma^2(t) - \sigma^2(0)\right)\mathbf{I}\right)$	$\mathcal{N}\left(\mathbf{x}_t; \mathbf{x}_0 e^{-\frac{1}{2}\int_0^t \beta(\tau)\,\mathrm{d}\tau}, \mathbf{I} - \mathbf{I}e^{-\int_0^t \beta(\tau)\,\mathrm{d}\tau}\right)$
p_{prior}	$\mathcal{N}(\mathbf{0}, \sigma^2(T)\mathbf{I})$	$\mathcal{N}(\mathbf{0}, \mathbf{I})$

4.3.1 VE SDE

VE SDE 包含以下组件：

- **漂移项**：零漂移项 $\mathbf{f} = \mathbf{0}$ 。
- **扩散项**：$g(t) = \sqrt{\frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t}}$ 对于某个函数 $\sigma(t)$ 。

前向 SDE 具有如下形式：

$$\mathrm{d}\mathbf{x}(t) = \sqrt{\frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t}}\,\mathrm{d}\mathbf{w}(t). \tag{4.3.1}$$

类似地，Section 4.3.3 的结果暗示了 VE SDE 的扰动核，并建议选择适当的先验分布：

- **微扰核**：

$$p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \mathbf{x}_0, \left(\sigma^2(t) - \sigma^2(0)\right)\mathbf{I}\right)$$

- **先验分布**：假设 $\sigma(t)$ 是 $t \in [0, T]$ 的增函数，且 $\sigma^2(T) \gg \sigma^2(0)$ 。先验分布为：

$$p_{\text{prior}} := \mathcal{N}(\mathbf{0}, \sigma^2(T)\mathbf{I}).$$

一个典型的 VE SDE 实例是 NCSN，其设计如下：

$$\sigma(t) := \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^t, \quad \text{for } t \in (0,1],$$

其中 $\sigma_{\min}$ 和 $\sigma_{\max}$ 为预先指定的常数。即，方差序列被设计为几何序列。通过这种方式，NCSN 被视为 VE SDE 的离散化版本，如 Section 4.1.1 所讨论。

4.3.2 副总裁软件开发工程师

令 $\beta\colon [0,T] \to \mathbb{R}_{\geq 0}$ 为 t 的非负函数。一个 VP SDE 由以下组成部分定义：

- **漂移项**：一个由 $\mathbf{f}(\mathbf{x},t) = -\frac{1}{2}\beta(t)\mathbf{x}$ 给出的线性漂移。
- **扩散项**：$g(t) = \sqrt{\beta(t)}$.

因此，前向 SDE 表示为：

$$\mathrm{d}\mathbf{x}(t) = -\frac{1}{2}\beta(t)\mathbf{x}(t)\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{w}(t). \tag{4.3.2}$$

利用 Section 4.3.3 的结果，我们可以推导出 VP SDE 的扰动核，并选择适当的先验分布：

- **微扰核**：

$$p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \mathbf{x}_0 e^{-\frac{1}{2}\int_0^t \beta(\tau)\,\mathrm{d}\tau}, \mathbf{I} - \mathbf{I}e^{-\int_0^t \beta(\tau)\,\mathrm{d}\tau}\right).$$

- **先验分布**：$p_{\text{prior}} := \mathcal{N}(\mathbf{0},\mathbf{I})$.

我们注意到，由于扰动核是具有已知均值和协方差的高斯分布，因此我们可以应用 Equation (D.2.5) 来计算其得分函数。

一个典型的 VP SDE 的例子是 DDPM，其中噪声调度 $\beta(t)$ 定义为：

$$\beta(t) := \beta_{\min} + t(\beta_{\max} - \beta_{\min}), \quad \text{for all } t \in [0,1].$$

此处，$\beta_{\min}$ 和 $\beta_{\max}$ 是预定义的常数。借助此设定，如 Section 4.1.1 所讨论，DDPM 可被解释为 VP SDE 的离散化。

4.3.3 （可选）微扰核 $p_t(\mathbf{x}_t|\mathbf{x}_0)$ 是如何推导的？

如果前向 SDE Equation (4.1.3) 中的漂移项关于 $\mathbf{x}$ 是线性的，其形式为

$$\mathbf{f}(\mathbf{x}, t) = f(t)\mathbf{x},$$

对于某个标量值的、与时间相关的函数 $f(t) \in \mathbb{R}$，那么 Equation (4.1.3) 变为一个线性 SDE：

$$\mathrm{d}\mathbf{x}(t) = f(t)\mathbf{x}(t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}(t).$$

即使初始分布 p_{data} 非高斯，漂移的线性性也保证了条件过程仍为高斯分布。特别地，对于 $t > 0$ ，转移核具有如下形式：

$$p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}(t), P(t)\mathbf{I}_D\right),$$

其中 $\mathbf{x}_0 \sim p_{\text{data}}$ ，以及 $\mathbf{m}(t) \in \mathbb{R}^D$ ，$P(t) \in \mathbb{R}_{\geq 0}$ 表示在给定 $\mathbf{x}_0$ 条件下的条件均值和（标量）方差，定义为：

$$\mathbf{m}(t) = \mathbb{E}\left[\mathbf{x}_t|\mathbf{x}(0) = \mathbf{x}_0\right], \qquad P(t)\mathbf{I}_D = \mathrm{Cov}\left[\mathbf{x}_t|\mathbf{x}(0) = \mathbf{x}_0\right].$$

这些一阶和二阶矩按照以下常微分方程演化 (**sarkka2019applied**)：

$$\begin{aligned} \frac{\mathrm{d}\mathbf{m}(t)}{\mathrm{d}t} &= f(t)\mathbf{m}(t), \\ \frac{\mathrm{d}P(t)}{\mathrm{d}t} &= 2f(t)P(t) + g^2(t), \end{aligned} \tag{4.3.3}$$

其中初始均值 $\mathbf{m}(0)$ 和方差 $P(0)$ 为有限值。

由于两个微分方程都是线性的，因此可以通过积分因子法得到闭式解。给定初值 $\mathbf{x}_0$ ，均值和方差的演化为

$$\mathbf{m}(t) = \mathcal{E}(0 \to t)\mathbf{x}_0, \qquad P(t) = \int_0^t \mathcal{E}^2(s \to t)g(s)^2\,\mathrm{d}s, \tag{4.3.4}$$

其中 $\mathbf{m}(0) = \mathbf{x}_0$ 和 $P(0) = 0$ 。这里 $\mathcal{E}(s \to t)$ 表示指数积分因子

$$\mathcal{E}(s \to t) := \exp\left(\int_s^t f(u)\,\mathrm{d}u\right),$$

这捕捉了从时间 s 到 t 的漂移的累积效应。因此，转移核 $p_t(\mathbf{x}_t|\mathbf{x}_0)$ 也具有闭式

表达式。

我们将在 Section C.1.5 中通过伊藤微积分来证明，在具有独立坐标和扩散系数 $g(t)\mathbf{I}_D$ 的 D 维维纳过程下，$p_t(\mathbf{x}_t|\mathbf{x}_0)$ 的条件协方差是各向同性的，即 $\mathrm{Cov}[\mathbf{x}_t|\mathbf{x}_0] = P(t)\mathbf{I}_D$，同时给出 Equation (4.3.3) 的推导。

Example: VE SDE's Transition Kernel

In the special case of VE SDE: $\mathbf{f} \equiv 0$ and $g(t) = \sqrt{\frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t}}$, the mean and covariance of the solution to the SDE evolve as follows.

Mean.

$$\frac{\mathrm{d}\mathbf{m}(t)}{\mathrm{d}t} = 0, \quad \text{with} \quad \mathbf{m}(0) = \mathbf{x}_0 \Longrightarrow \mathbf{m}(t) = \mathbf{x}_0.$$

Variance.

$$\frac{\mathrm{d}P(t)}{\mathrm{d}t} = \frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t}, \quad \text{with} \quad P(0) = 0 \Longrightarrow P(t) = \sigma^2(t) - \sigma^2(0).$$

Therefore

$$p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \mathbf{x}_0, \left(\sigma^2(t) - \sigma^2(0)\right)\mathbf{I}_D\right).$$

■

Example: VP SDE's Transition Kernel

In the VP SDE case with drift $\mathbf{f}(\mathbf{x}, t) = -\frac{1}{2}\beta(t)\mathbf{x}$ and diffusion $g(t) = \sqrt{\beta(t)}$:

Mean $\mathbf{m}(t)$.

$$\frac{\mathrm{d}\mathbf{m}}{\mathrm{d}t} = -\frac{1}{2}\beta(t)\mathbf{m}(t), \quad B(t) \coloneqq \int_0^t \beta(s)\,\mathrm{d}s, \quad \mathbf{m}(t) = e^{-\frac{1}{2}B(t)}\mathbf{x}_0.$$

Variance $P(t)$. The variance satisfies

$$\frac{\mathrm{d}P}{\mathrm{d}t} = -\beta(t)P(t) + \beta(t).$$

Applying the integrating factor $e^{B(t)}$ with $B(t) = \int_0^t \beta(s)\,\mathrm{d}s$, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[P(t)e^{B(t)}\right] = \beta(t)e^{B(t)}.$$

Integrating both sides gives

$$P(t) = 1 - e^{-B(t)}.$$

Hence the covariance is isotropic with

$$\mathbf{P}(t) = P(t)\mathbf{I}_D = \big(1 - e^{-B(t)}\big)\mathbf{I}_D.$$

Final Closed-Form Transition Kernel.

$$p_t(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \underbrace{e^{-\frac{1}{2}B(t)}\mathbf{x}_0}_{\mathbf{m}(t)}, \underbrace{\big(1 - e^{-B(t)}\big)\mathbf{I}_D}_{P(t)\mathbf{I}_D}\right), \quad B(t) = \int_0^t \beta(s)\,\mathrm{d}s.$$

4.4 (Optional) Rethinking Forward Kernels in Score-Based and Variational Diffusion Models

DDPM 和 Score SDE 通常通过前向转移核 $p(\mathbf{x}_t|\mathbf{x}_{t-\Delta t})$ 引入，DDPM 中以离散形式定义，Score SDE 中则以连续时间 SDE 形式定义。然而，在实际应用中，尤其是在它们的损失函数 (Equations (2.2.8) and (4.2.1)) 中，最为相关的是从数据出发的累积转移核 $p_t(\mathbf{x}_t|\mathbf{x}_0)$。两个框架最终都依赖于该核，要么通过递归计算（DDPM），要么通过求解 ODE，如 Section 4.3.3(Score SDE) 所述。

在本节中，我们首先定义 $p_t(\mathbf{x}_t|\mathbf{x}_0)$（在连续时间下），这提供了一个更简洁、更直接的视角。总体而言，尽管 $p(\mathbf{x}_t|\mathbf{x}_{t-\Delta t})$ 与 $p_t(\mathbf{x}_t|\mathbf{x}_0)$ 在理论上等价，但定义后者通常能得到更清晰且更易解释的表达形式。特别是，$p_t(\mathbf{x}_t|\mathbf{x}_0)$ 为先验提供了直接的洞察，即 $t \to T$ ，并且与实际的损失设计自然契合。

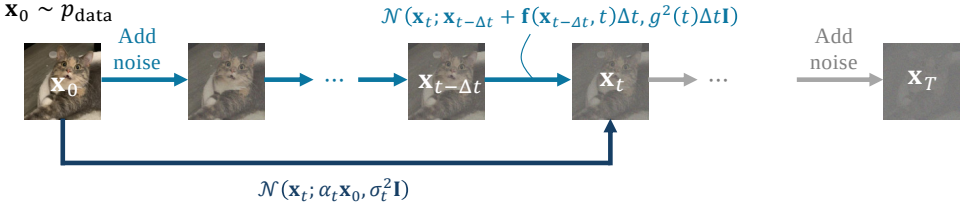


图 4.6: 引理 4.4.1 的示意图。通过连续时间随机微分方程 ($\Delta t \to 0$) 实现的增量噪声注入与对 Equation (4.4.1) 的直接扰动在数学上等价。

4.4.1 一个通用的仿射前向过程 $p_t(\mathbf{x}_t|\mathbf{x}_0)$

我们首先定义一个通用的前向扰动核：

$$p_t(\mathbf{x}_t|\mathbf{x}_0) := \mathcal{N}\big(\mathbf{x}_t; \alpha_t\mathbf{x}_0, \sigma_t^2\mathbf{I}\big), \tag{4.4.1}$$

其中 $\mathbf{x}_0 \sim p_{\text{data}}$ ，以及 α_t ， σ_t 是关于 $t \in [0,T]$ 的非负标量函数，满足：

(i) $\alpha_t > 0$ 和 $\sigma_t > 0$ 对所有 $t \in (0,1]$ （允许 $\sigma_0 = 0$ ），以及

(ii) 通常，$\alpha_0 = 1$ 和 $\sigma_0 = 0$。

也就是说，$\mathbf{x}_t \sim p_t(\mathbf{x}_t|\mathbf{x}_0)$ 可以被采样为

$$\mathbf{x}_t = \alpha_t\mathbf{x}_0 + \sigma_t\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

该框架涵盖了多个著名的实例，包括 VE（例如，NCSN）、VP（例如，DDPM）以及前向核 (**lipman2022flow**; **liu2022rectified**) 的流匹配（FM），其在线性插值 $\mathbf{x}_0$ 与 $\boldsymbol{\epsilon}$ 之间进行（见后续 Section 5.2）。

- **VE (NCSN) 核函数**：$\alpha_t \equiv 1$, $\sigma_T \gg 1$;
- **VP (DDPM) 核**：$\alpha_t := \sqrt{1-\sigma_t^2}$ ，使得 $\alpha_t^2 + \sigma_t^2 = 1$;
- **FM 核心**：$\alpha_t = 1 - t$ ， $\sigma_t = t$ 。

4.4.2 与得分 SDE 的连接

对于得分 SDE，以线性形式指定 $p_t(\mathbf{x}_t|\mathbf{x}_0)$ 会自然地诱导出具有仿射系数的 SDE，这为从漂移项和扩散项出发并求解矩的常微分方程提供了一个更直观的替代方法（见 Section 4.3.3）。

给定前向扰动核在 Equation (4.4.1)，对应的前向随机微分方程（SDE）具有关于 $\mathbf{x}$ 的线性形式，如 Equation (4.3.2) 所示：

$$\mathrm{d}\mathbf{x}(t) = \underbrace{f(t)\mathbf{x}(t)}_{\mathbf{f}(\mathbf{x}(t),t)}\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}(t),$$

其中 $f, g\colon [0,T] \to \mathbb{R}$ 为时间的实值函数。系数 $f(t)$ 和 $g(t)$ 可以用 α_t 和 σ_t 解析地表示，如以下引理所述。

Lemma 4.4.1: Forward Perturbation Kernel ⇔ Linear SDE

Define $\lambda_t := \log \frac{\alpha_t}{\sigma_t}$ for $t \in (0, T]$. Given the forward perturbation kernel

$$\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

the linear SDE

$$\mathrm{d}\mathbf{x}(t) = f(t)\mathbf{x}(t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}(t),$$

with coefficients

$$\begin{aligned} f(t) &= \frac{\mathrm{d}}{\mathrm{d}t}\log \alpha_t, \\ g^2(t) &= \frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t} - 2\frac{\mathrm{d}}{\mathrm{d}t}\log \alpha_t \sigma_t^2 = -2\sigma_t^2 \frac{\mathrm{d}}{\mathrm{d}t}\lambda_t, \end{aligned} \tag{4.4.2}$$

has the conditional transition $p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\big(\mathbf{x}_t; \alpha_t \mathbf{x}_0, \sigma_t^2 \mathbf{I}\big)$ for all $t \in (0, T]$. Conversely, if a linear SDE has conditional transitions $\mathcal{N}(\mathbf{x}_t; \alpha_t \mathbf{x}_0, \sigma_t^2 \mathbf{I})$ so that $\alpha_t > 0$ and $\sigma_t > 0$ for all $t \in (0, T]$, then its coefficients satisfy Equation (4.4.2) for $t \in (0, T]$.

Proof for Lemma.

From Section 4.3.3, the proof matches the mean and covariance ODEs $\mathbf{m}'(t) = f(t)\mathbf{m}(t)$, $\mathbf{P}'(t) = 2f(t)\mathbf{P}(t) + g^2(t)\mathbf{I}$ with $\mathbf{m}(t) = \alpha_t \mathbf{x}_0$ and $\mathbf{P}(t) = \sigma_t^2 \mathbf{I}$ on $(0, T]$.

Remark.

为确切的匹配终端时刻的高斯先验，该过程必须完全遗忘 $\mathbf{x}_0$ 并达到目标方差；这要求 $\alpha_T = 0$ 和 σ_T^2 等于先验方差。在随机微分方程框架中，有

$$\alpha_t = \exp\Big(\int_0^t f(u)\,\mathrm{d}u\Big).$$

因此，在有限时刻 T 强制实现 $\alpha_T = 0$ 会要求

$$\int_0^T f(u)\,\mathrm{d}u = -\infty,$$

> 意味着当 $t \to T$ 时漂移项必须以无穷大速度收缩。同时，扩散项必须发散以维持设定的方差，这体现为
>
> $$g^2(t) = \sigma_t^{2\prime} - 2\frac{\alpha_t'}{\alpha_t}\sigma_t^2 \to \infty \quad 当 t \to T.$$
>
> 若 f 和 g 在区间 $[0,T]$ 上保持有界，则必然有 $\alpha_T > 0$ 且存在对 $\mathbf{x}_0$ 的残余依赖。此时高斯先验仅能渐近达到：要么在极限情况 $t \to T$ 下实现（无法精确达到），要么通过适当的 $T \to \infty$ 时间重参数化在无限时域上精确实现。

由上述引理可知，通过系数为 $f(t)$ 和 $g(t)$ 的线性随机微分方程（SDE）指定增量噪声注入，在数学上等价于用参数 α_t 和 σ_t 定义扰动核。在扩散模型文献中，这两种观点被交替使用。因此，我们得出结论：

观察 4.4.1:

定义 $p_t(\mathbf{x}_t|\mathbf{x}_0)$ 等价于指定线性 SDE 的系数 $f(t)$ 和 $g(t)$ 。

4.4.3 基于变分的扩散模型连接

我们重新审视 DDPM 中的一个基本身份，该身份通过贝叶斯法则推导得出：

$$p(\mathbf{x}_{t-\Delta t}|\mathbf{x}_t, \mathbf{x}) = p(\mathbf{x}_t|\mathbf{x}_{t-\Delta t}) \cdot \frac{p_{t-\Delta t}(\mathbf{x}_{t-\Delta t}|\mathbf{x})}{p_t(\mathbf{x}_t|\mathbf{x})}, \tag{4.4.3}$$

对于任意 $\mathbf{x}$ （通常为 $\mathbf{x} \sim p_{\text{data}}$ ）。这种反向条件 $p(\mathbf{x}_{t-\Delta t}|\mathbf{x}_t, \mathbf{x})$ 在建模中至关重要，它既支持易处理的训练目标，又实现了高效的采样。

尽管 DDPM 通常首先定义增量核 $p(\mathbf{x}_t|\mathbf{x}_{t-\Delta t})$，但累积转移 $p_t(\mathbf{x}_t|\mathbf{x}_0)$ 往往能提供更具可解释性和实用性的表述，尤其是在先验和损失设计方面。

推导转移核。 我们现在将其扩展到连续时间情景。设 $0 \le t < s \le T$ 为两个（连续）时间点。给定扰动核 $p_t(\mathbf{x}_t|\mathbf{x}_0)$ ，我们可以对任意 $\mathbf{x}$ 通过应用 Equation (4.4.3) 计算反向条件 $p(\mathbf{x}_t|\mathbf{x}_s, \mathbf{x})$ 。[11]，使用前向核 $p(\mathbf{x}_s|\mathbf{x}_t)$ 作为中间步骤。下面的引理总结了这一推导过程，扩展了引理 2.2.2 而无需假设 $\alpha_t^2 + \sigma_t^2 = 1$。

[11] 该恒等式可自然地推广到连续时间情形，其中将 s 视为一个一般性的较早时刻。

Lemma 4.4.2: Reverse Conditional Transition Kernels

Let $0 \leq t < s \leq T$. The reverse *conditional* transition kernel is:

$$p(\mathbf{x}_t|\mathbf{x}_s, \mathbf{x}) = \mathcal{N}\left(\mathbf{x}_t; \boldsymbol{\mu}(\mathbf{x}_s, \mathbf{x}; s, t), \sigma^2(s,t)\mathbf{I}\right),$$

where

$$\boldsymbol{\mu}(\mathbf{x}_s, \mathbf{x}; s, t) := \frac{\alpha_{s|t}\sigma_t^2}{\sigma_s^2}\mathbf{x}_s + \frac{\alpha_t \sigma_{s|t}^2}{\sigma_s^2}\mathbf{x}, \quad \sigma^2(s,t) := \sigma_{s|t}^2 \frac{\sigma_t^2}{\sigma_s^2}. \tag{4.4.4}$$

Here, $\alpha_{s|t}$ and $\sigma_{s|t}$ are defined as:

$$\alpha_{s|t} := \frac{\alpha_s}{\alpha_t}, \qquad \sigma_{s|t}^2 := \sigma_s^2 - \alpha_{s|t}^2 \sigma_t^2.$$

Proof for Lemma.

We first compute the forward transition kernel:

$$p(\mathbf{x}_s|\mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_s; \alpha_{s|t}\mathbf{x}_t, \sigma_{s|t}^2\mathbf{I}\right). \tag{4.4.5}$$

The reverse kernel then follows from Bayes' rule, and since all involved distributions are Gaussian, the result can be derived by direct computation. For further details, see Appendix A of (**kingma2021variational**).

尽管 $p(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t)$ 和 $p_t(\mathbf{x}_t|\mathbf{x}_0)$ 在理论上等价，$p_t(\mathbf{x}_t|\mathbf{x}_0)$ 通常扮演更为重要的角色。Equation (4.4.5) 中的逐步转移主要目的在于获得闭式反向核。因此，近期的研究 (**kingma2021variational**) 更倾向于直接指定 $p_t(\mathbf{x}_t|\mathbf{x}_0)$，以提升清晰度和可解释性。

反向过程建模、训练与采样。 在我们推广的情景下，训练目标（Equation (2.2.13) 中的 ELBO）以及 Section 2.2 中引入的建模框架仍然适用。为清晰起见，我们采用 $\mathbf{x}$ -预测公式，记为 $\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_s, s)$，参考 **kingma2021variational**。然而，由于存在关联性（如 Equation (2.2.12) 所示），等价的 $\boldsymbol{\epsilon}$ -预测视角，表示为 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_s, s)$，同样有效。

$$\mathbf{x}_s = \alpha_s \mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_s, s) + \sigma_s \boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_s, s), \quad \text{for any given } \mathbf{x}_s \sim q_s.$$

建模与扩散损失 $\mathcal{L}_{\text{diffusion}}$ 。 与 DDPM 类似，Equation (4.4.4) 中的条件分布 $p(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x})$ 激发了用可学习的预测器 $\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_s,s)$ 替代干净信号 $\mathbf{x}$ ，从而得到如下参数化的反向模型：

$$p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_s) := \mathcal{N}\left(\mathbf{x}_t;\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_s,s,t),\sigma^2(s,t)\mathbf{I}\right), \tag{4.4.6}$$

均值参数化为：

$$\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_s,s,t) = \frac{\alpha_{s|t}\sigma_t^2}{\sigma_s^2}\mathbf{x}_s + \frac{\alpha_t\sigma_{s|t}^2}{\sigma_s^2}\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_s,s).$$

给定前向核在 Equation (4.4.1)，$\mathcal{L}_{\text{diffusion}}(\mathbf{x};\boldsymbol{\phi})$ 中的 KL 散度简化为加权回归损失：

$$\begin{aligned}\mathcal{D}_{\text{KL}}\big(p(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0)\|p_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_s)\big) &= \frac{1}{2\sigma^2(s,t)}\left\|\boldsymbol{\mu}(\mathbf{x}_s,\mathbf{x}_0;s,t)-\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_s,s,t)\right\|_2^2 \\ &= \frac{1}{2}\big(\text{SNR}(t)-\text{SNR}(s)\big)\left\|\mathbf{x}_0-\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_s,s)\right\|_2^2,\end{aligned} \tag{4.4.7}$$

其中 $\mathbf{x}_s=\alpha_s\mathbf{x}_0+\sigma_s\boldsymbol{\epsilon}$ ，在 $\mathbf{x}_0\sim p_{\text{data}}$ 、$\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$ 和 $\text{SNR}(s):=\alpha_s^2/\sigma_s^2$ 的条件下，表示时间 s 处的信噪比。

Remark.

> 在 (**kingma2021variational**) 中，作者研究了当 $t\to s$ 时 Equation (4.4.7) 的连续时间极限，得出：
>
> $$\mathcal{L}_{\text{VDM}}^{\infty}(\mathbf{x}_0) = -\frac{1}{2}\mathbb{E}_{s,\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\text{SNR}'(s)\left\|\mathbf{x}_0-\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_s,s)\right\|_2^2.$$
>
> 该设置还引入了可学习的噪声调度，虽然其泛化能力可延伸至连续数据之外，但此类扩展已超出当前讨论范围。

采样。 采样过程与 DDPM 类似，使用来自 Equation (4.4.6) 的参数化核函数：

$$\mathbf{x}_t = \underbrace{\frac{\alpha_{s|t}\sigma_t^2}{\sigma_s^2}\mathbf{x}_s + \frac{\alpha_t\sigma_{s|t}^2}{\sigma_s^2}\mathbf{x}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s,s)}_{\boldsymbol{\mu}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s,t,s)} + \sigma_{s|t}\frac{\sigma_t}{\sigma_s}\boldsymbol{\epsilon}_s, \quad \boldsymbol{\epsilon}_s\sim\mathcal{N}(\mathbf{0},\mathbf{I}). \tag{4.4.8}$$

4.5 （可选）通过边缘化与贝叶斯法则的福克-普朗克方程与反向时间随机微分方程

在本节中，我们从概率角度探讨福克-普朗克方程和反向时间随机微分方程的结构。通过运用边缘化技巧和贝叶斯规则等基本工具，我们揭示了随机过程的统计表述与其对应微分方程之间的联系。

我们强调，此处提出的“推导”并非数学上的严格证明，而是一种旨在传达潜在联系的启发式论述。

4.5.1 从转移核的边缘化得到的福克-普朗克方程

给定如 Equation (4.1.2) 所示的前向转移概率

$$p(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t+\Delta t}; \mathbf{x}_t + \mathbf{f}(\mathbf{x}_t, t)\Delta t, g^2(t)\Delta t\mathbf{I}\right),$$

以及边缘分布

$$p_t(\mathbf{x}_t), \quad p_{t+\Delta t}(\mathbf{x}_{t+\Delta t}),$$

我们旨在推导描述边缘分布 p_t 随时间演化的福克-普朗克方程。

变量变换。 根据马尔可夫性质，时间 $t+\Delta t$ 的边缘分布可以表示为对前一状态 $\mathbf{x}_t$ 的积分（即查普曼-科尔莫戈罗夫方程）：

$$p_{t+\Delta t}(\mathbf{x}) = \int \mathcal{N}(\mathbf{x}; \mathbf{y} + \mathbf{f}(\mathbf{y}, t)\Delta t, g^2(t)\Delta t\mathbf{I}) p_t(\mathbf{y})\, \mathrm{d}\mathbf{y}.$$

我们引入一个新变量

$$\mathbf{u} := \mathbf{y} + \mathbf{f}(\mathbf{y}, t)\Delta t,$$

因此高斯分布以 $\mathbf{u}$ 为中心。当 Δt 较小时，该映射是可逆的，其逆为

$$\mathbf{y} = \mathbf{u} - \mathbf{f}(\mathbf{u}, t)\Delta t + \mathcal{O}(\Delta t^2), \quad \left|\det \frac{\partial \mathbf{y}}{\partial \mathbf{u}}\right| = 1 - (\nabla_{\mathbf{u}} \cdot \mathbf{f})(\mathbf{u}, t)\Delta t + \mathcal{O}(\Delta t^2).$$

因此，变量变换公式得出：

$$p_{t+\Delta t}(\mathbf{x}) = \int \mathcal{N}\left(\mathbf{x}; \mathbf{u}, g^2(t)\Delta t \mathbf{I}\right) \cdot \left[p_t(\mathbf{u}) - \Delta t \mathbf{f}(\mathbf{u}, t) \cdot \nabla_{\mathbf{u}} p_t(\mathbf{u}) - \Delta t (\nabla_{\mathbf{u}} \cdot \mathbf{f})(\mathbf{u}, t) p_t(\mathbf{u})\right] \mathrm{d}\mathbf{u} + \mathcal{O}(\Delta t^2),$$

泰勒展开 对于任意光滑函数 $\phi : \mathbb{R}^D \to \mathbb{R}$ 和尺度 $\sigma > 0$，若 $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$，则以下近似成立（称为 泰勒–高斯平滑公式）：

$$\int \mathcal{N}(\mathbf{x}; \mathbf{u}, \sigma^2 \mathbf{I}) \phi(\mathbf{u}) \, \mathrm{d}\mathbf{u} = \mathbb{E}\left[\phi(\mathbf{x} + \sigma \mathbf{z})\right] = \phi(\mathbf{x}) + \frac{\sigma^2}{2} \Delta_{\mathbf{x}} \phi(\mathbf{x}) + \mathcal{O}(\sigma^4).$$

这是因为泰勒展开式：

$$\phi(\mathbf{x} + \sigma \mathbf{z}) = \phi(\mathbf{x}) + \sigma \nabla_{\mathbf{x}} \phi(\mathbf{x}) \cdot \mathbf{z} + \frac{\sigma^2}{2} \mathbf{z}^\top \nabla_{\mathbf{x}}^2 \phi(\mathbf{x}) \mathbf{z} + \mathcal{O}(\sigma^3)$$

以及 $\mathbb{E}[\mathbf{z}] = \mathbf{0}$，$\mathbb{E}[\mathbf{z}\mathbf{z}^\top] = \mathbf{I}$。

将此应用于 $\phi = p_t$、$\phi = \mathbf{f} \cdot \nabla_{\mathbf{u}} p_t$ 和 $\phi = (\nabla_{\mathbf{u}} \cdot \mathbf{f}) p_t$，并使用 $\sigma^2 = g^2(t)\Delta t$，我们可以得到

$$\begin{aligned} & p_{t+\Delta t}(\mathbf{x}) - p_t(\mathbf{x}) \\ =& -\Delta t \mathbf{f}(\mathbf{x}, t) \cdot \nabla_{\mathbf{x}} p_t(\mathbf{x}) - \Delta t (\nabla_{\mathbf{x}} \cdot \mathbf{f})(\mathbf{x}, t) p_t(\mathbf{x}) + \frac{g^2(t)}{2} \Delta t \Delta_{\mathbf{x}} p_t(\mathbf{x}) + \mathcal{O}(\Delta t^2) \\ =& -\Delta t \nabla_{\mathbf{x}} \cdot \left(\mathbf{f}(\mathbf{x}, t) p_t(\mathbf{x})\right) + \frac{g^2(t)}{2} \Delta t \Delta_{\mathbf{x}} p_t(\mathbf{x}) + \mathcal{O}(\Delta t^2). \end{aligned}$$

除以 Δt，令 $\Delta t \to 0$，得到福克-普朗克方程。

在 Section C.1.4 中，我们给出了基于 Itô 的推导，以补充上述离散时间视角。

4.5.2 为什么反向时间 SDE 具有这种形式？

逆时间 SDE 的严格推导是技术性的，需要深入探讨福克-普朗克方程的性质。然而，通过贝叶斯定理可以直观地理解逆时间 SDE 的形式。在此，我们给出一种启发式的推导，以揭示 Equation (4.1.6) 为何具有该形式，其中出现了评分函数[12].

[12] 该推导受到 this post 中方法的启发。

使用贝叶斯规则进行反演。 我们的目标是通过首先考虑离散时间情形来确定反向时间转移核：

$$p(\mathbf{x}_t|\mathbf{x}_{t+\Delta t}),$$

然后取 $\Delta t \to 0$ 以得到连续时间形式。使用贝叶斯定理，我们表示：

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{x}_{t+\Delta t}) &= p(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t)\frac{p_t(\mathbf{x}_t)}{p_{t+\Delta t}(\mathbf{x}_{t+\Delta t})} \\ &= p(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t)\exp\left(\log p_t(\mathbf{x}_t) - \log p_{t+\Delta t}(\mathbf{x}_{t+\Delta t})\right). \end{aligned} \tag{4.5.1}$$

假设前向转移核如 Equation (4.1.2): 所示。

$$p(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t+\Delta t}; \mathbf{x}_t + \mathbf{f}(\mathbf{x}_t, t)\Delta t, g^2(t)\Delta t\mathbf{I}\right)$$

泰勒展开 为了处理指数项，我们应用一阶泰勒展开。关键思想是在空间和时间两个方向上围绕点 $(\mathbf{x}_t, t)$ 进行展开：

$$\begin{aligned} \log p_{t+\Delta t}(\mathbf{x}_{t+\Delta t}) = \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t) \cdot (\mathbf{x}_{t+\Delta t} - \mathbf{x}_t) \\ + \frac{\partial \log p_t(\mathbf{x}_t)}{\partial t}\Delta t + \mathcal{O}(\|\mathbf{h}\|_2^2) \end{aligned}$$

其中 $\mathbf{h} := (\mathbf{x}_{t+\Delta t} - \mathbf{x}_t, \Delta t)$ 。因此：

$$\begin{aligned} \log p_t(\mathbf{x}_t) - \log p_{t+\Delta t}(\mathbf{x}_{t+\Delta t}) = -\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t) \cdot (\mathbf{x}_{t+\Delta t} - \mathbf{x}_t) \\ - \frac{\partial \log p_t(\mathbf{x}_t)}{\partial t}\Delta t + \mathcal{O}(\|\mathbf{h}\|_2^2) \end{aligned} \tag{4.5.2}$$

对于具有有限漂移和扩散的前向过程，我们有 $\mathbb{E}[\|\mathbf{x}_{t+\Delta t} - \mathbf{x}_t\|_2^2] = \mathcal{O}(\Delta t)$ ，这确保了余项在期望上为 $\mathcal{O}((\Delta t)^2)$ 。

代入反向转移。 将方程 Equation (4.1.2) 和 Equation (4.5.2) 代入 Equation (4.5.1)：

$$\begin{aligned} & p(\mathbf{x}_t|\mathbf{x}_{t+\Delta t}) \\ =& \frac{1}{(2\pi g^2(t)\Delta t)^{D/2}} \exp\left(-\frac{\|\mathbf{x}_{t+\Delta t} - \mathbf{x}_t - \mathbf{f}(\mathbf{x}_t, t)\Delta t\|_2^2}{2g^2(t)\Delta t}\right) \\ & \cdot \exp\left(-\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t) \cdot (\mathbf{x}_{t+\Delta t} - \mathbf{x}_t) - \frac{\partial \log p_t(\mathbf{x}_t)}{\partial t}\Delta t + \mathcal{O}((\Delta t)^2)\right). \end{aligned}$$

代数运算。 关键步骤是在指数部分配方。我们有：

$$
\begin{aligned}
&-\frac{\|\mathbf{x}_{t+\Delta t}-\mathbf{x}_t-\mathbf{f}(\mathbf{x}_t,t)\Delta t\|_2^2}{2g^2(t)\Delta t}-\nabla_{\mathbf{x}}\log p_t(\mathbf{x}_t)\cdot(\mathbf{x}_{t+\Delta t}-\mathbf{x}_t)\\
=&-\frac{[\|\mathbf{x}_{t+\Delta t}-\mathbf{x}_t-\mathbf{f}(\mathbf{x}_t,t)\Delta t\|_2^2+2g^2(t)\Delta t\nabla_{\mathbf{x}}\log p_t(\mathbf{x}_t)\cdot(\mathbf{x}_{t+\Delta t}-\mathbf{x}_t)]}{2g^2(t)\Delta t}
\end{aligned}
$$

设 $\boldsymbol{\delta}:=\mathbf{x}_{t+\Delta t}-\mathbf{x}_t$ 和 $\boldsymbol{\mu}:=\mathbf{f}(\mathbf{x}_t,t)\Delta t$ 。则：

$$
\begin{aligned}
&\|\boldsymbol{\delta}-\boldsymbol{\mu}\|_2^2+2g^2(t)\Delta t\nabla_{\mathbf{x}}\log p_t(\mathbf{x}_t)\cdot\boldsymbol{\delta}\\
=&\|\boldsymbol{\delta}\|_2^2-2\boldsymbol{\delta}\cdot\boldsymbol{\mu}+\|\boldsymbol{\mu}\|_2^2+2g^2(t)\Delta t\nabla_{\mathbf{x}}\log p_t(\mathbf{x}_t)\cdot\boldsymbol{\delta}\\
=&\|\boldsymbol{\delta}\|_2^2-2\boldsymbol{\delta}\cdot[\boldsymbol{\mu}-g^2(t)\Delta t\nabla_{\mathbf{x}}\log p_t(\mathbf{x}_t)]+\|\boldsymbol{\mu}\|_2^2\\
=&\|\boldsymbol{\delta}-[\boldsymbol{\mu}-g^2(t)\Delta t\nabla_{\mathbf{x}}\log p_t(\mathbf{x}_t)]\|_2^2-\|g^2(t)\Delta t\nabla_{\mathbf{x}}\log p_t(\mathbf{x}_t)\|_2^2
\end{aligned}
$$

代回后：

$$
\begin{aligned}
&\|\boldsymbol{\delta}-[\mathbf{f}(\mathbf{x}_t,t)\Delta t-g^2(t)\Delta t\nabla_{\mathbf{x}}\log p_t(\mathbf{x}_t)]\|_2^2\\
=&\|\mathbf{x}_{t+\Delta t}-\mathbf{x}_t-[\mathbf{f}(\mathbf{x}_t,t)-g^2(t)\nabla_{\mathbf{x}}\log p_t(\mathbf{x}_t)]\Delta t\|_2^2.
\end{aligned}
$$

因此，

$$
\begin{aligned}
&p(\mathbf{x}_t|\mathbf{x}_{t+\Delta t})\\
=&\frac{1}{(2\pi g^2(t)\Delta t)^{D/2}}\\
&\cdot\exp\left(-\frac{\|\mathbf{x}_{t+\Delta t}-\mathbf{x}_t-[\mathbf{f}(\mathbf{x}_t,t)-g^2(t)\nabla_{\mathbf{x}}\log p_t(\mathbf{x}_t)]\Delta t\|_2^2}{2g^2(t)\Delta t}\right)\\
&\cdot\exp(\mathcal{O}(\Delta t))\\
=&\mathcal{N}\left(\mathbf{x}_t;\mathbf{x}_{t+\Delta t}-[\mathbf{f}(\mathbf{x}_t,t)-g^2(t)\nabla_{\mathbf{x}}\log p_t(\mathbf{x}_t)]\Delta t,g^2(t)\Delta t\mathbf{I}\right)\\
&\cdot(1+\mathcal{O}(\Delta t)).
\end{aligned}
$$

完成平方产生的附加项 $\|g^2(t)\Delta t\nabla_{\mathbf{x}}\log p_t(\mathbf{x}_t)\|_2^2$ 为 $\mathcal{O}((\Delta t)^2)$ ，可以被吸收到误差项中。类似地，时间导数项 $\frac{\partial\log p_t(\mathbf{x}_t)}{\partial t}\Delta t$ 为 $\mathcal{O}(\Delta t)$ ，在连续极限下将消失。

取 $\Delta t \to 0$ 极限。 当 $\Delta t \approx 0$ 时，在光滑性假设下，以下近似成立：

$$\begin{aligned}
\mathbf{f}(\mathbf{x}_t, t) &\approx \mathbf{f}(\mathbf{x}_{t+\Delta t}, t+\Delta t), \\
g(t) &\approx g(t+\Delta t), \\
\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t) &\approx \nabla_{\mathbf{x}} \log p_{t+\Delta t}(\mathbf{x}_{t+\Delta t}) \\
&= \mathbf{s}(\mathbf{x}_{t+\Delta t}, t+\Delta t).
\end{aligned}$$

利用这些近似值和一些代数变换，我们得到：

$$\begin{aligned}
& p(\mathbf{x}_t|\mathbf{x}_{t+\Delta t}) \\
\approx & \frac{1}{(2\pi g^2(t)\Delta t)^{D/2}} \exp\Bigg(\\
& - \frac{\left\| \mathbf{x}_t - \left(\mathbf{x}_{t+\Delta t} - \left[\mathbf{f}(\mathbf{x}_{t+\Delta t}, t+\Delta t) - g^2(t+\Delta t)\mathbf{s}(\mathbf{x}_{t+\Delta t}, t+\Delta t) \right] \Delta t \right) \right\|_2^2}{2g^2(t+\Delta t)\Delta t} \\
& \Bigg).
\end{aligned}$$

这意味着 $p(\mathbf{x}_t|\mathbf{x}_{t+\Delta t})$ 大致服从一个正态分布，其：

$$\begin{aligned}
&\textbf{Mean:} \mathbf{x}_{t+\Delta t} - \left[\mathbf{f}(\mathbf{x}_{t+\Delta t}, t+\Delta t) - g^2(t+\Delta t)\mathbf{s}(\mathbf{x}_{t+\Delta t}, t+\Delta t) \right] \Delta t, \\
&\textbf{Covariance:} g^2(t+\Delta t)\Delta t \mathbf{I}.
\end{aligned}$$

令 $\Delta t \to 0$ 趋于极限，我们“推导出”式 Equation (4.1.6) 中给出的反向时间连续 SDE。

4.6 闭幕词

本章标志着我们旅程中的一个关键转折点，将变分法和得分函数视角下的离散时间扩散过程统一到一个简洁优美的连续时间框架中。我们证明了 DDPM 和 NCSN 均可被理解为具有不同漂移/波动系数的随机微分方程（SDE）的离散化形式。

该框架的核心在于存在一个对应的反向时间 SDE，它形式上定义了一个逆转噪声污染的生成过程。关键的是，该反向过程的漂移项仅依赖于一个未知量：在每个时间点上边际数据分布的得分函数，$\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$。这一洞察强化了得分函数在生成式建模中的核心作用。

此外，我们引入了一个纯确定性的对应模型——概率流常微分方程（PF-ODE），其解轨迹沿着与 SDE 相同的边缘密度 $\{p_t\}$ 演化。这种显著的一致性由潜在的福克-普朗克方程保证。其深刻含义在于，生成这一复杂任务在本质上等价于求解一个微分方程。训练过程简化为学习定义方程向量场的评分函数，而采样则转化为数值积分问题。

PF-ODE 的引入，即一种纯粹确定性的流，为扩散模型的第三种也是最后一种视角提供了有力的桥梁。学习由速度场控制的确定性变换这一概念，是近年来主流生成式模型的核心原理。在下一章中，我们将：

1. 从其在归一化流（Normalizing Flows）和神经微分方程（Neural ODEs）中的起源出发，探索这一基于流的视角。

2. 展示这一观点如何导出现代流匹配框架，该框架直接学习一个速度场，以实现样本在分布之间的迁移。

最终，我们将看到，这个从随机原理推导出的确定性 PF-ODE，如何从这种完全不同的基于流的原点出发被构建并实现泛化，从而完成对扩散建模的统一描述。

5

基于流的视角：从归一化流到流匹配

Everything flows.

Heraclitus

变量变换公式，概率论的基石之一 (**tabak2010density**; **tabak2013family**)，在现代生成式建模中焕发了新的生机。尽管得分 SDEs 通过福克-普朗克方程 (Section 4.1.5) 提供了连接数据分布与先验分布的微分方程框架，但这种连续演化本质上仍是同一基本原理的动态形式。

密度的变量变换公式。 给定一个可逆变换 $\mathbf{f}$，当 $\mathbf{z} \sim p_{\text{prior}}$ 时，$\mathbf{x} = \mathbf{f}(\mathbf{z})$ 的密度为：

$$p(\mathbf{x}) = p_{\text{prior}}(\mathbf{z}) \left| \det \frac{\partial \mathbf{f}^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right|, \quad \text{where } \mathbf{z} = \mathbf{f}^{-1}(\mathbf{x}). \tag{5.0.1}$$

这个看似简单的公式在 $\mathbf{f}$ 为易处理的情况下，能够实现密度和样本的精确、双向传输，这正是我们在 Section 5.1 中将要介绍的归一化流的基础。但如果我们将这一思想通过连续时间变换的视角重新审视呢?

在本章中，我们基于这一核心原理，探索扩散模型的一种新视角：流匹配 (见 Section 5.2)。流匹配自然地源于（连续）归一化流，深化了我们将扩散视为一种强大密度传输过程的理解。

为了更好地理解本章内容，我们在 Chapter B 中提供了一个直观且自包含的概述，介绍了变量变换公式的不同变体，从基本情形逐步推进到连续性方程，最终到福克-普朗克方程。

5.1 基于流的模型：归一化流与神经微分方程

在本节中，我们将介绍基于流的模型，包括归一化流（NFs）(**rezende2015variational**) 和神经微分方程（NODEs）(**chen2018neural**)。

神经网络流（NFs）通过将一系列可逆变换应用于简单的基础分布，实现了灵活且易处理的概率密度估计。节点网络（NODEs）将这一框架扩展到连续时间，其中变换由常微分方程（ODE）控制。通过将变换视为连续时间动力学，NODEs 为神经网络流范式提供了平滑且可扩展的延伸。

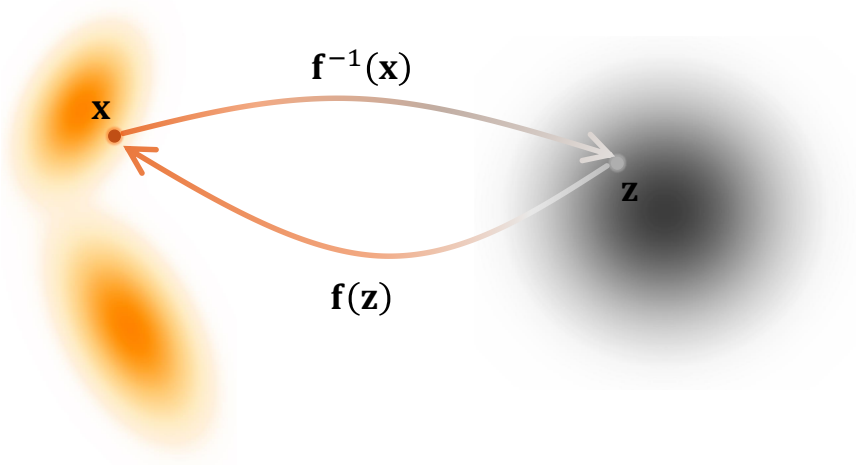


图 5.1: NF 在可逆映射下的样本运动示意图。它由一系列可逆函数 $\mathbf{f} : \mathbf{z} \mapsto \mathbf{x}$ 组成，这些函数将潜变量 $\mathbf{z}$ 变换为数据 $\mathbf{x}$，以及对应的逆映射 $\mathbf{f}^{-1} : \mathbf{x} \mapsto \mathbf{z}$ 用于重构数据。NF 类似于编码器–解码器结构，但编码器由光滑的可逆映射实现，而解码器则恰好为其逆映射。相应的密度变化可通过变量变换公式计算，如 Equation (5.0.1) 所示。

5.1.1 归一化流

NFs (**rezende2015variational**) 通过可逆映射将一个简单的先验 $p_{\text{prior}}(\mathbf{z})$（例如标准高斯分布 $\mathcal{N}(\mathbf{0}, \mathbf{I})$）变换为复合数据分布 $p_{\text{data}}(\mathbf{x})$，从而对复杂数据分布 $p_{\text{data}}(\mathbf{x})$ 建模

$$\mathbf{f}_{\phi} : \mathbb{R}^D \to \mathbb{R}^D,$$

与 $\mathbf{x} = \mathbf{f}_{\boldsymbol{\phi}}(\mathbf{z})$ 和 $\mathbf{z} \sim p_{\text{prior}}$。这里，$\mathbf{x}$ 和 $\mathbf{z}$ 共享相同的维度。使用 Equation (5.0.1) 中的变量变换公式，模型似然为[1]

$$\log p_{\boldsymbol{\phi}}(\mathbf{x}) = \log p_{\text{prior}}(\mathbf{z}) + \log \left| \det \frac{\partial \mathbf{f}_{\boldsymbol{\phi}}^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right|. \tag{5.1.1}$$

训练目标。 参数 $\boldsymbol{\phi}$ 通过在数据上最大化似然来学成：

$$\mathcal{L}_{\text{NF}}(\boldsymbol{\phi}) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \left[\log p_{\boldsymbol{\phi}}(\mathbf{x}) \right]. \tag{5.1.2}$$

在 Equation (5.1.1) 中计算雅克比行列式可能代价较高，通常其复杂度为 $\mathcal{O}(D^3)$。

构造可逆变换。 单个复杂的可逆网络由于其雅克比行列式可能导致成本较高。相反，简单的变换（例如线性变换）虽然高效，但表达能力不足。

为了平衡这一点，NFs 采用了一系列 K 可训练的可逆映射 $\{\mathbf{f}_k\}_{k=0}^{L-1}$，每个映射都具有高效可计算的雅克比。

$$\mathbf{f}_{\boldsymbol{\phi}} = \mathbf{f}_{L-1} \circ \mathbf{f}_{L-2} \circ \cdots \circ \mathbf{f}_0.$$

每个 $\mathbf{f}_k$ 都由一个神经网络参数化，尽管为了记号简洁，我们省略了对 $\boldsymbol{\phi}$ 的显式依赖。

样本通过变换

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k), \quad k = 0, \ldots, L-1, \tag{5.1.3}$$

使用 $\mathbf{z} = \mathbf{x}_0 \sim p_{\text{prior}}$ 和 $\mathbf{x} = \mathbf{x}_L$，分别对应数据。得到的（对数）密度推导如下

$$\begin{aligned} p_{\boldsymbol{\phi}}(\mathbf{x}) &= p_{\text{prior}}(\mathbf{x}_0) \prod_{k=0}^{L-1} \left| \det \frac{\partial \mathbf{f}_k}{\partial \mathbf{x}_k} \right|^{-1}, \text{ or equivalently,} \\ \log p_{\boldsymbol{\phi}}(\mathbf{x}) &= \log p_{\text{prior}}(\mathbf{x}_0) + \sum_{k=0}^{L-1} \log \left| \det \frac{\partial \mathbf{f}_k}{\partial \mathbf{x}_k} \right|^{-1}. \end{aligned} \tag{5.1.4}$$

[1] 如果该映射进一步被约束为某个凸势函数的梯度，$\mathbf{f}_{\boldsymbol{\phi}} = \nabla \psi_{\boldsymbol{\phi}}$ 且 $\psi_{\boldsymbol{\phi}}$ 凸，则 Equation (5.1.1) 简化为 Equation (7.2.4) 中的 Monge–Ampère 关系。此偏微分方程刻画了在二次代价下将一个分布最优变换为另一个分布的变换。详见 Chapter 7 和 (**huangconvex**)。

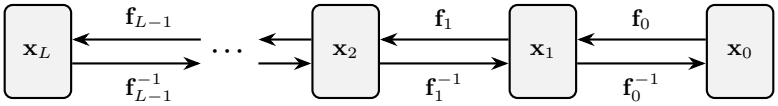


图 5.2: NF 的示意图。 NF 由一系列可逆映射 $\mathbf{f}_\phi = \mathbf{f}_{L-1} \circ \mathbf{f}_{L-2} \circ \cdots \circ \mathbf{f}_0$ 组成。变换将潜在样本 $\mathbf{x}_0 \sim p_{\text{prior}}$ 映射到数据样本 $\mathbf{x}_L \sim p_{\text{data}}$ 。

可逆流的示例。 大量文献集中于设计能够高效计算雅克比的单层流结构。以下，我们介绍两种代表性类型：平面流 (**rezende2015variational**) 和残差流 (**chen2019residual behrmann2019invertible**)，后者推动了 Section 5.1.2 中的发展。

平面流： 它应用一种简单的变换

$$\mathbf{f}(\mathbf{z}) = \mathbf{z} + \mathbf{u}h(\mathbf{w}^\top \mathbf{z} + b),$$

其中 $\mathbf{u}, \mathbf{w} \in \mathbb{R}^D$ 、$b \in \mathbb{R}$ 和 $h(\cdot)$ 为活性值。雅克比行列式为

$$\left|1 + \mathbf{u}^\top h'(\mathbf{w}^\top \mathbf{z} + b)\mathbf{w}\right|.$$

残差流： 将变换 $\mathbf{f}$ 定义为

$$\mathbf{f}(\mathbf{z}) = \mathbf{z} + \mathbf{v}(\mathbf{z}), \tag{5.1.5}$$

其中 $\mathbf{v}$ 为收缩映射（Lipschitz 常数为 < 1 ）。这保证了通过巴拿赫不动点定理可求逆。

雅克比矩阵的对数行列式可简化为迹展开：

$$\begin{aligned}
\log \left| \det \frac{\partial \mathbf{f}(\mathbf{z})}{\partial \mathbf{z}} \right| &= \log \left(\det \frac{\partial \mathbf{f}(\mathbf{z})}{\partial \mathbf{z}} \right) \\
&= \operatorname{Tr} \left(\log \left(\frac{\partial \mathbf{f}(\mathbf{z})}{\partial \mathbf{z}} \right) \right) \\
&= \operatorname{Tr} \left(\log \left(\mathbf{I} + \frac{\partial \mathbf{v}(\mathbf{z})}{\partial \mathbf{z}} \right) \right) \\
&= \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k} \operatorname{Tr} \left(\left(\frac{\partial \mathbf{v}(\mathbf{z})}{\partial \mathbf{z}} \right)^k \right),
\end{aligned} \tag{5.1.6}$$

通过迹估计量提高评估效率 (**hutchinson1989stochastic**).

采样与推断。 从 NF 中采样是直接的：抽取 $\mathbf{x}_0 \sim p_{\text{prior}}$ 并计算 $\mathbf{x} = \mathbf{f}_{\boldsymbol{\phi}}(\mathbf{x}_0)$。确切的似然值通过 Equation (5.1.4) 获得。

5.1.2 神经 ODE

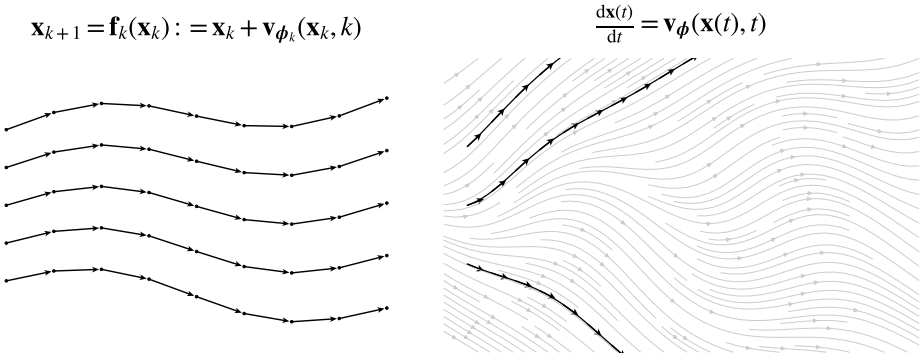


图 5.3: 离散型与连续型归一化流。(左) 离散型归一化流通过有限序列的可逆映射 $\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k)$ 传输样本，产生分步的、不交叉的轨迹（带箭头的点）。(右) 连续型归一化流（神经微分方程）沿 $\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}(t), t)$ 的积分曲线演化状态，其中黑色路径及其切向箭头显示在灰色向量场之上。

从离散时间神经网络到连续时间神经网络（神经微分方程）。 NFs 通常被表述为一系列 L 离散的、可逆变换。从 Equation (5.1.3) 和 Equation (5.1.5) 中的“残差流”表述视角来看，每一层可以表示为以下形式：

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) := \mathbf{x}_k + \mathbf{v}_{\boldsymbol{\phi}_k}(\mathbf{x}_k, k),$$

其中 $\mathbf{v}_{\boldsymbol{\phi}_k}(\cdot, k)$ 是一个由神经网络参数化的与层相关的速度场。直观上，这个速度场是一个学成的向量值函数，它在输入空间中“推动”数据点以小而平滑的步骤前进。每次变换都沿着该速度场建议的方向移动点，逐步将简单的先验分布变形为目标分布。

这种表述实际上对应于带有可学习参数 $\boldsymbol{\phi}$ 的连续时间微分方程的欧拉离散化：

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}(t), t).$$

在层的数量趋于无穷且步长趋近于零（$\Delta t \to 0$）的极限情况下，离散的 NFs 收敛到一个连续模型，从而得到 *神经微分方程*（NODEs）（**chen2018neural**）的框架，也称为 *连续归一化流*（CNFs）。

神经微分方程的形式化设定。 神经微分方程通过以下方式定义连续变换：

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}(t), t), \quad t \in [0, T] \tag{5.1.7}$$

其中：

- $\mathbf{x}(t) \in \mathbb{R}^D$ 是时间 t 时的状态；为简洁起见，我们有时写作 $\mathbf{x}_t$ 。
- $\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}(t), t)$ 是一个由 $\boldsymbol{\phi}$ 参数化的神经网络。

NODE 的目标。 从初始条件 $\mathbf{x}(0) \sim p_{\text{prior}}$ 出发，微分方程在时间上连续地演化状态，从而诱导出一族边缘分布 $p_{\boldsymbol{\phi}}(\mathbf{x}_t, t)$ （类似于 PF-ODE!）[2].

目标是学习神经向量场 $\mathbf{v}_{\boldsymbol{\phi}}$ ，其直观地表示在数据空间中沿连续轨迹传输点的速度。通过学习该速度，$t = 0$ 时刻的终态分布将匹配目标分布 $p_{\text{data}}(\cdot)$ 。这种连续变换将离散归一化流和神经微分方程统一在一个框架之中。

连续时间变量变换公式。 类似于 Equation (5.0.1) 或 Equation (5.1.4)，**chen2018neural** 推导出了变量变换公式的一个连续时间版本。对于在 Equation (5.1.7) 作用下演化的过程 $\mathbf{x}(t)$ 的时变密度 $p_{\boldsymbol{\phi}}(\mathbf{x}(t), t)$，所谓的 瞬时变量变换公式为：

$$\frac{\mathrm{d}}{\mathrm{d}t} \log p_{\boldsymbol{\phi}}(\mathbf{x}(t), t) = -\nabla_{\mathbf{x}} \cdot \mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}(t), t).$$

因此，给定先验 $p_{\text{prior}}(\mathbf{x}(T), T)$ ，由神经微分方程诱导的终止状态 $\mathbf{x}(T)$ 的对数密度为

$$\log p_{\boldsymbol{\phi}}(\mathbf{x}(T), T) = \log p_{\text{prior}}(\mathbf{x}(0), 0) - \int_0^T \nabla_{\mathbf{x}} \cdot \mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}(t), t) \mathrm{d}t. \tag{5.1.8}$$

该表达式通过数值求解常微分方程实现了确切似然评估，从而使得模型的极大似然训练成为可能。我们稍后将详细讨论这一点。

尽管起初可能显得不熟悉，这种瞬时变量变换公式是福克-普朗克方程的一个特例，即其确定性形式，被称为 连续性方程（参见 Chapter B）。它也可以被解释为 Equation (5.1.4) 的连续时间极限。我们在下面的引理中总结这一结果及其推导过程：

[2]我们采用翻转的时间约定，其中 $t = 0$ 表示先验（源）分布，$t = 1$ 表示数据（目标）分布。先验可交替写作 $p_{\boldsymbol{\phi}}(\mathbf{x}(0), 0)$、$p_{\text{prior}}(\mathbf{x}(0), 0)$，或简写为 $p_{\text{prior}}(\mathbf{z})$。

Lemma 5.1.1: Instantaneous Change of Variables

Let $\mathbf{z}(t)$ be a continuous random process with time-dependent density $p(\mathbf{z}(t), t)$, and suppose it evolves according to the ODE

$$\frac{\mathrm{d}\mathbf{z}(t)}{\mathrm{d}t} = \mathbf{F}(\mathbf{z}(t), t).$$

Assuming $\mathbf{F}$ is uniformly Lipschitz in $\mathbf{z}$ and continuous in t, the time derivative of the log-density satisfies:

$$\frac{\partial \log p(\mathbf{z}(t), t)}{\partial t} = -\nabla_{\mathbf{z}} \cdot \mathbf{F}(\mathbf{z}(t), t). \tag{5.1.9}$$

Proof for Lemma.

We present two alternative derivations in Section D.3.

与离散时间公式的联系。 在 Equation (5.1.8) 中的 NODE 似然，

$$\log p_{\boldsymbol{\phi}}(\mathbf{x}(T), T) = \log p_{\text{prior}}(\mathbf{x}(0), 0) - \int_0^T \nabla_{\mathbf{x}} \cdot \mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}(t), t)\, \mathrm{d}t,$$

可以看作是公式 Equation (5.1.4) 中离散归一化流公式的连续时间类比。

$$\log p_{\boldsymbol{\phi}}(\mathbf{x}_L) = \log p_{\text{prior}}(\mathbf{x}_0) - \sum_{k=0}^{L-1} \log \left| \det \frac{\partial \mathbf{f}_k}{\partial \mathbf{x}_k} \right|.$$

积分反映了求和，且迹算子取代了对数行列式，如 Equation (5.1.6) 中所讨论。这些类比在引理的证明中得到了进一步探讨。

训练结点。 基于 Equation (5.1.8)，结点学习一个参数化的速度场 $\mathbf{v}_{\boldsymbol{\phi}}$，使得终态分布

$$p_{\boldsymbol{\phi}}(\cdot, T) \approx p_{\text{data}},$$

其中轨迹从潜变量 $\mathbf{x}(0) \sim p_{\text{prior}}$ 通过常微分方程流演化。训练遵循 Equation (1.1.2) 中的极大似然估计框架：

$$\mathcal{L}_{\text{NODE}}(\boldsymbol{\phi}) := \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \big[\log p_{\boldsymbol{\phi}}(\mathbf{x}, T) \big].$$

确切的对数似然计算。 为了计算数据点 $\mathbf{x}$ 的 $\log p_{\boldsymbol{\phi}}(\mathbf{x}, T)$ ，我们对变量变换公式 equation 5.1.8 进行积分：

$$\log p_{\boldsymbol{\phi}}(\mathbf{x}, T) = \log p_{\text{prior}}(\mathbf{z}(0)) - \int_0^T \nabla_{\mathbf{z}} \cdot \mathbf{v}_{\boldsymbol{\phi}}(\mathbf{z}(t), t)\, \mathrm{d}t. \tag{5.1.10}$$

此处，$\mathbf{z}(t)$ 从 $t = T$ 逆向求解常微分方程至 $t = 0$：

$$\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} = \mathbf{v}_{\boldsymbol{\phi}}(\mathbf{z}(t), t)$$

其中 $\mathbf{z}(T) = \mathbf{x}$。先验项 $\log p_{\text{prior}}(\mathbf{z}(0))$ 对于标准分布是易处理的。这使得神经微分方程能够进行基于确切似然的训练和评估。

基于梯度的最优化。 最大化 $\mathcal{L}_{\text{NODE}}$ 需要通过常微分方程求解器进行反向传播。伴随敏感性方法 (**pontryagin2018mathematical**; **chen2018neural**) 通过一个辅助常微分方程计算梯度，具有 $\mathcal{O}(1)$ 的内存复杂度，但由于每一步都需要数值积分，结点训练仍然成本高昂。

使用结点进行推理。 使用训练好的模型 $\mathbf{v}_{\boldsymbol{\phi}^\times}$ 进行采样，通过绘制 $\mathbf{x}(0) \sim p_{\text{prior}}$ 并向前积分（通过数值求解器）：

$$\mathbf{x}(T) = \mathbf{x}(0) + \int_0^T \mathbf{v}_{\boldsymbol{\phi}^\times}(\mathbf{x}(t), t)\, \mathrm{d}t.$$

终止状态 $\mathbf{x}(T)$ 近似于来自 p_{data} 的一个样本。

此外，我们注意到对于任意向量场 $\mathbf{F}$ ，以下恒等式成立：

$$\operatorname{Tr}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{z}(t)}\right) = \nabla_{\mathbf{z}} \cdot \mathbf{F}.$$

因此，散度可以使用随机迹估计量（如 Hutchinson 估计量 (**hutchinson1989stochastic**)）高效地进行估计，这使得在高维情景下进行确切似然计算变得更加易处理。

5.2 Flow Matching Framework

得分 SDEs (Chapter 4) 和 NODEs (Section 5.1) 为生成式建模提供了另一种视角：学习一种连续时间流，无论是随机的还是确定性的，该流将简单的高斯先验样本 $\boldsymbol{\epsilon} \sim p_{\text{prior}}$ 转换为来自 p_{data} 的数据类样本。

流匹配（Flow Matching, FM）框架 (**lipman2022flow**; **lipman2024flow**; **tongimproving**) 基于这一思想，但将其泛化为学习两个任意固定端点分布之间的流：一个源分布 p_{src} 和一个目标分布 p_{tgt} ，两者均假设易于采样。在这一更广泛的设置中，生成任务成为一种特殊情况，其中 p_{src} 为高斯先验，p_{tgt} 为数据分布。

在本节中，我们采用 FM 观点。[3]，强调其核心原理：学习一个随时间变化的向量场 $\mathbf{v}_t(\mathbf{x}_t)$，其对应的常微分方程流与预定义的概率路径 $\{p_t\}_{t\in[0,1]}$ 相匹配，且满足边界条件

$$p_0 = p_{\text{src}}, \quad p_1 = p_{\text{tgt}}.$$

当 p_{src} 服从高斯分布时，我们将此情景称为 高斯流匹配。与经典扩散模型相比，流匹配（FM）能够仅使用端点样本，实现对广泛一类传输问题的高效、无需仿真的训练。

5.2.1 基于得分方法的启示

我们重新审视 Score SDE 框架 (Chapter 4)，采用一种略有不同但等价的表述方式，以提取关键见解，从而启发 FM 方法。该分析揭示了扩散模型如何隐式地学习概率流，并促使提出一种更直接的表述方式。

第一步：定义条件路径及其边缘密度。 一个扩散模型指定了一个连续时间密度族 $\{p_t\}_{t\in[0,1]}$ ，该族将简单的先验 p_{prior} （例如，高斯分布）在 $t = 1$ 处（用作源）传输到目标数据分布 p_{data} 在 $t = 0$ 处：

$$p_1(\mathbf{x}_1) = p_{\text{prior}}(\mathbf{x}_1), \quad p_0(\mathbf{x}_0) = p_{\text{data}}(\mathbf{x}_0).$$

[3]几种相关的方法共享了利用连续时间流在端点分布之间进行传输的核心思想，尽管其表述略有不同。这些方法包括流匹配（FM） (**lipman2022flow**; **neklyudov2023action**)、校正流（RF） (**liu2022rectified**; **heitz2023iterative**) 和随机插值 (**albergo2023stochastic**; **albergobuilding**; **ma2024sit**)。在此，我们采用 FM 的术语作为统一的表示。

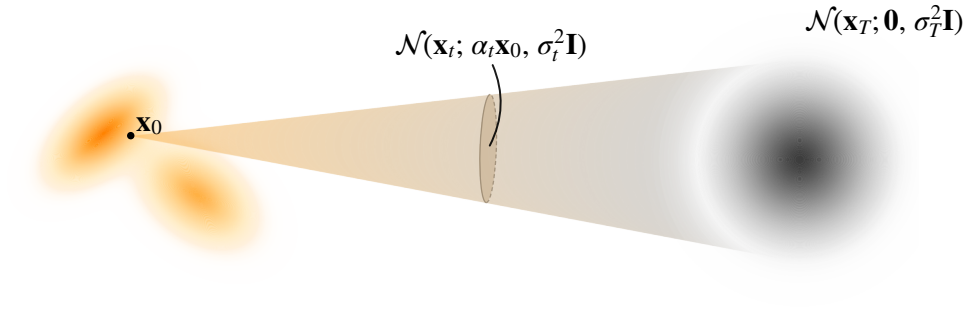


图 5.4: 条件转移分布的示意图。 $p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \alpha_t\mathbf{x}_0, \sigma_t^2\mathbf{I})$，定义了从数据样本 $\mathbf{x}_0 \sim p_{\text{data}}$（左侧）到高斯先验 p_{prior}（右侧）的（高斯）条件概率路径。

该路径通过前向条件分布隐式定义

$$p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \alpha_t\mathbf{x}_0, \sigma_t^2\mathbf{I}), \quad \mathbf{x}_0 \sim p_{\text{data}} \tag{5.2.1}$$

这导致了边缘密度

$$p_t(\mathbf{x}_t) := \int p_t(\mathbf{x}_t|\mathbf{x}_0)p_{\text{data}}(\mathbf{x}_0)\mathrm{d}\mathbf{x}_0.$$

条件高斯分布的方差 σ_t^2 越来越大，推动 p_t 向高斯先验演化。

步骤 2：速度场。 边际密度 p_t 的时间演化由速度场 $\mathbf{v}_t : \mathbb{R}^D \to \mathbb{R}^D$ 控制，该速度场来源于福克-普朗克方程：

$$\mathbf{v}_t(\mathbf{x}) := f(t)\mathbf{x} - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x}), \tag{5.2.2}$$

这定义了通过 PF-ODE 的确定性粒子流：

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \underbrace{f(t)\mathbf{x}(t) - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}} \log p_t\big(\mathbf{x}(t)\big)}_{\mathbf{v}_t(\mathbf{x}(t))}.$$

该常微分方程将初始随机变量 $\mathbf{x}(0) \sim p_{\text{data}}$ 向前或向后时间传输至 $\mathbf{x}(1) \sim p_{\text{prior}}$，使得 $\mathbf{x}(t)$ 的演化边缘密度在每个 $t \in [0, 1]$ 处均与 p_t 相匹配（参见下方“潜在规则”）。

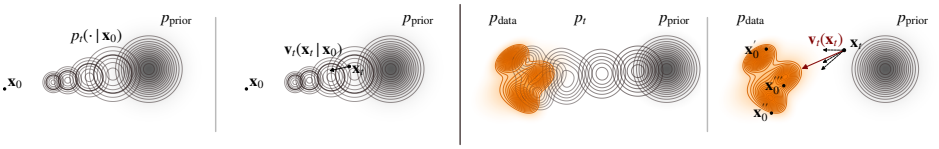


图 5.5: 扩散过程中条件视角与边际视角的示意图。(本图受 **lipman2024flow** 启发。)(1) 条件高斯路径 $p_t(\cdot|\mathbf{x}_0)$，显示从固定 $\mathbf{x}_0$ 向先验扩展的密度变化。(2) 条件速度 $\mathbf{v}_t(\mathbf{x}_t|\mathbf{x}_0) = f(t)\mathbf{x}_t - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{x}_0)$。(3) 边际密度 p_t，将数据分布 p_{data}（橙色）传输至先验 p_{prior}（灰色）。(4) 边际速度 $\mathbf{v}_t(\mathbf{x}_t) = f(t)\mathbf{x}_t - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)$，通过对从 $\mathbf{x}_t$ 到多个可能原点（虚线）的条件方向进行平均得到，表现为红色箭头。在单侧条件化的 FM 框架中 $\mathbf{z} = \mathbf{x}_0$，该示意图同样适用于 $\mathbf{v}_t(\mathbf{x}_t|\mathbf{x}_0)$ 和 $\mathbf{v}_t(\mathbf{x}_t)$，且无需显式地用得分 $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{x}_0)$ 或 $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)$ 表示它们。

标量函数 $f(t)$ 和 $g(t)$ 由相关前向 SDE 的系数决定，或等价地由条件路径中定义的高斯核参数 α_t 和 σ_t 决定（见引理 4.4.1）。

第三步：通过条件策略进行学习。 目标是使用神经网络 $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)$ 通过期望平方误差进行训练，来逼近虚设速度场 $\mathbf{v}_t(\mathbf{x}_t)$。

$$\mathcal{L}_{\text{SM}}(\boldsymbol{\phi}) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\mathbf{x}_t\sim p_t}\left[\left\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t,t) - \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)\right\|^2\right].$$

由于边际得分 $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)$ 不可获取，我们利用易处理的条件分布来定义条件速度：

$$\mathbf{v}_t(\mathbf{x}_t|\mathbf{x}_0) := f(t)\mathbf{x}_t - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{x}_0).$$

根据全期望定律，边际得分被恢复为

$$\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t) = \mathbb{E}_{\mathbf{x}_0\sim p(\cdot|\mathbf{x}_t)}\left[\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{x}_0)\right]. \tag{5.2.3}$$

这证明了代理训练目标的合理性：

$$\mathcal{L}_{\text{SM}}(\boldsymbol{\phi}) = \underbrace{\mathbb{E}_{t,\mathbf{x}_0\sim p_{\text{data}},\mathbf{x}_t\sim p_t(\cdot|\mathbf{x}_0)}\left[\left\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t,t) - \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{x}_0)\right\|^2\right]}_{\mathcal{L}_{\text{DSM}}(\boldsymbol{\phi})} + C,$$

其中 C 是与 $\boldsymbol{\phi}$ 无关的常数。最小化器 $\mathbf{s}^*(\mathbf{x}_t, t)$ 满足

$$\mathbf{s}^*(\mathbf{x}_t,t) = \mathbb{E}_{\mathbf{x}_0\sim p(\cdot|\mathbf{x}_t)}\left[\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{x}_0)\right] = \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t),$$

其中第二个等式由 Equation (5.2.3) 推出，从而验证了条件训练目标。

潜在规则：福克-普朗克方程。 边际密度 p_t 按照福克-普朗克方程演化：

$$\frac{\partial p_t(\mathbf{x})}{\partial t} + \nabla \cdot \left(\underbrace{\left(f(t)\mathbf{x} - \frac{1}{2} g^2(t) \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right)}_{\mathbf{v}_t(\mathbf{x})} p_t(\mathbf{x}) \right) = 0.$$

该偏微分方程确保由 PF-ODE 给出的密度与前向 SDE 的边缘分布相匹配。要理解这一点，回顾一下如 Equation (4.1.9) 中所定义的 PF-ODE 的流映射 $\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s)$，它将时间 s 处的初始状态 $\mathbf{x}_s$ 直接映射到时间 t 的状态。从 $t=1$ 到 $t=0$ 反向运行 PF-ODE，起始于 $\mathbf{x}_1 \sim p_{\text{prior}}$，我们通过前推公式得到随时间变化的密度：

$$p_t^{\text{rev}}(\mathbf{x}) = \int \delta\left(\mathbf{x} - \boldsymbol{\Psi}_{1\to t}(\mathbf{x}_1)\right) p_{\text{prior}}(\mathbf{x}_1) \mathrm{d}\mathbf{x}_1. \tag{5.2.4}$$

福克-普朗克方程确保诱导密度路径与同一演化密度相一致：

$$p_t^{\text{rev}} = p_t. \tag{5.2.5}$$

特别地，这暗示了 $p_0^{\text{rev}} = p_0 = p_{\text{data}}$ ，从而恢复了时间 $t=0$ 处的数据分布。由于常微分方程解映射是双向的，我们可以类似地从 $\mathbf{x}_0 \sim p_{\text{data}}$ 开始初始化并向前求解常微分方程，从而实现并行分析。

5.2.2 流匹配框架

Section 5.2.1 中的分析表明，扩散模型通过学习一个速度场（具体来说是得分）来实现成功，该速度场在满足边界条件的同时实现了分布之间的转换。Equation (5.2.1) 中设计的高斯条件路径，其方差 σ_t^2 逐渐增加，隐式地将一个端点锚定在高斯先验上，同时允许条件密度在整个空间上定义，从而实现了基于得分的梯度计算。

在本小节中，我们介绍了 FM 框架，该框架基于这一洞察（Figure 5.5 中的示意图同样适用于 FM 框架），并将其扩展至学习连续流，以在两个任意分布 p_{src} 和 p_{tgt} 之间传输样本。

第一步：定义条件路径及其边缘密度。 考虑任意源概率分布 p_{src} 和目标概率分布 p_{tgt} 在 $\mathbb{R}^D$ 上。我们设定[4]

$$p_0(\mathbf{x}) = p_{\text{src}}(\mathbf{x}), \quad p_1(\mathbf{x}) = p_{\text{tgt}}(\mathbf{x}). \tag{5.2.6}$$

FM 隐式地定义了一个连续的中间密度族 $\{p_t\}_{t\in[0,1]}$，该族在这些端点之间进行插值。每个边缘分布 p_t 通过一个从已知分布 $\pi(\mathbf{z})$ 中抽取的潜变量 $\mathbf{z}$ 以及一个条件分布 $p_t(\mathbf{x}_t|\mathbf{z})$ 来表示。

$$p_t(\mathbf{x}_t) = \int p_t(\mathbf{x}_t|\mathbf{z})\pi(\mathbf{z})\,\mathrm{d}\mathbf{z}, \tag{5.2.7}$$

其中 $(\pi(\mathbf{z}), \{p_t(\cdot|\mathbf{z})\})$ 的选取需满足 Equation (5.2.6) 中的边界条件。

我们注意到，通常情况下，边缘密度 p_t 并不易处理，因为它们需要对 $\pi(\mathbf{z})$ 进行积分，而 $\pi(\mathbf{z})$ 以及条件分布 $p_t(\mathbf{x}_t|\mathbf{z})$ 都可能非常复杂。尽管如此，对潜在变量 $\mathbf{z}$ 进行条件化使得 FM 能够灵活地建模超出 Section 5.2.1 所讨论范围的广泛插值路径。$\mathbf{z}$ 的常见选择包括：

- **双边条件**：$\mathbf{z} = (\mathbf{x}_0, \mathbf{x}_1) \sim p_{\text{src}}(\mathbf{x}_0)p_{\text{tgt}}(\mathbf{x}_1)$，其中 π 将源分布和目标分布耦合在一起。这使得 FM 能够定义任意分布之间的传输。
- **单边条件化**：$\mathbf{z} = \mathbf{x}_0$ 或 $\mathbf{z} = \mathbf{x}_1$。当源分布选择为高斯分布时，它尤其能恢复扩散类似的情形。

在所有情况下，条件分布 $p_t(\mathbf{x}_t|\mathbf{z})$ 应具有易处理的闭式表达式。我们始终做此假设，并在 Section 5.3.2 中给出具体构造，Figure 5.6 中附有图示说明。

步骤 2：速度场。 在标准扩散模型或高斯 FM 中，中间密度 $\{p_t\}_{t\in[0,1]}$ 的构建方式是将一个端点设为标准高斯分布。在此情景下，速度场 $\mathbf{v}_t$ 被唯一确定，并且与得分相关联，存在闭式表达式（见 Equation (5.2.2)）。

相比之下，通用 FM 在通用源分布 p_{src} 和目标分布 p_{tgt} 之间进行插值，此时速度场不再唯一确定（稍后将解释原因）。

目标是找到一个速度场 $\mathbf{v}_t(\mathbf{x})$，使得由此产生的微分方程能够实现样本相关的变换。

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{v}_t(\mathbf{x}(t)), \quad t \in [0,1],$$

[4]为了与 FM 文献中的标准记法一致，我们相对于前面各节反转了时间轴：$t = 0$ 对应源分布，$t = 1$ 对应目标。

在每个时间点 t，无论从 $\mathbf{x}(0) \sim p_{\text{src}}$ 正向积分还是从 $\mathbf{x}(1) \sim p_{\text{tgt}}$ 反向积分，均能产生与 p_t 一致的 $\mathbf{x}(t)$ 的边缘分布（详见 Section 5.2.4 的更正式讨论）。

此要求由*连续性方程*描述[5]：

$$\frac{\partial p_t(\mathbf{x})}{\partial t} + \nabla \cdot \big(\mathbf{v}_t(\mathbf{x}) p_t(\mathbf{x})\big) = 0. \tag{5.2.8}$$

任何满足 Equation (5.2.8) 的速度场 $\mathbf{v}_t$ 确保了常微分方程流以确切的方式传输样本，遵循预设的 p_t（详见 Section 5.2.4）。因此，求解常微分方程可实现从 p_{src} 到 p_{tgt} 的传输，同时匹配所有中间分布。

直观地，许多不同的流可以诱导出相同的边缘演化。这是因为 Equation (5.2.8) 是一个标量方程，而 $\mathbf{v}_t$ 是 $\mathbb{R}^D$ 中的一个向量场，所以该方程有无限多个解。例如，如果 $\mathbf{v}_t$ 解该方程，那么...

$$\mathbf{v}_t + \frac{1}{p_t}\tilde{\mathbf{v}}_t,$$

对于任意无散度向量场 $\tilde{\mathbf{v}}_t$（即 $\nabla \cdot \tilde{\mathbf{v}}_t = 0$）。因此，FM 寻求一个特定的速度场 $\mathbf{v}_t$，使其满足 Equation (5.2.8)，从而实现样本沿路径 $\{p_t\}$ 的连续传输。然而，对于任意分布，p_t 和 $\mathbf{v}_t$ 通常无法以闭式表达。作为一个具体示例，在 Section 5.3.1 中我们考虑高斯到高斯的桥接过程，其中这两个量均可显式计算。

第三步：通过条件策略进行学习。 FM 训练的目标是使用神经网络 $\mathbf{v}_\phi$ 近似原始速度场 $\mathbf{v}_t$，通过最小化期望平方误差：

$$\mathcal{L}_{\text{FM}}(\phi) = \mathbb{E}_{t,\mathbf{x}_t \sim p_t}\left[\|\mathbf{v}_\phi(\mathbf{x}_t, t) - \mathbf{v}_t(\mathbf{x}_t)\|^2\right].$$

我们将这种神经网络参数化称为 $\mathbf{v}$ -预测（速度预测），其目标是直接学习微分方程的漂移项。

如 Section 5.2.1 所示，最优速度 $\mathbf{v}_t(\mathbf{x})$ 通常难以处理。为解决此问题，我们引入一个潜变量 $\mathbf{z} \sim \pi(\mathbf{z})$，并通过构造定义一个条件速度场 $\mathbf{v}_t(\mathbf{x}|\mathbf{z})$。这使我们能够通过全期望定律重写损失[6]：

[5] 无扩散项的 Fokker–Planck 方程的确定性类比。

[6] 这遵循标准的分部积分论证，如 Equation (3.3.3) 的推导所示。同样地，Equation (5.2.11) 也是在分数匹配框架内采用类似方法推导得出。

$$\mathcal{L}_{\mathrm{FM}}(\boldsymbol{\phi}) = \underbrace{\mathbb{E}_{t,\mathbf{z}\sim\pi(\mathbf{z}),\mathbf{x}_t\sim p_t(\cdot|\mathbf{z})}\left[\left\|\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t,t)-\mathbf{v}_t(\mathbf{x}_t|\mathbf{z})\right\|^2\right]}_{\mathcal{L}_{\mathrm{CFM}}(\boldsymbol{\phi})} + C, \tag{5.2.9}$$

其中 C 是与 $\boldsymbol{\phi}$ 无关的常数。主项 $\mathcal{L}_{\mathrm{CFM}}$ 称为 条件流匹配。

也就是说，最小化 $\mathcal{L}_{\mathrm{FM}}(\boldsymbol{\phi})$ 等价于最小化 $\mathcal{L}_{\mathrm{CFM}}(\boldsymbol{\phi})$ ，而后者提供了更易处理的公式。为了使 $\mathcal{L}_{\mathrm{CFM}}(\boldsymbol{\phi})$ 实现无需仿真的易处理训练，必须满足两个条件：

(i) 从条件概率路径 $p_t(\mathbf{x}_t|\mathbf{z})$ 采样应该是直接的（无需仿真）。

(ii) 条件速度 $\mathbf{v}_t(\mathbf{x}_t|\mathbf{z})$ ，用作回归目标，必须具有简单的闭式表达式。

我们将提供满足这些条件的显式构造方法，详见 Section 5.3.2。这种条件视角使得训练成为可行：模型不再需要学习难以处理的无条件速度场 $\mathbf{v}_t(\cdot)$ ，而是学习易处理的条件场 $\mathbf{v}_t(\cdot|\mathbf{z})$ ： 这与降噪分数匹配具有直接类比关系。

尽管存在无穷多个与给定 p_t 一致的无条件速度场，但可以通过对条件速度场进行边缘化来恢复其中一种场：

$$\mathbf{v}_t(\mathbf{x}_t) := \mathbb{E}_{\mathbf{z}\sim p(\cdot|\mathbf{x}_t)}\left[\mathbf{v}_t(\mathbf{x}_t|\mathbf{z})\right], \tag{5.2.10}$$

其中期望是关于 $p(\mathbf{z}|\mathbf{x}_t)$ 取的。我们可以证明，条件流匹配目标在 Equation (5.2.9) 中的最小化器 $\mathbf{v}^*$ 恢复了该边缘速度：

$$\mathbf{v}^*(\mathbf{x}_t,t) = \mathbf{v}_t(\mathbf{x}_t). \tag{5.2.11}$$

因此，学习匹配条件速度场 $\mathbf{v}_t(\cdot|\mathbf{z})$ 就足以恢复一个有效的无条件速度场。

我们将上述讨论总结如下：

Theorem 5.2.1: $\mathcal{L}_{\text{FM}}$ 与 $\mathcal{L}_{\text{CFM}}$ 的等价性

下列关系成立：

$$\mathcal{L}_{\text{FM}}(\boldsymbol{\phi}) = \mathcal{L}_{\text{CFM}}(\boldsymbol{\phi}) + C,$$

其中 C 是与参数 $\boldsymbol{\phi}$ 无关的常数。此外，两个损失的最小化子 $\mathbf{v}^*$ 满足

$$\mathbf{v}^*(\mathbf{x}_t, t) = \mathbf{v}_t(\mathbf{x}_t), \quad \text{对几乎所有} \mathbf{x}_t \sim p_t,$$

其中 $\mathbf{v}_t(\mathbf{x}_t)$ 在 Equation (5.2.10) 中定义。

Proof for Theorem.

该最小化子的论证和推导完全遵循命题 4.2.1 中分数匹配情形的相同推理过程。

这标志着条件技巧第三次产生易处理的训练目标。值得注意的是，变分法、基于得分的方法以及基于流的方法均反映了相同的潜在原理。

Remark.

取 $\pi = p_{\text{data}}$ ，我们可以应用贝叶斯规则：

$$p(\mathbf{x}_0|\mathbf{x}_t) = \frac{p_t(\mathbf{x}_t|\mathbf{x}_0)p_{\text{data}}(\mathbf{x}_0)}{p_t(\mathbf{x}_t)},$$

基于得分的模型中出现了类似的 Equation (5.2.10) 分解：

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) &= \mathbb{E}_{\mathbf{x}_0 \sim p(\cdot|\mathbf{x}_t)} \left[\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)\right] \\ &= \mathbb{E}_{\mathbf{x}_0 \sim p_{\text{data}}} \left[\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0) \cdot \frac{p_t(\mathbf{x}_t|\mathbf{x}_0)}{p_t(\mathbf{x}_t)}\right],\end{aligned}$$

这反映了 Equation (5.2.10) 中的边缘化策略。

如 Section 5.2.1 所示，条件密度 $p_t(\mathbf{x}_t|\mathbf{z})$ 和条件速度场 $\mathbf{v}_t(\mathbf{x}_t|\mathbf{z})$ 必须显式指定，对于 $p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \alpha_t\mathbf{x}_0, \sigma_t^2\mathbf{I})$ 和 $\mathbf{v}_t(\mathbf{x}_t|\mathbf{x}_0) = f(t)\mathbf{x}_t - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)$，一般条件流匹配框架同样需要这两个组成部分。然而，在此一般情况下，我们尚未构造出条件密度 $p_t(\mathbf{x}_t|\mathbf{z})$ 或条件速度场 $\mathbf{v}_t(\mathbf{x}_t|\mathbf{z})$。在下一节中，我们将介绍这些组成部分的几种常见实例。

5.2.3 扩散模型、通用流匹配与 NODEs 的比较

扩散模型与通用流匹配的比较。 Section 5.2.1 提供的洞察促使我们扩展了 FM 框架，该框架保留了相同的潜在原则。为了突出它们的相似性，我们在 Table 5.1 中对其进行总结。

表 5.1: 扩散模型（或高斯 FM）与通用 FM 框架之间的比较。此处，通用 FM 框架指的是双侧条件设置，其中 $\mathbf{x}_0 \sim p_{\text{src}}$ 和 $\mathbf{x}_1 \sim p_{\text{tgt}}$ 被独立采样。

Aspect	Diffusion Model	General FM
Source dist. p_{src}	Gaussian prior	Any
Target dist. p_{tgt}	Data distribution	Any
Latent dist. $\pi(\mathbf{z})$	p_{data}	See Section 5.3.2
Cond. dist. $p_t(\mathbf{x}_t\|\mathbf{z})$	$\mathcal{N}(\mathbf{x}_t; \alpha_t\mathbf{x}_0, \sigma_t^2\mathbf{I})$	See Section 5.3.2
Marginal dist. $p_t(\mathbf{x}_t)$	$\int p_t(\mathbf{x}_t\|\mathbf{x}_0)p_{\text{data}}(\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_0$	$\int p_t(\mathbf{x}_t\|\mathbf{z})\pi(\mathbf{z})\,\mathrm{d}\mathbf{z}$
Cond. velocity $\mathbf{v}_t(\mathbf{x}\|\mathbf{z})$	$f(t)\mathbf{x} - \frac{1}{2}g^2(t)\nabla \log p_t(\mathbf{x}\|\mathbf{x}_0)$	See Section 5.3.2
Marginal velocity $\mathbf{v}_t(\mathbf{x})$	$f(t)\mathbf{x} - \frac{1}{2}g^2(t)\nabla \log p_t(\mathbf{x})$	See Equation (5.2.10)
Learning objective	$\mathcal{L}_{\text{SM}} = \mathcal{L}_{\text{DSM}} + C$	$\mathcal{L}_{\text{FM}} = \mathcal{L}_{\text{CFM}} + C$
Underlying Rule	Fokker-Planck / Continuity Equation	

我们注意到，由于高斯 FM 在本质上等价于标准扩散模型（详见 Chapter 6），除非特别说明，否则我们将不区分二者。

连接到结点。 FM 可被视为一种无需仿真的 NODEs 替代方法，如 Section 5.1.2 所介绍。与 CNFs 在极大似然训练过程中需要求解常微分方程（ODE）不同，后者计算开销较大，而 FM 通过简单的回归损失直接回归预设的速度场，从而避免了这一过程。其核心思想在于：当连接源分布和目标分布的边际密度路径固定时，训练期间无需进行确切的仿真。

5.2.4 （可选）潜在规则

连续性方程：质量守恒准则。 类似于 Section 5.2.1 中的 PF-ODE 和 Fokker–Planck 分析，我们现在提出一个准则，用于验证由 ODE 流诱导的密度路径是否与预设路径 $\{p_t\}_{t\in[0,1]}$ 一致。

考虑描述粒子在时变速度场 $\mathbf{v}_t$ 作用下的运动的常微分方程：

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{v}_t\left(\mathbf{x}(t)\right).$$

如 Equation (5.2.4) 所示，该常微分方程为任意 $s, t \in [0,1]$ 定义了一个流映射

$\mathbf{\Psi}_{s\to t}(\mathbf{x}_0)$，特别地，它将初始点 $\mathbf{x}_0 \sim p_{\text{src}}$ 在时间 0 时的状态传输到时间 t 时的状态。在时间 t 时诱导的分布由前推给出

$$p_t^{\text{fwd}}(\mathbf{x}) = \int \delta\left(\mathbf{x} - \mathbf{\Psi}_{0\to t}(\mathbf{x}_0)\right) p_{\text{src}}(\mathbf{x}_0)\mathrm{d}\mathbf{x}_0 =: \mathbf{\Psi}_{0\to t}\#p_{\text{src}}, \tag{5.2.12}$$

使得 $\mathbf{\Psi}_{0\to t}(\mathbf{x}_0) \sim p_t^{\text{fwd}}$ 当且仅当 $\mathbf{x}_0 \sim p_{\text{src}}$ 。类似地，可以通过 $\mathbf{\Psi}_{1\to 0}(\mathbf{x}_1)$ 从 $\mathbf{x}_1 \sim p_{\text{tgt}}$ 反向传输到 p_{src} 。

假设我们给定一个预设的密度路径 $\{p_t\}_{t\in[0,1]}$ ，并构造一个速度场 $\{\mathbf{v}_t\}_{t\in[0,1]}$ 以定义粒子流。这自然引出了一个问题：

Question 5.2.1

> 在什么条件下，流动诱导的密度 p_t^{fwd} 对于所有 $t \in [0,1]$ 都恰好等于目标密度 p_t ？

当两个密度演化对齐后，我们可以通过求解微分方程来灵活地在 p_{src} 和 p_{tgt} 之间传输样本。

如 Equation (5.2.5) 所示，验证此对齐的一种合理方法是通过 连续性方程，该方程描述了随时间演化的密度中的质量守恒：

Theorem 5.2.2: 质量守恒准则

对于所有 $t \in [0,1]$，流诱导密度 p_t^{fwd} 等于指定路径 p_t；即

$$p_t^{\text{fwd}} = p_t, \quad \text{对所有} t \in [0,1],$$

当且仅当向量对 $(p_t, \mathbf{v}_t)$ 满足连续性方程：

$$\partial_t p_t(\mathbf{x}) + \nabla_{\mathbf{x}} \cdot (p_t(\mathbf{x})\mathbf{v}_t(\mathbf{x})) = 0,$$

对于所有 $t \in [0,1]$ 和 $\mathbf{x}$ 成立。

Proof for Theorem.

> 概念推导见 Section D.3.2，更严格的论述可参考 (**villani2008optimal**)(see "Mass Conservation Formula")。

从条件路径到边缘路径。 如 Section 5.2.2 所示，我们首先定义一个条件概率路径 $p_t(\cdot|\mathbf{z})$ 和相应的条件速度场 $\mathbf{v}_t(\cdot|\mathbf{z})$。然后通过以下方式构建边缘速度场：

$$\mathbf{v}_t(\mathbf{x}) = \int \mathbf{v}_t(\mathbf{x}|\mathbf{z}) \frac{p_t(\mathbf{x}|\mathbf{z})\pi(\mathbf{z})}{p_t(\mathbf{x})} \, \mathrm{d}\mathbf{z},$$

如 Equation (5.2.10) 所示。然而，我们仍需确保所得的边缘速度 $\mathbf{v}_t$ 诱导出一个其密度路径与预设 p_t 一致的 ODE 流。幸运的是，这一验证可以在条件层面完全完成：如果每个条件速度场 $\mathbf{v}_t(\cdot|\mathbf{z})$ 诱导出条件密度路径 $p_t(\cdot|\mathbf{z})$，那么所得的边缘速度 $\mathbf{v}_t$ 也会诱导出正确的边缘路径。形式上，可表述如下：

Proposition 5.2.3: 边缘速度场生成给定边缘密度

若条件速度场 $\mathbf{v}_t(\cdot|\mathbf{z})$ 诱导的条件密度路径与 $p_t(\cdot|\mathbf{z})$ 匹配（从 $p_0(\cdot|\mathbf{z})$ 出发），则 Equation (5.2.10) 中定义的边缘速度场 $\mathbf{v}_t(\cdot)$ 会诱导出与 $p_t(\cdot)$ 对齐的边缘密度路径，且起始于 $p_0(\cdot)$。

Proof for Proposition.

该结论可通过验证 $(p_t, \mathbf{v}_t)$ 满足连续性方程而得。我们采用逆向论述方式，以直观说明边缘化速度场为何呈现 Equation (5.2.10) 中的形式。由于条件速度场 $\mathbf{v}_t(\cdot|\mathbf{z})$ 诱导的密度路径与条件密度 $p_t(\cdot|\mathbf{z})$ 匹配（其中 $\mathbf{z} \sim \pi$），每个条件对都满足连续性方程：

$$\frac{\mathrm{d}}{\mathrm{d}t} p_t(\mathbf{x}|\mathbf{z}) = -\nabla_{\mathbf{x}} \cdot \big(\mathbf{v}_t(\mathbf{x}|\mathbf{z}) p_t(\mathbf{x}|\mathbf{z})\big). \tag{5.2.13}$$

我们的目标是找到速度场 $\mathbf{v}_t(\cdot)$，使其诱导的密度与边缘密度 p_t 对齐，即满足：

$$\frac{\mathrm{d}}{\mathrm{d}t} p_t(\mathbf{x}) = -\nabla_{\mathbf{x}} \cdot \big(\mathbf{v}_t(\mathbf{x}) p_t(\mathbf{x})\big). \tag{5.2.14}$$

从 Equation (5.2.7) 中 p_t 的定义出发：

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} p_t(\mathbf{x}) &= \int \frac{\mathrm{d}}{\mathrm{d}t} p_t(\mathbf{x}_t|\mathbf{z}) \pi(\mathbf{z}) \, \mathrm{d}\mathbf{z} \\
&= -\int \nabla_{\mathbf{x}} \cdot \Big(\mathbf{v}_t(\mathbf{x}|\mathbf{z}) p_t(\mathbf{x}|\mathbf{z})\Big) \pi(\mathbf{z}) \, \mathrm{d}\mathbf{z} \\
&= -\nabla_{\mathbf{x}} \cdot \Big(\int \mathbf{v}_t(\mathbf{x}|\mathbf{z}) p_t(\mathbf{x}|\mathbf{z}) \pi(\mathbf{z}) \, \mathrm{d}\mathbf{z} \Big),
\end{aligned}$$

其中第二个等式通过应用 Equation (5.2.13) 得到。将此式与 Equation (5.2.14) 右侧对比可知，除去无散项后：

$$\mathbf{v}_t(\mathbf{x})p_t(\mathbf{x}) = \int \mathbf{v}_t(\mathbf{x}|\mathbf{z})p_t(\mathbf{x}|\mathbf{z})\pi(\mathbf{z})\,\mathrm{d}\mathbf{z}.$$

因此可定义：

$$\mathbf{v}_t(\mathbf{x}) := \int \mathbf{v}_t(\mathbf{x}|\mathbf{z})\frac{p_t(\mathbf{x}|\mathbf{z})}{p_t(\mathbf{x})}\pi(\mathbf{z})\,\mathrm{d}\mathbf{z},$$

此即 Equation (5.2.10) 中的形式。本定理的证明实质上遵循上述论证的逆过程。

这种联系使我们能够将潜在的难处理的边缘速度场的构造，简化为定义更简单的条件场 $\mathbf{v}_t(\cdot|\mathbf{z})$，这些条件场在构造上更容易处理。

5.3 构建分布之间的概率路径与速度

流匹配的本质在于将源分布逐步变换为目标分布。为了引导这一变换过程，需要两个关键要素：*概率路径* p_t ，它在每个时间点 t 提供演化分布的快照；以及 *速度场* $\mathbf{v}_t$ ，它描述了单个粒子沿路径的运动方式。这两个对象并非独立存在；它们通过连续性方程相互关联，确保粒子动力学与分布演化保持一致。因此，学习任务简化为寻找一个能够准确驱动该过程的速度场 $\mathbf{v}_t$ 。然而，难点在于，对于一般且复杂的分布，真实的边际速度 $\mathbf{v}_t$ 是未知的，导致我们面对一个无法直接获取的难处理目标。

条件流匹配的核心思想是通过构建一个人工但易处理的过程来解决真实边缘速度的不可处理性。为此，我们引入一个条件变量 $\mathbf{z}$ ，并设计条件速度 $\mathbf{v}_t(\mathbf{x}_t|\mathbf{z})$ 和/或条件路径 $p_t(\mathbf{x}_t|\mathbf{z})$ ，这些被特意选择为简单形式。

由于这些条件对象具有闭式表达，它们可作为模型回归的目标。这将导致一个有效的训练损失 $\mathcal{L}_{\text{CFM}}$ ，前提是满足两个实际要求：(i) 我们能够从 $p_t(\cdot|\mathbf{z})$ 中高效采样，以及 (ii) 相应的速度 $\mathbf{v}_t(\cdot|\mathbf{z})$ 具有闭式表达。

我们应如何设计一个表现良好的条件过程？作为灵感来源，我们转向唯一完全理解的情形：高斯到高斯的桥接过程（Section 5.3.1）。该例子突显了两种自然的设计策略：在每个时间点 t 采用高斯概率路径，或规定一个仿射速度场，这两种方法都是易处理的。

基于这一洞察，我们扩展到一般的端点分布，并从两个互补视角（参见 Section B.1.2）出发，构建条件路径和速度：

- **条件概率路径优先（欧拉视角）**。它从一个条件概率路径 $p_t(\cdot|\mathbf{z})$ 开始，并推导出相应的条件速度场。

- **条件流优先（拉格朗日视角）**。从一个条件流 $\boldsymbol{\Psi}_{0\to t}(\cdot|\mathbf{z})$ 开始，通常为仿射形式，通过沿轨迹对时间求导来推导出条件速度场。

在 Section 5.3.2 中，我们详细介绍了第一种方法，其与 Section 5.2.1 中讨论的扩散模型构建具有密切类比关系；而在 Section 5.3.3 中，我们提出了第二种方法。这两种视角共同提供了一个实用的框架，用于定义 $p_t(\mathbf{x}_t|\mathbf{z})$ 和 $\mathbf{v}_t(\mathbf{x}_t|\mathbf{z})$，从而实现无需仿真的训练，并构建任意源分布与目标分布之间的流。

5.3.1 一个关键的特殊情形：高斯到高斯桥中的边际 $p_t(\mathbf{x}_t)$ 与速度 $\mathbf{v}_t(\mathbf{x}_t)$

我们从高斯-端点情形开始，此时可以解析地计算边缘密度 $p_t(\mathbf{x}_t)$ 和速度场 $\mathbf{v}_t(\mathbf{x}_t)$ 。这为一般情形下条件密度 $p_t(\mathbf{x}_t|\mathbf{z}_t)$ 和速度场 $\mathbf{v}_t(\mathbf{x}_t|\mathbf{z}_t)$ 的构造提供了一个模板。

当源分布和目标分布 p_{src} 与 p_{tgt} 均为高斯分布时，速度场 $\mathbf{v}_t(\cdot)$ 有闭式表达式。我们考虑插值的边缘密度路径：

$$p_t(\mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_t; \boldsymbol{\mu}(t), \sigma^2(t)\mathbf{I}\right), \tag{5.3.1}$$

随时间变化的均值 $\boldsymbol{\mu}(t)$ 和方差 $\sigma^2(t) > 0$ 。两个端点由

$$p_{\text{src}} = p_0 = \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}(0), \sigma^2(0)\mathbf{I}\right), \quad p_{\text{tgt}} = p_1 = \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}(1), \sigma^2(1)\mathbf{I}\right),$$

使得路径 $\{p_t\}_{t\in[0,1]}$ 连接这些分布。

给定路径 $\{p_t\}_{t\in[0,1]}$ ，确实存在许多速度场可以诱导一个 ODE 流 $\boldsymbol{\Psi}_{0\to t}(\mathbf{x})$ ，使得 $\mathbf{x} \sim p_0$ 蕴含 $\boldsymbol{\Psi}_{0\to t}(\mathbf{x}) \sim p_t$ 。对于这个高斯路径，一个特别简单的实现由下式给出[7]：

$$\boldsymbol{\Psi}_{0\to t}(\mathbf{x}) := \boldsymbol{\mu}(t) + \sigma(t)\left(\frac{\mathbf{x} - \boldsymbol{\mu}(0)}{\sigma(0)}\right). \tag{5.3.2}$$

对于定义的高斯路径 p_t （对所有 t 的高斯路径），诱导 Equation (5.3.2) 中常微分方程流的速度场 $\mathbf{v}_t(\cdot)$ 被唯一且解析地表征如下 (**lipman2022flow**)：

Proposition 5.3.1: 高斯密度路径的闭式速度场

设 p_t 为 Equation (5.3.1) 中的高斯路径。则生成 ODE 流 Equation (5.3.2) 的速度场 $\mathbf{v}_t(\cdot)$ 对于定义的 $\boldsymbol{\Psi}_{0\to t}$ 是唯一的，并具有闭式表达式：

$$\mathbf{v}_t(\mathbf{x}) = \frac{\sigma'(t)}{\sigma(t)}\left(\mathbf{x} - \boldsymbol{\mu}(t)\right) + \boldsymbol{\mu}'(t).$$

Proof for Proposition.

[7]在 (**lipman2022flow**) 中，作者考虑了 $\boldsymbol{\Psi}_{0\to t}(\mathbf{x}) = \boldsymbol{\mu}(t) + \sigma(t)\mathbf{x}$ ，这要求对 $\boldsymbol{\mu}(t)$ 和 $\sigma(t)$ 施加约束以确保边界条件。我们采用了一种等价的规范化公式，避免了此类约束。

考虑具有初始条件 $\mathbf{y}$ 的 ODE:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\Psi}_{0\to t}(\mathbf{y}) = \mathbf{v}_t(\boldsymbol{\Psi}_{0\to t}(\mathbf{y})).$$

由于 $\boldsymbol{\Psi}_{0\to t}$ 可逆（因为 $\sigma(t) > 0$ ），可设 $\mathbf{x} = \boldsymbol{\Psi}_{0\to t}(\mathbf{y})$ 与 $\mathbf{y} = \boldsymbol{\Psi}_{0\to t}^{-1}(\mathbf{x}) = \boldsymbol{\Psi}_{t\to 0}(\mathbf{x})$ 得到

$$\boldsymbol{\Psi}'_{0\to t}(\boldsymbol{\Psi}_{0\to t}^{-1}(\mathbf{x})) = \mathbf{v}_t(\mathbf{x}).$$

对 Equation (5.3.2) 关于 t 求导可得

$$\boldsymbol{\Psi}'_{0\to t}(\mathbf{x}) = \boldsymbol{\mu}'(t) + \sigma'(t)\left(\frac{\mathbf{x} - \boldsymbol{\mu}(0)}{\sigma(0)}\right).$$

解 $\mathbf{y} = \boldsymbol{\Psi}_{0\to t}^{-1}(\mathbf{x})$ 得

$$\mathbf{y} = \boldsymbol{\mu}(0) + \sigma(0)\left(\frac{\mathbf{x} - \boldsymbol{\mu}(t)}{\sigma(t)}\right).$$

将其代入 $\boldsymbol{\Psi}'_{0\to t}(\mathbf{x})$ 可得

$$\mathbf{v}_t(\mathbf{x}) = \frac{\sigma'(t)}{\sigma(t)}\left(\mathbf{x} - \boldsymbol{\mu}(t)\right) + \boldsymbol{\mu}'(t),$$

得证。

我们注意到，对于固定的流映射 $\boldsymbol{\Psi}_{0\to t}$（流优先视角），速度由

$$\mathbf{v}_t = \partial_t \boldsymbol{\Psi}_{0\to t} \circ \boldsymbol{\Psi}_{0\to t}^{-1}.$$

在此构造下，对偶 $(p_t, \mathbf{v}_t)$ 自动满足连续性方程。相比之下，对于给定的密度路径 $t \mapsto p_t$ 而不固定 $\boldsymbol{\Psi}_{0\to t}$ （概率路径优先视角），速度场并不唯一。

这一区别精确地刻画了流优先与概率路径优先视角之间的差异。

这种闭式表征在对潜变量 $\mathbf{z}$ 进行条件化时仍然有效。接下来，我们将这一见解扩展，以构建一个条件高斯路径 $p_t(\cdot|\mathbf{z})$ ，并推导出一般边缘情景下的相应条件速度场 $\mathbf{v}_t(\cdot|\mathbf{z})$ 。

5.3.2 条件概率-路径优先构造 $\mathbf{v}_t(\cdot|\mathbf{z})$ 与 $p_t(\cdot|\mathbf{z})$

我们旨在首先构造一个条件密度路径 $p_t(\cdot|\mathbf{z})$，然后通过命题 5.3.1 推导出其对应的条件速度场 $\mathbf{v}_t(\cdot|\mathbf{z})$，条件为 $\pi(\mathbf{z})$。根据 $\mathbf{z}$ 的选取方式，存在两种自然情

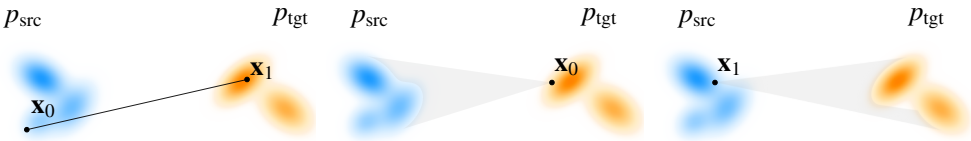


图 5.6: 两种常见条件概率路径的示意图。包括：(1) 双边，在 $\mathbf{x}_0 \sim p_{\text{tgt}}$ 和 $\mathbf{x}_1 \sim p_{\text{src}}$ 条件下，具有通用端点分布；(2) 单边，在 $\mathbf{x}_0 \sim p_{\text{tgt}}$ 或 $\mathbf{x}_1 \sim p_{\text{src}}$ 条件下。

形：(i) 双侧条件化，其中 $\mathbf{z} = (\mathbf{x}_0, \mathbf{x}_1)$，或 (ii) 单侧条件化，其中 $\mathbf{z} = \mathbf{x}_0$ 或 $\mathbf{x}_1$。在任一种情况下，构造必须与边界分布相匹配：

$$p_{\text{src}}(\mathbf{x}_0) = \int p_0(\mathbf{x}_0|\mathbf{z})\pi(\mathbf{z})\,\mathrm{d}\mathbf{z}, \quad p_{\text{tgt}}(\mathbf{x}_1) = \int p_1(\mathbf{x}_1|\mathbf{z})\pi(\mathbf{z})\,\mathrm{d}\mathbf{z}.$$

由于一旦指定了具体的构造，验证这些约束就变得很简单，因此我们在此不强调验证步骤。

I. 两侧 $\mathbf{z} = (\mathbf{x}_0, \mathbf{x}_1)$ —“梁型”路径。

$\pi(\mathbf{z})$ 的选择。 考虑定义在 $\mathbb{R}^D$ 上的两个一般分布 p_{src} 和 p_{tgt}。令 $\mathbf{z} = (\mathbf{x}_0, \mathbf{x}_1)$，其中 $\mathbf{x}_0 \sim p_{\text{src}}$ 且 $\mathbf{x}_1 \sim p_{\text{tgt}}$ 独立，即

$$\pi(\mathbf{z}) = p_{\text{src}}(\mathbf{x}_0)p_{\text{tgt}}(\mathbf{x}_1).$$

条件路径的选择 $p_t(\cdot|\mathbf{z})$。 通过固定方差 $\sigma > 0$ 的线性插值定义条件路径：

$$p_t(\mathbf{x}_t|\mathbf{z} = (\mathbf{x}_0, \mathbf{x}_1)) = \mathcal{N}(\mathbf{x}_t; a_t\mathbf{x}_0 + b_t\mathbf{x}_1, \sigma^2\mathbf{I}),$$

其中 a_t 和 b_t 是满足 $a_0 = 1, b_0 = 0$ 和 $a_1 = 0, b_1 = 1$ 的时变函数。(**lipman2022flow**; **liu2022rectified**) 建议的选择为 $a_t = 1 - t$，$b_t = t$。在确定性情况下 $\sigma = 0$，我们得到

$$p_t(\mathbf{x}_t|\mathbf{z}) = \delta\big(\mathbf{x}_t - [a_t\mathbf{x}_0 + b_t\mathbf{x}_1]\big),$$

描述从 $\mathbf{x}_0$ 到 $\mathbf{x}_1$ 的确定性插值路径。

推导出的条件速度 $\mathbf{v}_t(\cdot|\mathbf{z})$。 根据命题 5.3.1，条件速度为

$$\mathbf{v}_t(\mathbf{x}|\mathbf{z}) = a_t'\mathbf{x}_0 + b_t'\mathbf{x}_1.$$

CFM 损失。 当 $\sigma = 0$ 使得 $\mathbf{x}_t = a_t\mathbf{x}_0 + b_t\mathbf{x}_1$ 时，CFM 损失简化为

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t,\mathbf{x}_0\sim p_{\text{src}},\mathbf{x}_1\sim p_{\text{tgt}}} \left\|\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - (a_t'\mathbf{x}_0 + b_t'\mathbf{x}_1)\right\|^2.$$

由 Equations (5.2.10) and (5.2.11)，最优速度场为

$$\mathbf{v}^*(\mathbf{x}_t, t) = \mathbb{E}\left[\mathbf{x}_t'|\mathbf{x}_t\right] = \mathbb{E}\left[a_t'\mathbf{x}_0 + b_t'\mathbf{x}_1|\mathbf{x}_t\right].$$

此处，期望是关于 $p(\mathbf{x}_0, \mathbf{x}_1|\mathbf{x}_t)$ 取的，即关于可能产生在时间 t 观测到的插值 $\mathbf{x}_t = a_t\mathbf{x}_0 + b_t\mathbf{x}_1$ 的源-目标对 $(\mathbf{x}_0, \mathbf{x}_1)$ 的条件分布。

II. 单侧 $\mathbf{z} = \mathbf{x_0}$ 或 $\mathbf{x_1}$ — "探照灯式" 路径。 我们以单边情景为例，说明条件概率–路径–优先的构造方法，考虑标准生成设置中的 $p_{\text{src}} = \mathcal{N}(\mathbf{0}, \mathbf{I})$ 和 $p_{\text{tgt}} = p_{\text{data}}$ 。关键的是，这种高斯源并非额外假设，而是下方定义的条件路径的直接结果。关于任意端点的更一般处理将在 Section 5.3.3 中给出。

$\boldsymbol{\pi}(\mathbf{z})$ 的选择。 我们取 $\mathbf{z} = \mathbf{x}_1$ ，其中 $\pi(\mathbf{z}) = p_{\text{data}}(\mathbf{x}_1)$（情况 $\mathbf{z} = \mathbf{x}_0 \sim p_{\text{prior}}$ 的情形类似）。

条件路径的选择 $\boldsymbol{p_t(\cdot|\mathbf{z})}$ 。 对于固定的 $\mathbf{x}_1 \sim p_{\text{data}}$ ，定义

$$p_t(\mathbf{x}_t|\mathbf{z} = \mathbf{x}_1) = \mathcal{N}\big(\mathbf{x}_t; b_t\mathbf{x}_1, a_t^2\mathbf{I}\big),$$

其中 $a_0 = 1, b_0 = 0, a_1 = 0, b_1 = 1$ （通常被解释为极限）。在边界处，

$$p_0(\cdot|\mathbf{z} = \mathbf{x}_1) = \mathcal{N}(\cdot; \mathbf{0}, \mathbf{I}), \qquad p_1(\cdot|\mathbf{z} = \mathbf{x}_1) = \delta(\cdot - \mathbf{x}_1).$$

对 $\mathbf{x}_1$ 进行边缘化得到 $\{p_t\}_{t\in[0,1]}$，其中 $p_0 = \mathcal{N}(\mathbf{0}, \mathbf{I})$（与 p_{data} 无关）和 $p_1 = p_{\text{data}}$。

推导出的条件速度 $\mathbf{v_t}(\cdot|\mathbf{z})$ 。 对于 $t \in (0, 1)$ 且 $b_t > 0$ ，将命题 5.3.1 应用于条件高斯路径得到

$$\mathbf{v}_t(\mathbf{x}|\mathbf{x}_1) = b_t'\mathbf{x}_1 + \frac{a_t'}{a_t}\big(\mathbf{x} - b_t\mathbf{x}_1\big).$$

单边 CFM 目标。 使用 $t \sim \mathcal{U}(0,1)$ （或任意固定采样分布）和 $\mathbf{x}_1 \sim p_{\text{data}}$ ，CFM 损失变为

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t,\mathbf{x}_1}\mathbb{E}_{\mathbf{x}_t\sim p_t(\cdot|\mathbf{x}_1)}\left\|\mathbf{v}_\phi(\mathbf{x}_t,t) - \left[b_t'\mathbf{x}_1 + \tfrac{a_t'}{a_t}\big(\mathbf{x}_t - b_t\mathbf{x}_1\big)\right]\right\|_2^2. \tag{5.3.3}$$

根据 MSE 最优性，唯一的最小化器是边缘速度场。

$$\mathbf{v}^*(\mathbf{x},t) = \mathbb{E}\left[\mathbf{v}_t(\mathbf{x}|\mathbf{x}_1)|\mathbf{x}_t = \mathbf{x}\right] = \mathbb{E}\left[a_t'\mathbf{x}_0 + b_t'\mathbf{x}_1|\mathbf{x}_t = \mathbf{x}\right].$$

等价于双侧目标。 对于配对样本 $(\mathbf{x}_0,\mathbf{x}_1)$ 且 $\mathbf{x}_t = a_t\mathbf{x}_0 + b_t\mathbf{x}_1$ ，

$$\mathbf{v}_t(\mathbf{x}_t|\mathbf{x}_1) = b_t'\mathbf{x}_1 + \tfrac{a_t'}{a_t}\big(\mathbf{x}_t - b_t\mathbf{x}_1\big) = a_t'\mathbf{x}_0 + b_t'\mathbf{x}_1.$$

因此，单边损失会回归到给定 $\mathbf{x}_t$ 的双边目标的 条件期望：

$$\mathbf{v}^*(\mathbf{x},t) = \mathbb{E}\big[a_t'\mathbf{x}_0 + b_t'\mathbf{x}_1|\mathbf{x}_t = \mathbf{x}\big],$$

因此，单边和双边 CFM 目标函数具有相同的最小值点。

高斯 FM = 扩散模型。 我们使用 FM 约定，其中 $t=0$ 表示源/先验，$t=1$ 表示目标/数据：

$$p_{\text{src}} = p_{\text{prior}}, \qquad p_{\text{tgt}} = p_{\text{data}}.$$

相比之下，扩散模型通常从数据到噪声对时间进行索引（即 $t=0$ 为数据，$t=1$ 为先验）。在此，我们始终采用 FM 的索引方式以避免混淆。若进一步 $p_{\text{src}} = \mathcal{N}(\mathbf{0},\mathbf{I})$ ，则对于固定的条件 $\mathbf{x}_1 \sim p_{\text{data}}$ ，条件路径 $p_t(\cdot|\mathbf{x}_1)$ 自然选择为高斯分布，而目标分布 p_{tgt} 本身无需为高斯分布。部分文献通常将此情景称为 高斯 *FM*。

选择 $a_t = 1-t$ 和 $b_t = t$ （等价于在扩散模型中重新标记 $a_t := \sigma_t$, $b_t := \alpha_t$ 时的 $\alpha_t = t$ 和 $\sigma_t = 1-t$ ）可恢复熟悉的 FM/RF 调度 (**lipman2022flow**; **liu2022rectified**)。

在高斯流匹配（Gaussian FM）情景下，束状和点状的条件路径均导向与标准扩散损失相似的训练目标。正如我们将在 Chapter 6 中详细阐述的，高斯流匹配实际上可以等价地解释为一个扩散模型，其目标是预测 速度，在线性调度 $a_t = 1-t$ 与 $b_t = t$ 下进行训练。这一视角表明，流匹配与扩散模型并非本质不同，而是两种可相互转换的等价表述。高斯流匹配的目标在实践中尤为吸引人：其损失函数 ($\mathbb{E}_{t,\mathbf{x}_t}\left[\|\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t,t) - (\mathbf{x}_1 - \mathbf{x}_0)\|_2^2\right]$) 简单易用，并且已被证明在大规

表 5.2: 不同插值函数在 FM 规范下的总结 $\mathbf{x}_t = a_t\mathbf{x}_0 + b_t\mathbf{x}_1$ ，其中 $\mathbf{x}_0 \sim p_{\text{src}} = p_{\text{prior}}$ ， $\mathbf{x}_1 \sim p_{\text{tgt}} = p_{\text{data}}$ 。VE/VP 通过 $a_t := \sigma_t$, $b_t := \alpha_t$ 从它们的扩散规范（数据 → 噪声）转换而来。

	VE	**VP**	**FM/RF**	**Trig.** (**albergo2023stochastic**)
a_t (prior coeff.)	a_t	$\sqrt{1-b_t^2}$	$1-t$	$\cos\left(\frac{\pi}{2}t\right)$
b_t (data coeff.)	1	b_t	t	$\sin\left(\frac{\pi}{2}t\right)$
a_0	0	0	1	1
b_0	1	1	0	0
a_1	a_1	1	0	0
b_1	1	0	1	1
p_{prior}	$\mathcal{N}(\mathbf{0}, a_1^2\mathbf{I})$	$\mathcal{N}(\mathbf{0}, \mathbf{I})$	$\mathcal{N}(\mathbf{0}, \mathbf{I})$	$\mathcal{N}(\mathbf{0}, \mathbf{I})$

模场景下能够实现具有竞争力的性能 (**esser2024scaling**)。

Remark.

> 值得强调的是，部分先验研究 (**liu2022rectified**; **lipman2022flow**) 提出采用正则仿射流 $a_t = 1-t$ 和 $b_t = t$ 可产生" 直线型" 常微分方程轨迹，从而实现更快采样。然而该论断并不具有普适性。此公式中的速度场由条件期望给出：
>
> $$\mathbf{v}(\mathbf{x}, t) = \mathbb{E}[\mathbf{x}_1 - \mathbf{x}_0 | \mathbf{X}_t = \mathbf{x}],$$
>
> 其取值依赖于 t ，因此并不总是与朴素方向 $\mathbf{x}_1 - \mathbf{x}_0$ 保持一致。实践中，时间加权函数与参数化的选择会显著影响训练动态，并能提升实证性能，但这些改进不能归因于调度器 ($a_t = 1-t$, $b_t = t$) 所宣称的" 直线性"。

5.3.3 条件流优先构建 $\mathbf{v}_t(\cdot|\mathbf{z})$ 和 $p_t(\cdot|\mathbf{z})$

我们处理端点 p_{src} （位于 $t = 0$ ）和 p_{tgt} （位于 $t = 1$ ）为任意的一般情况。我们的目标是在轨迹空间中直接设计一种条件流，将样本从 p_{src} 传输到 p_{tgt} ，并得到一个可用于回归目标的闭式 $\mathbf{v}_t(\mathbf{x}_t|\mathbf{z})$ 。

动机。 与其先设计条件密度路径，不如直接指定一个条件流映射 $\mathbf{\Psi}_{0\to t}(\cdot;\mathbf{z})$，该映射沿轨迹移动样本。这具有两个实际优势：(i) 可通过轨迹上的时间导数立即得到用于训练的回归目标；(ii) 在几何结构化空间（黎曼流形、李群或约束子流形）上，通常可以直接从几何结构（如测地线、指数映射或预度量）构造条件流映射 $\mathbf{\Psi}_{0\to t}$ (**lipman2024flow**)，从而在训练中获得解析的、无需仿真的目标速

度。

条件仿射流（参见命题 5.3.1）。 我们固定一个条件变量 $\mathbf{z} \sim \pi$（例如，单边“聚光灯”训练中的 $\mathbf{z} = \mathbf{x}_1 \sim p_{\text{tgt}}$），并通过时变的 *条件仿射流* 将 $\mathbf{x}_0 \sim p_{\text{src}}$ 进行前向传播

$$\boldsymbol{\Psi}_{0\to t}(\mathbf{x}_0; \mathbf{z}) := \boldsymbol{\mu}_t(\mathbf{z}) + \mathbf{A}_t(\mathbf{z})\mathbf{x}_0, \qquad t \in [0,1],$$

其中 $\boldsymbol{\mu}_t(\mathbf{z}) \in \mathbb{R}^D$ 和 $\mathbf{A}_t(\mathbf{z}) \in \mathbb{R}^{D\times D}$ 在 $t \in (0,1)$ 下可求逆。边界 $\mathbf{A}_0(\mathbf{z}) = \mathbf{I}$，$\boldsymbol{\mu}_0(\mathbf{z}) = \mathbf{0}$ 在 $t = 0$ 处恢复 p_{src}。当 $t \to 1$ 时，将边界解释为极限是标准做法（终端映射可能将质量集中在低维度集合或一点上）。[8].

诱导条件路径 $p_t(\cdot|\mathbf{z})$。 该构造定义了

$$p_t(\cdot|\mathbf{z}) = \big(\boldsymbol{\Psi}_{0\to t}(\cdot; \mathbf{z})\big)_{\#} p_{\text{src}}, \qquad p_t(\cdot) = \int p_t(\cdot|\mathbf{z})\pi(\mathbf{z})\,\mathrm{d}\mathbf{z}.$$

在 $\mathcal{L}_{\text{CFM}}$ 中最终重要的是如何从中进行采样：首先抽取 $\mathbf{z} \sim \pi$，然后抽取 $\mathbf{x}_0 \sim p_{\text{src}}$，最后设定

$$\mathbf{x}_t = \boldsymbol{\mu}_t(\mathbf{z}) + \mathbf{A}_t(\mathbf{z})\mathbf{x}_0.$$

我们注意到，当 $\boldsymbol{\Psi}_{0\to t}$ 关于 $\mathbf{x}_0$ 为仿射时，$p_t(\cdot|\mathbf{z})$ 为高斯分布当且仅当 p_{src} 为高斯分布。特别地，对于任意（非高斯）p_{src}，仿射变换通常会产生一个非高斯的 $p_t(\cdot|\mathbf{z})$。

推导出的条件速度 $\mathbf{v}_t(\cdot|\mathbf{z})$。 条件速度 $\mathbf{v}_t(\cdot|\mathbf{z})$ 通过对方程 t -对条件流映射 $\boldsymbol{\Psi}_{0\to t}$ 进行微分得到。根据命题 5.3.1 中的推导，考虑由流映射 $\boldsymbol{\Psi}_{0\to t}(\mathbf{y}; \mathbf{z})$ 定义的 条件 ODE，初始条件为 $\mathbf{y}$，其目标是确定相应的条件速度场 $\mathbf{v}_t(\cdot|\mathbf{z})$：

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\Psi}_{0\to t}(\mathbf{y}; \mathbf{z}) = \mathbf{v}_t\Big(\underbrace{\boldsymbol{\Psi}_{0\to t}(\mathbf{y}; \mathbf{z})}_{\mathbf{x}}\Big|\mathbf{z}\Big).$$

由于 $\boldsymbol{\Psi}_{0\to t}(\cdot; \mathbf{z})$ 对于 $t \in (0,1)$ 可求逆，我们可以将 $\mathbf{y}$ 表示为当前状态 $\mathbf{x} := \boldsymbol{\Psi}_{0\to t}(\mathbf{y}; \mathbf{z})$ 的函数，即 $\mathbf{y} = \boldsymbol{\Psi}^{-1}_{0\to t}(\mathbf{x}; \mathbf{z}) = \boldsymbol{\Psi}_{t\to 0}(\mathbf{x}; \mathbf{z})$。将其代入微分方程，得到条件速度场的如下构造：

$$\mathbf{v}_t(\mathbf{x}|\mathbf{z}) := \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\Psi}_{0\to t}\big(\boldsymbol{\Psi}_{t\to 0}(\mathbf{x}; \mathbf{z}); \mathbf{z}\big),$$

[8] 允许 $\mathbf{A}_1(\mathbf{z})$ 为奇异的（例如 $\mathbf{0}$）与 $(0,1)$ 上的可逆性相容，并导致路径在 $t = 1$ 处收缩到指定的端点。

这明确表明，导数必须沿着在时间 t 到达空间点 $\mathbf{x}$ 的轨迹来计算。

由于 $\mathbf{x}_t = \boldsymbol{\mu}_t(\mathbf{z}) + \mathbf{A}_t(\mathbf{z})\mathbf{x}_0$ 和 $\mathbf{A}_t(\mathbf{z})$ 在 $(0,1)$ 上可逆，我们得到 $\mathbf{x}_0 = \mathbf{A}_t(\mathbf{z})^{-1}\big(\mathbf{x} - \boldsymbol{\mu}_t(\mathbf{z})\big)$ ，从而

$$\mathbf{v}_t(\mathbf{x}|\mathbf{z}) = \boldsymbol{\mu}_t'(\mathbf{z}) + \mathbf{A}_t'(\mathbf{z})\mathbf{A}_t(\mathbf{z})^{-1}\big(\mathbf{x} - \boldsymbol{\mu}_t(\mathbf{z})\big).$$

单侧条件化 ($\mathbf{z} = \mathbf{x}_1$)。 选择 $\boldsymbol{\mu}_t(\mathbf{z}) = b_t\mathbf{z}$ 和 $\mathbf{A}_t(\mathbf{z}) = a_t\mathbf{I}$ 使得 $a_0 = 1, a_1 = 0$ 与 $b_0 = 0, b_1 = 1$ （对于 $t \in (0,1)$ 使用 $a_t > 0$ ）成立，可得

$$\mathbf{x}_t = a_t\mathbf{x}_0 + b_t\mathbf{x}_1, \qquad \mathbf{v}_t(\mathbf{x}|\mathbf{x}_1) = b_t'\mathbf{x}_1 + \frac{a_t'}{a_t}\big(\mathbf{x} - b_t\mathbf{x}_1\big).$$

对于配对样本 $(\mathbf{x}_0, \mathbf{x}_1)$（具有 $\mathbf{x}_t = a_t\mathbf{x}_0 + b_t\mathbf{x}_1$），这简化为通常的 CFM 目标：

$$\mathbf{v}_t(\mathbf{x}_t|\mathbf{x}_1) = a_t'\mathbf{x}_0 + b_t'\mathbf{x}_1.$$

双向条件化 ($\mathbf{z} = (\mathbf{x}_0, \mathbf{x}_1)$)。 使用相同的模板与 $\boldsymbol{\mu}_t(\mathbf{x}_0, \mathbf{x}_1) = b_t\mathbf{x}_1$ 和 $\mathbf{A}_t(\mathbf{x}_0, \mathbf{x}_1) = a_t\mathbf{I}$ ，使得条件路径 确定性：

$$\mathbf{x}_t = a_t\mathbf{x}_0 + b_t\mathbf{x}_1, \qquad p_t(\cdot|\mathbf{x}_0, \mathbf{x}_1) = \delta\left(\cdot - (a_t\mathbf{x}_0 + b_t\mathbf{x}_1)\right),$$

且条件速度为

$$\mathbf{v}_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_1) = a_t'\mathbf{x}_0 + b_t'\mathbf{x}_1,$$

即标准的双侧 CFM 目标。

无条件高斯路径作为特例。 若 $\boldsymbol{\mu}_t$ 与 $\mathbf{z}$ 独立（记作 $\boldsymbol{\mu}(t)$ ）且 $\mathbf{A}_t = \frac{\sigma(t)}{\sigma(0)}\mathbf{I}$ ，则

$$\boldsymbol{\Psi}_{0\to t}(\mathbf{x}_0) = \boldsymbol{\mu}(t) + \sigma(t)\frac{\mathbf{x}_0 - \boldsymbol{\mu}(0)}{\sigma(0)}, \quad \mathbf{v}_t(\mathbf{x}) = \boldsymbol{\mu}'(t) + \frac{\sigma'(t)}{\sigma(t)}\big(\mathbf{x} - \boldsymbol{\mu}(t)\big),$$

，从而恢复了高斯密度路径和命题 5.3.1 中的闭式速度。

5.3.4 概率路径优先与流量优先构建

两种构造均旨在通过条件动力学将源分布 p_{src} 与目标分布 p_{tgt} 联系起来。概率路径优先（欧拉）视角首先假设一个条件密度路径 $p_t(\cdot|\mathbf{z})$ ，通常选择高斯

或仿射族，以便关联的速度 $\mathbf{v}_t(\cdot|\mathbf{z})$ 可以解析求解。*流优先*（拉格朗日）视角则直接指定一个条件流映射 $\boldsymbol{\Psi}_{0\to t}(\cdot|\mathbf{z})$，并通过沿粒子轨迹的微分直接获得速度。尽管两者在正则条件下产生等价的传输效果，但在可识别性、计算便捷性以及端点约束的施加方式上存在差异。下表总结了这些对比。要点：当条件路径具有闭式速度时，路径优先是自然的选择；当对轨迹具有强结构先验时，流优先更为自然。

轴	条件概率-路径优先	条件流-优先
给定	条件密度路径 $p_t(\cdot\|\mathbf{z})$ 。	条件流映射 $\boldsymbol{\Psi}_{0\to t}(\cdot\|\mathbf{z})$ （轨迹，对每个固定的 $\mathbf{z}$ ）。
获取速度	对每个 $\mathbf{z}$ ，寻找 $\mathbf{v}_t(\cdot\|\mathbf{z})$ 使得 $$\partial_t p_t(\cdot\|\mathbf{z}) + \nabla\cdot(p_t(\cdot\|\mathbf{z})\mathbf{v}_t(\cdot\|\mathbf{z})) = 0;$$ 不唯一：若 $\nabla\cdot(p_t\mathbf{w}_t) = 0$ 则 $\mathbf{v}_t + \mathbf{w}_t$ 产生相同的 p_t 。	沿路径（对每个 $\mathbf{z}$ ）： $$\mathbf{v}_t\left(\boldsymbol{\Psi}_{0\to t}(\cdot\|\mathbf{z})|\mathbf{z}\right) = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\Psi}_{0\to t}(\cdot|\mathbf{z})$$ 当 $\boldsymbol{\Psi}_{0\to t}$ 可逆时，可求解 $$\mathbf{v}_t(\mathbf{x}|\mathbf{z}) = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\Psi}_{0\to t}(\boldsymbol{\Psi}_{0\to t}^{-1}(\mathbf{x})|\mathbf{z})$$
$\mathbf{v}_t(\cdot\|\mathbf{z})$ 的闭式表达	当 $p_t(\cdot\|\mathbf{z})$ 为高斯/指数族时方便；否则获得 $\mathbf{v}_t(\cdot\|\mathbf{z})$ 非常困难。	当 $\boldsymbol{\Psi}_{0\to t}(\cdot\|\mathbf{z})$ 具有结构（仿射/低秩）时方便；避免密度评估。
$\mathbf{v}_t(\cdot\|\mathbf{z})$ 的唯一性	对每个 $\mathbf{z}$ ， $\mathbf{v}_t(\cdot\|\mathbf{z})$ 在未施加选择规则（如势流/最小动能）时欠定。	给定 $\boldsymbol{\Psi}_{0\to t}(\cdot\|\mathbf{z})$ ， $p_t(\cdot\|\mathbf{z}) = (\boldsymbol{\Psi}_{0\to t}(\cdot\|\mathbf{z}))_{\#}p_0$ 和 $\mathbf{v}_t(\cdot\|\mathbf{z})$ 均被确定；非可逆映射仍沿轨迹定义 $\mathbf{v}_t(\cdot\|\mathbf{z})$ ，而可逆映射使其唯一。
可实现性	必须验证构造的 $\mathbf{v}_t(\cdot\|\mathbf{z})$ 是否在目标支撑集上满足连续性方程。	构造保证成立： $$p_t(\cdot\|\mathbf{z}) = (\boldsymbol{\Psi}_{0\to t}(\cdot\|\mathbf{z}))_{\#}\,p_0(\cdot\|\mathbf{z})$$
匹配 $(p_{\mathrm{src}}, p_{\mathrm{tgt}})$	混合条件： $p_{\mathrm{src}} = \int p_0(\cdot\|\mathbf{z})\pi(\mathbf{z})\,\mathrm{d}\mathbf{z}$, $p_{\mathrm{tgt}} = \int p_1(\cdot\|\mathbf{z})\pi(\mathbf{z})\,\mathrm{d}\mathbf{z}$. 在高斯-仿射条件路径且 $\mathbf{z}$ -无关系数下, p_{src} 可强制为高斯分布。对于一般固定终点 p_{src} （可能非高斯），$p_t(\cdot\|\mathbf{z})$ 的选择通常不能确定 p_{src} 。	设定 $\boldsymbol{\Psi}_{0\to 0} = \mathrm{Id}$ 并选择边界条件以命中任意 p_{tgt} 。
优选场景	扩散型构造；通过条件高斯分布实现分析目标 $p_t(\cdot\|\mathbf{z})$ 。	通过映射施加强结构先验 $\boldsymbol{\Psi}_{0\to t}(\cdot\|\mathbf{z})$ ；易于边界控制；可处理奇异/低维终点；适用于基于映射的正则化/输运成本。

5.4 (Optional) Properties of the Canonical Affine Flow

给定两个端点分布 $p_0 = p_{\text{src}}$ 和 $p_1 = p_{\text{tgt}}$,在流匹配 (FM) (**lipman2022flow**) 和修正流 (RF) (**liu2022flow**) 中，定义条件路径的一种自然且广泛使用的选择是线性插值

$$a_t = 1 - t, \qquad b_t = t,$$

从而得到插值函数

$$\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1, \qquad \mathbf{x}_0 \sim p_{\text{src}},\ \mathbf{x}_1 \sim p_{\text{tgt}}.$$

在这种选择下，训练目标简化为

$$\mathbb{E}_{t\sim\mathcal{U}[0,1]}\mathbb{E}_{\mathbf{x}_0,\mathbf{x}_1}\left[\left\|\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t,t) - (\mathbf{x}_1 - \mathbf{x}_0)\right\|_2^2\right].$$

这种线性流具有多个吸引人的性质。特别是，它支持一种迭代优化方案，称为 *Reflow*，该方案在保持边缘分布不变的同时，逐步使分布之间的路径变得更为直线化。

5.4.1 校正流：从噪声猜测到结构化配对

从噪声到数据的相干路径。 考虑生成任务中，p_{src} 为先验，p_{tgt} 为真实数据。我们希望找到一条连续路径，将噪声传输至数据。一种简单的方法是独立地采样 $\mathbf{z}_0 \sim p_{\text{src}}$ 与 $\mathbf{x}_1 \sim p_{\text{tgt}}$，进行插值（例如 $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1$），并拟合该直线上的速度场。这会导致不一致的配对：不同迭代间的端点彼此无关，使得轨迹波动剧烈，方差急剧增大，收敛变慢，样本质量下降。

独立耦合为何不足。 使用独立抽样的条件流匹配

$$\pi(\mathbf{z}) = p_{\text{src}}(\mathbf{x}_0)p_{\text{tgt}}(\mathbf{x}_1),$$

或单边变体。这类耦合方式有利于采样，但会导致路径锯齿状且方差较高，使速度场难以建模。

通过耦合依赖关系校正流。 而不是依赖任意的配对，我们使用预训练的扩散模型 $\mathbf{v}_{\boldsymbol{\phi}^\times}(\cdot,t)$ 作为 PF-ODE 中的漂移项，以确定性方式传输每个源点。从 $\mathbf{z}(0) =$

$\mathbf{z}_0 \sim p_{\text{src}}$ 开始，我们进行积分

$$\frac{\mathrm{d}\mathbf{z}(t)}{\mathrm{d}t} = \mathbf{v}_{\boldsymbol{\phi}^\times}(\mathbf{z}(t), t), \qquad t \in [0,1],$$

为了获得 $\hat{\mathbf{z}}_1 := \mathbf{z}(1)$，使其靠近从预训练模型中学成的数据空间。由此形成的配对 $(\mathbf{z}_0, \hat{\mathbf{z}}_1)$ 构成了一个 依赖耦合：它遵循一种结构化、由模型引导的路径，而非任意的插值。这一思想可自然推广到形如 $\mathbf{x}_t = a_t\mathbf{x}_0 + b_t\mathbf{x}_1$ 的仿射参考路径，其中 $\mathbf{x}_0 \sim p_{\text{src}}$ 且 $\mathbf{x}_1 \sim p_{\text{tgt}}$。

Algorithm 2 `Rectify` Operation

Input: Reference path $\{\mathbf{x}_t\}_{t\in[0,1]}$ (e.g. $\mathbf{x}_t = a_t\mathbf{x}_0 + b_t\mathbf{x}_1$)

1: **Pre-Train Diffusion.** Fit $\mathbf{v}_{\boldsymbol{\phi}^\times}$ on the chosen path by minimizing

$$\boldsymbol{\phi}^\times \in \arg\min_{\boldsymbol{\phi}} \mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_1}\left[\left\|\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}\right\|_2^2\right].$$

2: **Rectify.** Sample $\mathbf{z}_0 \sim p_{\text{src}}$ and integrate

$$\frac{\mathrm{d}\mathbf{z}(t)}{\mathrm{d}t} = \mathbf{v}_{\boldsymbol{\phi}^\times}(\mathbf{z}(t), t), \qquad \mathbf{z}(0) = \mathbf{z}_0, \quad t \in [0,1],$$

to obtain $\hat{\mathbf{z}}_1 = \mathbf{z}(1)$ and the trajectory $\{\mathbf{z}(t)\}_{t\in[0,1]}$.

Output: Dependent (coherent) pair $(\mathbf{z}_0, \hat{\mathbf{z}}_1)$ or the full trajectory.

有效原因：**保持边际结构**。 设 $\boldsymbol{\Phi}_{0\to t}$ 表示由上述常微分方程生成的流映射，该常微分方程由预训练的扩散模型 $\mathbf{v}_{\boldsymbol{\phi}^\times}$ 定义；则 $\mathbf{z}(t) = \boldsymbol{\Phi}_{0\to t}(\mathbf{z}_0)$ 和 $\hat{\mathbf{z}}_1 = \boldsymbol{\Phi}_{0\to 1}(\mathbf{z}_0)$。`Rectify` 过程将每个源点与其流终点配对，从而得到确定性联合分布

$$\pi_{\text{Rectify}}(\mathbf{z}_0, \mathbf{z}_1) = p_{\text{src}}(\mathbf{z}_0)\delta\big(\mathbf{z}_1 - \boldsymbol{\Phi}_{0\to 1}(\mathbf{z}_0)\big).$$

我们有两个直接的后果：

- **源边缘保持**：$\displaystyle\int \pi_{\text{Rectify}}(\mathbf{z}_0, \mathbf{z}_1)\,\mathrm{d}\mathbf{z}_1 = p_{\text{src}}(\mathbf{z}_0)$.
- **沿流的前推**：$(\boldsymbol{\Phi}_{0\to t})_{\#}p_{\text{src}} = \text{Law}(\mathbf{z}(t))$ ，即时间–t 分布是 p_{src} 通过 $\boldsymbol{\Phi}_{0\to t}$ 的前推。

如果 $\mathbf{v}_{\phi^\times}$ 与给定参考路径 $\mathbf{x}_t$ 的预言漂移相匹配，则所有中间边缘分布均一致：

$$\mathrm{Law}(\mathbf{z}(t)) = \mathrm{Law}(\mathbf{x}_t), \quad \text{for all } t \in [0,1], \quad \text{and} \quad (\mathbf{\Phi}_{0\to 1})_{\#} p_{\mathrm{src}} = p_{\mathrm{tgt}}.$$

摘要。 修正用平滑的教师引导轨迹替代了噪声干扰的独立配对，降低了方差，简化了最优化过程，并提升了样本质量。该思想涵盖了正则线性路径 $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1$ 和一般仿射形式 $\mathbf{x}_t = a_t\mathbf{x}_0 + b_t\mathbf{x}_1$ 。

对于正则路径，反复应用 `Rectify`（“*Reflow*”）可以进一步使轨迹变直而不增加传输成本，从而使训练仍然更简单。

5.4.2 `Reflow`：迭代直线化流

为什么选择 `Reflow`？ 独立配对通常会在 p_{src} 和 p_{tgt} 之间引发不规则且蜿蜒的常微分方程轨迹，这会增加仿真过程中的离散化误差和方差。这引出一个自然的问题：

Question 5.4.1

我们能否学习到一种耦合方式，使得传输路径更接近两个分布之间的直线，同时仍保持正确的边缘分布？

这促使我们提出 *Reflow*：反复应用 `Rectify` 来更新耦合，使得后续的流更易于积分。

核心思想：通过 `Rectify` 实现递归修正。 从乘积耦合上的正则插值开始 $\pi^{(0)} := p_{\mathrm{src}}(\mathbf{x}_0)p_{\mathrm{tgt}}(\mathbf{x}_1)$ ，

$$\mathbf{x}_t = t\mathbf{x}_0 + (1-t)\mathbf{x}_1.$$

应用 `Rectify` 将独立配对替换为依赖配对 $(\mathbf{z}_0, \hat{\mathbf{z}}_1)$ ，经验上诱导出学成场下的更低曲率路径。迭代此更新逐步降低路径曲率（不强制要求直线），提升了数值稳定性与对齐性。

`Reflow` 流程。 每次迭代执行两个步骤：

- **重拟合流程**：从当前耦合的样本中训练一个新的速度场：

$$\boldsymbol{\phi}_{k+1} = \arg\min_{\boldsymbol{\phi}} \mathcal{L}\left(\boldsymbol{\phi} \middle| \pi^{(k)}\right), \quad \text{where}$$
$$\mathcal{L}\left(\boldsymbol{\phi} \middle| \pi^{(k)}\right) := \mathbb{E}_{t,(\mathbf{z}_0^{(k)}, \hat{\mathbf{z}}_1^{(k)})\sim\pi^{(k)}}\left[\left\|\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{z}_t, t) - (\hat{\mathbf{z}}_1^{(k)} - \mathbf{z}_0^{(k)})\right\|^2\right] \tag{5.4.1}$$

with $\mathbf{z}_t = t\mathbf{z}_0^{(k)} + (1-t)\hat{\mathbf{z}}_1^{(k)}$.

- **生成新的耦合**：从新的源样本 $\mathbf{z}_0^{(k+1)} \sim p_{\text{src}}$ 出发求解学成的 ODE：

$$\hat{\mathbf{z}}_1^{(k+1)} \leftarrow \mathbf{z}_0^{(k+1)} + \int_0^1 \mathbf{v}_{\boldsymbol{\phi}_{k+1}}(\mathbf{z}(t), t)\mathrm{d}t,$$

并定义更新后的耦合：

$$\pi^{(k+1)}(\mathbf{z}_0, \mathbf{z}_1) := p_{\text{src}}(\mathbf{z}_0)\delta\left(\mathbf{z}_1 - \hat{\mathbf{z}}_1^{(k+1)}\right).$$

换句话说，`Reflow` 可以看作是重复应用 `Rectify` 算子，生成一系列逐步优化的匹配关系：

$$\pi^{(k+1)} = \texttt{Rectify}\left(\pi^{(k)}\right) \tag{5.4.2}$$

从而使流动和耦合共同演化，形成越来越稳定的传输路径。

5.4.3 `Reflow` 的性质

两个关键的理论性质推动了 `Reflow` 的实用性：它降低了传输成本，并使轨迹变得笔直。

I. `Reflow` 不会增加运输成本。 令 $c(\mathbf{y})$ 为一个凸代价函数（例如，$\|\mathbf{y}\|_2^p$ 且 $p \geq 1$）。每个 `Rectify` 步骤生成一个新的耦合 $(\mathbf{z}_0, \hat{\mathbf{z}}_1)$ ，其代价不劣于原始情况：

Proposition 5.4.1: 修正**可能降低传输成本**

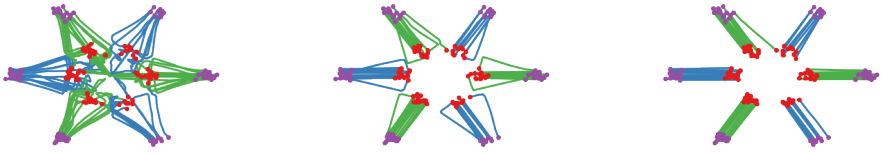
假设存在理想速度场 $\mathbf{v}^* = \mathbf{v}_{\boldsymbol{\phi}^\times}$ ，则有：

$$\mathbb{E}\left[c\left(\hat{\mathbf{z}}_1 - \mathbf{z}_0\right)\right] \leq \mathbb{E}\left[c\left(\mathbf{x}_1 - \mathbf{x}_0\right)\right].$$

Proof for Proposition.

由詹森不等式可得。完整推导参见 **liu2022flow** 。

将此结果递归应用，表明 `Reflow` 过程不会增加运输成本。



(a) 1-修正流 (b) 2-修正流 (c) 3-修正流

图 5.7: 从 liu2022flow 体现的 `Reflow` 示意图。经过 `Rectify` 处理后，路径逐渐变得笔直。

II. `Reflow` 使路径变直。 我们迭代 `Reflow` 的次数越多，学成的轨迹可能就越直。为了衡量这一点，定义路径 $\mathbf{Y} = \{\mathbf{y}_t\}_{t\in[0,1]}$ 的 直度泛函为

$$\mathcal{S}(\mathbf{Y}) := \int_0^1 \mathbb{E}\left[\left\|\mathbf{y}_1 - \mathbf{y}_0 - \frac{\mathrm{d}\mathbf{y}_t}{\mathrm{d}t}\right\|_2^2\right] \mathrm{d}t.$$

如果 $\mathcal{S}(\mathbf{Y}) = 0$ ，则 $\mathbf{Y}$ 恰好是一条直线。

Proposition 5.4.2: `Reflow` 拉直随机路径

对于修正路径 $\mathbf{Z}^{(k)}$ ，有：

$$\min_{k\in\{0,\dots,K\}} \mathcal{S}(\mathbf{Z}^{(k)}) \le \frac{\mathbb{E}\left[\|\mathbf{x}_1 - \mathbf{x}_0\|^2\right]}{K}.$$

Proof for Proposition.

参见 **liu2022rectified** 的定理 3.7。

带有线性插值内核的 FM 或 RF 公式，结合 `Reflow` 过程，提供了更简单的训练目标和一种实用的方法来优化随机耦合。有关理论细节，我们建议读者参考 (**liu2022flow**; **liu2022rectified**)。

三、与最优传输的联系。 最后，我们注意到直线耦合在最优传输（OT）的意义上并不一定是最优的。这涉及一些将在 Section 7.2 中介绍的术语，因此不熟悉 OT 的读者可参考该部分。

二次代价最优传输的一个显著特征是粒子沿直线运动：一个位于 $\mathbf{x}_0$ 的粒子通过 $\mathbf{T}(\mathbf{x}_0)$ 移动到 $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{T}(\mathbf{x}_0)$ ，其中 $\mathbf{T}$ 是最优传输映射。然而，并非每一个生成此类直线路径的映射 $\mathbf{S}$ ，即满足 $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{S}(\mathbf{x}_0)$ 的映射，都是最优的。只有当映射 $\mathbf{S}$ 最小化 Monge 代价 $\mathbb{E}[\|\mathbf{x}_0 - \mathbf{S}(\mathbf{x}_0)\|^2]$ 时，才能

得到最优流。因此，虽然直线路径是必要的，但并不充分；最优性还取决于正确的终点映射 $\mathbf{T}$ 。

Example: Straight Couplings Need Not Be Optimal

Let $p_{\text{src}} = p_{\text{tgt}} = \mathcal{N}(\mathbf{0}, \mathbf{I})$. For the cost $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^p$ with $p > 0$, the c-optimal coupling is the identity coupling π^*, where π^* is the law of $(\mathbf{x}, \mathbf{x})$ with $\mathbf{x} \sim p_{\text{src}}$.

Now consider the coupling $\pi_{\mathbf{A}}$ defined as the law of $(\mathbf{x}, \mathbf{A}\mathbf{x})$, where $\mathbf{x} \sim p_{\text{src}}$ and $\mathbf{A}$ is a rotation matrix satisfying $\mathbf{A}^\top \mathbf{A} = \mathbf{I}$, $\det(\mathbf{A}) = 1$, $\mathbf{A} \neq \mathbf{I}$, and -1 is not an eigenvalue. Then $\pi_{\mathbf{A}}$ is a valid coupling of p_{src} and p_{tgt}, and corresponds to straight-line paths between $\mathbf{x}$ and $\mathbf{A}\mathbf{x}$, but it is not c-optimal for any twice-differentiable strictly convex cost c with invertible Hessian. The suboptimality arises from the rotational transformation. As discussed in Equation (7.5.2), even removing the rotation may not lead to an optimal coupling. ■

我们将继续探讨与 OT 在 Section 7.5.2中的联系。

5.5 闭幕词

本章阐明了扩散模型的第三种也是最后一种基础视角，这种视角建立在确定性流的原理之上。我们的探索始于归一化流（NFs），它利用变量变换公式，学习从简单先验到数据分布的确切、可逆映射。随后，这一概念演变为连续时间过程，即神经微分方程（Neural ODEs），其中学习得到的速度场控制着变换过程。然而，这种方法存在一个显著缺点，即在训练环中需要进行昂贵的微分方程仿真。

现代的流匹配（Flow Matching, FM）框架被提出作为解决这一挑战的一种优雅且高效的方法。通过预先定义一个概率路径 $\{p_t\}_t$ 以及满足连续性方程的相应速度场，FM 为常微分方程（ODE）流确立了一个明确的目标。关键在于，正如我们在变分和基于得分的观点中所见，FM 采用了一种强大的条件化技巧。这将难以处理的边际速度场匹配问题转化为对已知条件速度的简单且易处理的回归问题，使得训练完全无需仿真。从这一视角来看，扩散模型本身可被视为学习一种确定性流以将高斯先验传输至数据分布的一个特例。

随着基于流的视角的引入，我们对扩散建模三大概念支柱的综述现已完成。在整个过程中，一个显著的模式逐渐显现：尽管各个框架在变分自编码器、能量模型或流模型中有着不同的起源，但它们最终都趋向于连续时间生成过程，并依赖于一种条件化策略以实现易处理的学习。

在下一章中，我们将最终将这些并行的线索整合成一个单一的、统一的框架。我们将：

1. 严格证明变分、基于得分和基于流的观点不仅类比，而且在根本层面具有数学等价性。

2. 展示福克-普朗克方程如何作为普遍定律，统一支配三种视角下的密度演化，揭示它们仅仅是描述同一核心生成原理的不同视角。

这一统一的视角将提供对现代扩散范式的全面而系统的理解。

6

扩散模型的统一与系统性视角

> 数学是给不同事物赋予相同名称的艺术。
>
> 亨利 · 庞加莱

本章提出了一种系统性的观点，将变分、得分基础和流基础的视角统一在一个连贯的框架中。尽管其出发点各异，这些方法均聚焦于现代扩散方法背后的同一核心机制。基于 Chapters 2 to 5，我们观察到一种通用范式：定义一个前向破坏过程，该过程追踪一系列边缘分布的路径，然后学习一个随时间变化的向量场，沿着此路径将简单的先验分布传输至数据分布。

所有视角中的一个关键要素是 Section 6.1 中引入的条件化技巧，该技巧将难以处理的边缘目标变换为易处理的条件目标，从而实现稳定且高效的训练。

在 Section 6.2 中，我们以系统化的方式分析训练目标，识别其基本组成部分，并阐明在变分、基于得分和基于流的观点下损失函数的构建方式。

Section 6.3 表明，任何形式的仿射前向噪声注入 $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}$ 都可以等价地变换为标准线性调度 $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\boldsymbol{\epsilon}$。此外，常见的参数化方法如噪声预测、干净数据预测、得分预测和速度预测在梯度层面是可互换的。因此，噪声调度和参数化的选择均遵循相同的建模原则。

最后，Section 6.4 将讨论整合起来，并识别出支配性规则：福克-普朗克方程。无论从变分方案（离散时间降噪）、得分方法（SDE 公式）还是流基方法（ODE 公式）的角度来看，每种方法都构建了一个生成器，其边缘分布遵循相同

的密度演化过程。因此，福克-普朗克方程成为三种观点共同遵守的通用约束，差异仅体现在参数化方式和训练目标上。

6.1 条件技巧：扩散模型的秘诀

迄今为止，我们已经从三种看似不同的起源探讨了扩散模型：变分、基于得分和基于流的视角。每种方法最初都是由不同的目标所驱动，并导致了各自独立的训练目标（固定 t ）：

- **变分视角**：通过最小化以下目标来学习一个参数化的密度 $p_{\boldsymbol{\phi}}(\mathbf{x}_{t-\Delta t}|\mathbf{x}_t)$，以逼近最优的逆转移 $p(\mathbf{x}_{t-\Delta t}|\mathbf{x}_t)$：

$$\mathcal{J}_{\mathrm{KL}}(\boldsymbol{\phi}) := \mathbb{E}_{p_t(\mathbf{x}_t)}\left[\mathcal{D}_{\mathrm{KL}}\big(p(\mathbf{x}_{t-\Delta t}|\mathbf{x}_t)\|p_{\boldsymbol{\phi}}(\mathbf{x}_{t-\Delta t}|\mathbf{x}_t)\big)\right];$$

- **基于得分的视角**：学习一个得分模型 $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)$ 以通过以下方式近似边际得分 $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)$ ：

$$\mathcal{J}_{\mathrm{SM}}(\boldsymbol{\phi}) := \mathbb{E}_{p_t(\mathbf{x}_t)}\left[\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)\|_2^2\right];$$

- **基于流的视角**：学习一个速度模型 $\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)$ 以匹配真实速度 $\mathbf{v}_t(\mathbf{x}_t)$（例如，由 Equation (5.2.10) 定义）通过最小化：

$$\mathcal{J}_{\mathrm{FM}}(\boldsymbol{\phi}) := \mathbb{E}_{p_t(\mathbf{x}_t)}\left[\|\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \mathbf{v}_t(\mathbf{x}_t)\|_2^2\right].$$

乍看之下，这些目标似乎难以处理，因为它们都要求访问在一般情况下根本无法获知的预言量。但接下来便是令人兴奋的转折：每种方法独立地得出了同一个优雅的解决方案：*基于数据 $\mathbf{x}_0$ 进行条件化*。该技术将每个难以处理的训练目标转化为易处理的目标。

这种优雅的“条件化技巧”将目标重写为已知高斯条件分布 $p_t(\mathbf{x}_t|\mathbf{x}_0)$ 上的期望，从而得到梯度等价的闭式回归目标和易处理的训练目标：

- **变分视角** (Equation (2.2.3))**:**

$$\mathcal{J}_{\mathrm{KL}}(\boldsymbol{\phi}) = \underbrace{\mathbb{E}_{\mathbf{x}_0}\mathbb{E}_{p_t(\mathbf{x}_t|\mathbf{x}_0)}\left[\mathcal{D}_{\mathrm{KL}}\big(p(\mathbf{x}_{t-\Delta t}|\mathbf{x}_t, \mathbf{x}_0)\|p_{\boldsymbol{\phi}}(\mathbf{x}_{t-\Delta t}|\mathbf{x}_t)\big)\right]}_{\mathcal{J}_{\mathrm{CKL}}(\boldsymbol{\phi})} + C;$$

- **基于得分的视角** (Equation (3.3.3))**:**

$$\mathcal{J}_{\mathrm{SM}}(\boldsymbol{\phi}) = \underbrace{\mathbb{E}_{\mathbf{x}_0}\mathbb{E}_{p_t(\mathbf{x}_t|\mathbf{x}_0)}\left[\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)\|_2^2\right]}_{\mathcal{J}_{\mathrm{DSM}}(\boldsymbol{\phi})} + C;$$

■ **基于流的视角** (Equation (5.2.9))**:**

$$\mathcal{J}_{\mathrm{FM}}(\boldsymbol{\phi}) = \underbrace{\mathbb{E}_{\mathbf{x}_0}\mathbb{E}_{p_t(\mathbf{x}_t|\mathbf{x}_0)}\left[\left\|\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t,t)-\mathbf{v}_t(\mathbf{x}_t|\mathbf{x}_0)\right\|^2\right]}_{\mathcal{J}_{\mathrm{CFM}}(\boldsymbol{\phi})} + C.$$

为了构建统一的视角，我们接下来以系统化的方式重新审视条件 KL、得分和速度目标。关键的是，这些目标不仅易处理，而且在常数垂直平移的意义下与原始形式等价。条件版本（$\mathcal{J}_{\mathrm{CKL}}$，$\mathcal{J}_{\mathrm{DSM}}$，$\mathcal{J}_{\mathrm{CFM}}$）与原始版本（$\mathcal{J}_{\mathrm{KL}}$，$\mathcal{J}_{\mathrm{SM}}$，$\mathcal{J}_{\mathrm{FM}}$）仅存在这一平移差异，这使得梯度保持不变，从而保留了最优化地形。因此，最小化点仍能唯一地对应于真实的预言机目标，因为每个问题都简化为一个最小二乘回归问题，其解可恢复相应的条件期望：

$$\begin{aligned}
p^*(\mathbf{x}_{t-\Delta t}|\mathbf{x}_t) &= \mathbb{E}_{\mathbf{x}_0\sim p(\cdot|\mathbf{x}_t)}\left[p(\mathbf{x}_{t-\Delta t}|\mathbf{x}_t,\mathbf{x}_0)\right] &&= p(\mathbf{x}_{t-\Delta t}|\mathbf{x}_t),\\
\mathbf{s}^*(\mathbf{x}_t,t) &= \mathbb{E}_{\mathbf{x}_0\sim p(\cdot|\mathbf{x}_t)}\left[\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{x}_0)\right] &&= \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t), \qquad (6.1.1)\\
\mathbf{v}^*(\mathbf{x}_t,t) &= \mathbb{E}_{\mathbf{x}_0\sim p(\cdot|\mathbf{x}_t)}\left[\mathbf{v}_t(\mathbf{x}_t|\mathbf{x}_0)\right] &&= \mathbf{v}_t(\mathbf{x}_t).
\end{aligned}$$

这并非巧合：通过使训练变得易处理，这些条件形式揭示了深刻的合一性。变分扩散、基于得分的随机微分方程和流匹配只是同一原理的不同方面。三种视角，一个洞见，优雅地相互关联。

我们将继续在本章余下部分探讨它们的等价性。

6.2 阐明扩散模型训练损失的路线图

本节构建了扩散模型训练损失的系统性视角。在 Section 6.2.1 中，我们将标准的三个目标扩展到更广泛的四组参数化形式，展示了它们如何从不同的建模视角中产生。在 Section 6.2.2 中，我们进一步将这些结果提炼为一个通用框架，解耦了扩散目标的结构，为 Section 6.3 中的等价性结果奠定了基础。

6.2.1 扩散模型中的四种常见参数化方法

在本节中，我们考虑前向扰动核

$$p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \alpha_t \mathbf{x}_0, \sigma_t^2 \mathbf{I}\right),$$

其中 $\mathbf{x}_0 \sim p_{\text{data}}$ 按照 Equation (4.4.1) 中的定义，除非另有说明。

令 $\omega : [0, T] \to \mathbb{R}_{>0}$ 表示一个正的时间权重函数。四种标准参数化（噪声 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}}$，纯净 $\mathbf{x}_{\boldsymbol{\phi}}$，得分 $\mathbf{s}_{\boldsymbol{\phi}}$，以及速度 $\mathbf{v}_{\boldsymbol{\phi}}$），连同它们各自的极小化器 $\boldsymbol{\epsilon}^*$，$\mathbf{x}^*$，$\mathbf{s}^*$，和 $\mathbf{v}^*$，如下所示，以方便清晰说明并促进进一步讨论。

变分视角。 基于 DDPM 中的 KL 散度（见 Sections 2.2.4 and 4.4.3），该方法简化为预测产生 $\mathbf{x}_t$ 的预期噪声，或预测 $\mathbf{x}_t$ 被扰动前的预期干净信号。

1. **$\boldsymbol{\epsilon}$ -预测（噪声预测）** (**ho2020denoising**):

$$\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) \approx \mathbb{E}[\boldsymbol{\epsilon}|\mathbf{x}_t] = \boldsymbol{\epsilon}^*(\mathbf{x}_t, t) \tag{6.2.1}$$

   以训练目标为导向

$$\mathcal{L}_{\text{noise}}(\boldsymbol{\phi}) := \mathbb{E}_t\left[\omega(t)\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left\|\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \boldsymbol{\epsilon}\right\|_2^2\right].$$

   此处，$\boldsymbol{\epsilon}^*$ 表示为获得给定的 $\mathbf{x}_t$ 而注入的平均噪声。

2. **$\mathbf{x}$ -预测（干净预测）** (**kingma2021variational**; **karras2022elucidating**; **song2023consistency**):

$$\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) \approx \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] = \mathbf{x}^*(\mathbf{x}_t, t) \tag{6.2.2}$$

以训练目标为导向

$$\mathcal{L}_{\text{clean}}(\boldsymbol{\phi}) := \mathbb{E}_t \left[\omega(t) \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left\| \mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \mathbf{x}_0 \right\|_2^2 \right].$$

此处，$\mathbf{x}^*$ 表示在给定噪声观测 $\mathbf{x}_t$ 的条件下，所有合理干净猜测的平均值。

基于得分的视角。 在噪声水平 t 处预测评分函数，该函数指向平均方向，将 $\mathbf{x}_t$ 逐步去噪，使其回到所有可能生成它的干净样本：

3. **得分预测** (**song2019generative**; **song2020score**):

$$\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = \mathbb{E}\left[\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{x}_0) | \mathbf{x}_t \right] = \mathbf{s}^*(\mathbf{x}_t, t) \tag{6.2.3}$$

以训练目标为导向

$$\mathcal{L}_{\text{score}}(\boldsymbol{\phi}) := \mathbb{E}_t \left[\omega(t) \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left\| \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{x}_0) \right\|_2^2 \right],$$

其中，条件得分满足 $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{x}_0) = -\frac{1}{\sigma_t} \boldsymbol{\epsilon}$。

基于流的视图。 预测数据随 $\mathbf{x}_t$ 演变过程中的瞬时平均速度：

4. **v -预测(速度预测)** (**lipman2022flow**; **liu2022rectified**; **salimans2021progressive**; **albergo2023stochastic**):

$$\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) \approx \mathbb{E}\left[\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} \middle| \mathbf{x}_t \right] = \mathbf{v}^*(\mathbf{x}_t, t) \tag{6.2.4}$$

以训练目标为导向

$$\mathcal{L}_{\text{velocity}}(\boldsymbol{\phi}) := \mathbb{E}_t \left[\omega(t) \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left\| \mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \mathbf{v}_t(\mathbf{x}_t | \mathbf{x}_0, \boldsymbol{\epsilon}) \right\|_2^2 \right],$$

其中，条件速度为 $\mathbf{v}_t(\mathbf{x}_t | \mathbf{x}_0, \boldsymbol{\epsilon}) = \alpha_t' \mathbf{x}_0 + \sigma_t' \boldsymbol{\epsilon}$ 。

此处，$\mathbf{v}^*$ 表示经过观测点 $\mathbf{x}_t$ 的平均速度向量。

基于 Equation (6.1.1) 的洞察，所有四种预测类型最终都旨在近似一种条件期望，其形式为给定观测到的 $\mathbf{x}_t$ 时的平均噪声、干净数据、得分或速度。

6.2.2 解耦扩散模型的训练目标

如 Section 6.2.1 所示，四种预测类型的目标函数在扩散模型训练中通常共享以下模板形式：

$$\mathcal{L}(\boldsymbol{\phi}) := \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}} \underbrace{\mathbb{E}_{p_{\text{time}}(t)}}_{\substack{\text{time}\\ \text{distribution}}} \Big[\underbrace{\omega(t)}_{\substack{\text{time}\\ \text{weighting}}} \underbrace{\Big\| \text{NN}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - (A_t\mathbf{x}_0 + B_t\boldsymbol{\epsilon}) \Big\|_2^2}_{\text{MSE part}} \Big]. \tag{6.2.5}$$

在此，为了提高训练效率并优化扩散模型学习流水线，几个关键的设计选择至关重要 (**karras2022elucidating**; **lu2024simplifying**)：

(A) 前向过程中的 $\mathbf{x}_t$ 通过 α_t 和 σ_t 的噪声调度；

(B) $\text{NN}_{\boldsymbol{\phi}}$ 的预测类型及其相关的回归目标 $(A_t\mathbf{x}_0 + B_t\boldsymbol{\epsilon})$；

(C) 时间权重函数 $\omega(\cdot) : [0, T] \to \mathbb{R}_{\geq 0}$；

(D) 时间分布 p_{time} .

我们在此详细阐述这四个组成部分，以作为后续各部分内容讨论的指南。

(A) 噪声调度 α_t 和 σ_t 。 用户可以根据其应用需求灵活选择合适的调度方案，常见的例子总结在 Table 5.2 中。重要的是，正如我们在 Equations (6.3.3) and (6.3.5) 中将要证明的，所有形如 $\mathbf{x}_t = \alpha_t\mathbf{x}_0 + \sigma_t\boldsymbol{\epsilon}$ 的仿射流都是数学等价的。具体而言，通过适当的时序重参数化和空间再缩放，任何此类插值都可以转换为正则的线性调度 ($\alpha_t = 1 - t$, $\sigma_t = t$) 或三角调度 ($\alpha_t = \cos t$, $\sigma_t = \sin t$)。

(B) 参数化 NN_ϕ 和训练目标 $A_t\mathbf{x}_0 + B_t\boldsymbol{\epsilon}$. 用户可以灵活选择模型的预测目标：干净信号、噪声、得分或速度预测。如 Section 6.2.1 所述，所有这些预测类型共享一种形式为

$$\text{Regression Target} = A_t\mathbf{x}_0 + B_t\boldsymbol{\epsilon},$$

其中，系数 A_t 和 B_t 依赖于所选择的预测类型以及调度 (α_t, σ_t)。这些关系在 Table 6.1 中进行了总结。

尽管这四种参数化形式看似不同，我们将在 Equation (6.3.1) 中证明它们可以通过简单的代数变换相互转化。此外，我们还将在 Equation (6.3.6) 中表明，

Equation (6.2.5) 中的平方-ℓ_2 损失项在所有预测类型中保持梯度等价性，仅通过一个依赖于噪声调度 (α_t, σ_t) 的时间加权因子（超出 $\omega(t)p_{\text{time}}(t)$ ）有所差异。

表 6.1: 不同参数化方法之间的关系总结。所有四种参数化方法在数学上是等价的，可以通过简单的代数变换相互转换。

Regression Target =	$A_t\mathbf{x}_0 + B_t\boldsymbol{\epsilon}$	
	A_t	B_t
Clean	1	0
Noise	0	1
Conditional Score	0	$-\frac{1}{\sigma_t}$
Conditional Velocity	α_t'	σ_t'

(C) 时间分布 $p_{\text{time}}(t)$. 由于训练损失是关于 t 的期望，从 $p_{\text{time}}(t)$ 采样多次在数学上等价于将每个 t 的均方误差按 $p_{\text{time}}(t)$ 加权；该因子可以并入现有的时间加权 $\omega(t)$ [1]. 然而，实证证据[2] 表明不同的 $p_{\text{time}}(t)$ 选择会影响性能。因此，我们分别讨论时间分布 $p_{\text{time}}(t)$ and 时间权重函数 $\omega(t)$ 。

时间分布的常见选择是区间 $[0, T]$ (**ho2020denoising**; **song2020score**; **lipman2022flow**; **liu2022rectified**) 上的均匀分布。其他可选方案包括对数正态分布 (**karras2022elucidating**) 和自适应重要性采样法 (**song2021maximum**; **kingma2021variational**)。

(D) 时间加权函数 $\omega(t)$. 一种常见的权重函数选择是常数权重 $\omega \equiv 1$ (**ho2020denoising**; **karras2022elucidating**; **lipman2022flow**; **liu2022rectified**)，尽管也提出了自适应权重方案 (**karras2023analyzing**)。某些 $\omega(t)$ 的选择可将 Equation (6.2.5) 转化为负对数似然的更紧上界，从而将目标重新表述为极大似然训练。对于 $\omega(t)$ 的显著权重方案包括设置 $\omega(t) = g^2(t)$ (**song2021maximum**)，其中 g 为前向 SDE 中的扩散系数，见 Equation (4.1.3)。其他方法采用信噪比（SNR）权

[1] 我们的目标总体随时间变化的目标是一个如下形式的积分

$$\mathcal{L} = \int_0^T \omega(t)\mathtt{mse}(t)\,\mathrm{d}t,$$

其中 $\mathtt{mse}(t)$ 表示每个 t 的均方误差类项。如果在训练过程中进行 $t \sim p_{\text{time}}(t)$，通过

$$\widehat{\mathcal{L}} = \mathbb{E}_{t\sim p_{\text{time}}}\left[\tfrac{\omega(t)}{p_{\text{time}}(t)}\mathtt{mse}(t)\right],$$

可得到 $\mathcal{L}$ 的无偏蒙特卡罗估计量，即采样与加权可通过重要性加权相互替换。

[2] 在实际应用中，我们通过在离散的时间点上使用小批量随机梯度下降来近似训练目标。在这种近似下，$p_{\text{time}}(t)$ 的不同选择会同时影响梯度的方差以及每个时间步的有效权重。因此，我们分别讨论 $p_{\text{time}}(t)$（采样）和 $\omega(t)$（加权）。

重 (**kingma2021variational**) 或单调权重函数 (**kingma2023understanding**)，其中 $\omega(t)$ 为时间的一个单调函数。

总体而言，无论选择何种噪声调度、预测类型或时间采样分布，这些因子在理论上都会影响目标函数中的时间加权。这种时间加权可能会影响实际训练过程的格局，进而影响模型性能。

6.3 扩散模型中的等价性

Section 6.2.1 中引入的四种预测类型将在后续 (Section 6.3.1) 中被证明在梯度最小化下是等价的。随后我们在 Section 6.3.3 中扩展了这一观点，表明不同的前向噪声调度可通过简单的时空再缩放相互关联。

6.3.1 四种预测类型等价

我们首先分析 Equation (6.2.5) 中组件 (B) 的设计选择。

我们已经看到，四种预测类型并非独立的选择，而是同一潜在量的不同视角。例如，噪声预测与干净预测直接相关 (Section 2.2.4)，得分预测与噪声预测也是如此 (Section 3.4)。这种反复出现的模式指向一个更深层次的原则：这四种参数化在代数上是等价的，可以通过简单的变换相互转换。为了精确描述这一联系，我们提出以下命题，如 Figure 6.1所示，参考 (**kingma2021variational**)。

Proposition 6.3.1: 参数化等价性

令最小化各自目标的预测分别为

$$\boldsymbol{\epsilon}^*(\mathbf{x}_t, t), \quad \mathbf{x}^*(\mathbf{x}_t, t), \quad \mathbf{s}^*(\mathbf{x}_t, t), \quad \mathbf{v}^*(\mathbf{x}_t, t),$$

对应噪声、干净数据、得分和速度参数化。这些预测满足以下等价关系：

$$\begin{aligned}
\boldsymbol{\epsilon}^*(\mathbf{x}_t, t) &= -\sigma_t \mathbf{s}^*(\mathbf{x}_t, t), \\
\mathbf{x}^*(\mathbf{x}_t, t) &= \frac{1}{\alpha_t}\mathbf{x}_t + \frac{\sigma_t^2}{\alpha_t}\mathbf{s}^*(\mathbf{x}_t, t), \\
\mathbf{v}^*(\mathbf{x}_t, t) &= \alpha_t' \mathbf{x}^* + \sigma_t' \boldsymbol{\epsilon}^* = f(t)\mathbf{x}_t - \frac{1}{2}g^2(t)\mathbf{s}^*(\mathbf{x}_t, t).
\end{aligned} \tag{6.3.1}$$

此处，$f(t)$ 和 $g(t)$ 通过引理 4.4.1 与 α_t 和 σ_t 相关联。此外，这些最小化满足 Equations (6.2.1) to (6.2.4) 中给出的恒等式。

Proof for Proposition.

证明过程与定理 4.2.1 类似，该定理分析了 DSM 目标下各种匹配损失的全局最优解。详见 Section D.4。

Equation (6.3.1) 在每个 t 上（给定前向加噪系数）诱导出四种参数化之间

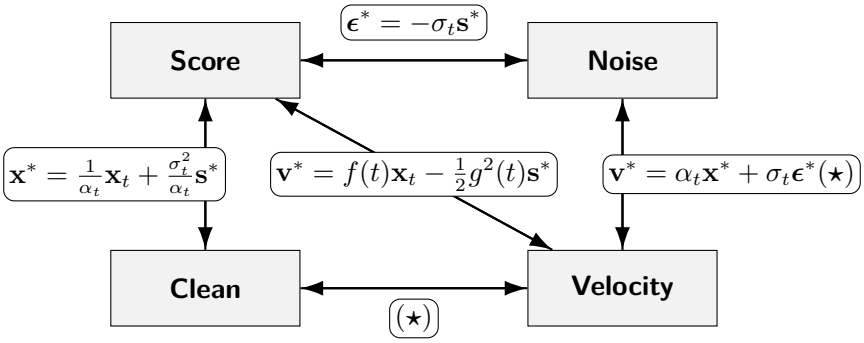


图 6.1: 四种参数化之间的等价关系。 $\mathbf{v}$ -预测由 $\mathbf{v}^* = \alpha_t \mathbf{x}^* + \sigma_t \boldsymbol{\epsilon}^*$ 给出，其中干净的和 $\boldsymbol{\epsilon}$ -预测可通过 $\mathbf{x}_t = \alpha_t \mathbf{x}^* + \sigma_t \boldsymbol{\epsilon}^*$ 互相转换。

的双射转换

$$\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_t, t), \quad \mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_t, t), \quad \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t), \quad \mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t, t).$$

在实际中，我们训练一个单一的网络以一种参数化方式（例如，$\boldsymbol{\epsilon}_{\boldsymbol{\phi}}$）。然后，其他量通过 Equation (6.3.1) 中的转换被后验定义。

6.3.2 不同参数化下的 PF-ODE

PF-ODE 允许多种等价的参数化形式（得分、噪声、去噪和速度）。尽管在原则上可以互换，但选择不同参数化会产生实际影响：它会改变向量场的刚性、离散化误差的行为以及最优化的难易程度。对于使用高级 ODE 求解器进行快速采样（见 Chapter 9），从业者通常采用 $\boldsymbol{\epsilon}$ 或 $\mathbf{x}$ 预测，因为它们与求解器输入匹配良好，并能减少误差累积。对于仅使用少量函数评估进行训练的生成器（见 Chapter 11），$\mathbf{x}$ 或 $\mathbf{v}$ 预测通常能得到更平滑的目标函数并提升步间一致性。

我们针对每种参数化形式写出 PF-ODE，并使用 Equation (6.3.1) 明确表示转换过程。结果汇总如下命题。

Proposition 6.3.2: PF-ODE in Different Parameterizations

Let α_t and σ_t be the forward perturbation schedules, and denote time derivatives by $\alpha_t' := \frac{\mathrm{d}\alpha_t}{\mathrm{d}t}$ and $\sigma_t' := \frac{\mathrm{d}\sigma_t}{\mathrm{d}t}$. Then the empirical PF-ODE admits the equivalent forms

$$\begin{aligned}\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} &= \frac{\alpha_t'}{\alpha_t}\mathbf{x}(t) - \sigma_t\left(\frac{\alpha_t'}{\alpha_t} - \frac{\sigma_t'}{\sigma_t}\right)\boldsymbol{\epsilon}^*(\mathbf{x}(t),t)\\ &= \frac{\sigma_t'}{\sigma_t}\mathbf{x}(t) + \alpha_t\left(\frac{\alpha_t'}{\alpha_t} - \frac{\sigma_t'}{\sigma_t}\right)\mathbf{x}^*(\mathbf{x}(t),t)\\ &= \frac{\alpha_t'}{\alpha_t}\mathbf{x}(t) + \sigma_t^2\left(\frac{\alpha_t'}{\alpha_t} - \frac{\sigma_t'}{\sigma_t}\right)\mathbf{s}^*(\mathbf{x}(t),t)\\ &= \alpha_t'\mathbf{x}^*(\mathbf{x}(t),t) + \sigma_t'\boldsymbol{\epsilon}^*(\mathbf{x}(t),t)\\ &= \mathbf{v}^*(\mathbf{x}(t),t).\end{aligned} \tag{6.3.2}$$

为了理解 Score SDE 的记号，我们回顾引理 4.4.1。如果我们设定

$$f(t) = \frac{\alpha_t'}{\alpha_t}, \quad g^2(t) = \frac{\mathrm{d}}{\mathrm{d}t}(\sigma_t^2) - 2\frac{\alpha_t'}{\alpha_t}\sigma_t^2 = 2\sigma_t\sigma_t' - 2\frac{\alpha_t'}{\alpha_t}\sigma_t^2,$$

然后 PF-ODE 可以写成熟悉的得分 SDE 形式：

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = f(t)\mathbf{x}(t) - \frac{1}{2}g^2(t)\mathbf{s}^*(\mathbf{x}(t),t).$$

为了具体说明 PF-ODE 是如何为采样进行离散化的，我们将在 Section 9.2中展示一种广泛使用的基于扩散的 ODE 采样器 DDIM 方案的更新规则。这个例子将展示欧拉离散化如何自然地与 PF-ODE 相联系。

6.3.3 所有仿射流都等价

我们接下来分析 Equation (6.2.5) 中组件 (A) 的设计选择。

状态层面的等价性。 在 FM (**lipman2022flow**) 和 RF (**liu2022rectified**) 中使用的一种方便的正则插值是

$$\mathbf{x}_t^{\mathrm{FM}} = (1-t)\mathbf{x}_0 + t\boldsymbol{\epsilon} = \mathbf{x}_0 + t(\boldsymbol{\epsilon} - \mathbf{x}_0),$$

其速度为常向量 $\boldsymbol{\epsilon} - \mathbf{x}_0$ 。本小节的关键在于，这种选择的表面简单性并非本质：任何仿射插值

$$\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}$$

可以写成正则路径的时间重参数化和缩放版本。定义

$$c(t) := \alpha_t + \sigma_t, \qquad \tau(t) := \frac{\sigma_t}{\alpha_t + \sigma_t} \quad \big(c(t) \neq 0\big).$$

直接的代数重写得到

$$\begin{aligned} \mathbf{x}_t &= \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon} \\ &= \big(\alpha_t + \sigma_t\big) \left(\frac{\alpha_t}{\alpha_t + \sigma_t} \mathbf{x}_0 + \frac{\sigma_t}{\alpha_t + \sigma_t} \boldsymbol{\epsilon} \right) \\ &= c(t) \left(\big(1 - \tau(t)\big) \mathbf{x}_0 + \tau(t) \boldsymbol{\epsilon} \right) = c(t) \mathbf{x}^{\mathrm{FM}}_{\tau(t)}. \end{aligned}$$

因此，每条仿射路径都是正则 FM 路径在变量变换 $t \mapsto \tau(t)$ 与空间再缩放 $\mathbf{x} \mapsto c(t)\mathbf{x}$ 下的像。该等式逐点成立，因此在分布意义下也成立。

对于相关的速度，对 $\mathbf{x}_t = c(t)\mathbf{x}^{\mathrm{FM}}_{\tau(t)}$ 应用链式法则：

$$\begin{aligned} \mathbf{v}(\mathbf{x}_t, t) &:= \frac{\mathrm{d}}{\mathrm{d}t} \left(\alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon} \right) \\ &= \frac{\mathrm{d}}{\mathrm{d}t} \big(c(t) \mathbf{x}^{\mathrm{FM}}_{\tau(t)} \big) \\ &= c'(t) \mathbf{x}^{\mathrm{FM}}_{\tau(t)} + c(t) \tau'(t) \frac{\mathrm{d}}{\mathrm{d}s} \mathbf{x}^{\mathrm{FM}}_{s} \Big|_{s = \tau(t)} \\ &= c'(t) \mathbf{x}^{\mathrm{FM}}_{\tau(t)} + c(t) \tau'(t) \mathbf{v}^{\mathrm{FM}} \Big(\mathbf{x}^{\mathrm{FM}}_{\tau(t)}, \tau(t) \Big), \end{aligned}$$

由于 $\mathbf{v}^{\mathrm{FM}}(\mathbf{x}^{\mathrm{FM}}_{\tau}, \tau) = -\mathbf{x}_0 + \boldsymbol{\epsilon}$ 沿着正则路径。

我们将上述推导总结为以下命题中的正式声明。

Proposition 6.3.3: Equivalence of Affine Flows

Let $\mathbf{x}_t^{\mathrm{FM}} = (1-t)\mathbf{x}_0 + t\boldsymbol{\epsilon}$ and $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}$ with $c(t) := \alpha_t + \sigma_t \neq 0$ and $\tau(t) := \sigma_t/(\alpha_t + \sigma_t)$. Then

$$\begin{aligned} \mathbf{x}_t &= c(t)\mathbf{x}_{\tau(t)}^{\mathrm{FM}}, \\ \mathbf{v}(\mathbf{x}_t, t) &= c'(t)\mathbf{x}_{\tau(t)}^{\mathrm{FM}} + c(t)\tau'(t)\mathbf{v}^{\mathrm{FM}}\left(\mathbf{x}_{\tau(t)}^{\mathrm{FM}}, \tau(t)\right). \end{aligned} \tag{6.3.3}$$

In particular, all affine interpolations are equivalent up to time reparameterization and spatial rescaling.

三角函数流的等价性。 另一种广泛使用的仿射流是三角插值 (**salimans2021progressiv albergo2023stochastic**; **lu2024simplifying**)。作为一个具体例子，我们还证明了任意仿射流都可以用这种形式表示。三角函数路径定义为

$$\mathbf{x}_u^{\mathrm{Trig}} := \cos(u)\mathbf{x}_0 + \sin(u)\boldsymbol{\epsilon}. \tag{6.3.4}$$

设 $R_t := \sqrt{\alpha_t^2 + \sigma_t^2}$ 且假设 $R_t > 0$ 。选取一个角度 τ_t ，使得

$$\cos\tau_t = \frac{\alpha_t}{R_t}, \qquad \sin\tau_t = \frac{\sigma_t}{R_t}.$$

然后每个仿射插值 $\mathbf{x}_t = \alpha_t\mathbf{x}_0 + \sigma_t\boldsymbol{\epsilon}$ 都是一个缩放和重新定时的三角路径：

$$\mathbf{x}_t = \alpha_t\mathbf{x}_0 + \sigma_t\boldsymbol{\epsilon} = R_t\left(\frac{\alpha_t}{R_t}\mathbf{x}_0 + \frac{\sigma_t}{R_t}\boldsymbol{\epsilon}\right) = R_t\mathbf{x}_{\tau_t}^{\mathrm{Trig}}. \tag{6.3.5}$$

该对 (α_t, σ_t) 是平面上的一个点。通过 R_t 进行归一化，将其置于单位圆上，这确定了角度 τ_t ，从而确定了状态 $\mathbf{x}_{\tau_t}^{\mathrm{Trig}}$ ；半径 R_t 给出了整体的尺度。

对 $\mathbf{x}_u^{\mathrm{Trig}}$ 关于 u 求导得到其速度，

$$\mathbf{v}_u^{\mathrm{Trig}} = -\sin(u)\mathbf{x}_0 + \cos(u)\boldsymbol{\epsilon}.$$

通过与 equation 6.3.5 中相同的变量变换，此关系式为速度（以及其他参数化形式类似）提供了闭式转换。

总结上述讨论，我们得出以下结论：

Conclusion 6.3.1:

无论调度 (α_t, σ_t) 为何，包括 VE、VP（如三角函数）、FM 或 RF，仿射插值均可通过适当的时间变量变换和标量再缩放相互转换。

四种参数化方案的训练目标。 设 $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}$ 且 $\sigma_t > 0$，在 (α_t, σ_t) 上可微，满足 $\alpha_t' \sigma_t - \alpha_t \sigma_t' \neq 0$。考虑预言机目标

$$\boldsymbol{\epsilon}^*(\mathbf{x}_t, t) = \mathbb{E}[\boldsymbol{\epsilon}|\mathbf{x}_t], \quad \mathbf{x}_0^*(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t], \quad \mathbf{v}^*(\mathbf{x}_t, t) = \mathbb{E}[\alpha_t' \mathbf{x}_0 + \sigma_t' \boldsymbol{\epsilon}|\mathbf{x}_t].$$

由命题 6.3.1，它们满足

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = -\frac{1}{\sigma_t} \boldsymbol{\epsilon}^*(\mathbf{x}_t, t) = \frac{\alpha_t}{\sigma_t^2} \left(\mathbf{x}_0^*(\mathbf{x}_t, t) - \frac{\mathbf{x}_t}{\alpha_t} \right), \quad \mathbf{v}^* = \alpha_t' \mathbf{x}_0^* + \sigma_t' \boldsymbol{\epsilon}^*.$$

在头部转换下

$$\mathbf{s}_{\boldsymbol{\phi}} \equiv -\frac{1}{\sigma_t} \boldsymbol{\epsilon}_{\boldsymbol{\phi}} \equiv \frac{\alpha_t}{\sigma_t^2} \left(\mathbf{x}_{\boldsymbol{\phi}} - \frac{\mathbf{x}_t}{\alpha_t} \right),$$

速度与得分的转换是

$$\mathbf{s}_{\boldsymbol{\phi}} = \frac{\alpha_t}{\sigma_t(\alpha_t' \sigma_t - \alpha_t \sigma_t')} \mathbf{v}_{\boldsymbol{\phi}} - \frac{\alpha_t'}{\sigma_t(\alpha_t' \sigma_t - \alpha_t \sigma_t')} \mathbf{x}_t,$$

每个样本的平方损失与时间相关的权重相匹配：

$$\begin{aligned} \left\| \mathbf{s}_{\boldsymbol{\phi}} - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \right\|_2^2 &= \frac{1}{\sigma_t^2} \left\| \boldsymbol{\epsilon}_{\boldsymbol{\phi}} - \boldsymbol{\epsilon}^* \right\|_2^2 \\ &= \frac{\alpha_t^2}{\sigma_t^4} \left\| \mathbf{x}_{\boldsymbol{\phi}} - \mathbf{x}_0^* \right\|_2^2 \\ &= \left(\frac{\alpha_t}{\sigma_t(\alpha_t' \sigma_t - \alpha_t \sigma_t')} \right)^2 \left\| \mathbf{v}_{\boldsymbol{\phi}} - \mathbf{v}^* \right\|_2^2. \end{aligned} \tag{6.3.6}$$

由命题 6.3.3，任意仿射流 $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}$ 可通过 $\mathbf{x}_t = c(t) \mathbf{x}_{\tau(t)}^{\mathrm{FM}}$ 转移至正则的 FM 路径，其中 $c(t) = \alpha_t + \sigma_t$ 且 $\tau(t) = \sigma_t / (\alpha_t + \sigma_t)$。求导得

$$\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) = c'(t) \mathbf{x}_{\tau(t)}^{\mathrm{FM}} + c(t) \tau'(t) \mathbf{v}_{\boldsymbol{\phi}}^{\mathrm{FM}} \left(\mathbf{x}_{\tau(t)}^{\mathrm{FM}}, \tau(t) \right), \qquad \mathbf{x}_{\tau(t)}^{\mathrm{FM}} = \frac{\mathbf{x}_t}{c(t)},$$

且该关系对 $\mathbf{v}^*$ 也成立。因此速度损失的变换为

$$\begin{aligned}&\left\|\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t,t)-\mathbf{v}^*(\mathbf{x}_t,t)\right\|_2^2\\=&\left(c(t)\tau'(t)\right)^2\left\|\mathbf{v}_{\boldsymbol{\phi}}^{\mathrm{FM}}\left(\frac{\mathbf{x}_t}{c(t)},\tau(t)\right)-\left(\mathbf{v}^{\mathrm{FM}}\right)^*\left(\frac{\mathbf{x}_t}{c(t)},\tau(t)\right)\right\|_2^2.\end{aligned}$$

基于上述观察，我们得出以下结论：

Conclusion 6.3.2:

得分、噪声、干净和速度训练目标在理论上是等价的，仅相差时间相关的权重（对于速度目标，还涉及由 (α_t,σ_t) 确定的仿射头转换，包含 $\mathbf{x}_t$ ）。

6.3.4 （可选）参数化与正则流的概念分析

尽管我们在前几节中已经证明了四种参数化在数学上是等价的，可以相互变换，且前向仿射噪声注入流等价于正则形式。

$$\mathbf{x}_t^{\mathrm{FM}}=(1-t)\mathbf{x}_0+t\boldsymbol{\epsilon},$$

在本小节中，我们提供进一步的直观理解，并分析将 $\mathbf{v}$ -预测参数化与这种正则仿射流结合使用的潜在优势。

本小节提出一个简单问题：不同的参数化方式和调度策略如何影响模型的学习过程以及我们的采样方式？我们分三个步骤进行：

- **回归目标与调度策略**。我们关注为何将 $\mathbf{v}$ -预测与正则的线性调度 $(\alpha_t,\sigma_t)=(1-t,t)$ 结合是自然的：它能在时间上保持稳定的目标尺度，并消除动力学中的曲率效应。

- **求解器影响**。我们探讨这一参数化概念上如何与数值积分方法相互作用，而具体的例子（如欧拉求解器和赫恩方法）则留待 Sections 9.2.2 and 9.4.5。

在继续之前，我们区分两种类型的速度场以避免分歧。条件速度，作为易处理的训练目标，定义为

$$\mathbf{v}_t(\mathbf{x}_t|\mathbf{z})=\mathbf{x}_t'=\alpha_t'\mathbf{x}_0+\sigma_t'\boldsymbol{\epsilon},\quad\text{where }\mathbf{z}=(\mathbf{x}_0,\boldsymbol{\epsilon}),$$

而用于 PF-ODE 求解推理过程中移动样本的 *Oracle*（边缘化）速度为

$$\mathbf{v}^*(\mathbf{x},t)=\mathbb{E}\big[\mathbf{v}_t(\cdot|\mathbf{z})\big|\mathbf{x}_t=\mathbf{x}\big].$$

视角 1：为什么 $(\alpha_t,\sigma_t)=(1-t,t)$ 是一种自然的调度。 将 $\sigma_t:=\rho(t)$ 和 $\alpha_t:=1-\rho(t)$ 写为随时间变化的 $\rho(t)$，条件速度变为

$$\mathbf{v}_t(\mathbf{x}_t|\mathbf{z})=\rho'(t)(\boldsymbol{\epsilon}-\mathbf{x}_0),\quad \text{where } \mathbf{z}=(\mathbf{x}_0,\boldsymbol{\epsilon}).$$

单元尺度回归目标。 对于正则调度 $\rho(t)=t$，条件速度 $\mathbf{v}_t(\cdot|\mathbf{z})$ 满足

$$\mathbb{E}\left[\|\mathbf{v}_t(\cdot|\mathbf{z})\|_2^2\right]=\mathbb{E}_{\boldsymbol{\epsilon}}\|\boldsymbol{\epsilon}\|_2^2+\mathbb{E}_{\mathbf{x}_0}\|\mathbf{x}_0\|_2^2=D+\underbrace{\operatorname{Tr}\operatorname{Cov}[\mathbf{x}_0]}_{\text{total variance}}+\underbrace{\|\mathbb{E}\mathbf{x}_0\|_2^2}_{\text{mean}}. \tag{6.3.7}$$

因此，期望的目标幅度在 t 中保持恒定。在将数据标准化为零均值和单位协方差（即 $\operatorname{Cov}[\mathbf{x}_0]=\mathbf{I}$）后，两个分量 $\alpha_t'\mathbf{x}_0$ 和 $\sigma_t'\boldsymbol{\epsilon}$ 对所有 t 的贡献相当，从而避免了在端点附近出现梯度爆炸/消失。为了理解这一点，我们考虑扩散模型的训练目标：

$$\mathcal{L}_{\text{velocity}}(\boldsymbol{\phi})=\mathbb{E}_t\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t,t)-\mathbf{v}_t(\mathbf{x}_t|\mathbf{z})\right\|_2^2\right].$$

通过应用链式法则，该损失关于模型参数 $\boldsymbol{\phi}$ 的梯度为

$$\nabla_{\boldsymbol{\phi}}\mathcal{L}_{\text{velocity}}(\boldsymbol{\phi})=2\mathbb{E}_t\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\partial_{\boldsymbol{\phi}}\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t,t)^\top\left(\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t,t)-\mathbf{v}_t(\mathbf{x}_t|\mathbf{z})\right)\right].$$

因此，目标 $\|\mathbf{v}_t(\mathbf{x}_t|\mathbf{z})\|_2$ 的尺度会影响梯度的稳定性：如果在某个 t 处目标坍缩为 0（或爆炸），则梯度倾向于消失（或爆炸），其他条件相同时。采用正则选择 $\rho(t)=t$ 时，Equation (6.3.7) 提供了一个与 t 无关的目标幅度，因此回归信号不会引起端点（$t=0$ 或 $t=1$）的坍缩或爆炸（假设 $\mathbb{E}\|\partial_{\boldsymbol{\phi}}\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t,t)\|^2$ 成立，且任意时间权重均受控）。

正则调度与 v-预测的相互作用。 在仿射路径 $\mathbf{x}_t=\alpha_t\mathbf{x}_0+\sigma_t\boldsymbol{\epsilon}$ 下，预言速度可分解为

$$\mathbf{v}^*(\mathbf{x},t)=\alpha_t'\mathbf{x}^*(\mathbf{x},t)+\sigma_t'\boldsymbol{\epsilon}^*(\mathbf{x},t),$$

其中 $\mathbf{x}^* = \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t = \mathbf{x}]$ 与 $\boldsymbol{\epsilon}^* = \mathbb{E}[\boldsymbol{\epsilon}|\mathbf{x}_t = \mathbf{x}]$。在固定 $\mathbf{x}$ 的条件下进行微分，可得

$$\partial_t \mathbf{v}_t^* = \underbrace{\alpha_t'' \mathbf{x}^* + \sigma_t'' \boldsymbol{\epsilon}^*}_{\text{schedule curvature}} + \alpha_t' \partial_t \mathbf{x}^* + \sigma_t' \partial_t \boldsymbol{\epsilon}^*.$$

使用线性调度 $\alpha_t = 1 - t$, $\sigma_t = t$ 时，曲率项消失 $(\alpha_t'' = \sigma_t'' = 0)$，因此 $\mathbf{v}_t^*$ 的时间变化主要反映后验演化 $(\partial_t \mathbf{x}^*, \partial_t \boldsymbol{\epsilon}^*)$ ，而非调度本身。这一效应在 $\mathbf{v}$ -预测中尤为清晰：系数 α_t', σ_t' 为常数 $(-1$ 与 $+1)$，避免了漂移中额外的与 t 有关的再缩放。相比之下，得分、$\mathbf{x}_0$ 或 $\boldsymbol{\epsilon}$ -参数化通常引入如 σ_t'/σ_t 或 α_t'/α_t 这类比值，即使在采用线性调度时，这些比值在端点附近也可能剧烈变化。因此，尽管并非原则上唯一，但线性 $(1-t, t)$ 调度与 v -预测相结合，为预言速度提供了特别稳定且透明的时间依赖性。

最小化条件能量。 接下来，我们从更理论的角度探讨最优传输（见 Chapter 7）。这里的 *条件动能*衡量了在前向路径上条件速度的总期望运动量，即从 $\mathbf{x}_0$ 到 $\boldsymbol{\epsilon}$ 所需的瞬时移动量（或动能消耗）：

$$\mathcal{K}[\rho] := \int_0^1 \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}}[\|\mathbf{v}_t(\cdot|\mathbf{z})\|_2^2] \, \mathrm{d}t = \left(D + \operatorname{Tr} \operatorname{Cov}[\mathbf{x}_0] + \|\mathbb{E}\mathbf{x}_0\|_2^2\right) \int_0^1 \left(\rho'(t)\right)^2 \mathrm{d}t.$$

最小化 $\mathcal{K}[\rho]$ 等价于在期望下寻找最平滑、能量最低的路径。在边界条件 $\rho(0) = 0$ 和 $\rho(1) = 1$ 下，欧拉-拉格朗日方程 $\rho''(t) = 0$ 给出了极小化器 $\rho(t) = t$ ，对应一条直线条件路径。这意味着，在所有连接 $\mathbf{x}_0$ 和 $\boldsymbol{\epsilon}$ 的光滑插值中，正则流 $\rho(t) = t$ 是在两者之间移动时能量效率最高的方式。我们将在命题 7.5.1 中对此点进行更详细的讨论。

关于 Oracle 速度的说明。 如果改用边际速度定义的能量进行评估

$$\int_0^1 \mathbb{E}_{\mathbf{x}_t}\left[\|\mathbf{v}^*(\mathbf{x}_t, t)\|^2\right] \mathrm{d}t,$$

然后使用 $\mathbf{z} = (\mathbf{x}_0, \boldsymbol{\epsilon})$ 和 $\mathbf{v}_t(\mathbf{x}_t|\mathbf{z}) = \rho'(t)(\boldsymbol{\epsilon} - \mathbf{x}_0)$，

$$\mathbf{v}^*(\mathbf{x}, t) = \mathbb{E}[\mathbf{v}_t(\cdot|\mathbf{z})|\mathbf{x}_t = \mathbf{x}] = \rho'(t)\mathbb{E}[\boldsymbol{\epsilon} - \mathbf{x}_0|\mathbf{x}_t = \mathbf{x}];$$

因此，边缘速度的能量变为

$$\int_0^1 \mathbb{E}_{\mathbf{x}_t\sim p_t}\left[\|\mathbf{v}^*(\mathbf{x}_t,t)\|_2^2\right]\mathrm{d}t = \int_0^1 \mathbb{E}_{\mathbf{x}_t}\left[\|\rho'(t)\mathbb{E}[\boldsymbol{\epsilon}-\mathbf{x}_0|\mathbf{x}_t]\|_2^2\right]\mathrm{d}t = \int_0^1 \left(\rho'(t)\right)^2\kappa(t)\,\mathrm{d}t,$$

其中 $\kappa(t) := \mathbb{E}_{\mathbf{x}_t\sim p_t}\left[\left\|\mathbb{E}[\boldsymbol{\epsilon}-\mathbf{x}_0|\mathbf{x}_t]\right\|_2^2\right]$ 。

因此，*marginal*-最优调度 $\rho(t)$ 未必是线性的。当且仅当 $\kappa(t)$ 为常数时，它才是线性的；一般情况下，欧拉-拉格朗日条件

$$(\kappa(t)\rho'(t))' = 0 \Rightarrow \rho'(t) \propto \frac{1}{\kappa(t)}$$

这意味着最优预言机调度自适应地重新参数化时间。直观上，$\kappa(t)$ 量化了从 $\mathbf{x}_t\sim p_t$ 可以预测多少标签 $(\boldsymbol{\epsilon}-\mathbf{x}_0)$ ：当 $\kappa(t)$ 较大时，预言机流速减慢，反映出预言机速度的期望幅值较高的区域；当 $\kappa(t)$ 较小时，流速加快。因此，尽管条件流使用线性调度 $(1-t,t)$ ，但相应的边际（预言机）动力学通常是非线性的。

视角二：为何速度预测可被视为采样的自然选择。

PF–ODE 在 $\mathbf{x}$ -、$\boldsymbol{\epsilon}$ - 和 $\mathbf{s}$ -预测下的半线性形式。 在干净、噪声和得分参数化下，漂移项呈现半线性形式（参见 Equation (6.3.2) 中的前三个恒等式）：

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \underbrace{L(t)\mathbf{x}(t)}_{\text{linear part}} + \underbrace{\mathbf{N}_{\boldsymbol{\phi}}(\mathbf{x}(t),t)}_{\text{nonlinear part}}, \quad \mathbf{N}_{\boldsymbol{\phi}} \in \{\mathbf{x}_{\boldsymbol{\phi}}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}, \mathbf{s}_{\boldsymbol{\phi}}\}.$$

当线性漂移 $L(t)\mathbf{x}(t)$ 在某些方向上相对于非线性部分以非常不同的速率驱动 $\mathbf{x}(t)$ 的变化时，系统是 刚性的，这意味着漂移的雅克比（在 $\mathbf{x}$ 中）

$$\mathbf{J}(\mathbf{x},t) := L(t) + \nabla_{\mathbf{x}}\mathbf{N}_{\boldsymbol{\phi}}(\mathbf{x},t)$$

特征值的实部相差若干数量级（较大数值对应更快的方向）[3]. 例如，动力学可能

[3]设 PF–ODE 漂移为 $\mathbf{F}(\mathbf{x},t) = L(t)\mathbf{x} + \mathbf{N}_{\boldsymbol{\phi}}(\mathbf{x},t)$ ，并假设 $\mathbf{N}_{\boldsymbol{\phi}}$ 在 $\mathbf{x}$ 上（局部）关于 $\mathrm{Lip}_{\mathbf{N}_{\boldsymbol{\phi}}}(t)$ 满足 Lipschitz 条件。对于邻近状态 $\mathbf{x},\mathbf{y}$ ，

$$\|\tilde{\mathbf{f}}(\mathbf{x},t) - \tilde{\mathbf{f}}(\mathbf{y},t)\| \le \underbrace{\left(\|L(t)\| + \mathrm{Lip}_{\mathbf{N}_{\boldsymbol{\phi}}}(t)\right)}_{=:\,C(t)}\|\mathbf{x}-\mathbf{y}\|.$$

等价地，雅克比（关于 $\mathbf{x}$ ）

$$\mathbf{J}(\mathbf{x},t) = L(t) + \nabla_{\mathbf{x}}\mathbf{N}_{\boldsymbol{\phi}}(\mathbf{x},t)$$

满足 $\|\mathbf{J}(\mathbf{x},t)\|_{\mathrm{op}} \le C(t)$ （即由 $\mathbb{R}^D$ 上的欧几里得范数诱导的算子范数）。因此，$\mathbf{J}$ 的所有特征值的实部在绝对

涉及 $\mathbf{x}(t)$ 中的“快速线性”变化与“慢速非线性”变化并存。在这种情况下，显式求解器必须采用非常小的时间步长才能保持数值稳定。

为解决这一不平衡问题，高阶稳定求解器通常采用一个积分因子，该因子对线性项 $L(t)\mathbf{x}$ 进行解析处理，仅对较慢的非线性余项进行离散化，尽管这会带来额外的代数和实现复杂性。Chapter 9专门针对此主题进行了详细讨论。

PF–ODE 在 $\mathbf{v}$ -预测下。 采用 $\mathbf{v}$ -预测时，模型直接学习速度场并进行积分

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}(t), t) \approx \mathbf{v}^*(\mathbf{x}(t), t).$$

在此公式中，显式的线性项被吸收进一个学成的场中，因此动力学不再分解为分离的部分。步长由此由学成场 $\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}, t)$ 随 $\mathbf{x}$ 和 t 变化的平滑程度决定，而非由预设标量系数 $L(t)$ 的大小决定。换句话说，潜在的快速线性漂移被整合进一个统一的速度场中，减少了时间尺度差异并简化了数值积分。

稍后在 Section 9.2.2 中，我们将通过一个简单示例说明，在采样过程中 $\mathbf{v}$ -预测的结构简洁性。为了获得与 DDIM (**song2020denoising**)(one of the most widely used fast samplers in diffusion modeling) 相同的 PF-ODE 离散化更新，仅使用 $\boldsymbol{\epsilon}$ -、$\mathbf{x}$ - 或 $\mathbf{s}$ -参数化下的普通欧拉步长仅能近似线性项，而非精确计算（见 Equation (9.1.8)）。因此，这些参数化需要采用更高级的方法——指数积分器，以分离并精确计算线性项。相比之下，使用 $\mathbf{v}$ -预测时，PF-ODE 的漂移项中不存在需分离的独立线性项，因此普通欧拉更新自然与 DDIM 公式一致。一个密切相关的类比出现在 Section 9.4.5：二阶 DPM-Solver (**lu2022dpm**) 与经典 Heun 方法一致：对于 $\mathbf{v}$ -预测，这是普通的 Heun 法；而对于 $\boldsymbol{\epsilon}$ -、$\mathbf{x}$ - 或 $\mathbf{s}$ -预测，则为指数 Heun 法。详细讨论我们留到各自章节中进行。

我们注意到，生成性能的任何提升（例如在 PF-ODE 求解中以更少的模型评估次数获得更高样本质量）都取决于 $\mathbf{v}_{\boldsymbol{\phi}}$ 对最优速度近似的准确性，以及采样算法（包括数值积分器、离散化方案和步长控制）与其交互的有效性。因此，采用 $\mathbf{v}$ 参数化本身并不能保证更好的采样性能。

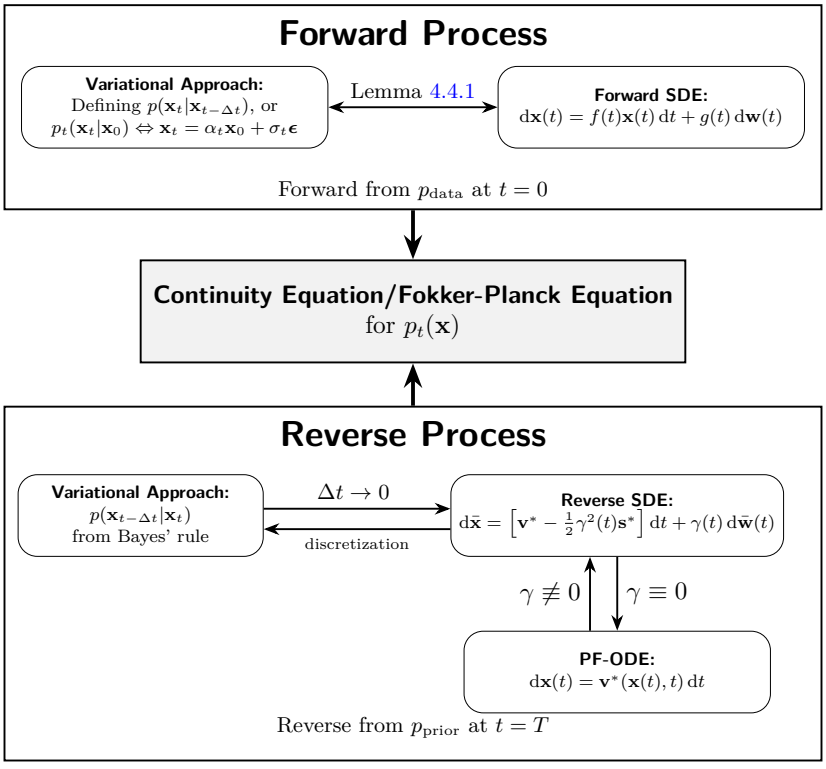
结论。 虽然 $\mathbf{v}$-预测与正则线性调度相结合在理论上具有一些优势，例如恒定的目标幅值和无调度曲率，但这些特性并不一定使其普遍更优。在实际应用中，模型性能取决于一系列相互作用的因素，包括网络结构、规范化方案、随时间变化的损失加权、采样器和离散化步数的选择、引导强度、正则化策略、数据缩放以

值上均被 $C(t)$ 所限制。故较大的 $C(t)$ 意味着快速的局部速率，因此显式求解器需要较小的步长（$h \lesssim 1/C(t)$）。

及整体训练预算。不同的数据集和目标可能更倾向于其他参数化方式或调度策略，而最优配置最终是一个需要通过验证和消融实验来解决的经验性问题。

6.4 其下之源：福克-普朗克方程

图 6.2: 通过连续性方程将变分、随机微分方程（SDE）和常微分方程（ODE）公式统一起来的视角，其中所有 $p_t(\mathbf{x})$ 在共享动力学下演化。速度场 $\mathbf{v}^*(\mathbf{x},t) = f(t)\mathbf{x} - \frac{1}{2}g^2(t)\mathbf{s}^*(\mathbf{x},t)$ 由评分函数 $\mathbf{s}^*(\mathbf{x},t) := \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ 决定。系数 $f(t)$ 、$g(t)$ 、σ_t 和 α_t 是预定义的时间相关函数，$\gamma(t)$ 是可调的时间相关超参数。



在本节中，我们表明扩散模型的三个主要视角——变分法、得分函数法和基于流的方法——并非相互独立的构造，而是源于一个统一的原则：即在选定的前向过程中，密度演化所遵循的连续性（福克-普朗克）方程。

首先，我们回顾一下，Section 4.5 中的分析将基于离散内核和贝叶斯规则的变分视角，与连续动力学的基于得分的 SDE 视角统一起来。我们通过证明变分模型是潜在前向和反向 SDE 的一致离散化来建立这一联系。具体而言，通过离散内核逐步计算的边缘分布演化方式与控制连续时间动力学的福克-普朗克方程一致。这证实了两种视角在本质上是等价的。

然后我们连接基于流和基于得分的观点。在 Section 6.4.1 中，我们表明一个常微分方程流确定了一条密度路径，其边缘分布总能由一族随机过程实现。这

将确定性流与随机随机微分方程置于同一类中。

这些结果将三种观点统一到一个框架之下（见 Figure 6.2）。最后，我们在 Section 6.5 结束本章。

6.4.1 基于流的方法与得分 SDE 的连接

扩散模型的一个显著特点在于，不同的动力系统（确定性或随机性）能够演化出相同概率分布的轨迹。在本节中，我们揭示了基于常微分方程（ODE）的流 Section 5.2 与得分随机微分方程（Score SDEs）之间一种自然且优美的联系。具体而言，我们表明定义生成型 ODE 的速度场可以转化为一个遵循相同福克-普朗克动力学的随机对应形式，从而为确定性插值与随机采样之间建立起一个合理的桥梁。这为我们提供了一类连续的模型家族，从 ODE 到 SDE，均能生成相同的路径数据分布。

我们考虑连续时间情形，其中扰动核由下式给出

$$p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \alpha_t \mathbf{x}_0, \sigma_t^2 \mathbf{I}),$$

其中 $\mathbf{x}_0 \sim p_{\text{data}}$。此条件分布如常诱导出边缘密度路径 $p_t(\mathbf{x}_t) = \mathbb{E}_{\mathbf{x}_0 \sim p_{\text{data}}}[p(\mathbf{x}_t|\mathbf{x}_0)]$，其中 $p_T \approx p_{\text{prior}}$。

为了匹配这条密度路径，考虑常微分方程

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{v}_t(\mathbf{x}(t)), \quad t \in [0, T], \tag{6.4.1}$$

其中 $\mathbf{v}_t(\mathbf{x}) = \mathbb{E}\left[\alpha_t' \mathbf{x}_0 + \sigma_t' \boldsymbol{\epsilon} | \mathbf{x}\right]$ 是如 Equation (5.2.10)(noting that time is flipped to follow the diffusion model convention) 所示的预言机速度。从 $\mathbf{x}(T) \sim p_{\text{prior}}$ 开始将 equation 6.4.1 向后积分，可得到 p_0 的样本。

尽管该常微分方程足以生成高质量的样本，但引入随机性可能提升样本的多样性。这引出了以下问题：

Question 6.4.1

是否存在一个随机微分方程，其从 p_{prior} 出发的动力学行为能够产生与 Equation (6.4.1) 中常微分方程相同的边缘密度？

该声明确认了存在一族反向时间随机微分方程，其诱导的边缘密度路径与相应的 PF-ODE 完全相同。这些随机微分方程所诱导的密度满足同一 Fokker–Planck

方程，因此它们在任意时刻的边缘分布均与给定的插值路径 $\{p_t\}_{t\in[0,T]}$ 一致。[4].

Proposition 6.4.1: 反向时间随机微分方程生成与插值相同的边缘分布

设 $\gamma(t)\geq 0$ 为任意时间相关系数。考虑反向时间随机微分方程

$$\mathrm{d}\bar{\mathbf{x}}(t)=\left[\mathbf{v}^*(\bar{\mathbf{x}}(t),t)-\tfrac{1}{2}\gamma^2(t)\mathbf{s}^*(\bar{\mathbf{x}}(t),t)\right]\mathrm{d}\bar{t}+\gamma(t)\,\mathrm{d}\bar{\mathbf{w}}(t), \tag{6.4.2}$$

该过程从 $\bar{\mathbf{x}}(T)\sim p_T$ 反向演化至 $t=0$ 。则此过程 $\{\bar{\mathbf{x}}(t)\}_{t\in[0,T]}$ 与常微分方程密度路径导出的指定边缘分布 $\{p_t\}_{t\in[0,T]}$ 相匹配。此处 $\mathbf{s}(\mathbf{x},t):=\nabla_{\mathbf{x}}\log p_t(\mathbf{x})$ 为评分函数，其与速度场 $\mathbf{v}(\mathbf{x},t)$ 的关系由下式给出：

$$\mathbf{v}^*(\mathbf{x},t)=f(t)\mathbf{x}-\frac{1}{2}g^2(t)\mathbf{s}^*(\mathbf{x},t),\quad \mathbf{s}^*(\mathbf{x},t)=\frac{1}{\sigma_t}\frac{\alpha_t\mathbf{v}^*(\mathbf{x},t)-\alpha_t'\mathbf{x}}{\alpha_t'\sigma_t-\alpha_t\sigma_t'}. \tag{6.4.3}$$

Proof for Proposition.

对应于 Equation (6.4.2) 的反向时间福克-普朗克方程为

$$\partial_{\bar{t}}p=-\nabla\cdot\left(\left[\mathbf{v}^*-\tfrac{1}{2}\gamma^2\mathbf{s}^*\right]p\right)+\tfrac{1}{2}\gamma^2\Delta p.$$

利用恒等式 $\nabla\cdot(\mathbf{s}^*p)=\Delta p$ （因 $\mathbf{s}^*=\nabla\log p$ ），二阶项相互抵消，可得

$$\partial_{\bar{t}}p=-\nabla\cdot(\mathbf{v}^*p),$$

即对应于概率流常微分方程的一阶（仅漂移项）福克-普朗克方程。因此反向时间随机微分方程与常微分方程导出相同的边缘密度路径 $\{p_t\}$ 。详见 **ma2024sit** 的附录 A.2–A.3 节。

超参数 $\gamma(t)$ 可以任意选择，与 α_t 和 σ_t 无关，甚至在训练之后也可以如此，因为它不会影响速度 $\mathbf{v}(\mathbf{x},t)$ 或得分 $\mathbf{s}(\mathbf{x},t)$。以下是一些示例：

- 情景 $\gamma(t)=0$ 恢复了 Equation (6.4.1) 中的常微分方程。
- 当 $\gamma(t)=g(t)$ 时，Equation (6.4.2) 变为 Equation (4.1.6) 中的反向时间

[4]为了完整性，前向 SDE 表示（此处不需要）为

$$\mathrm{d}\mathbf{x}(t)=f(t)\mathbf{x}(t)\,\mathrm{d}t+g(t)\,\mathrm{d}\mathbf{w}(t),$$

其中 $f(t)$ 和 $g(t)$ 通过 Equation (4.4.2) 与 (α_t,σ_t) 相关联。

SDE，因为最优速度 $\mathbf{v}(\mathbf{x},t)$ 满足（见命题 6.3.1）：

$$\mathbf{v}^*(\mathbf{x},t) = f(t)\mathbf{x} - \frac{1}{2}g^2(t)\mathbf{s}^*(\mathbf{x},t).$$

- 对 $\gamma(t)$ 的其他选择也已进行探索；例如，**ma2024sit** 选择 $\gamma(t)$ 以最小化 p_{data} 与通过求解 Equation (6.4.2) 从 $t=T$ 得到的 $t=0$ 密度之间的 KL 散度。

遵循得分 SDE，训练好的速度场 $\mathbf{v}_{\phi^\times}(\mathbf{x},t)$ 可通过 Equation (6.4.3) 转换为参数化的得分函数 $\mathbf{s}_{\phi^\times}(\mathbf{x},t)$。将其代入 Equation (6.4.2) 定义了一个经验反向时间 *SDE*，可通过从 $t=T$ 数值积分并使用 $\bar{\mathbf{x}}(T)\sim p_{\text{prior}}$ 进行采样。

该命题突显了扩散模型的显著灵活性：一旦固定了 边缘密度路径 $\{p_t\}_{t\in[0,T]}$，便存在一整类动力学系统能够重现它，其中包括 PF-ODE 和反向时间 SDE。

$$\mathrm{d}\bar{\mathbf{x}}(t) = \left[\mathbf{v}^*(\bar{\mathbf{x}},t) - \tfrac{1}{2}\gamma^2(t)\mathbf{s}^*(\bar{\mathbf{x}},t)\right]\mathrm{d}\bar{t} + \gamma(t)\,\mathrm{d}\bar{\mathbf{w}}(t), \qquad \gamma(t)\geq 0.$$

所有此类动力学均满足相同的反向时间福克-普朗克方程，因此产生相同的边缘演化。函数 $\gamma(t)$ 连续调节随机性的水平，而不影响单时间分布，揭示了确定性流基常微分方程与其随机微分方程对应物之间的深刻联系，如 Figure 6.2 所示。

6.5 闭幕词

本章作为我们理论探索的基石，将变分法、基于得分的方法和基于流的方法综合成一个统一而连贯的框架。我们已经证明，这三种看似不同的方法不仅并非彼此平行，而且在深层和根本上紧密相连。

我们的合一建立在两个核心洞见之上。首先，我们发现所有框架共有的秘诀：一种条件化技巧，能够将难以处理的边缘训练目标变换为易处理的条件目标，从而实现稳定且高效的學習。其次，我们确立了福克-普朗克方程是支配概率密度演化的普遍规律。三种视角各自以不同方式构建了一个符合这一基本动态的生成过程。

此外，我们证明了各种模型参数化方式——即噪声、干净数据、得分或速度预测——都是可互换的。这表明预测目标的选择更多是实现方式和稳定性的问题，而非根本性的建模差异。最终的启示是，尽管现代扩散方法来源多样，但它们都体现了同一个核心原理：学习一个随时间变化的向量场，将简单的先验分布转换为数据分布。

在这一统一且严谨的基础牢固建立之后，我们现已具备从基础理论迈向扩散模型的实际应用与加速的条件。生成过程等价于求解微分方程这一核心洞察，为控制与最优化提供了强大平台。本专著的后续部分将利用这一统一认知，解决关键的实际挑战：

1. 第三部分将专注于改进推理阶段的采样过程。我们将探索如何引导生成轨迹以实现可控生成（Chapter 8），并研究先进的数值求解器以显著加速缓慢的迭代采样过程（Chapter 9）。

2. 第 D 部分将超越迭代求解器，直接学习快速生成器。我们将研究能够在仅一步或几步内生成高质量样本的方法，这些方法要么通过从教师模型进行蒸馏实现（Chapter 10），要么通过从零开始训练实现（Chapter 11）。

在统一了扩散模型的什么和为什么之后，我们现在将注意力转向令人兴奋且实用的如何领域。

7

（可选）扩散模型与最优传输

将一个分布映射到另一个分布（生成作为特例）是一个核心挑战。流匹配通过学习一个随时间变化的速度场来解决这一问题，该速度场将质量从源分布传输到目标分布。这自然地与输运理论相联系：经典最优输运寻找分布之间的最小成本路径，而其熵正则化形式——薛定谔桥，则选择相对于参考过程（如布朗运动）最可能的受控扩散。

在本章中，我们回顾了最优传输、熵正则最优传输以及薛定谔桥作为分布到分布问题的表述基础。这引出了一个核心问题：扩散模型在多大程度上实现了这种最优传输？它们有两种视角：一种是作为通过前向和反向随机微分方程定义的*随机过程*，另一种是作为由 PF-ODE 给出的*确定性过程*。随机视角直接与熵正则最优传输相一致，而 PF-ODE 通常并不对应于任何已知的传输目标。这一差距留下了一个开放性问题：在何种条件下，扩散模型可被视为求解最优传输问题？

7.1 分布到分布翻译导言

扩散模型将终态分布固定为标准高斯分布，p_{prior}。然而，许多应用需要 *分布到分布的转换*：将源分布 p_{src} 转换为不同的目标 p_{tgt}。例如，将草图转换为逼真的图像或在不同艺术风格之间进行转换。

现代扩散方法提供了实现这一目标的实用途径。*单端点方法*，如 SDEdit (**meng2021sded**)，从源图像 $t = 0$ 开始，将其扩散至中间步骤 t，然后使用预训练的扩散模型对目标领域进行逆过程操作。这将生成与目标分布风格和内容相匹配的输出。

双端点方法，如双重扩散桥 (**su2022dual**)，通过一个共享的潜在分布（通常为 $t = 1$ 处的高斯分布）连接两个领域。前向概率流常微分方程将样本从 p_{src} 传输到该潜在空间，而基于目标领域的反向常微分方程则将其映射回 p_{tgt}。除了这类采样时间方法外，Section 5.2 中描述的流匹配框架提供了一种基于训练的替代方案：它直接学习一个常微分方程流，连续地将质量从 p_{src} 移动到 p_{tgt}。

至关重要的是，分布之间的变换不仅仅需要两个独立训练的模型。这要求一种有原则的映射，能够同时在两端对齐动力学特性，并以“最便宜”（成本高效）的方式实现。

在本节中，我们不再列举众多基于扩散的翻译应用，而是将重点转向这一经典分布到分布问题的数学基础。特别是，我们强调最优传输（OT）及其熵正则化变体——薛定谔桥（SB），它们长期以来在理论界被视为成本高效（从数学意义上讲）分布变换的正则表述。

其核心问题在于：

Question 7.1.1

> 给定两个概率分布，如何以最高效的方式将一个变换为另一个，同时最小化总的 代价？

其中，代价 $c(\mathbf{x}, \mathbf{y})$ 是一个非负函数，用于为将单位质量从点 $\mathbf{x}$ 移动到点 $\mathbf{y}$ 分配惩罚。一种常见选择是平方距离，$c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$。

本节提供了一个简要概述，以阐明基于扩散的方法（包括流匹配）如何与经典及正则化最优传输相联系。我们旨在探讨的核心问题是：

Question 7.1.2

> 扩散模型是否是一种将 p_{data} 与 p_{prior} 相连接的最优传输形式，以及在何种意义上？

为回答这一问题，我们首先阐明在 Section 7.2 中“最优性”的含义。我们回顾静态 Monge–Kantorovich 形式下的经典 *最优传输 (OT)* (Equation (7.2.1))，

以及其动态的 Benamou–Brenier 形式 (Equation (7.2.3)；在连续性方程约束下最小化动能)，还有熵正则化变体 (熵正则 *OT*)，即 Equation (7.2.5)，其等价于 *Schrödinger* 桥问题 (Equation (7.2.8))。从动态视角看，OT 诱导一个满足连续性方程的确定性流，而 SB 诱导一个受控扩散过程，其边缘分布由 Fokker–Planck 方程演化。我们在 Section 7.3 中提供了这些形式之间的高层次映射。

随后，我们将讨论分为两部分。首先，在 Section 7.4 中，我们说明标准扩散模型中使用的固定前向加噪 SDE 本身并不是任意 p_{src} 与 p_{tgt} 之间的 Schrödinger 桥：前向过程是一种选定的参考扩散，且前向时间或反向时间 SDE 通常并不强制确切地匹配到预定的目标。因此，除非显式地求解具有这些端点的 SB 问题，否则它并非熵正则 OT 的最优解；而由于其以一个起始点为锚点，它却是半桥问题的最优解。

其次，在 Section 7.5 中，我们回到生成情景，其中包含 $p_{\mathrm{src}} = p_{\mathrm{prior}}$（高斯分布）和 $p_{\mathrm{tgt}} = p_{\mathrm{data}}$。PF-ODE 通过构造定义了一个确定性映射，将 p_{prior} 映射到 p_{data}。然而，这种流通常不是针对指定传输成本（例如二次型 W_2）的最优传输映射：它只是众多可接受的确定性耦合中的一种，并未最小化 Benamou–Brenier 动作。接下来我们将讨论“校正流”过程 (Section 5.4.1) 是否能导出最优传输映射；然而，一般情况下并无此类理论保证。因此，扩散模型的 PF-ODE 映射与最优传输之间的确切关系仍然是一个具有挑战性且尚未解决的问题。

7.2 问题设置的分类

在本节中，我们介绍从 p_{src} 到 p_{tgt} 运输质量的最“高效”或“最优”方式的概念。这些概念包括经典的最优运输（OT）及其熵正则化变体，后者具有一个等价表述，称为施罗丁格桥。这一分类体系为后续阐明与扩散模型之间的联系提供了背景。

7.2.1 最优传输（OT）

Monge–Kantorovich（静态）最优传输问题表述。 我们固定一个代价函数 $c : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$，该函数指定了将概率质量从 $\mathbf{x}$ 传输到 $\mathbf{y}$ 的开销。目标是尽可能以最低的成本将源分布 p_{src} 转换为目标分布 p_{tgt}。

为了定义一个代价，我们必须知道哪些配对 $(\mathbf{x}, \mathbf{y})$ 是匹配的。这一作用由一个 *耦合* 所承担：定义在 $\mathbb{R}^D \times \mathbb{R}^D$ 上的联合分布 γ，其边缘分布分别为 p_{src} 和 p_{tgt}。换句话说，采样 $(\mathbf{x}, \mathbf{y}) \sim \gamma$ 意味着将来自源的 $\mathbf{x}$ 与来自目标的 $\mathbf{y}$ 进行匹配。如果 γ 相对于勒贝格测度存在密度 $\gamma(\mathbf{x}, \mathbf{y})$，则边缘约束可表示为

$$\int_{\mathbb{R}^D} \gamma(\mathbf{x}, \mathbf{y})\, \mathrm{d}\mathbf{y} = p_{\mathrm{src}}(\mathbf{x}), \qquad \int_{\mathbb{R}^D} \gamma(\mathbf{x}, \mathbf{y})\, \mathrm{d}\mathbf{x} = p_{\mathrm{tgt}}(\mathbf{y}).$$

也就是说，对 $\mathbf{y}$ 积分可恢复 $\mathbf{x}$ 中的源密度，而对 $\mathbf{x}$ 积分可恢复 $\mathbf{y}$ 中的目标密度。

我们举两个标准例子来说明：

1. **离散支撑.** 若 p_{src} 和 p_{tgt} 在有限多个点上具有支撑，则耦合由一个非负矩阵 (γ_{ij}) 表示，其行和等于 $p_{\mathrm{src}}(i)$，列和等于 $p_{\mathrm{tgt}}(j)$。每个元素 γ_{ij} 表示从 i 发送到 j 的质量。

2. **确定性映射**。若存在一个可测映射 $\mathbf{T}$ 满足 $\mathbf{T}_{\#}p_{\mathrm{src}} = p_{\mathrm{tgt}}$，则 $\gamma = (\mathbf{I}, \mathbf{T})_{\#}p_{\mathrm{src}}$ 是一个确定性耦合，它将每个点 $\mathbf{x}$ 直接移动到 $\mathbf{T}(\mathbf{x})$。

一旦耦合 γ 固定，运输成本即为此方案下的平均单位成本：

$$\int c(\mathbf{x}, \mathbf{y})\, \mathrm{d}\gamma(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma}\big[c(\mathbf{x}, \mathbf{y})\big].$$

在离散情形下，这简化为 $\sum_{i,j} c_{ij}\gamma_{ij}$，而在连续情形下则变为双重积分。在接下来的内容中，我们将仅关注连续情形。

最优运输问题则是从所有可接受的耦合中选择一个使期望成本最小的。

$$\mathrm{OT}(p_{\mathrm{src}}, p_{\mathrm{tgt}}) := \inf_{\gamma \in \Gamma(p_{\mathrm{src}}, p_{\mathrm{tgt}})} \int c(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\gamma(\mathbf{x}, \mathbf{y}), \tag{7.2.1}$$

其中，可行集仅施加边际约束或质量守恒约束：

$$\Gamma(p_{\mathrm{src}}, p_{\mathrm{tgt}}) = \Big\{ \gamma \in \mathcal{P}(\mathbb{R}^D \times \mathbb{R}^D) : \int \gamma(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{y} = p_{\mathrm{src}}(\mathbf{x}), \ \int \gamma(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{x} = p_{\mathrm{tgt}}(\mathbf{y}) \Big\},$$

其中 $\mathcal{P}(\mathbb{R}^D \times \mathbb{R}^D)$ 表示在 $\mathbb{R}^D \times \mathbb{R}^D$ 上所有概率测度的集合。

特殊情况：Wasserstein-2 距离。 Wasserstein-2 距离是带有二次代价 $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$ 的 Monge–Kantorovich 问题的一个特例。它将两个概率分布之间的距离度量为：

$$\mathcal{W}_2^2(p_{\mathrm{src}}, p_{\mathrm{tgt}}) := \inf_{\gamma \in \Gamma(p_{\mathrm{src}}, p_{\mathrm{tgt}})} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} \left[\|\mathbf{x} - \mathbf{y}\|^2 \right].$$

在对 p_{src} 和 p_{tgt} 作出适当假设的前提下，Brenier 定理（见 Theorem 7.1）[1] 保证了二次代价的最优耦合 γ 位于某个确定性映射 $\mathbf{T} : \mathbb{R}^D \to \mathbb{R}^D$ 的图像上。因此，Wasserstein-2 距离可等价地表示为[2]：

$$\mathcal{W}_2^2(p_{\mathrm{src}}, p_{\mathrm{tgt}}) = \inf_{\substack{\mathbf{T}: \mathbb{R}^D \to \mathbb{R}^D, \\ \text{s.t. } \mathbf{T} \# p_{\mathrm{src}} = p_{\mathrm{tgt}}}} \mathbb{E}_{\mathbf{x} \sim p_{\mathrm{src}}} \left[\|\mathbf{T}(\mathbf{x}) - \mathbf{x}\|^2 \right]. \tag{7.2.2}$$

此处，$\mathbf{T} \# p_{\mathrm{src}} = p_{\mathrm{tgt}}$ 表示 $\mathbf{T}$ 将 p_{src} 推向前方至 p_{tgt}，即 $\mathbf{T}(\mathbf{x}) \sim p_{\mathrm{tgt}}$ 对 $\mathbf{x} \sim p_{\mathrm{src}}$。

因此，Wasserstein-2 距离表示在所有匹配给定边缘分布的耦合或传输映射中，最小的期望平方运输成本。用 $\mathbf{T}^*(\mathbf{x})$ 表示的最优传输映射，被称为 *Monge* 映射，它给出了将 p_{src} 高效转换为 p_{tgt} 的最佳方式。

贝纳穆-布伦耶（动态）最优传输公式 与其像蒙日-坎托罗维奇公式那样以静态方式直接映射分布，运输也可以被建模为连续时间流：

$$p_0 := p_{\mathrm{src}} \to p_t \to p_1 := p_{\mathrm{tgt}}, \quad t \in [0, 1].$$

[1] Brenier 定理涉及二次代价下最优传输映射的存在性与结构。特别地，如果 p_{src} 不在维度至多为 $D-1$ 的集合上赋予质量，则最优传输映射 $\mathbf{T}^*$ 唯一存在。

[2] 有三种常用的 $\mathcal{W}_2$ 距离表述形式：Monge 表述（基于最优传输映射）、Kantorovich 表述（基于耦合）以及 Benamou–Brenier 动态表述（见 Equation (7.2.3)）。在适当的正则性条件下，它们是等价的。

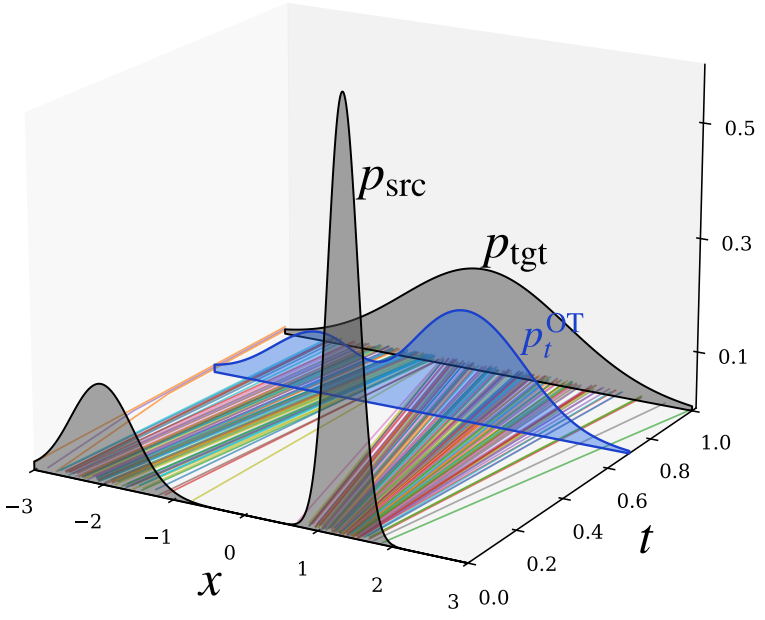


图 7.1: OT 动态视角示意图。插值 p_t^{OT} 随时间连续演化，提供了将 p_{src} 确定性地映射到 p_{tgt} 的最低成本运输方案（McCann 位移插值）。

这种由 **benamou2000computational** 引入的最优传输动态表述，旨在寻找一个光滑的速度场 $\mathbf{v}_t(\mathbf{x})$，用以描述 $p_t(\mathbf{x})$ 中质量随时间的演化。

贝纳莫-布伦耶公式[3] 表明，对于二次代价函数 $c(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_2^2$（即 $\mathcal{W}_2$ 距离），Equation (7.2.1) 中静态最优传输问题的最优值等于动能最小化问题的最优值：

$$\mathcal{W}_2^2(p_{\text{src}}, p_{\text{tgt}}) = \min_{\substack{(p_t,\mathbf{v}_t)\ \text{s.t.}\ \partial_t p_t + \nabla\cdot(p_t\mathbf{v}_t)=0,\\ p_0=p_{\text{src}},\ p_1=p_{\text{tgt}}}} \int_0^1 \int_{\mathbb{R}^D} \|\mathbf{v}_t(\mathbf{x})\|^2 p_t(\mathbf{x})\,\mathrm{d}\mathbf{x}\,\mathrm{d}t \tag{7.2.3}$$

其中 p_t 是一个概率分布在 $\mathbb{R}^D$ 上，对每个 $t \in [0,1]$ 。特别地，最优传输流 $p_t(\mathbf{x})$

[3] Benamou–Brenier 公式通过在测度和速度的连续路径上最小化动能来描述如何计算 $\mathcal{W}_2$ 距离。

遵循 *McCann* 的位移插值：

$$\mathbf{T}_t^*(\mathbf{x}) = (1-t)\mathbf{x} + t\mathbf{T}^*(\mathbf{x}),$$

其中 $\mathbf{T}^*(\mathbf{x})$ 是将 p_{src} 映射到 p_{tgt} 的 OT 映射。这种线性插值沿直线以恒定速度移动质量：对每个 $t \in [0,1]$ ，有 $p_t = \mathbf{T}_t^* \# p_{\text{src}}$ 。

最优传输映射 $\mathbf{T}^*$ 满足 *Monge–Ampère 方程*：

$$p_{\text{tgt}}\left(\nabla\psi(\mathbf{x})\right) \det\left(\nabla^2\psi(\mathbf{x})\right) = p_{\text{src}}(\mathbf{x}), \tag{7.2.4}$$

其中 $\mathbf{T}^*(\mathbf{x}) = \nabla\psi(\mathbf{x})$ 由某个凸函数 $\boldsymbol{\psi}$ 通过 Brenier 定理得出。然而，这种非线性偏微分方程通常难以显式求解。需要注意的是，这正是归一化流（normalizing flows）所使用的变量变换关系（参见 Equation (5.0.1)）：流通过参数化一个可逆的传输映射，并保证雅克比行列式易处理，但一般不强制梯度势结构 $\mathbf{T}^* = \nabla\psi$；因此，训练得到的流与 Brenier/最优传输映射可能存在显著差异。

7.2.2 熵正则化最优传输（EOT）

为了具体说明 EOT，考虑由样本构建的经验分布。假设 p_{src} 支持在点 $\{\mathbf{x}^{(i)}\}_{i=1}^n \subset \mathbb{R}^D$ 上，权重为 a_i ，p_{tgt} 支持在 $\{\mathbf{y}^{(j)}\}_{j=1}^n \subset \mathbb{R}^D$ 上，权重为 b_j 。一个耦合即是一个 $n \times n$ 的非负矩阵 $\gamma = (\gamma_{ij})$ ，其行和匹配 a ，列和匹配 b 。每个元素 γ_{ij} 表示从 $\mathbf{x}^{(i)}$ 运输到 $\mathbf{y}^{(j)}$ 的质量。[4] .

为什么要对 OT 进行正则化？ 在该离散情形下的经典 OT（通过在连续公式 Equation (7.2.1) 中取计数测度得到）简化为最小化

$$\min_{\gamma=(\gamma_{ij})} \sum_{i,j} C_{ij}\,\gamma_{ij},$$

所有可行耦合 $\gamma = (\gamma_{ij})$ 上，其中 $C_{ij} = c(\mathbf{x}^{(i)}, \mathbf{y}^{(j)})$ 表示将一单位质量从源点 $\mathbf{x}^{(i)}$ 移动到目标点 $\mathbf{y}^{(j)}$ 的成本，针对给定的基本成本 $c : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}_{\geq 0}$ （例如 $c(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_2^2$ ）。

两个主要问题随之产生：

[4] 经验（离散）测度为连续分布提供了合理的代理。当基代价为 $c(x,y) = d(x,y)^p$ 时（此时 OT 值等于 W_p^p ），且测度具有有限的 p 阶矩时，经验测度以定量速率收敛到总体分布于 W_p ；详见 **fournier2015rate** 及 **peyre2019computational** 中的综述。

1. **非唯一性与不稳定性**：极小点 γ^* 未必是唯一的。例如，如果两个运输方案达到相同的极小成本，求解器可能选择其中任意一个。因此，输入 (a, b, C) 的微小变化（如移动一个样本、调整权重或轻微扰动成本）可能导致解的突变。

2. **计算成本高**：该问题是一个含有 n^2 个变量和 $2n$ 个约束的线性规划问题。实际求解器（例如，匈牙利算法、网络单纯形 (**peyre2019computational**)）的复杂度通常为 $\mathcal{O}(n^3)$ ，对于大规模的 n 来说这是不可行的。

为克服这些瓶颈，EOT 目标函数在经典 OT 问题中引入了一个由参数 $\varepsilon > 0$ 控制的正则化项：

$$\mathrm{EOT}_\varepsilon(p_{\mathrm{src}}, p_{\mathrm{tgt}}) := \min_{\gamma \in \Gamma(p_{\mathrm{src}}, p_{\mathrm{tgt}})} \int c(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\gamma(\mathbf{x}, \mathbf{y}) + \varepsilon \mathcal{D}_{\mathrm{KL}}(\gamma \| M) . \tag{7.2.5}$$

参考测度 M 通常选择为边缘分布的乘积，$p_{\mathrm{src}} \otimes p_{\mathrm{tgt}}$ 。KL 散度项与传输计划 γ 的香农熵直接相关：

$$\mathcal{D}_{\mathrm{KL}}(\gamma \,\|\, p_{\mathrm{src}} \otimes p_{\mathrm{tgt}}) = -\mathcal{H}(\gamma) + \mathrm{Constant},$$

其中 $\mathcal{H}(\gamma) := -\int \gamma(\mathbf{x}, \mathbf{y}) \log \gamma(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{y}$ 。

加入此项正则化项带来了若干理论和实际优势，我们在此简要概述如下：

为什么熵正则化项有帮助？

1. **质量扩散**。由于 $t \mapsto t \log t$ 是凸函数且在 t 较大时增长迅速，最小化 $\int \gamma \log \gamma$ 会惩罚 尖锐的耦合（某些 $\gamma(\mathbf{x}, \mathbf{y})$ 很大，其余接近零）。对于固定的总质量 $\int \gamma = 1$ ，它倾向于使 $\gamma(\mathbf{x}, \mathbf{y})$ 在 $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^D \times \mathbb{R}^D$ 上分布得更加均匀的方案。等价地，最大化香农熵有助于提高 “不确定性”（弥散性）。

2. **严格凸性与唯一性**。由于 $\mathcal{H}$ 严格凹，Equation (7.2.5) 中的目标函数在 γ 上严格凸，从而产生一个 唯一的极小化子 γ_ε^* ，其关于 $(p_{\mathrm{src}}, p_{\mathrm{tgt}}, c)$ 连续依赖。

3. **Sinkhorn 形式与正性**。在较弱条件下[5]，优化器具有 *Schrödinger/Sinkhorn* 形式

$$\gamma_\varepsilon^*(\mathbf{x},\mathbf{y}) = u(\mathbf{x})\,\exp\!\big(-\tfrac{c(\mathbf{x},\mathbf{y})}{\varepsilon}\big)\,v(\mathbf{y})p_{\mathrm{src}}(\mathbf{x})p_{\mathrm{tgt}}(\mathbf{y}),$$

对于正的缩放函数 u, v（至多相差一个全局因子）。在实际应用中，连续形式通过有限样本进行近似，将 EOT 简化为有限（采样）的 Sinkhorn 迭代。熵项目标函数是严格凸的，且缩放（Sinkhorn/IPFP）算法能够高效求解 (**sinkhorn1964relationship**; **cuturi2013sinkhorn**)。对于具有每边缘 n 个支撑点的稠密问题（一个 $n \times n$ 核），每次 Sinkhorn 迭代的时间复杂度为 $\mathcal{O}(n^2)$，空间复杂度为 $\mathcal{O}(n^2)$，使得该方法更具可扩展性和实用性 (**altschuler2017near**)。

4. **在 ε 中的极限**。正如 $\varepsilon \to 0$ 所示，最优计划 γ_ε^* 逐渐变得集中，趋近于一个（可能奇异的）经典传输耦合（我们将在 Section 7.3.2 中重新讨论这一联系）。随着 ε 增大，γ_ε^* 逐渐分散，并趋近于独立耦合 $p_{\mathrm{src}} \otimes p_{\mathrm{tgt}}$ 。

7.2.3 薛定谔桥（SB）

SB 的 KL 表述。 薛定谔桥（SB）问题由埃尔温 · 薛定谔在 20 世纪 30 年代提出，其问题是：假设粒子按照某种简单的参考动力学运动，例如布朗运动。现在设想我们在两个时间点观察这些粒子：在 $t = 0$ 时，它们的分布为 p_{src} ，而在 $t = 1$ 时，分布为 p_{tgt} 。在所有能够连接这两个分布的随机过程中，哪一个与参考动力学的偏离最小？这里的“偏离”通过 KL 散度来衡量，因此 SB 问题的解即为将布朗运动变形为满足给定边界条件的最可能过程。

为了使其精确，令 $\mathbf{x}_{0:T} := \{\mathbf{x}_t\}_{t\in[0,T]}$ 表示该过程的完整轨迹。我们将 P 记为 *轨迹的律*，即整个样本路径上的概率分布。P 的时间-t 边缘分布记为 p_t（或 P_t），它描述了单个时间点上状态 $\mathbf{x}_t$ 的分布。形式上，对于一个可测集 $A \subseteq \mathbb{R}^D$ ，

$$p_t(A) = P(\mathbf{x}_t \in A).$$

换句话说，p_t 可以看作是从 P 采样多个完整轨迹后，在时间 t 收集的状态所得到的经验分布——例如，当状态为一维时，可以表示为直方图。

[5]我们假设 $c < \infty$ 在 $p_{\mathrm{src}} \otimes p_{\mathrm{tgt}}$ -几乎处处成立，且边缘核积分是有限且为正的。为了简化，我们专注于 γ_ε^* 、p_{src} 和 p_{tgt} 相对于勒贝格测度具有密度的情况。

考虑一个由 SDE 控制的参考扩散 $\{\mathbf{x}_t\}_{t\in[0,T]}$

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\ \mathrm{d}t + g(t)\ \mathrm{d}\mathbf{w}_t, \tag{7.2.6}$$

其中 $\mathbf{f}\colon \mathbb{R}^D \times [0,T] \to \mathbb{R}^D$ 、$g\colon [0,T] \to \mathbb{R}$ 和 $\{\mathbf{w}_t\}_{t\in[0,T]}$ 是标准布朗运动。令 R 表示完整轨迹 $\mathbf{x}_{0:T} := \{\mathbf{x}_t\}_{t\in[0,T]}$ 的路径律（联合分布）；该 R 将作为参考轨迹分布。

使用此记号，薛定谔桥（SB）问题寻求一个轨迹律 P，使其在 KL 散度上最接近 R，同时匹配给定的端点边缘分布：

$$\mathrm{SB}(p_{\mathrm{src}}, p_{\mathrm{tgt}}) := \min_{P}\ \mathcal{D}_{\mathrm{KL}}(P\|R) \quad \text{s.t.} \quad P_0 = p_{\mathrm{src}},\ \ P_T = p_{\mathrm{tgt}}. \tag{7.2.7}$$

优化器 P^* 依赖于所选择的参考过程 R 。

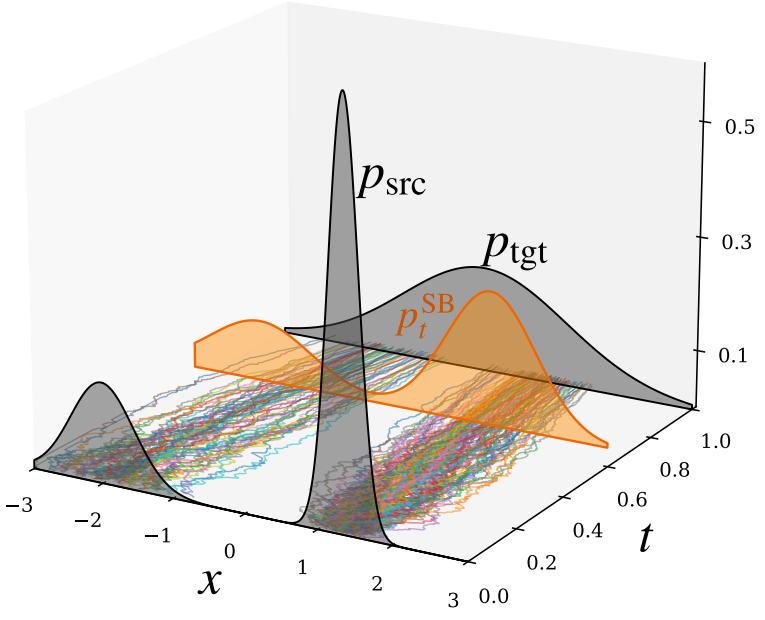


图 7.2: SB 的随机控制视角示意图。该桥梁寻找偏离参考路径最小的随机路径，以连接 p_{src} 和 p_{tgt} 。

随机控制视角下的 SB。 与其在 Equation (7.2.7) 中优化任意路径分布 P，不如采用一种更易处理的方法：将参考动力学作为锚点并允许其漂移。这通过引入时间依赖的漂移 $\mathbf{v}_t(\mathbf{x}_t)$ 实现，该漂移扰动参考过程并生成一族候选轨迹分布。最终的动力学呈现为 *受控扩散*：

$$\mathrm{d}\mathbf{x}_t = \big[\mathbf{f}(\mathbf{x}_t, t) + \mathbf{v}_t(\mathbf{x}_t)\big]\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t,$$

其中 $\mathbf{v}_t : \mathbb{R}^D \to \mathbb{R}^D$ 是稍后将要优化的漂移项 (Equation (7.2.8))。在标准可积性条件下（例如 Novikov 条件），并根据 Girsanov 定理（见 Section C.2.1），受控律 P 与参考律 R 之间的 KL 散度具有动态（动能）形式

$$\mathcal{D}_{\mathrm{KL}}(P\|R) = \mathbb{E}_P\left[\frac{1}{2}\int_0^T \frac{\|\mathbf{v}_t(\mathbf{x}_t)\|^2}{g^2(t)}\,\mathrm{d}t\right] = \frac{1}{2}\int_0^T \int_{\mathbb{R}^D} \frac{\|\mathbf{v}_t(\mathbf{x})\|^2}{g^2(t)} p_t(\mathbf{x})\,\mathrm{d}\mathbf{x}\,\mathrm{d}t,$$

其中 p_t 是受控过程下的 $\mathbf{x}_t$ 的时间–t 边缘分布。第二个等式由全期望公式得出。

因此，SB 问题可以重新表述为在所有可接受的漂移 $\mathbf{v}_t$ 上最小化将过程从 p_{src} 在 $t = 0$ 处驱动到 p_{tgt} 在 $t = T$ 处的期望控制能量 (**dai1991stochastic**; **pra1990markov**; **pavon1991free**; **chen2016relation**)。这导出了 随机控制公式：

$$\begin{aligned}&\mathrm{SB}_\varepsilon(p_{\mathrm{src}}, p_{\mathrm{tgt}})\\ &= \min_{\substack{\mathbf{v}_t \text{ s.t. } \mathrm{d}\mathbf{x}_t = [\mathbf{f}(\mathbf{x}_t,t)+\mathbf{v}_t(\mathbf{x}_t)]\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t,\\ \mathbf{x}_0\sim p_{\mathrm{src}},\ \mathbf{x}_T\sim p_{\mathrm{tgt}}}} \frac{1}{2}\int_0^T \int_{\mathbb{R}^D} \frac{\|\mathbf{v}_t(\mathbf{x})\|^2}{g^2(t)} p_t(\mathbf{x})\,\mathrm{d}\mathbf{x}\,\mathrm{d}t,\end{aligned} \tag{7.2.8}$$

重要的是，端点分布 p_{src} 和 p_{tgt} 是任意的；控制 $\mathbf{v}_t$ 被精确地选择以 “连接” 这两个边缘分布之间的参考动力学，同时尽可能接近（在 KL 散度意义下）参考过程 R。

一个特殊的布朗运动参考系。 Equation (7.2.8) 类似于 Equation (7.2.3) 中的 Benamou–Brenier 最优传输（OT）公式，特别是当参考过程 R^ε（带有 $\varepsilon > 0$）被选择为布朗运动时：

$$\mathrm{d}\mathbf{x}_t = \sqrt{\varepsilon}\,\mathrm{d}\mathbf{w}_t,$$

使得 $\mathbf{f} \equiv \mathbf{0}$ 和 $g(t) \equiv \sqrt{\varepsilon}$。

在此情景中，SB 问题寻求一条路径分布 P，使其在 KL 散度意义下尽可能接近布朗运动参考路径 R^ε，同时匹配端点边缘分布：

$$\mathrm{SB}_\varepsilon(p_{\mathrm{src}}, p_{\mathrm{tgt}}) := \min_P \mathcal{D}_{\mathrm{KL}}(P\|R^\varepsilon) \quad \text{s.t.} \quad P_0 = p_{\mathrm{src}}, P_T = p_{\mathrm{tgt}}. \tag{7.2.9}$$

于是，等价的随机控制表述变为

$$\mathrm{SB}_\varepsilon(p_{\mathrm{src}}, p_{\mathrm{tgt}}) = \min_{\substack{\mathbf{v}_t \text{ s.t. } \mathrm{d}\mathbf{x}_t = \sqrt{\varepsilon}\,\mathrm{d}\mathbf{w}_t, \\ \mathbf{x}_0 \sim p_{\mathrm{src}},\ \mathbf{x}_T \sim p_{\mathrm{tgt}}}} \frac{1}{2\varepsilon} \int_0^T \int_{\mathbb{R}^D} \|\mathbf{v}_t(\mathbf{x})\|^2 \, p_t(\mathbf{x}) \, \mathrm{d}\mathbf{x} \, \mathrm{d}t. \tag{7.2.10}$$

为何我们需要指定一个参考分布？ 与经典最优传输不同，由于随机性，SB 问题需要一个参考分布。在 OT 中，代价函数（例如 $c(\mathbf{x},\mathbf{y}) \propto \|\mathbf{x}-\mathbf{y}\|^2$ ）隐式地定义了一条唯一的确定性测地路径，因此无需参考分布。相比之下，SB 情景允许存在无限多条连接边缘分布的随机过程，且不存在“自然”路径的内在概念。参考测度 R 编码了系统的潜在物理或几何结构（如布朗运动），并定义了基于 KL 散度的最优化目标 $\mathcal{D}_{\mathrm{KL}}(P\|R)$ ，若无此参考，则最优性的概念将无法定义。

耦合偏微分方程表征。 描述 SB 解的一种便捷方式是通过两个时空势 $\Psi(x,t)$ 和 $\widehat{\Psi}(x,t)$。令 p_t^{SB} 表示在 Equation (7.2.7) 中，时间 $t \in [0,T]$ 处的最优轨迹律 P^* 的边际。则有对称因子分解 (**dai1991stochastic**)

$$p_t^{\mathrm{SB}}(x) = \Psi(x,t)\widehat{\Psi}(x,t), \tag{7.2.11}$$

其中 Ψ 和 $\widehat{\Psi}$ 求解 (线性的) 薛定谔系统 (**caluya2021wasserstein**; **chen2021stochastic**; **chenlikelihood**)：

$$\begin{aligned}
\frac{\partial \Psi}{\partial t}(\mathbf{x},t) &= -\nabla_{\mathbf{x}}\Psi(\mathbf{x},t)\cdot\mathbf{f}(\mathbf{x},t) - \frac{g^2(t)}{2}\Delta_{\mathbf{x}}\Psi(\mathbf{x},t), \\
\frac{\partial \widehat{\Psi}}{\partial t}(\mathbf{x},t) &= -\nabla_{\mathbf{x}}\cdot\big(\widehat{\Psi}(\mathbf{x},t)\,\mathbf{f}(\mathbf{x},t)\big) + \frac{g^2(t)}{2}\Delta_{\mathbf{x}}\widehat{\Psi}(\mathbf{x},t)
\end{aligned} \tag{7.2.12}$$

subject to

$$\Psi(\mathbf{x},0)\,\widehat{\Psi}(\mathbf{x},0) = p_{\mathrm{src}}(\mathbf{x}), \quad \Psi(\mathbf{x},T)\,\widehat{\Psi}(\mathbf{x},T) = p_{\mathrm{tgt}}(\mathbf{x}).$$

前向时间薛定谔桥随机微分方程。 一旦 Ψ 确定，最优动力学即为由时空因子 Ψ 倾斜的参考扩散过程：

$$\mathrm{d}\mathbf{x}_t = \left[\mathbf{f}(\mathbf{x}_t, t) + g^2(t)\,\nabla_{\mathbf{x}} \log \Psi(\mathbf{x}_t, t)\right] \mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_0 \sim p_{\mathrm{src}}. \tag{7.2.13}$$

令 Q 表示 Equation (7.2.13)(so $Q_0 = p_{\mathrm{src}}$ and $Q_T = p_{\mathrm{tgt}}$ by Equations (7.2.11) and (7.2.12)) 的轨迹定律。则 $Q = P^*$ 以及 Equation (7.2.8) 的极小化元 $\mathbf{v}^*$ 为（见 (**chen2021stochastic**) 的第 4.6 节）：

$$\mathbf{v}_t^*(\mathbf{x}) = g^2(t)\nabla_{\mathbf{x}} \log \Psi(\mathbf{x}, t).$$

也就是说，漂移校正 $g^2\nabla_{\mathbf{x}} \log \Psi$ 恰好是使参考路径的终点边缘分布相匹配所需的最小 KL 扰动。

反向时间薛定谔桥随机微分方程。 相同的最优路径律也可以逆向生成。一种方便的方法是概念性地使用扩散过程的标准时间反演恒等式：

$$\mathbf{b}^-(\mathbf{x}, t) = \mathbf{b}^+(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t^{\mathrm{SB}}(\mathbf{x}),$$

其中 $\mathbf{b}^+ = \mathbf{f} + g^2\nabla \log \Psi$ 和 $p_t = \Psi\,\widehat{\Psi}$。这给出了

$$\mathbf{b}^-(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log \widehat{\Psi}(\mathbf{x}, t).$$

因此，反向时间 SDE 可表示为

$$\mathrm{d}\mathbf{x}_t = \left[\mathbf{f}(\mathbf{x}_t, t) - g^2(t)\nabla_{\mathbf{x}} \log \widehat{\Psi}(\mathbf{x}_t, t)\right] \mathrm{d}t + g(t)\ \mathrm{d}\bar{\mathbf{w}}_t, \quad \mathbf{x}_T \sim p_{\mathrm{tgt}}. \tag{7.2.14}$$

等价地，通过由 $\mathbf{y}_\tau := \mathbf{x}_{T-\tau}$ 重新参数化时间，使得 τ 从 0 增加到 T。然后，$\mathbf{y}_\tau$ 沿 τ 从 $\mathbf{y}_0 \sim p_{\mathrm{tgt}}$ 开始向前演化。

$$\begin{aligned}\mathrm{d}\mathbf{y}_\tau = \big[-\mathbf{f}(\mathbf{y}_\tau, T-\tau) + g^2(T-\tau)\nabla_{\mathbf{y}} \log \widehat{\Psi}(\mathbf{y}_\tau, T-\tau)\big]\,\mathrm{d}\tau \\ + g(T-\tau)\,\mathrm{d}\mathbf{w}_\tau.\end{aligned} \tag{7.2.15}$$

在 Equation (7.2.8)(same quadratic energy with the reversed clock): 的反向时间随机控制公式中，

$$\min_{\substack{\mathbf{u}_\tau \text{ s.t. } \mathrm{d}\mathbf{y}_\tau=[-\mathbf{f}(\mathbf{y}_\tau,T-\tau)+\mathbf{u}_\tau(\mathbf{y}_\tau)]\,\mathrm{d}\tau+g(T-\tau)\,\mathrm{d}\mathbf{w}_\tau,\\ \mathbf{y}_0\sim p_{\mathrm{tgt}},\ \mathbf{y}_T\sim p_{\mathrm{src}}}} \frac{1}{2}\int_0^T\int_{\mathbb{R}^D}\frac{\|\mathbf{u}_\tau(\mathbf{y})\|^2}{g^2(T-\tau)}\,p_{T-\tau}(\mathbf{y})\,\mathrm{d}\mathbf{y}\,\mathrm{d}\tau. \tag{7.2.16}$$

最优控制是

$$\mathbf{u}_t^*(\mathbf{x}) = -g^2(t)\nabla_{\mathbf{x}}\log\widehat{\Psi}(\mathbf{x},t).$$

正向和反向的描述均得出相同的最优路径律 P^* ，它们通过

$$\nabla\log p_t^{\mathrm{SB}} = \nabla\log\Psi + \nabla\log\widehat{\Psi}, \qquad \mathbf{b}^- = \mathbf{b}^+ - g^2\,\nabla\log p_t^{\mathrm{SB}},$$

因此，它们的边缘分布处处与 p_t^{SB} 一致。额外的漂移项 $g^2\nabla\log\Psi$（前向）和 $-g^2\nabla\log\widehat{\Psi}$（反向时间）作为控制力，引导参考扩散过程匹配终点边缘分布，同时在相对熵意义下保持最接近参考过程。

耦合偏微分方程方法的实际障碍。 基于 Equation (7.2.14) 构建生成过程，必须求解 Equation (7.2.12) 中的耦合偏微分方程，以获得后向薛定谔势 $\widehat{\Psi}$ 。然而，这些偏微分方程即使在低维情形下也以难以求解著称，这使得其在生成式建模中的直接应用面临挑战。为克服这一问题，已有若干研究提出了替代策略：利用得分随机微分方程技术迭代求解每个半桥问题（$p_{\mathrm{tgt}}\leftarrow p_{\mathrm{src}}$ 和 $p_{\mathrm{tgt}}\rightarrow p_{\mathrm{src}}$）(**de2021diffusion**)；优化代理似然界 (**chenlikelihood**; **liu20232**)；或设计无需仿真的训练方法，基于样本对 $(\mathbf{x}_0,\mathbf{x}_T)\sim p_{\mathrm{src}}\otimes p_{\mathrm{tgt}}$ 的后验解析解 $\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T$ (**liu20232**)。此处不深入探讨具体技术细节，仅简要讨论扩散模型与 SB 之间的联系 Section 7.4。

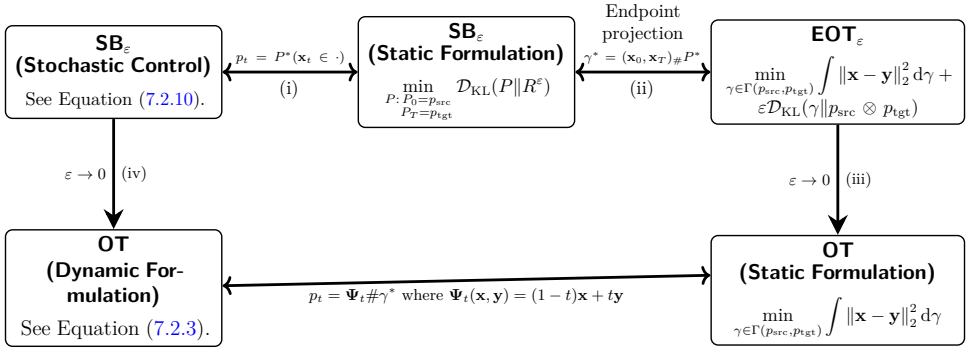
7.2.4 全局推动与局部动力学：生成模型的最优传输类比

从最优传输视角（见 Equation (7.2.1)），可以利用深度生成模型来学习一个从简单先验到数据的传输（前推）映射，即 $\mathbf{G}_{\boldsymbol{\phi}\#}p_{\mathrm{prior}}\approx p_{\mathrm{data}}$。尽管 $\mathbf{G}_{\boldsymbol{\phi}}$ 通常并不等同于最优传输映射（除非在文献 (**genevay2018learning**; **onken2021ot**) 中在适当条件下引入了 OT 目标），但 Benamou–Brenier 公式（见 Equation (7.2.3)）提供了一个互补的动态视角。与其直接学习单一全局映射，不如将其描述为由时变局部向量场生成的连续流，沿着从 p_{prior} 到 p_{data} 的平滑路径进行传输。这一

动态公式与静态 Schrödinger 桥问题（见 Equation (7.2.7)）及其随机控制对应形式（见 Equation (7.2.8)）之间的关系相类似，其中最优耦合被实现为一个受控扩散过程。在生成式建模中也出现了类似的类比：标准 DGMs（如 GAN 或 VAE）学习一个全局前推映射，而扩散模型则学习一个驱动生成动力学的时变局部向量场。

7.3 变体最优传输公式的关联

图 7.3: 最优传输不同形式之间的关系以及 SB 中的 $c(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_2^2$ 与参考文献 R^ε 。我们总结了如下等价关系：(i) SB_ε (随机控制) $\Leftrightarrow$ SB_ε (静态形式)，其中 p_t 为路径测度 P 的时间–t 边缘分布；(ii) SB_ε (静态形式) $\Leftrightarrow$ EOT_ε (见 Section 7.3.1)；(iii) EOT_ε (静态形式) $\Leftrightarrow$ OT_ε (静态) (见 Section 7.3.2)；(iv) SB_ε (随机控制) $\Leftrightarrow$ OT (动态) (见 Section 7.3.3)。



在深入技术细节之前，厘清最优传输及其熵正则化不同表述之间的联系是有帮助的。从高层次上看，这些问题可以被视为相互关联的（参见 Figure 7.3 中的连接示意图）：

(i) SB 问题 SB_ε ，其特定参考系 R^ε 由布朗运动给出

$$\mathrm{d}\mathbf{x}_t = \sqrt{\varepsilon}\,\mathrm{d}\mathbf{w}_t$$

等价于其静态形式：演化边缘分布 p_t 恰好是最优路径测度 P 在时间–t 切片上的取值（见 Section 7.2.3）；

(ii) SB_ε 的静态形式直接关联到熵性最优传输问题，EOT_ε（参见 Section 7.3.1）；

(iii) EOT_ε ，反过来可以与熵形式的静态最优传输问题 OT_ε 相关联（见 Section 7.3.2）；

(iv) SB_ε 的随机控制视角也可以与经典最优传输的动态公式联系起来（见 Section 7.3.3）。

这些非平凡的关系共同提供了对随机控制、熵正则化和经典最优传输框架的紧凑视角。

7.3.1 SB 和 EOT 是（双重）等效的

在本节中，我们提出了两种互补的视角，表明 SB 本质上等价于 EOT。与产生单一确定性映射的经典最优传输不同，SB 产生了一种 随机的粒子流：质量以概率方式被传输，边缘分布遵循类似扩散的动力学演化。

从静态视角来看，SB 与 EOT 一致，其目标是找到两个端点分布之间的耦合，以平衡运输成本与熵。从动态视角来看，SB 描述了一个受控的扩散过程，该过程尽可能接近一个简单的参考过程（如布朗运动），同时仍能匹配预期的端点。这两种视角各自独立地确立了等价性，提供了两种一致的方法来理解 SB/EOT 作为分布到分布变换的正则形式。

静态薛定谔桥。 让

$$\widetilde{R}^{\varepsilon}(\mathbf{x},\mathbf{y}) := \frac{1}{Z_{\varepsilon}}\, e^{-c(\mathbf{x},\mathbf{y})/\varepsilon}\, p_{\mathrm{src}}(\mathbf{x})\, p_{\mathrm{tgt}}(\mathbf{y}),$$

带有归一化常数：

$$Z_{\varepsilon} := \iint e^{-c(\mathbf{x},\mathbf{y})/\varepsilon} p_{\mathrm{src}}(\mathbf{x}) p_{\mathrm{tgt}}(\mathbf{y}) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y}.$$

然后是熵正则化的最优传输目标

$$\begin{aligned}\min_{\gamma\in\Gamma(p_{\mathrm{src}},p_{\mathrm{tgt}})} \Big\{ \int c\mathrm{d}\gamma + \varepsilon \mathcal{D}_{\mathrm{KL}}\big(\gamma \| p_{\mathrm{src}} \otimes p_{\mathrm{tgt}}\big) \Big\} = \varepsilon \min_{\gamma\in\Gamma(p_{\mathrm{src}},p_{\mathrm{tgt}})} \mathcal{D}_{\mathrm{KL}}\big(\gamma \| \widetilde{R}^{\varepsilon}\big) \\ - \varepsilon \log Z_{\varepsilon},\end{aligned} \tag{7.3.1}$$

因此，它等价于（相差一个常数）静态薛定谔桥（见 Equation (7.2.9)）：

$$\min_{\gamma\in\Gamma} \mathcal{D}_{\mathrm{KL}}(\gamma \| \widetilde{R}^{\varepsilon}).$$

动态对等性（布朗运动参照）。 我们也可以从动态等价性来理解这一点，经典结果 (**mikami2006duality**) 表明，带有二次代价的熵正则化最优传输

$$c(\mathbf{x},\mathbf{y}) = \frac{\|\mathbf{y}-\mathbf{x}\|^2}{2T}$$

与 SB 问题仿射等价，其中参考路径律 R^{ε} 是在 $[0,T]$ 上的布朗运动。

$$\mathrm{d}\mathbf{x}_t = \sqrt{\varepsilon}\, \mathrm{d}\mathbf{w}_t.$$

此处，“仿射等价”表示最优值之间仅相差一个正缩放因子和一个与决策变量无关的常数项，因此最优解保持一致。特别地，设 P^* 为 SB 的最优路径分布，γ^* 为 EOT 的最优传输方案。若 $\mathbf{x}_{[0:T]} \sim P^*$ ，则端点对 $(\mathbf{x}_0, \mathbf{x}_T)$ 服从分布 γ^* ：

$$P^* \text{ solves SB} \iff \gamma^* \text{ solves EOT and } (\mathbf{x}_0, \mathbf{x}_T) \sim \gamma^*.$$

简言之，动态（SB）问题的最优过程诱导出静态（EOT）问题的最优耦合。反之，在热核满足较弱条件的情况下，任何最优静态耦合都可以被看作是某个最优 SB 过程端点的实现。

推导这一结论的关键思想是，路径上的 KL 散度可以依据端点进行分解，这意味着 Schrödinger 桥问题可简化为仅关于 $(\mathbf{x}_0, \mathbf{x}_T)$ 的联合分布的 KL 散度。对于布朗运动而言，从 $\mathbf{x}$ 到 $\mathbf{y}$ 的转移密度具有高斯形式，因此其负对数为二次型：

$$-\varepsilon \log p_T(\mathbf{y} \mid \mathbf{x}) = \frac{\|\mathbf{y} - \mathbf{x}\|^2}{2T} + \text{const}.$$

这表明，端点 KL 与带有二次代价的熵正则化最优传输目标完全相同，仅相差一个无关紧要的常数。

基于通用参考的 SB 确定了 EOT 成本。 正如我们在 Equation (7.2.7) 中讨论的，SB 问题并不仅限于布朗运动；它可以定义在任何（适定的）参考过程上。这种选择唯一地决定了相应 EOT 问题中的代价函数。关键联系在于，SB 的 *参考动力学*引出了 EOT 的 *代价函数*。

令参考过程由定义在 $[0, T]$ 上的 SDE 控制，产生转移密度 $p_T(\mathbf{y}|\mathbf{x})$ ，即从时间 0 的 $\mathbf{x}$ 到达时间 T 的 $\mathbf{y}$ 的似然。那么，EOT 代价函数（至多一个缩放常数）为

$$c(\mathbf{x}, \mathbf{y}) \propto -\log p_T(\mathbf{y}|\mathbf{x}).$$

在此代价下，求解 SB 问题等价于求解 EOT 问题。简而言之，在 SB 中选择参考动力学在数学上等价于在 EOT 中指定传输代价。根据 Equation (7.3.1)，熵正则化 OT 目标与静态 SB 目标不同；因此这两个问题等价且具有相同的极小化器。

7.3.2 EOT_ε 被简化为 OT，其中 $\varepsilon \to 0$

令 γ_ε^* 表示 EOT_ε 的最优计划，令 γ^* 为 Equation (7.2.1) 中无正则化 OT 问题的最优计划。以下结果 (**mikami2008optimal**; **peyre2019computational**)

表明，当 $\varepsilon \to 0$ 时，熵正则化最优计划 γ_ε^* （以适当的方式）收敛到 OT 计划 γ^*，且 EOT 代价收敛到 OT 代价。

这个收敛结果既具有基础性又具有实际重要性。其中一个原因是，熵正则化最优传输（OT）问题 EOT_ε 可以通过诸如 Sinkhorn 等算法得到高效的数值解。因此，该结果为使用 EOT_ε（其中 ε 较小）作为经典 OT 问题在 Equation (7.2.1) 中的计算上易处理的替代方案提供了理论依据，即使代价函数 $c(\mathbf{x}, \mathbf{y})$ 比二次情形更一般。

Theorem 7.3.1: (非正式) EOT_ε 收敛到 OT。

当 $\varepsilon \to 0$ 时，最优值收敛：

$$\lim_{\varepsilon \to 0} \mathrm{EOT}_\varepsilon(p_{\mathrm{src}}, p_{\mathrm{tgt}}) = \mathrm{OT}(p_{\mathrm{src}}, p_{\mathrm{tgt}}).$$

此外，最优方案 γ_ε^* 弱收敛到 γ^* 。即，对所有有界连续（测试）函数 $g : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ ，

$$\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma_\varepsilon^*}[g(\mathbf{x}, \mathbf{y})] \to \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma^*}[g(\mathbf{x}, \mathbf{y})],$$

Proof for Theorem.

对于严格证明，我们参考文献 (**mikami2008optimal**; **peyre2019computational**)。下面我们将给出最优值收敛的一个启发式推导。为记号简便起见，记相应的最优值为

$$V_\varepsilon := \mathrm{EOT}_\varepsilon(p_{\mathrm{src}}, p_{\mathrm{tgt}}), \quad V_0 := \mathrm{OT}(p_{\mathrm{src}}, p_{\mathrm{tgt}})$$

上界。 由 γ_ε^* 的最优性，其值 V_ε 被使用方案 γ^* 的代价所限制：

$$V_\varepsilon \le \int c \, \mathrm{d}\gamma^* + \varepsilon \mathcal{D}_{\mathrm{KL}}(\gamma^* \| p_{\mathrm{src}} \otimes p_{\mathrm{tgt}}).$$

假设 KL 项是一个有限常数 K，则得到 $V_\varepsilon \le V_0 + \varepsilon K$。取上极限得 $\limsup_{\varepsilon \to 0} V_\varepsilon \le V_0$。**下界。** 由于 KL 散度非负，$V_\varepsilon \ge \int c \, \mathrm{d}\gamma_\varepsilon^*$。根据 V_0 作为最小传输代价的定义，任何方案的代价至少为 V_0，因此 $\int c \, \mathrm{d}\gamma_\varepsilon^* \ge V_0$。这意味着对所有 $\varepsilon > 0$ 都有 $V_\varepsilon \ge V_0$，从而 $\liminf_{\varepsilon \to 0} V_\varepsilon \ge V_0$。结合上界与下界可知最优值的收敛性 $\lim_{\varepsilon \to 0} V_\varepsilon = V_0$。至于最优方案本身的弱收敛性 $\gamma_\varepsilon^* \to \gamma^*$，这是 Γ-收敛理论中一个更深入的结果，此处省略。

7.3.3 SB_ε 被简化为 OT，即 $\varepsilon \to 0$

对于每个 $\varepsilon > 0$，令 $\mathbf{v}_t^\varepsilon$ 为如 Equation (7.2.10) 所述的 SB 问题的最小化元，令 p_t^ε 为由 $\mathbf{v}_t^\varepsilon$ 诱导的受控 SDE $\mathbf{x}_t$ 的边缘分布。则 p_t^ε 满足相应的福克-普朗克方程。相比之下，记 $(p_t^0, \mathbf{v}_t^0)$ 为最优传输的本莫-布伦纳公式的一个最小化元（见 Equation (7.2.3)）。

以下定理[6] 当 $\varepsilon \to 0$ 时，SB 问题收敛到 OT 问题。该结果在实践上具有重要意义，原因与定理 7.3.1 中所述类似。目标函数 SB_ε 可以通过 Sinkhorn 型算法高效求解，从而得到最优传输的一个数值上易处理且可微的代理。这在高维或大规模情景下尤其有价值，因为在这些情况下，直接求解器（例如基于 Benamou–Brenier 公式的方法）会变得计算成本高昂。

Theorem 7.3.2: (非正式) SB_ε 收敛到 OT。

当 $\varepsilon \to 0$ 时，我们有：

$$\lim_{\varepsilon \to 0} \mathrm{SB}_\varepsilon(p_{\mathrm{src}}, p_{\mathrm{tgt}}) = \mathrm{OT}(p_{\mathrm{src}}, p_{\mathrm{tgt}}),$$

其中 OT 是如 Equation (7.2.3) 中的 Benamou–Brenier 形式。此外，p_t^ε 在相应的函数空间中弱收敛于 p_t^0，且 $\mathbf{v}_t^\varepsilon$ 弱收敛于 $\mathbf{v}_t^0$。

Proof for Theorem.

该收敛结果的完整严格证明超出了我们的范围；我们建议读者参考 **leonard2012schrodin** **leonard2014survey** 以获得详细推导。尽管如此，我们可以从启发式角度理解这一收敛为何成立。在 SB 问题的随机控制表述 Equation (7.2.10) 中，受控 SDE 为：

$$\mathrm{d}\mathbf{x}_t = \mathbf{v}_t^\varepsilon(\mathbf{x}_t)\mathrm{d}t + \sqrt{2\varepsilon}\mathrm{d}\mathbf{w}_t.$$

当 $\varepsilon \to 0$ 时，噪声项消失，SDE 形式上趋近于一个确定性 ODE：

$$\mathrm{d}\mathbf{x}_t = \mathbf{v}_t^0(\mathbf{x}_t)\mathrm{d}t.$$

[6] 我们注意到，定理中最优值的收敛性是Γ -收敛意义下的，而非经典的逐点极限。尽管这需要更多的技术背景，但此处省略细节，仅陈述概念性结果。

这表明 SB 问题的最优值收敛到最优传输问题的最优值：

$$\lim_{\varepsilon \to 0} \mathrm{SB}_\varepsilon(p_{\mathrm{src}}, p_{\mathrm{tgt}}) = \mathrm{OT}(p_{\mathrm{src}}, p_{\mathrm{tgt}}).$$

同时，边缘密度 p_t^ε 满足 Fokker–Planck 方程：

$$\partial_t p_t^\varepsilon + \nabla \cdot (p_t^\varepsilon \mathbf{v}_t^\varepsilon) = \varepsilon \Delta p_t^\varepsilon.$$

同样地，当 $\varepsilon \to 0$ 时，扩散项消失，方程形式上退化为连续性方程：

$$\partial_t p_t^0 + \nabla \cdot (p_t^0 \mathbf{v}_t^0) = 0.$$

到目前为止，我们已经展示了在各自假设下 EOT 与 SB 之间的基本等价性，以及它们通过极限过程与 OT 的重要联系，如 Figure 7.3所示。接下来，我们将探讨扩散模型如何与这些概念相联系。

7.4 扩散模型的 SDE 是否为 SB 问题的最优解？

7.4.1 扩散模型作为薛定谔桥的一种特殊情形

SB 框架通过在任意源分布与目标分布之间实现非线性插值，扩展了（基于得分的）扩散模型。其通过添加源自标量势函数 $\Psi(\mathbf{x},t)$ 与 $\widehat{\Psi}(\mathbf{x},t)$ 的控制漂移项来实现这一目标，这些项引导参考扩散过程以匹配预定的端点边缘分布（见 Equation (7.2.12)），并遵循如下分解：

$$\nabla \log \Psi(x,t) + \nabla \log \hat{\Psi}(x,t) = \nabla \log p_t^{\mathrm{SB}}(\mathbf{x}).$$

这种泛化使模型能够超越标准的高斯先验，从更广泛的分布中生成样本。

与扩散模型的连接。 扩散模型是 SB 框架的一个特例。假设势能为常数，$\Psi(\mathbf{x},t) \equiv 1$。在此假设下，Equation (7.2.12) 中的第二个偏微分方程退化为标准的福克-普朗克方程，其解即为参考过程的边缘密度：

$$\widehat{\Psi}(\mathbf{x},t) = p_t^{\mathrm{SB}}(\mathbf{x}). \tag{7.4.1}$$

因此，相应的 SB 前向 SDE 便成为不受控的参考过程：

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t,t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t,$$

且该后向 SDE 简化为：

$$\mathrm{d}\mathbf{x}_t = \left[\mathbf{f}(\mathbf{x}_t,t) - g^2(t)\nabla \log p_t^{\mathrm{SB}}(\mathbf{x}_t)\right]\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}_t,$$

这与 Anderson 在扩散模型中使用的反向时间 SDE 相匹配。这种对应关系表明，扩散模型可以被解释为 SB 的零控制极限，其中势能不会引入额外的漂移。

边界条件与普适性。 上述约化仅在边界约束相容时才是纯粹形式上的。对于任意的源/目标 $(p_{\mathrm{src}}, p_{\mathrm{tgt}})$，通常情况下，选择 $\Psi \equiv 1$ 无法满足 Equation (7.2.12) 中的偏微分方程边界条件。全量 SB 通过学习非平凡势能来解决此问题，这些势能诱导非线性控制漂移，使参考动力学弯曲以匹配任何指定的端点。相比之下，扩散模型将一个端点固定为简单的先验（通常是高斯分布），仅学习反向时间得分以达到数据。从这一视角看，SB 是更灵活的统一框架：使用非平凡势能时，它

可连接任意端点；当 $\Psi \equiv 1$ 时，其退化为上述扩散模型情形。此外，我们还注意到，在标准线性扩散模型中，$p_T \approx p_{\text{prior}}$ 仅在 $T \to \infty$ 时成立，因此与先验的匹配仅为近似。

7.4.2 扩散模型作为薛定谔半桥

在本节中，我们解释为何扩散模型并非完整的薛定谔桥，而可以通过更宽松的概念——薛定谔半桥来理解。半桥仅施加一个端点约束（p_{prior} 或 p_{data}），而非同时施加两个，因此是完整桥接的一侧变体。在正式建立这一联系之前，我们首先引入薛定谔半桥的定义，基于 Equation (7.2.7) 中的一般形式，并允许任意的 p_{src} 和 p_{tgt} 。随后，我们将回到扩散模型，展示当端点由 p_{prior} 和 p_{data} 给定时，半桥视角如何自然适用。

薛定谔半桥 SB 问题要求找到一个随机过程，其分布与一个简单的参考过程在 KL 散度意义下最接近，同时满足两个端点分布 p_{src} 和 p_{tgt} 的匹配。求解完整的桥接过程需要同时满足两个边界条件，这通常计算上较为困难。一种有用的简化是 半桥问题：不需同时匹配两个端点，只需匹配其中一个即可。

形式上，设 R 为参考路径分布。前向半桥寻求一个路径分布 P，使其最小化

$$\min_{P:P_0=p_{\text{src}}} \mathcal{D}_{\text{KL}}(P \,\|\, R),$$

受单个约束 $P_0 = p_{\text{src}}$ 限制。类似地，反向半桥仅约束终端分布，

$$\min_{P:P_T=p_{\text{tgt}}} \mathcal{D}_{\text{KL}}(P \,\|\, R).$$

简而言之，前向半桥的问题是：在所有从期望的初始分布出发的过程之中，哪一个最接近参考过程？后向半桥则针对以期望的终止分布结束的过程提出同样的问题。通过迭代地结合这两个松弛过程，可以近似得到完整的最优传输路径。

扩散模型未能精确匹配终点。 扩散模型与 SB 框架的一个关键区别在于对终态分布 p_T 的处理。在标准扩散模型中，前向 SDE 通常关于 $\mathbf{x}_t$ 是线性的（见 Equation (4.3.2)），并且设计为仅当 $T \to \infty$ 时，p_T 才近似于先验。

$$p_T \approx p_{\text{prior}}.$$

然而，在有限时间内，p_T 是一个参数依赖于 p_{data} 的高斯分布（参见 Section C.1.5）。因此，通常需要仔细调整才能使其与期望的先验相匹配。

相比之下，SB 框架通过引入形式为 $g^2(t)\nabla_{\mathbf{x}} \log \Psi(\mathbf{x}, t)$ 的额外控制漂移，在有限时间 T 处强制实现确切的边缘匹配。这确保了终端分布精确满足 $p_T = p_{\text{prior}}$，而与初始数据分布 $p_0 = p_{\text{data}}$ 无关。总结如下：

- **扩散模型**：$p_T \approx p_{\text{prior}}$，渐近地当 $T \to \infty$ 时，
- **Schrödinger Bridge**：$p_T = p_{\text{prior}}$ 在有限 T 时确切地实现，这是通过求解控制势 Ψ 和 $\widehat{\Psi}$ 获得的。

扩散薛定谔桥 标准扩散模型不强制执行 $P_T = p_{\text{prior}}$，因此仅解决从 p_{data} 到 p_{prior} 的薛定谔 半桥问题。

为解决这一问题，扩散薛定谔桥（DSB）（**de2021diffusion**）采用迭代比例拟合（IPF）算法的思想，通过交替投影方法来匹配两端的边缘分布。这将扩散模型扩展至解决完整的薛定谔桥（SB）问题，具体如下[7]：

- **步骤 0**：**参考过程**。以 $P^{(0)} := R_{\text{fwd}}$ 为初始值，即参考前向 SDE：

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_0 \sim p_{\text{data}}.$$

  这确保了 $P_0^{(0)} = p_{\text{data}}$，但通常为 $P_T^{(0)} \neq p_{\text{prior}}$。

- **步骤 1**：**反向传播**。计算过程 $P^{(1)}$，使其在时间 T 与 p_{prior} 匹配，同时尽可能接近 $P^{(0)}$：

$$P^{(1)} = \underset{P: P_T = p_{\text{prior}}}{\arg\min}\ \mathcal{D}_{\text{KL}}(P \| P^{(0)}).$$

  这通过使用神经网络 $\mathbf{s}_{\boldsymbol{\phi}^\times}$ 近似原始评分函数实现，从而得到反向时间 SDE：

$$\mathrm{d}\mathbf{x}_t = \left[\mathbf{f}(\mathbf{x}_t, t) - g^2(t)\mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t, t)\right]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}_t,$$

  从 $\mathbf{x}_T \sim p_{\text{prior}}$ 开始模拟向后。

- **迭代.** 该过程 $P^{(1)}$ 满足 $P_T^{(1)} = p_{\text{prior}}$，但其初始边缘分布 $P_0^{(1)}$ 通常偏离 p_{data}。IPF 通过学习一个前向 SDE 来将 $P_0^{(1)}$ 调整回 p_{data}，随后进行另一次反向传播以强制满足 p_{prior}。这种交替过程持续进行，逐步优化该过程

[7] 尽管此描述使用了 p_{data} 和 p_{prior}，但 DSB 框架适用于任意一对端点分布。

直至收敛到最优桥接过程 P^*，该过程同时满足 $P_0^* = p_{\text{data}}$ 和 $P_T^* = p_{\text{prior}}$。**de2021diffusion** 在较弱条件下证明了收敛性。

7.5 扩散模型的 ODE 是否为最优传输问题的最优映射？

在本节中，我们专注于二次代价最优传输问题。

7.5.1 PF-ODE 流通常不是最优传输

本节展示了 **lavenant2022flow** 的结果，该结果表明，在二次代价下，PF-ODE 的解映射通常不能得到最优传输映射。

设置 我们考虑一个 VP SDE，具体来说是 Ornstein–Uhlenbeck 过程，它将光滑的初始密度 p_0 渐近演化为标准高斯分布 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ ：

$$\mathrm{d}\mathbf{x}(t) = -\mathbf{x}(t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{w}(t), \quad \mathbf{x}(0) \sim p_0.$$

相关的 PF-ODE 表示为

$$\frac{\mathrm{d}\mathbf{S}_t(\mathbf{x})}{\mathrm{d}t} = -\mathbf{S}_t(\mathbf{x}) - \nabla \log p_t(\mathbf{S}_t(\mathbf{x})), \quad \mathbf{S}_0(\mathbf{x}) = \mathbf{x}.$$

此处，$\mathbf{S}_t$ 表示将 p_0 推前至边缘分布 p_t 的流映射：

$$(\mathbf{S}_t)\,\#p_0 = p_t, \quad \text{that is,} \quad p_t(\mathbf{y}) = \int_{\mathbb{R}^D} \delta(\mathbf{y} - \mathbf{S}_t(\mathbf{x}))p_0(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

这些密度 p_t 通过福克-普朗克方程演化：

$$\frac{\partial p_t}{\partial t} = \nabla \cdot (\mathbf{x}p_t) + \Delta p_t.$$

这相当于一个速度场的连续性方程：

$$\mathbf{v}_t(\mathbf{x}) = -\mathbf{x} - \nabla \log p_t(\mathbf{x}),$$

其流动由 $\mathbf{S}_t(\mathbf{x})$ 给出。换句话说，PF-ODE 可以表示为：

$$\frac{\mathrm{d}\mathbf{S}_t(\mathbf{x})}{\mathrm{d}t} = \mathbf{v}_t\left(\mathbf{S}_t(\mathbf{x})\right).$$

如 $t \to \infty$ 所示，该映射将初始分布传输至先验分布：

$$\mathbf{S}_\infty \# p_0 = \mathcal{N}(\mathbf{0}, \mathbf{I}) =: p_{\text{prior}}.$$

lavenant2022flow 的参数的目标。 lavenant2022flow 并不直接评估从 p_0 到高斯分布的终端映射 $\mathbf{S}_\infty$ 是否最优。相反，他们构造了一个特定的初始分布 p_0，并考察了整个 PF-ODE 轨迹。他们的关键观察是，最优性可能在流的某一点处失效。

他们考虑中间边缘 $p_t = \mathbf{S}_t \# p_0$，并将从 p_{t_0} 到高斯分布的残差传输映射定义为

$$\mathbf{T}_{t\to\infty} := \mathbf{S}_\infty \circ \mathbf{S}_t^{-1}, \quad \text{for all } t \geq 0.$$

他们的论点核心表明，对于一个精心选择的 p_0，存在一个时间 $t_0 \geq 0$，使得 $\mathbf{T}_{t_0\to\infty}$ 并非从新的起始分布 p_{t_0} 到 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ 的二次代价最优传输映射。

该结果表明，PF-ODE 流通常不能产生最优传输映射，且对于某些初始分布，最优性性质可能失效。

一些工具。 lavenant2022flow 的论证关键依赖于以下结果，称为 *Brenier* 定理：

Theorem 7.1 (Informal Brenier's Theorem)**.** 设 ν_1, ν_2 为定义在 $\mathbb{R}^D$ 上的两个具有光滑密度的概率分布。若光滑映射 $\mathbf{T} : \mathbb{R}^D \to \mathbb{R}^D$ 是从 ν_1 到 ν_2 的最优传输（在二次代价下），当且仅当存在某个凸函数 u，使得 $\mathbf{T} = \nabla u$。此时，$\mathrm{D}\mathbf{T}$ 是对称且半正定的，且 u 满足 Monge–Ampère 方程：

$$\det \mathrm{D}^2 u(\mathbf{x}) = \frac{\nu_1(\mathbf{x})}{\nu_2(\nabla u(\mathbf{x}))}.$$

该证明还隐式地使用了以下事实，我们不再每次重复：一个映射是两个分布之间的最优传输，当且仅当其逆映射在反方向上也是最优传输。

证明概要：PF-ODE 一般情况下并非 OT 映射。 lavenant2022flow 采用反证法：他们假设对于每个 $t \geq 0$，映射

$$\mathbf{T}_t = \mathbf{S}_t \circ \mathbf{S}_\infty^{-1}$$

是从 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ 到 p_t 的二次代价最优传输映射。

步骤 1：Brenier 定理。 根据 Brenier 定理，从高斯分布出发的任意最优传输映射的雅克比矩阵必须是对称且半正定的。因此，

$$\mathrm{D}\mathbf{T}_t(\mathbf{x}) = \mathrm{D}\mathbf{S}_t(\mathbf{S}_\infty^{-1}(\mathbf{x}))\mathrm{D}(\mathbf{S}_\infty^{-1})(\mathbf{x})$$

必须对所有 t 和 $\mathbf{x}$ 对称。此处，$\mathrm{D}\mathbf{T}_t(\mathbf{x})$ 表示关于 $\mathbf{x}$ 的全微分。

步骤 2：对称性条件的时间微分。 对时间进行微分：

$$\frac{\partial}{\partial t}\mathrm{D}\mathbf{T}_t(\mathbf{x}) = \left(\frac{\partial}{\partial t}\mathrm{D}\mathbf{S}_t\right)(\mathbf{S}_\infty^{-1}(\mathbf{x}))\mathrm{D}(\mathbf{S}_\infty^{-1})(\mathbf{x}).$$

由于对所有 t 都满足对称性，因此该导数保持对称。

利用流微分方程（对 $\mathbf{x}$ 求导），我们得到：

$$\frac{\partial(\mathrm{D}\mathbf{S}_t)}{\partial t} = \mathrm{D}\mathbf{v}_t(\mathbf{S}_t)\cdot\mathrm{D}\mathbf{S}_t = \left(-\mathbf{I} - \mathrm{D}^2\log p_t(\mathbf{S}_t)\right)\cdot\mathrm{D}\mathbf{S}_t.$$

综上所述，我们看到

$$\left(-\mathbf{I} - \mathrm{D}^2\log p_t(\mathbf{S}_t)\right)\cdot\mathrm{D}\mathbf{S}_t\cdot\mathrm{D}(\mathbf{S}_\infty^{-1})$$

对于所有 $t \geq 0$，其具有对称性。

在 $t = 0$ 处，我们有 $\mathbf{S}_0 = \mathbf{I}$ 和 $\mathrm{D}\mathbf{S}_0 = \mathbf{I}$ ，得到：

$$\left(-\mathbf{I} - \mathrm{D}^2\log p_0(\mathbf{S}_\infty^{-1}(\mathbf{x}))\right)\cdot\mathrm{D}(\mathbf{S}_\infty^{-1})(\mathbf{x}) \quad \text{is symmetric.}$$

步骤 3：交换条件。 由于假设 $\mathbf{T}_0 = \mathbf{S}_\infty^{-1}$ 为最优解，其雅克比矩阵 $D\mathbf{T}_0 = \mathrm{D}(\mathbf{S}_\infty^{-1})$ 为对称矩阵。此外，海森矩阵 $\mathrm{D}^2\log p_0$ 也是对称矩阵。回想一下，两个对称矩阵的乘积为对称矩阵当且仅当它们可交换。因此，对于所有 $\mathbf{x}\in\mathbb{R}^D$ ，

$$\mathrm{D}^2\log p_0\big(\mathbf{S}_\infty^{-1}(\mathbf{x})\big) \quad \text{must commute with} \quad \mathrm{D}(\mathbf{S}_\infty^{-1})(\mathbf{x}).$$

情景 $\mathbf{y} = \mathbf{S}_\infty^{-1}(\mathbf{x})$ 给出等价条件：对所有 $\mathbf{y}\in\mathbb{R}^D$ ，

$$\mathrm{D}^2\log p_0(\mathbf{y}) \quad \text{must commute with} \quad \mathrm{D}\mathbf{S}_\infty(\mathbf{y}).$$

现在，我们将此条件转化为更易于计算的形式。由于 $\mathbf{S}_\infty$ 在 p_0 和 $\mathcal{N}(\mathbf{0},\mathbf{I})$

之间是最优的，Brenier 定理保证了存在某个凸函数 u ，使得 $\mathbf{S}_\infty = \nabla u$ 。由 Monge–Ampère 方程可得：

$$\log p_0(\mathbf{y}) = \log\det(\mathrm{D}^2 u(\mathbf{y})) - \frac{1}{2}\|\nabla u(\mathbf{y})\|^2 + \text{Constant}.$$

条件变为（使用 $\mathrm{D}\mathbf{S}_\infty = \mathrm{D}^2 u$ ）：

$$\mathrm{D}^2\left(\log\det \mathrm{D}^2 u - \frac{1}{2}\|\nabla u\|^2\right) \quad \text{must commute with} \quad \mathrm{D}^2 u. \tag{7.5.1}$$

这给出了 $\mathbf{T}_t$ 为最优解的必要条件。

步骤 4：构造反例。 我们将展示如何利用这一必要条件推导出矛盾。

假设我们可以构造一个凸函数 u，使得

$$\mathrm{D}^2\left(\log\det \mathrm{D}^2 u(\mathbf{x}) - \frac{1}{2}|\nabla u(\mathbf{x})|^2\right)$$

与 $\mathrm{D}^2 u(\mathbf{x})$ 不可交换，对某些 $\mathbf{x} \in \mathbb{R}^D$ 。定义 $p_0 = (\nabla u)^{-1}\#\mathcal{N}(\mathbf{0}, \mathbf{I})$ ，Brenier 定理表明 ∇u 是从 p_0 到 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ 的最优传输。然而，Equation (7.5.1) 中的条件不成立，导致矛盾。因此，我们的目标是构造这样的函数。考虑

$$u(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2 + \varepsilon\phi(\mathbf{x}), \quad \text{for a small } \varepsilon.$$

然后 $\mathrm{D}^2 u(\mathbf{0}) = \mathbf{I} + \varepsilon \mathrm{D}^2\phi(\mathbf{0})$ ，在 $\mathbf{x} = \mathbf{0}$ 处的对易条件要求 $\mathrm{D}^2\phi(\mathbf{0})$ 与 $\mathrm{D}^2(\Delta\phi)(\mathbf{0})$ 对易。

例如，在 $\mathbb{R}^2$ 中选择

$$\phi(x_1, x_2) = x_1 x_2 + x_1^4$$

提供了一个反例，其中海森矩阵 $\mathrm{D}^2 \log p_0$ 与雅克比 $\mathrm{D}^2 u$ 不能交换。

这种矛盾表明，$\mathbf{T}_t$ 并非对所有 $t \geq 0$ 都是最优的。因此，存在某个 $t_0 \geq 0$，使得映射 $\mathbf{T}_{t_0 \to \infty}$ 并非最优。

7.5.2 正则线性流和反流能否导致最优传输映射？

我们已经看到，PF-ODE（尤其是在 VP 型前向核中）通常不是 OT 映射。一个自然的问题是：

Question 7.5.1

当线性插值流 $(1-t)\mathbf{x}_0 + t\mathbf{x}_1$ 作用于独立耦合 $\pi(\mathbf{x}_0, \mathbf{x}_1) = p_{\text{src}}(\mathbf{x}_0)p_{\text{tgt}}(\mathbf{x}_1)$，其中 $\mathbf{x}_0 \sim p_{\text{src}}$ 与 $\mathbf{x}_1 \sim p_{\text{tgt}}$，是否能恢复 *OT* 映射？

该问题的答案是否定的。

然而，将一条线性路径与给定的耦合相结合，可以为真实的最优传输成本提供一个实用的上界。在所有可能的路径中，线性插值能提供最紧的此类上界，这一点将在接下来的讨论中得以说明。

正则线性流与最优传输。聚焦于二次代价的最优传输问题，我们考虑 Equation (7.2.1) 的等价形式，即 Benamou–Brenier 形式化表达 Equation (7.2.3)：

$$\mathcal{K}\left(p_{\text{src}}, p_{\text{tgt}}\right) := \min_{\substack{(p_t, \mathbf{v}_t) \text{ s.t. } \partial_t p_t + \nabla\cdot(p_t \mathbf{v}_t) = 0, \\ p_0 = p_{\text{src}},\ p_1 = p_{\text{tgt}}}} \int_0^1 \int_{\mathbb{R}^D} \|\mathbf{v}_t(\mathbf{x})\|^2 p_t(\mathbf{x})\, \mathrm{d}\mathbf{x}\, \mathrm{d}t.$$

然而，直接求解此最小化问题通常难以处理，因为它需要求解一个高度非线性的偏微分方程，即蒙日-安培方程。

虽然求解 Benamou-Brenier 公式通常难以处理，但 **liu2022rectified**; **lipman2024flow** 表明其动能具有一个实用的上界。这是通过将搜索范围限制在一种更简单的 条件流类型中实现的，其中每条路径由来自源分布与目标分布耦合 $\pi_{0,1}$ 的固定端点 $(\mathbf{x}_0, \mathbf{x}_1)$ 定义。在此 条件流类型中，正则的线性插值成为最优选择，如下所述。

Proposition 7.5.1: 基于条件流的 OT 动能上界

设 $\pi_{0,1}$ 为 p_{src} 与 p_{tgt} 间的任意耦合。

(1) 动能受限于任意连接端点的条件流 $\boldsymbol{\Psi}_t(\mathbf{x}_0, \mathbf{x}_1)$ 的期望路径能量上界：

$$\mathcal{K}\left(p_{\mathrm{src}}, p_{\mathrm{tgt}}\right) \leq \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_1)\sim\pi_{0,1}}\left[\int_0^1 \left\|\boldsymbol{\Psi}_t'(\mathbf{x}_0, \mathbf{x}_1)\right\|^2 \mathrm{d}t\right].$$

(2) 使右侧上界最小化的唯一条件流 $\boldsymbol{\Psi}_t^*$ 为线性插值路径：

$$\boldsymbol{\Psi}_t^*(\mathbf{x}_0, \mathbf{x}_1) = (1-t)\mathbf{x}_0 + t\mathbf{x}_1.$$

代入此最优路径可得最紧形式的上界：

$$\mathcal{K}\left(p_{\mathrm{src}}, p_{\mathrm{tgt}}\right) \leq \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_1)\sim\pi_{0,1}}\|\mathbf{x}_1 - \mathbf{x}_0\|^2.$$

Proof for Proposition.

证明基于詹森不等式与条件期望的塔性质直接应用，随后通过欧拉-拉格朗日方程求解简化变分问题；完整论证参见 (**lipman2024flow**) 第 4.7 节。 ∎

换句话说，线性插值 $\boldsymbol{\Psi}_t^*$ （即流匹配和修正流所使用的前向核）对于任意选定的耦合 $\pi_{0,1}$ ，最小化了真实动能的上界。

我们强调，此类条件流的最优性并不能保证边缘分布的全局最优。

回流与最优传输。 两个分布之间最简单的运输方案是使用简单的独立耦合，将它们的样本用直线连接。然而，这种方法显然不是最优的，问题不在于直线路径本身，而在于点对点的初始配对效率低下。

Reflow 过程可能提供一种建设性的响应。它是一种专门设计用于纠正这种配对的迭代算法，关键在于每一步都保证成本不增加 (**liu2022flow**)。这一特性表明，Reflow 会系统地将运输计划推向更优的配置，这自然引出了其收敛性的核心问题。

Question 7.5.2

如果我们迭代应用 *Rectify* 算子，会发生什么？由此产生的运输计划序列是否能收敛到最优解，或者 *Reflow* 过程的不动点是否能给出最优传输映射？

简短的回答是：通常情况下并非如此。以下是可能出错的原因说明。回顾一下，Reflow 过程通过如下更新方式，迭代优化 p_{src} 与 p_{tgt} 之间的耦合：

$$\pi^{(k+1)} = \texttt{Rectify}(\pi^{(k)}),$$

以产品耦合 $\pi^{(0)} := p_{\mathrm{src}}(\mathbf{x}_0)p_{\mathrm{tgt}}(\mathbf{x}_1)$ 初始化。更精确地，`Rectify` 通过以下方式输出更新后的耦合 $\pi^{(k+1)}$：在每次迭代 $k = 0, 1, 2, \ldots$ 中，通过如下方式学习速度场 $\mathbf{v}_t^{(k)}$ ：

$$\mathbf{v}_t^{(k)} \in \arg\min_{\mathbf{u}_t} \mathcal{L}(\mathbf{u}_t | \pi^{(k)}),$$

其中 $\mathcal{L}(\mathbf{u}_t|\pi^{(k)})$ 是在 Equation (5.4.1) 中定义的损失（例如，RF 或 FM 损失）。这里，为简化符号表示，我们采用速度场的非参数形式，而非其他情境中使用的参数化形式 $\boldsymbol{\phi}$。更新后的耦合关系如下：

$$\pi^{(k+1)}(\mathbf{x}_0, \mathbf{x}_1) := p_{\mathrm{src}}(\mathbf{x}_0)\, \delta\big(\mathbf{x}_1 - \boldsymbol{\Psi}_1^{(k)}(\mathbf{x}_0)\big),$$

其中 $\boldsymbol{\Psi}_1^{(k)}$ 表示通过从初值 $\mathbf{x}_0$ 积分 $\mathbf{v}_t^{(k)}$ 得到的时间 $t = 1$ 处的解映射。

在 (**liu2022flow**) 中观察到，对于 p_{src} 与 p_{tgt} 之间的耦合 π ，存在一个使重流损失最小的速度场 $\mathbf{v}_t$ ，即满足 $\mathcal{L}(\mathbf{v}_t|\pi) = 0$ ，但这并不一定意味着运输是最优的。

受 Benamou–Brenier 框架的启发，最优传输速度已知为势函数的梯度，**liu2022rectified** 提出了一项额外约束：速度场 $\mathbf{v}_t$ 应为势场。相应地，Equation (5.4.1) 中的目标被修改为将 $\mathbf{v}_t$ 限制在梯度向量场的空间内，也称为势流：

$$\mathbf{w}_t^{(k)} \in \underset{\substack{\mathbf{u}_t:\, \mathbf{u}_t = \nabla\varphi \\ \text{for some } \varphi:\, \mathbb{R}^D \to \mathbb{R}}}{\arg\min} \mathcal{L}(\mathbf{u}_t | \pi^{(k)}), \tag{7.5.2}$$

其余过程与 `Rectify` 中相同。我们将此关联算子记为 $\texttt{Rectify}_\perp$，以强调其在无旋向量场上的投影特性。

设 π 为 p_{src} 与 p_{tgt} 之间的耦合。**liu2022flow** 提出如下等价性猜想，用于刻画最优性：

(*i*) π 是一个最优传输耦合。

(*ii*) π 是势能整流算子的不动点：

$$\pi = \texttt{Rectify}_\perp(\pi).$$

(*iii*) 存在一个梯度速度场 $\mathbf{v}_t = \nabla\varphi_t$，使得修正损失消失：

$$\mathcal{L}(\mathbf{v}_t|\pi) = 0.$$

然而，**hertrich2025relation** 展现出两种类型的反例：

1. 当中间分布 p_t 的支撑集不连通时，可以找到 $\texttt{Rectify}_\perp$ 的不动点，其重整流损失为零，梯度速度场也未产生最优耦合。

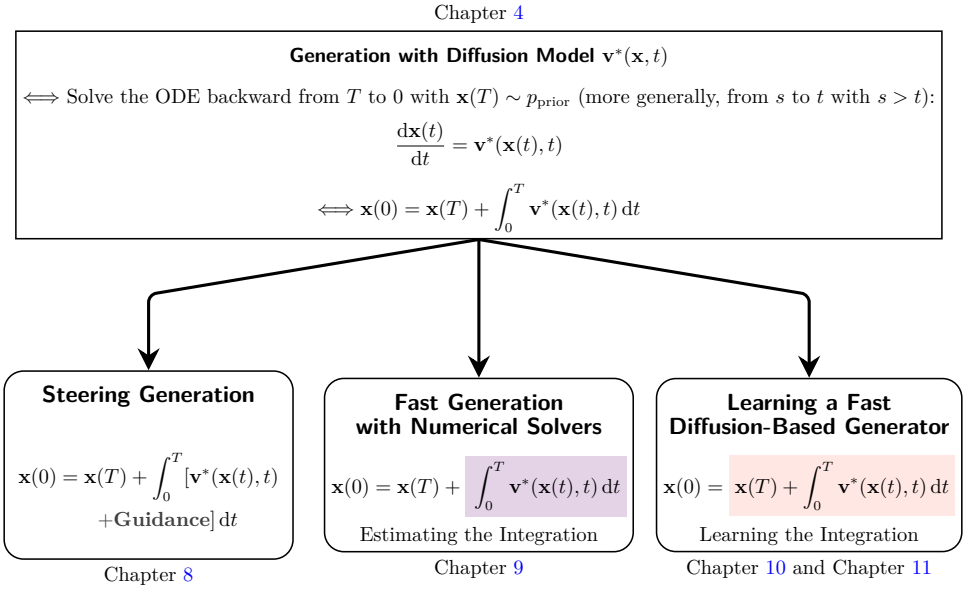
2. 即使两端的分布都是高斯分布，也存在损失任意小但与最优耦合偏差任意大的耦合。

因此，尽管修正流可能产生强大的生成式模型，但其作为最优传输求解器的可靠性仍然有限。这凸显了生成式建模与严谨最优传输理论之间的关键差距，也促使人们在两者的交叉领域开展进一步研究。

最后，我们注意到运输成本并不总是与下游性能相关；因此，计算确切的最优传输映射不一定能带来更好的实际效果。尽管如此，最优传输的变体仍然是科学和工程中许多问题的基础。扩散模型为探索这些挑战提供了一个强大的框架。

Part C

扩散模型的采样


Chapter 4
Generation with Diffusion Model $\mathbf{v}^*(\mathbf{x}, t)$
$\iff$ Solve the ODE backward from T to 0 with $\mathbf{x}(T) \sim p_{\text{prior}}$ (more generally, from s to t with $s > t$):
$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{v}^*(\mathbf{x}(t), t)$
$\iff \mathbf{x}(0) = \mathbf{x}(T) + \int_0^T \mathbf{v}^*(\mathbf{x}(t), t)\,\mathrm{d}t$
Steering Generation
$\mathbf{x}(0) = \mathbf{x}(T) + \int_0^T [\mathbf{v}^*(\mathbf{x}(t), t) + \mathbf{Guidance}]\,\mathrm{d}t$
Chapter 8
Fast Generation with Numerical Solvers
$\mathbf{x}(0) = \mathbf{x}(T) + \int_0^T \mathbf{v}^*(\mathbf{x}(t), t)\,\mathrm{d}t$
Estimating the Integration
Chapter 9
Learning a Fast Diffusion-Based Generator
$\mathbf{x}(0) = \mathbf{x}(T) + \int_0^T \mathbf{v}^*(\mathbf{x}(t), t)\,\mathrm{d}t$
Learning the Integration
Chapter 10 and Chapter 11

8

指导与可控生成

扩散模型是强大的生成框架。在无条件情景下，目标是学习 $p_{\text{data}}(\mathbf{x})$ 并生成样本而无需外部输入。

然而，许多应用需要条件生成，即生成的输出满足用户指定的标准。这可以通过引导一个无条件模型或直接学习条件分布 $p_0(\mathbf{x}|\mathbf{c})$ 来实现，其中条件 $\mathbf{c}$（例如标签、文本描述或草图）指导生成过程。

本章基于对条件得分的严谨观点，其可分解为一个*无条件方向*和一个*引导方向*，后者将样本推向条件的同时保持真实感。我们解释了引导为何至关重要，展示了条件得分如何作为控制的统一接口，并综述了近似引导项的各种方法。随后，我们将*控制*（满足条件）与*对齐*（在条件下满足人类偏好）区分开来，并描述了如何将偏好纳入同一框架。最后，我们讨论了无需额外奖励模型的偏好直接最优化（即，学成的评分器对更符合人类偏好的输出赋予更高得分）。

8.1 序言

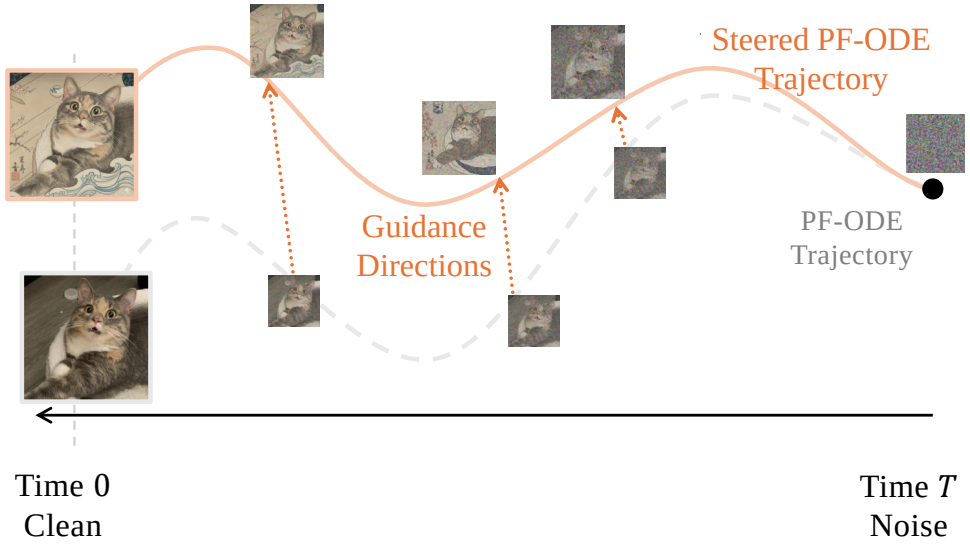


图 8.1: 引导扩散采样的示意图。反向时间 PF-ODE 采样从右侧的纯噪声 ($t = T$) 开始，逐步演化为左侧的干净样本 ($t = 0$)。在此过程中，由 w_t 加权的引导方向 $\nabla_{\mathbf{x}_t} \log \tilde{p}_t(\mathbf{c}|\mathbf{x}_t)$ 根据 $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + w_t \nabla_{\mathbf{x}_t} \log \tilde{p}_t(\mathbf{c}|\mathbf{x}_t)$ 修改速度场。这些额外的方向将轨迹引导至期望的属性（日本绘画风格），同时样本从粗略到精细细节逐步优化。

扩散模型的生成过程采用自粗至细的方式进行，提供了一个灵活的可控生成框架。在每一步中，去除少量噪声，样本逐渐变得清晰，结构和细节逐步显现。这一特性使得我们可以控制生成过程：通过向学成的、随时间变化的速度场添加引导项，可以引导生成轨迹以反映用户意图。

指导性采样在扩散模型中的合理基础是条件得分的贝叶斯分解。对于每个噪声水平 t ，

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{c}) = \underbrace{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)}_{\text{unconditional direction}} + \underbrace{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{c}|\mathbf{x}_t)}_{\text{guidance direction}}. \tag{8.1.1}$$

该恒等式表明，条件采样可以通过在无条件得分之上添加一个引导项 $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{c}|\mathbf{x}_t)$ 来实现。一系列可控生成方法（例如，分类器引导 (**dhariwal2021diffusion**)、通用无训练引导 (**ye2024tfg**)）可被解释为对该引导项的不同近似，因为 $p_t(\mathbf{c}|\mathbf{x}_t)$ 由于对 $\mathbf{x}_0$ 的边缘化而通常难以处理。

一旦获得这种近似，采样只需用其条件对应项替换无条件得分。使用 Equa-

tion (8.1.1)，PF-ODE 变为

$$\begin{aligned}\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} &= f(t)\mathbf{x}(t) - \frac{1}{2}g^2(t)\underbrace{\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}(t)|\mathbf{c})}_{\text{conditional score}} \\ &= f(t)\mathbf{x}(t) - \frac{1}{2}g^2(t)\Big[\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}(t)) + \nabla_{\mathbf{x}_t}\log p_t(\mathbf{c}|\mathbf{x}(t))\Big].\end{aligned} \tag{8.1.2}$$

我们强调，操控这些随时间变化的向量场从根本上依赖于它们的线性关系，因此下文以得分预测形式展开的讨论，通过其线性关系自然地延伸至 $\mathbf{x}$ -、$\boldsymbol{\epsilon}$ - 和 $\mathbf{v}$ -预测，如 Equation (6.3.1) 所示。

引导方向的实例。

1. **分类器引导（CG）**。在 Section 8.2 中，分类器引导（CG）在带噪声的数据 $\mathbf{x}_t$（通过在级别 t 下对干净的标注样本进行污染获得）上训练一个时间条件分类器 $p_{\psi}(\mathbf{c}|\mathbf{x}_t, t)$。在采样时，其输入梯度提供引导项：

   $$\nabla_{\mathbf{x}_t}\log p_{\psi}(\mathbf{c}|\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t}\log p_t(\mathbf{c}|\mathbf{x}_t),$$

   然后将其加到无条件得分 (**dhariwal2021diffusion**)。

2. **无分类器引导（CFG）**。在 Section 8.3 中，CFG 直接训练一个单一的条件模型

   $$\mathbf{s}_{\phi}(\mathbf{x}_t, t, \mathbf{c}) \approx \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{c}),$$

   其中，无条件模型通过在训练步骤中随机将条件替换为特殊的空标记来联合学习。

3. **无需训练（代理）引导**。条件 $p_t(\mathbf{c}|\mathbf{x}_t)$ 通常难以处理，因为其需要对干净的潜在变量 $\mathbf{x}_0$ 进行边缘化：

   $$p_t(\mathbf{c}|\mathbf{x}_t) = \int p(\mathbf{c}|\mathbf{x}_0)p(\mathbf{x}_0|\mathbf{x}_t)\mathrm{d}\mathbf{x}_0,$$

   在典型应用中，这些因子中至少有一个是未知的，使得该积分难以处理。

   在 Section 8.4.1 中，无需训练（基于损失的）引导避免了直接评估条件似然 $p_t(\mathbf{c}|\mathbf{x}_t)$。相反，它引入了一个现成的损失 $\ell(\mathbf{x}_t, \mathbf{c}; t)$，并定义了一个代理条件

分布 $\widetilde{p}_t(\mathbf{c}|\mathbf{x}_t)$ 为，

$$\widetilde{p}_t(\mathbf{c}|\mathbf{x}_t) \propto \exp\big(-\tau\,\ell(\mathbf{x}_t,\mathbf{c};t)\big), \quad \tau>0,$$

这起到了伪似然的作用。该公式规避了计算真实条件似然的不可行性，同时仍能通过所选损失的梯度进行引导。其条件得分仅通过损失函数关于 τ 的梯度来计算：

$$\nabla_{\mathbf{x}_t}\log\widetilde{p}_t(\mathbf{c}|\mathbf{x}_t) = -\tau\nabla_{\mathbf{x}_t}\ell(\mathbf{x}_t,\mathbf{c};t).$$

该项被加入无条件得分，权重为 w_t ：

$$\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t) + w_t\big[-\tau\nabla_{\mathbf{x}_t}\ell(\mathbf{x}_t,\mathbf{c};t)\big].$$

这正是所谓的 倾斜密度 $\widetilde{p}_t^{\text{tilt}}(\mathbf{x}_t|\mathbf{c})$ 的得分，定义为：

$$\widetilde{p}_t^{\text{tilt}}(\mathbf{x}_t|\mathbf{c}) \propto p_t(\mathbf{x}_t)\widetilde{p}_t(\mathbf{c}|\mathbf{x}_t)^{w_t} \propto p_t(\mathbf{x}_t)\exp\big(-w_t\tau\ell(\mathbf{x}_t,\mathbf{c};t)\big).$$

在实际操作中，我们将 Equation (8.1.2) 中采样条件得分替换为这种倾斜得分，并求解得到的微分方程以生成样本。

鉴于此，分类器引导本质上是通过学成的分类器 $\widetilde{p}_t(\mathbf{c}|\mathbf{x}_t) := p_{\boldsymbol{\psi}^\times}(\mathbf{c}|\mathbf{x}_t,t)$ 实现的代理引导，其表达式为：

$$\ell(\mathbf{x}_t,\mathbf{c};t) = -\log p_{\boldsymbol{\psi}^\times}(\mathbf{c}|\mathbf{x}_t,t), \quad \tau=1.$$

引导对采样轨迹的影响如 Figure 8.1 所示。

所有这些技术同样可以应用于条件模型之上，从而在生成过程中注入额外的控制信号。

Remark.

> 引导的 PF-ODE 在一般情况下不从倾斜族中采样。即使使用确切的得分和确切的 ODE 积分，用倾斜的得分替换得分也不会使时间–t 边缘分布等于 $\{\widetilde{p}_t^{\text{tilt}}(\cdot|\mathbf{c})\}_{t\in[0,1]}$ ，也不会使终端分布等于 $\widetilde{p}_0^{\text{tilt}}(\cdot|\mathbf{c})$ 。定义
>
> $$\mathbf{v}_t^{\text{orig}} = \mathbf{f} - \tfrac{1}{2}g^2(t)\nabla\log p_t, \quad \mathbf{h}_t(\mathbf{x}) = e^{-w_t\tau\ell(\mathbf{x},\mathbf{c};t)}, \quad \widetilde{p}_t^{\text{tilt}} = \tfrac{p_t\mathbf{h}_t}{Z_t}.$$

引导的 PF-ODE 使用

$$\mathbf{v}_t^{\text{tilt}} = \mathbf{f} - \tfrac{1}{2}g^2(t)\nabla \log \widetilde{p}_t^{\text{tilt}} = \mathbf{v}_t^{\text{orig}} - \tfrac{1}{2}g^2(t)\nabla \log \mathbf{h}_t.$$

如果 $\widetilde{p}_t^{\text{tilt}}$ 是真实的边缘分布，则它们应满足

$$\partial_t \widetilde{p}_t^{\text{tilt}} + \nabla\cdot(\widetilde{p}_t^{\text{tilt}}\mathbf{v}_t^{\text{tilt}}) = 0.$$

但直接计算得到残差

$$\begin{aligned}&\partial_t \widetilde{p}_t^{\text{tilt}} + \nabla\cdot(\widetilde{p}_t^{\text{tilt}}\mathbf{v}_t^{\text{tilt}})\\ =&\widetilde{p}_t^{\text{tilt}}\Big[\partial_t \log \mathbf{h}_t + \mathbf{v}_t^{\text{orig}}\cdot\nabla\log\mathbf{h}_t - \tfrac{1}{2}g^2(t)\big(\Delta\log\mathbf{h}_t + \|\nabla\log\mathbf{h}_t\|^2\big) - \tfrac{Z_t'}{Z_t}\Big].\end{aligned}$$

因此，$\widetilde{p}_t^{\text{tilt}}$ 与 PF-ODE 的边缘分布一致当且仅当该括号对所有 $\mathbf{x}$ 恒为零，即

$$\partial_t \log \mathbf{h}_t + \mathbf{v}_t^{\text{orig}}\cdot\nabla\log\mathbf{h}_t = \tfrac{1}{2}g^2(t)\big(\Delta\log\mathbf{h}_t + \|\nabla\log\mathbf{h}_t\|^2\big) + \tfrac{Z_t'}{Z_t}.$$

当 $\omega_t \equiv 0$ 时（无条件生成），此条件显然成立，但对于 $\mathbf{h}_t(\mathbf{x}) = e^{-w_t\tau\ell(\mathbf{x},\mathbf{c};t)}$ 几乎从不成立，除非在 w_t 或 ℓ 的某些非常特殊的情况下。因此，一般情况下，$\{\widetilde{p}_t^{\text{tilt}}\}$ 不是 PF-ODE 的边缘分布，终端样本的分布也不是 $\widetilde{p}_0^{\text{tilt}}(\mathbf{x}_0|\mathbf{c})$ 。

从控制到更优对齐：直接偏好最优化。 强控制可以是条件性成立但偏好性不成立：一个样本可能满足条件信号（例如提示），但仍偏离人类实际偏好。我们通过将条件目标倾斜一个偏好评分来形式化这一点[1]：

$$\widetilde{p}_0^{\text{tilt}}(\mathbf{x}_0|\mathbf{c}) \propto p_0(\mathbf{x}_0|\mathbf{c})\exp\big(\beta r(\mathbf{x}_0,\mathbf{c})\big),$$

其中 $r(\mathbf{x}_0,\mathbf{c})$ 是针对干净样本 $\mathbf{x}_0$ 和条件 $\mathbf{c}$ 的标量对齐评分（奖励）（较大的 r 表示更好的对齐）。在实际应用中，r 可能是：(i) 外部奖励/分类器的 Logit 或对数概率，(ii) 相似度度量（例如，CLIP/感知 (**radford2021learning**)），或 (iii) 学成的偏好模型。

现有实现此类可控性的方法通常收集人类对模型生成结果相对质量的标签，并微调条件扩散模型以匹配这些偏好，通常通过人类反馈的强化学习（RLHF）实现。然而，RLHF 是一种复杂且常常不稳定的流程：它首先拟合一个奖励模

[1]我们注意到，训练-free 的引导也可以在同一个框架下理解，即寻找一个带有损失引导的倾斜分布 $\ell(\mathbf{x}_t,\mathbf{c},t)$

型来捕捉人类偏好，然后利用强化学习微调条件扩散模型，以最大化该估计的奖励，同时限制策略从原始模型的漂移。

这自然引出了一个问题：*我们能否完全移除奖励模型训练阶段？*我们通过 Diffusion-DPO (**wallace2024diffusion**) 来解决这个问题，这是为大型语言模型开发的 Direct Preference Optimization (**rafailov2023direct**) 的一种适应。如 Section 8.5 所述，Diffusion-DPO 直接从成对选择中学习偏好倾斜，因此条件扩散模型被微调以对齐偏好，而无需分离的奖励模型。

8.2 分类器指导

8.2.1 分类器引导的基础

设 $\mathbf{c}$ 表示从分布 $p(\mathbf{c})$ 中抽取的条件变量，例如类标记、描述或其它辅助信息。我们的目标是从 $p_0(\mathbf{x}|\mathbf{c})$ 中抽取样本。在基于扩散的条件生成中，我们通过运行反向时间动力学来实现这一目标，其时间边缘分布为 $p_t(\cdot|\mathbf{c})$ 。这些动力学的漂移项依赖于条件得分

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{c}), \quad t \in [0, T].$$

因此，一条标准且有效的途径[2] 是用来估计这一数量的。

一个基于贝叶斯规则的基本见解是，条件得分可以分解为：

$$\begin{aligned}
\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{c}) &= \nabla_{\mathbf{x}_t} \log \left(\frac{p_t(\mathbf{x}_t) p_t(\mathbf{c}|\mathbf{x}_t)}{p(\mathbf{c})} \right) \\
&= \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{c}|\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log p(\mathbf{c}) \\
&= \underbrace{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)}_{\text{unconditional score}} + \underbrace{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{c}|\mathbf{x}_t)}_{\text{classifier gradient}},
\end{aligned} \tag{8.2.1}$$

其中 $p_t(\mathbf{c}|\mathbf{x}_t)$ 表示在 $\mathbf{x}_t$ 条件下 $\mathbf{c}$ 的概率，该条件从时间 t 的噪声输入 $\mathbf{x}_t$ 预测条件 $\mathbf{c}$。

这种分解[3] 激发了由 **dhariwal2021diffusion** 提出的 *分类器引导*（CG）方法，该方法利用一个预训练的时间相关分类器 $p_t(\mathbf{c}|\mathbf{x}_t)$ 来引导生成过程。具体而言，我们定义了一个单参数族的 *引导密度*（倾斜条件概率），其引导尺度为 $\omega \geq 0$：

$$p_t(\mathbf{x}_t|\mathbf{c}, \omega) \propto p_t(\mathbf{x}_t) p_t(\mathbf{c}|\mathbf{x}_t)^{\omega}, \tag{8.2.2}$$

这给出了评分函数：

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{c}, \omega) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \omega \nabla_{\mathbf{x}_t} \log p_t(\mathbf{c}|\mathbf{x}_t). \tag{8.2.3}$$

从几何上看，这会将无条件流倾斜到增加类别似然的方向。当 $\omega = 1$ 时，$p_t(\mathbf{x}_t|\mathbf{c}, \omega)$

[2] 原则上，若 $p(\mathbf{c}|\mathbf{x})$ 可用且校准良好，人们可以通过拒绝采样或重要性采样从无条件生成器中得到 $p_0(\mathbf{x}|\mathbf{c})$。这在高维或罕见条件下很少具有实际操作性。

[3] 在最后一个恒等式中，由于 $\nabla_{\mathbf{x}_t} \log p(\mathbf{c})$ 不依赖于 $\mathbf{x}_t$ ，因此在微分下消失。

与真实的条件 $p_t(\mathbf{x}_t|\mathbf{c})$ 重合；对于 $\omega \neq 1$ ，它是一种引导（加温）的重加权，而非字面意义上的条件。

标量 $\omega \geq 0$ 调节分类器的影响：

- $\omega = 1$ ： 恢复真实的条件得分 $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{c})$ 。
- $\omega > 1$ ： 增强分类器信号，通常会提高条件保真度（往往以牺牲多样性为代价）。
- $0 \leq \omega < 1$ ： 降低分类器信号的权重，通常会增加样本多样性，同时减弱条件控制。

计算机图形学中的实用近似。 在实际应用中，CG 是一种无需训练的方法（相对于扩散模型而言），用于引导预训练的无条件扩散模型，

$$\mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t).$$

CG 仅在采样时应用，不修改扩散模型本身。为此，训练一个时间相关的分类器 $p_{\boldsymbol{\psi}}(\mathbf{c}|\mathbf{x}_t, t)$ ，以从不同噪声水平 t 下的噪声输入 $\mathbf{x}_t$ 中预测条件 $\mathbf{c}$ 。该分类器通过最小化交叉熵损失以标准方式训练：

$$\mathbb{E}_{t\sim\mathcal{U}[0,T],(\mathbf{x},\mathbf{c})\sim p_{\text{data}},\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\Big[-\log p_{\boldsymbol{\psi}}(\mathbf{c}|\mathbf{x}_t, t)\Big], \tag{8.2.4}$$

其中 $(\mathbf{x}, \mathbf{c}) \sim p_{\text{data}}$ 表示成对标注的数据，$\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$ 是时间 t 时的噪声输入。由于分类器需要在所有噪声水平下都能可靠运行，因此必须显式地基于 t 进行条件化（例如，通过时间嵌入）。

训练完成后，分类器提供的得分可作为真实似然梯度的替代指标：

$$\nabla_{\mathbf{x}_t} \log p_{\boldsymbol{\psi}^\times}(\mathbf{c}|\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \log p_t(\mathbf{c}|\mathbf{x}_t).$$

8.2.2 使用 CG 进行推理

推理时，将分类器梯度 $\nabla_{\mathbf{x}_t} \log p_{\boldsymbol{\psi}^\times}(\mathbf{c}|\mathbf{x}_t, t)$ 加到无条件评分函数上，并通过指导权重 ω 缩放，从而得到对引导评分 $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{c}, \omega)$ 的近似，如 Equa-

tion (8.2.3) 所示：

$$\begin{aligned}\mathbf{s}^{\mathrm{CG}}(\mathbf{x}_t, t, \mathbf{c}; \omega) :=\ & \underbrace{\mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t, t)}_{\text{uncond. direction}} + \omega \underbrace{\nabla_{\mathbf{x}_t} \log p_{\boldsymbol{\psi}^\times}(\mathbf{c}|\mathbf{x}_t, t)}_{\text{guidance direction}} \\ \approx\ & \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{c}, \omega).\end{aligned}$$

相应地，只需将反向时间 SDE 或 PF-ODE 中的无条件评分函数 $\mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t, t)$ 替换为指定 ω 的引导评分 $\mathbf{s}^{\mathrm{CG}}(\mathbf{x}_t, t, \mathbf{c}; \omega)$，如 Equation (8.1.2) 所示，从而引导生成轨迹朝向与条件 $\mathbf{c}$ 一致的样本。

8.2.3 优势与局限性

CG 为条件生成提供了一种简单且灵活的机制，可通过 ω 显式控制条件强度。它可以与任何预训练的无条件扩散模型配合使用，仅需额外添加一个分类器用于条件控制。

然而，该方法存在明显的局限性：

- **训练成本**：分类器必须在所有噪声水平下进行训练以正常运行，这计算开销很大。
- **鲁棒性**：分类器必须在严重损坏的输入上表现出良好的泛化能力 $\mathbf{x}_t$，尤其是在大规模 t 的情况下，这可能具有挑战性。
- **分离训练**：由于分类器是独立于扩散模型进行训练的，因此它可能无法与学成的数据分布完全对齐。

8.3 无分类器指导

8.3.1 无分类器指导的基础

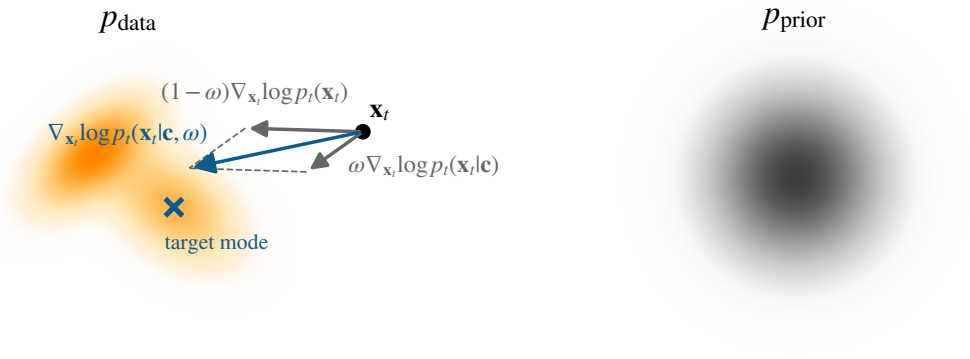


图 8.2: CFG 的示意图。调整后的得分 $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{c},\omega)$ 通过无条件得分 $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)$ 与条件得分 $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{c})$ 的线性插值获得，权重为 ω 。该结果方向将样本从先验引导至与目标条件一致的数据分布模式。

无分类器指引（CFG）（**ho2021classifier**）是一种简化的基于分类器的指引方法，消除了对独立分类器的需求。其核心思想是通过调整评分函数的梯度，使得在无需显式分类器的情况下实现有效的条件化。具体而言，条件分布的对数概率梯度被调整如下：

$$\nabla_{\mathbf{x}_t}\log p_t(\mathbf{c}|\mathbf{x}_t) = \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{c}) - \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t). \tag{8.3.1}$$

将该表达式代入 Equation (8.2.3) 得到条件得分的如下形式：

$$\begin{aligned}\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{c},\omega) &= \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t) + \omega\left(\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{c}) - \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)\right)\\ &= \omega\underbrace{\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{c})}_{\text{conditional score}} + (1-\omega)\underbrace{\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)}_{\text{unconditional score}}.\end{aligned} \tag{8.3.2}$$

超参数 ω 再次在控制条件信息的影响方面起着关键作用（我们取 $\omega \geq 0$ ）：

- 在 $\omega = 0$ 时，模型表现为一个无条件扩散模型，完全忽略条件信息。
- 在 $\omega = 1$ 时，模型使用条件得分而无需额外引导。
- 对于 $\omega > 1$，模型更加重视条件得分而减少对无条件得分的依赖，这增强了与 $\mathbf{c}$ 的一致性，但通常会降低多样性。

8.3.2 CFG 的训练与采样

通过 CFG 实现无条件与条件扩散模型的联合训练。 与 CG 不同，CFG 需要重新训练一个显式考虑条件变量 $\mathbf{c}$ 的扩散模型。然而，为条件评分函数和无条件评分函数分别训练两个模型在计算上通常不可行。为解决此问题，CFG 采用单个模型 $\mathbf{s}_\phi(\mathbf{x}_t, t; \mathbf{c})$ ，通过将 $\mathbf{c}$ 视为额外输入，在单一模型中学习两种评分函数。训练过程定义如下：

- 对于无条件训练，将一个空 token $\emptyset$ 作为条件输入的替代，得到 $\mathbf{s}_\phi(\mathbf{x}_t, t, \emptyset)$ 。
- 对于条件训练，真实条件变量 $\mathbf{c}$ 作为输入提供，从而得到 $\mathbf{s}_\phi(\mathbf{x}_t, t, \mathbf{c})$ 。

这两个训练策略通过以概率 p_{uncond} （一个用户定义的超参数，通常设为 0.1 ）随机将 $\mathbf{c}$ 替换为无输入 $\emptyset$ 来统一。这种联合训练策略使模型能够同时学习条件与无条件评分函数。完整的训练算法如 Algorithm 4 所示，其中还对比了标准无条件训练方法（见 Algorithm 3）。我们指出，在训练过程中，CFG 权重 ω 并未被使用。

Algorithm 3 Uncond. DM

Repeat
  $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$
  $t \sim \mathcal{U}[0, T]$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$
  Take gradient step on:
  $\nabla_\phi \left\| \mathbf{s}_\phi(\mathbf{x}_t, t) - \mathbf{s} \right\|^2$
until converged

Algorithm 4 CFG for Cond. DM

Input: p_{uncond}: prob. of unconditional dropout
Repeat
  $(\mathbf{x}, \mathbf{c}) \sim p_{\text{data}}(\mathbf{x}, \mathbf{c})$
  $\mathbf{c} \leftarrow \emptyset$ with prob. p_{uncond}
  $t \sim \mathcal{U}[0, T]$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$
  Take gradient step on:
  $\nabla_\phi \left\| \mathbf{s}_\phi(\mathbf{x}_t, t, \mathbf{c}) - \mathbf{s} \right\|^2$
until converged

带条件的采样与 CFG。 一旦模型 $\mathbf{s}_{\phi^\times}(\mathbf{x}_t, t, \mathbf{c})$ 被使用算法 4 训练完成，CFG 可以在采样过程中被应用。对数概率的梯度由下式给出：

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{c}, \omega) &= \omega \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{c}) + (1-\omega)\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \\ &\approx \omega \underbrace{\mathbf{s}_{\phi^\times}(\mathbf{x}_t, t, \mathbf{c})}_{\text{conditional score}} + (1-\omega) \underbrace{\mathbf{s}_{\phi^\times}(\mathbf{x}_t, t, \emptyset)}_{\text{unconditional score}} \\ &=: \mathbf{s}^{\text{CFG}}_{\phi^\times}(\mathbf{x}_t, t, \mathbf{c}; \omega).\end{aligned} \tag{8.3.3}$$

在采样过程中，应用固定的（或可选的时间相关）无分类器引导权重 ω。反向时间 SDE (Equation (4.1.6)) 或 PF-ODE (Equation (4.1.8)) 中的无条件得分 $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ 会被引导得分 $\mathbf{s}^{\text{CFG}}_{\phi^\times}(\mathbf{x}_t, t, \mathbf{c}; \omega)$ 替代，如 Equation (8.1.2) 所示，该方法以加权方式结合了条件得分与无条件得分。

该公式通过调整 ω 实现可控生成，使样本能够被引导至条件信号 $\mathbf{c}$ ，同时保持多样性。因此，CFG 提供了一种有效且计算高效的精确条件生成方法，因为它只需训练单一的扩散模型。

8.4 （可选）无需训练的引导

在本节中，我们介绍了广泛训练无关引导方法背后的高层理念 (**chung2023diffusion**; **ye2024tfg**; **he2024manifold**; **bansal2023universal**)。尽管实现方式和应用场景存在差异，这些方法由 Equation (8.1.1) 中表达的核心原则统一起来。我们首先在 Section 8.4.1 中介绍训练无关引导的高层方法，然后将这一思想扩展到训练无关的逆问题求解，Section 8.4.2 中提供了简要概述。

设置与记号。 令 $\mathbf{c}$ 表示一个条件变量。我们假设可以访问一个预训练的扩散模型 $\mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t, t)$，该模型以得分预测的形式表达。[4]. 此外，假设我们得到一个非负函数

$$\ell(\cdot, \mathbf{c})\colon \mathbb{R}^D \to \mathbb{R}_{\geq 0}$$

用于量化样本 $\mathbf{x} \in \mathbb{R}^D$ 与条件 $\mathbf{c}$ 的匹配程度，其中 $\ell(\mathbf{x}, \mathbf{c})$ 的较小值表示更强的匹配度。此类函数的具体示例包括：(i) $\mathbf{c}$ 为参考图像，$\ell(\cdot, \mathbf{c})$ 为衡量感知接近程度的相似度得分；(ii) $\ell(\cdot, \mathbf{c})$ 为通过预训练模型（如 CLIP (**radford2021learning**)）计算的基于特征的相似度得分。

考虑标准的线性-高斯前向加噪核 $p_t(\cdot|\mathbf{x}_0) := \mathcal{N}(\cdot; \alpha_t \mathbf{x}_0, \sigma_t^2 \mathbf{I})$ 。我们回顾 Equation (9.2.3) 中的 DDIM 更新，并将其作为一个例子：

$$\mathbf{x}_{t\to t-1} = \alpha_{t-1} \underbrace{\hat{\mathbf{x}}_0(\mathbf{x}_t)}_{\text{in data space}} - \sigma_{t-1}\sigma_t \underbrace{\hat{\mathbf{s}}(\mathbf{x}_t)}_{\text{in noise space}}, \tag{8.4.1}$$

其中 $\hat{\mathbf{x}}_0(\mathbf{x}_t) := \mathbf{x}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t, t)$ 为（干净的）$\mathbf{x}$ 预测，$\hat{\mathbf{s}}(\mathbf{x}_t) := \mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t, t)$ 为在时间层级 t 下由 $\mathbf{x}_t$ 得到的得分预测。

8.4.1 无训练引导的概念框架

大多数无需训练的引导方法 (**ye2024tfg**) 在 *数据空间* 或 *噪声空间* 中引入修正，以引导 DDIM 更新在 Equation (8.4.2) 中满足条件 $\mathbf{c}$。

$$\mathbf{x}_{t\to t-1} = \alpha_{t-1} \underbrace{\left(\hat{\mathbf{x}}_0(\mathbf{x}_t) + \eta_t^{\text{data}} \mathcal{G}_0\right)}_{\text{A. data space}} - \sigma_{t-1}\sigma_t \underbrace{\left(\hat{\mathbf{s}}(\mathbf{x}_t) + \eta_t^{\text{latent}} \mathcal{G}_t\right)}_{\text{B. noise space}}, \tag{8.4.2}$$

其中 $\eta_t^{\text{data}}, \eta_t^{\text{latent}} \geq 0$ 为随时间变化的引导强度，$\mathcal{G}_0$、$\mathcal{G}_t$ 为如下定义的修正项。

[4]此处，为简化数学表达，我们采用得分和 $\mathbf{x}$ -预测参数化；其他参数化方式（例如 $\boldsymbol{\epsilon}$ -预测）可类似处理。

A. 数据空间中的指导 沿负梯度方向下降

$$\mathcal{G}_0 := -\nabla_{\mathbf{x}_0}\ell(\mathbf{x}_0, \mathbf{c}),$$

数据空间中的修正清洁估计值，

$$\hat{\mathbf{x}}_0(\mathbf{x}_t) + \eta_t^{\text{data}}\mathcal{G}_0,$$

可以逐渐引导至更满足条件 $\mathbf{c}$ 的样本。该梯度下降方案可迭代应用，以逐步提高对齐效果。

代表性例子包括 MGPD (**he2023manifold**) 和 UGD (**bansal2023universal**)。

B. 噪声空间中的指导 如 Section 8.1 所述，条件得分 $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{c}|\mathbf{x}_t)$ 通常难以处理。一种实用的近似方法是引入一个代理似然 $\widetilde{p}_t(\mathbf{c}|\mathbf{x}_t)$ ：

$$\widetilde{p}_t(\mathbf{c}|\mathbf{x}_t) \propto \exp\big(-\eta\ell\left(\hat{\mathbf{x}}_0(\mathbf{x}_t), \mathbf{c}\right)\big)$$

带有重缩放常数 $\eta > 0$，使得

$$\nabla_{\mathbf{x}_t}\log\widetilde{p}_t(\mathbf{c}|\mathbf{x}_t) = -\eta\nabla_{\mathbf{x}_t}\ell\left(\hat{\mathbf{x}}_0(\mathbf{x}_t), \mathbf{c}\right) =: \mathcal{G}_t,$$

其中 $\hat{\mathbf{x}}_0(\mathbf{x}_t)$ 由扩散模型的预测得到。将其代入条件得分的贝叶斯法则，可得代理：

$$\begin{aligned}\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{c}) &\approx \underbrace{\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)}_{\text{unconditional}} + \underbrace{\nabla_{\mathbf{x}_t}\log\widetilde{p}_t(\mathbf{c}|\mathbf{x}_t)}_{\text{guidance}} \\ &\approx \hat{\mathbf{s}}(\mathbf{x}_t) + \eta_t^{\text{latent}}\mathcal{G}_t,\end{aligned}$$

这作为在噪声空间中引导下的校正。

然而，我们注意到评估 $\mathcal{G}_t$ 需要通过 $\mathbf{x}$ 预测进行反向传播，即

$$\nabla_{\mathbf{x}_t}\hat{\mathbf{x}}_0(\mathbf{x}_t)^\top \cdot \nabla_{\mathbf{x}_0}\log\ell_{\mathbf{c}}(\mathbf{x}_0)|_{\mathbf{x}_0=\hat{\mathbf{x}}_0(\mathbf{x}_t)},$$

这在实际中可能导致巨大的计算成本。

代表性例子包括 (**yu2023freedom**; **chung2022diffusion**; **bansal2023universal**)。

8.4.2 无训练方法在逆问题中的应用示例

Section 8.4.1 中介绍的原则在反问题中有重要应用。我们首先概述背景，然后提供几个具体示例，说明如何利用预训练扩散模型进行推理时的反问题求解。

逆问题背景。 设 $\mathcal{A}$ 为一种失真算子（可以是线性的或非线性的，已知的或未知的），例如模糊核或图像修复，设 $\mathbf{y}$ 为由以下失真模型生成的观测：

$$\mathbf{y} = \mathcal{A}(\mathbf{x}_0) + \sigma_{\mathbf{y}}\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{8.4.3}$$

逆问题的目标是从后验分布 $p_0(\mathbf{x}_0|\mathbf{y})$ 中进行采样，其中给定观测 $\mathbf{y}$ 可能存在无限多种对应的重构 $\mathbf{x}_0$。目标是恢复一个 $\mathbf{x}_0$，该 $\mathbf{x}_0$ 能够去除 $\mathbf{y}$ 中的噪声，同时保留其真实且语义上的特征。

求解逆问题的传统方法通常遵循监督框架，这需要收集退化样本与恢复样本的配对数据 $(\mathbf{y}, \mathbf{x})$，并依赖于最优化方法或神经网络的监督训练。此类方法在数据准备方面可能成本较高，并且可能缺乏对未见数据的泛化能力。

预训练扩散模型作为逆问题求解器。 如前所示，条件得分可通过贝叶斯规则分解：

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) = \underbrace{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)}_{\text{data score}} + \underbrace{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t)}_{\text{measurement alignment}}. \tag{8.4.4}$$

该分解将数据得分与一个测量对齐项分离，其中 $\mathbf{y}$ 与反问题相关。这使得能够通过建模干净数据分布 p_{data} 来无监督地求解 Equation (8.4.3)，且在反演过程中应用该模型。更具体地：

- **数据得分** $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$：使用在干净数据上训练的预训练扩散模型 $\mathbf{s}_{\phi^\times}(\mathbf{x}_t, t)$ 近似得到。
- **测量对齐** $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t)$：无法以闭式求解，因为它涉及对潜变量进行边缘化。

因此，大多数使用预训练扩散模型的无训练方法专注于近似 $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t)$。我们采用了一个常见的元形式，如 (**daras2024survey**) 所总结：

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \approx -\frac{\mathcal{P}_t \; \mathcal{M}_t}{\gamma_t}.$$

这里：

- $\mathcal{M}_t$：用于量化观测值 $\mathbf{y}$ 与估计信号之间不匹配程度的向量，
- $\mathcal{P}_t$：将 $\mathcal{M}_t$ 映射回 $\mathbf{x}_t$ 的环境空间的映射，
- γ_t：控制引导强度的标量。

代表性方法以不同方式实例化 $\mathcal{M}_t$ 、$\mathcal{P}_t$ 和 γ_t ，如下所示，通过颜色编码的组件突出显示。

基于扩散的逆问题求解器的实例。 我们展示了代表性方法，这些方法利用预训练的扩散模型来提供无监督方法（无需成对数据），可以使用相同的学成代理 p_{data} 灵活地应用于各种逆问题。

得分 SDE (song2020score). 早期基于扩散的逆问题求解器的工作之一。它考虑了一个已知的线性退化模型 $\mathbf{A}$，专注于噪声为零的情景，其中 $\sigma_{\mathbf{y}} = 0$。由于 $\mathbf{A}$ 是线性的，可以构建一个与噪声水平匹配的观测值

$$\mathbf{y}_t := \alpha_t \mathbf{y} + \sigma_t \boldsymbol{\epsilon},$$

并使用残差 $\mathbf{y}_t - \mathbf{A}\mathbf{x}_t$ （注意：一般情况下为 $\mathbf{y}_t \neq \mathbf{A}\mathbf{x}_t$）来驱动似然风格的修正。一种常见的近似（忽略乘性常数）是

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \approx - \mathbf{A}^\top (\mathbf{y}_t - \mathbf{A}\mathbf{x}_t).$$

迭代潜变量精炼（ILVR） (choi2021ilvr). 在与 ScoreSDE 情形相同的设置下，ILVR 估计：

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \approx -\mathbf{A}^\dagger(\mathbf{y}_t - \mathbf{A}\mathbf{x}_t) = - (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top (\mathbf{y}_t - \mathbf{A}\mathbf{x}_t),$$

其中 $\mathbf{A}^\dagger$ 为 Moore–Penrose 伪逆，且 $\mathbf{y}_t = \alpha_t \mathbf{y} + \sigma_t \boldsymbol{\epsilon}_t$。

扩散后验采样（DPS） (chung2022diffusion)。 对于已知非线性前向算子 $\mathcal{A}$ 且具有加性高斯噪声水平 $\sigma_{\mathbf{y}} \geq 0$ 的反问题，一种广泛应用的方法是 降噪后验得分（DPS），该方法近似

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \approx \nabla_{\mathbf{x}_t} \log p_t\big(\mathbf{y}|X_0 = \hat{\mathbf{x}}_0(\mathbf{x}_t)\big), \tag{8.4.5}$$

其中 $\hat{\mathbf{x}}_0(\mathbf{x}_t) := \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$ 表示在时间 t 给定噪声观测 $\mathbf{x}_t$ 时干净样本的条件均值，通常使用 Tweedie 公式 (Equation (3.3.6)) 从预训练的扩散模型中估计得到。

这个单点近似假设条件分布 $p(\mathbf{x}_0|\mathbf{x}_t)$ 高度集中，并由此得出：

$$\begin{aligned} p_t(\mathbf{y}|\mathbf{x}_t) &= \int p_t(\mathbf{y}|\mathbf{x}_t, \mathbf{x}_0) p(\mathbf{x}_0|\mathbf{x}_t) \mathrm{d}\mathbf{x}_0 \\ &= \int p_t(\mathbf{y}|\mathbf{x}_0) p(\mathbf{x}_0|\mathbf{x}_t) \mathrm{d}\mathbf{x}_0 \approx p_t\big(\mathbf{y}|X_0 = \hat{\mathbf{x}}_0(\mathbf{x}_t)\big), \end{aligned}$$

其中我们使用了 $\mathbf{y}$ 仅取决于 $\mathbf{x}_0$（不取决于 $\mathbf{x}_t$），给定 $\mathbf{x}_0$，并且近似在假设后验 $p(\mathbf{x}_0|\mathbf{x}_t)$ 在其均值附近尖锐集中的情况下成立。

自那时起

$$p_t\big(\mathbf{y}|X_0 = \hat{\mathbf{x}}_0(\mathbf{x}_t)\big) = \mathcal{N}\big(\mathbf{y}; \mathcal{A}(\hat{\mathbf{x}}_0(\mathbf{x}_t)), \sigma_{\mathbf{y}}^2 \mathbf{I}\big),$$

我们计算

$$\begin{aligned} \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) &\approx \nabla_{\mathbf{x}_t} \log \mathcal{N}\big(\mathbf{y}; \mathcal{A}(\hat{\mathbf{x}}_0), \sigma_{\mathbf{y}}^2 \mathbf{I}\big) \\ &= -\frac{1}{2\sigma_{\mathbf{y}}^2} \nabla_{\mathbf{x}_t} \big\|\mathbf{y} - \mathcal{A}(\hat{\mathbf{x}}_0)\big\|^2 \\ &= \frac{1}{\sigma_{\mathbf{y}}^2} \left[\mathcal{J}_{\mathcal{A}}\big(\hat{\mathbf{x}}_0(\mathbf{x}_t)\big) \cdot \nabla_{\mathbf{x}_t} \hat{\mathbf{x}}_0(\mathbf{x}_t)\right]^\top \big(\mathbf{y} - \mathcal{A}(\hat{\mathbf{x}}_0(\mathbf{x}_t))\big), \end{aligned}$$

其中 $\mathcal{J}_{\mathcal{A}}\big(\hat{\mathbf{x}}_0(\mathbf{x}_t)\big) := \nabla_{\mathbf{x}_0} \mathcal{A}(\mathbf{x})\big|_{\mathbf{x}=\hat{\mathbf{x}}_0(\mathbf{x}_t)}$ 表示前向算子对其输入的雅克比。该公式通过得分近似流水线传播梯度，反映了测量似然如何随噪声样本 $\mathbf{x}_t$ 的扰动而变化。

对于线性逆问题，这进一步简化为：

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \approx \frac{1}{\sigma_{\mathbf{y}}^2} \left[\mathbf{A} \cdot \nabla_{\mathbf{x}_t} \hat{\mathbf{x}}_0(\mathbf{x}_t)\right]^\top \Big(\mathbf{y} - \mathbf{A}(\hat{\mathbf{x}}_0(\mathbf{x}_t))\Big).$$

大量研究工作通过提出 $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t)$ 的各种近似方法，探讨了基于扩散的反问题求解器。为了全面了解，我们建议读者参考 **daras2024survey** 的综述。

8.5 从强化学习到直接偏好优化的模型对齐

在追求将生成式模型与人类意图对齐的过程中，主流范式是基于人类反馈的强化学习(RLHF)。尽管有效，但 RLHF 是一个复杂且多阶段的过程，可能不稳定。本节介绍了直接偏好优化(*DPO*) (**rafailov2023direct**)，这是一种更为简洁且稳定的方法，能够在不进行显式奖励建模或强化学习的情况下达到相同的目标。随后，我们概述了其扩展至扩散模型的方法——扩散 *DPO* (**wallace2024diffusion**)。

8.5.1 动机：规避 RLHF 的陷阱

对齐的目标是引导一个基础预训练模型（例如，SFT 模型）产生人类更偏好输出。RLHF 分为三个阶段。首先，监督微调（*SFT*）在提示-响应对上训练一个基础模型。其次，奖励建模（*RM*）在包含提示 $\mathbf{c}$ 和配对响应（一个偏好的“胜者” $\mathbf{x}_w$ 和一个不偏好的“败者” $\mathbf{x}_l$ ）的偏好数据上拟合一个模型，学习一个标量 $r(\mathbf{c}, \mathbf{x})$ 使得 $r(\mathbf{c}, \mathbf{x}_w) > r(\mathbf{c}, \mathbf{x}_l)$ 。第三，*RL* 微调优化 SFT 模型（策略 π [5]）使用 PPO (**schulman2017proximal**) 等算法，在最大化 r 的期望奖励的同时，通过 KL 惩罚项对 π 进行正则化，使其接近参考/SFT 分布。

尽管影响深远，该流水线仍存在一些缺陷：强化学习阶段不稳定且计算成本高昂，因为其采用同策略——每次更新都需要从当前模型中重新生成样本；它还要求训练和部署多个大型模型（SFT、奖励模型，有时还包括价值模型）；并且仅优化人类偏好的代理目标，因此奖励模型中的缺陷可能被利用。这引出了一个核心问题：

Question 8.5.1

> 我们能否消除显式的奖励建模和不稳定的强化学习步骤，直接基于偏好数据优化模型？

直接偏好优化（DPO）通过将多阶段的强化学习人类反馈（RLHF）流水线简化为单一的监督式步骤，实现了对齐过程的最优化。DPO 无需训练分离的奖励模型，也无需运行不稳定的强化学习算法（如 PPO），而是直接使用简单的对率损失将策略拟合到偏好对上，同时保持与固定参考模型的接近性。其核心思想是，可以重写带有 KL 正则化的 RLHF 目标，使策略与参考模型之间的对数似然比充当隐式奖励。这种方法在保留对参考策略的相同正则化效果的同时，避免了昂贵的模拟推演和显式的奖励建模。

[5] 策略将提示/历史（状态）映射到响应/动作的分布。

在 Section 8.5.2 中，我们简要回顾了 RLHF 流水线及其对大型奖励模型和强化学习微调的依赖。在 Section 8.5.3 中，我们介绍了 DPO，该方法最初为语言模型提出，能够绕过奖励模型训练并简化对齐微调。最后，在 Section 8.5.4 中，我们将这一思想扩展到扩散模型，提出了 Diffusion-DPO，作为一种在生成式建模情景下实用且稳定的对齐方法。

8.5.2 RLHF：Bradley-Terry 视角

强化学习人类反馈（RLHF）简介。 RLHF 从一个学成的裁判开始：一个奖励模型 r_{ψ} ，它为同一提示 $\mathbf{c}$ 的候选回复分配一个标量偏好得分 $\mathcal{D}$ 。数据集 $(\tilde{\mathbf{x}}, \mathbf{x})$ 包含由标签 y 标注的配对 $\tilde{\mathbf{x}}$ ，该标签表示 $\mathbf{x}$ 是否优于 $y \in \{0, 1\}$ 。标签可以是二元的 $y \in [0, 1]$ ，也可以是由多个评判者聚合得到的软值。训练目标是一个简单的对率损失

$$\begin{aligned}\mathcal{L}_{\mathrm{RM}}(\psi) = -\mathbb{E}_{(\mathbf{c},\tilde{\mathbf{x}},\mathbf{x},y)\sim\mathcal{D}}\Big[&y \log \sigma\big(r_{\psi}(\mathbf{c}, \tilde{\mathbf{x}}) - r_{\psi}(\mathbf{c}, \mathbf{x})\big) \\ &+ (1-y) \log\big(1 - \sigma\big(r_{\psi}(\mathbf{c}, \tilde{\mathbf{x}}) - r_{\psi}(\mathbf{c}, \mathbf{x})\big)\big)\Big],\end{aligned} \tag{8.5.1}$$

其中 $\sigma(u) = 1/(1 + e^{-u})$。在实际应用中，$\mathcal{D}$ 中的偏好对可能来自多种来源：精心整理的回复、不同检查点处的模型快照，或预训练条件扩散模型的生成结果。一种标准约定是将它们以有序格式存储 (winner, loser)。根据这一约定，我们简单地设定 $y = 1$，且 Equation (8.5.1) 退化为特殊情况（其中 $\tilde{\mathbf{x}} = \mathbf{x}^w$ 且 $\mathbf{x} = \mathbf{x}^l$）：

$$\mathcal{L}_{\mathrm{RM}}(\psi) = -\mathbb{E}_{(\mathbf{c},\mathbf{x}_w,\mathbf{x}_l)\sim\mathcal{D}}\Big[\log \sigma\big(r_{\psi}(\mathbf{c}, \mathbf{x}_w) - r_{\psi}(\mathbf{c}, \mathbf{x}_l)\big)\Big]. \tag{8.5.2}$$

布拉德利-特瑞统计观点与 KL 散度联系。 通常将其解释为

$$p_{r_{\psi}}(\tilde{\mathbf{x}} \succ \mathbf{x} | \mathbf{c}) := \sigma\big(r_{\psi}(\mathbf{c}, \tilde{\mathbf{x}}) - r_{\psi}(\mathbf{c}, \mathbf{x})\big)$$

通过 Bradley–Terry (BT) 模型 (**bradley1952rank**)，将两个标量得分转换为获胜概率。该公式突出了两个关键性质：(i) 仅得分的 *差值*重要（因此 $r_{\psi}(\mathbf{c}, \cdot)$ 为平移不变），以及 (ii) 损失会促使预测胜者得分高于败者得分。为了直观理解 (ii)，考虑一对标签为 $y \in \{0, 1\}$ 的样本，并定义

$$\Delta r := r_{\psi}(\mathbf{c}, \tilde{\mathbf{x}}) - r_{\psi}(\mathbf{c}, \mathbf{x}), \quad p := \sigma(\Delta r), \quad \sigma(u) = \tfrac{1}{1+e^{-u}}.$$

每个样本的对率损失为

$$\ell = -\big[y \log p + (1-y)\log(1-p)\big].$$

然后

$$\frac{\partial \ell}{\partial \Delta r} = \sigma(\Delta r) - y.$$

在步长为 $\eta > 0$ 的梯度下降下，得分差距的更新方式为

$$\Delta r \leftarrow \Delta r - \eta\big(\sigma(\Delta r) - y\big).$$

因此，如果 $y = 1$ （“$\tilde{\mathbf{x}}$ 获胜”），则 $\sigma(\Delta r) - 1 \le 0$ ，所以 Δr 增加（胜者上升，败者下降）；如果 $y = 0$ ，Δr 减少。

Equation (8.5.1) 中的每个样本项可被视为观测到的伯努利标签与模型预测胜率之间的交叉熵：

$$-\big[y \log p_{r_\psi} + (1-y)\log(1-p_{r_\psi})\big] = \mathcal{D}_{\mathrm{KL}}\big(\mathrm{Bern}(y)\big\|\mathrm{Bern}(p_{r_\psi})\big) + \mathcal{H}\big(\mathrm{Bern}(y)\big),$$

其中 $\mathcal{H}$ 为目标伯努利分布的熵。对数据集 $\mathcal{D}$ 取平均得到

$$\mathcal{L}_{\mathrm{RM}}(\boldsymbol{\psi}) = \mathbb{E}_{\mathcal{D}}\Big[\mathcal{D}_{\mathrm{KL}}\left(\mathrm{Bern}(y)\big\|\mathrm{Bern}(p_{r_\psi})\right)\Big] + \underbrace{\mathbb{E}_{\mathcal{D}}\big[\mathcal{H}(\mathrm{Bern}(y))\big]}_{\text{independent of } \boldsymbol{\psi}}. \tag{8.5.3}$$

因此，最小化对率损失等价于最小化人类标签的经验伯努利分布与模型预测的伯努利分布之间的 KL 散度。在二分类情况下（$y \in \{0,1\}$），这种等价关系是确切的；对于软标签（$y \in [0,1]$），结果在熵常数偏移的意义上成立。直观上，奖励模型被训练以调整其获胜概率，直到它们与数据集中观察到的经验人类胜率相一致。

从这一点开始，我们采用最常用的约定，其中 $\mathcal{D}$ 以有序格式存储配对数据：$(\mathbf{x}^w, \mathbf{x}^l, \mathbf{c}) \sim \mathcal{D}$。在此约定下，标签始终为 $y = 1$，损失简化为 Equation (8.5.2) 中给出的有序形式，我们将在后续讨论中使用该形式。

KL 约束策略最优化（固定奖励） 通过拟合的奖励 $r := r_{\psi^\times}$（通过 Equation (8.5.2) 训练得到），以及一个条件预训练扩散模型 $p_{\boldsymbol{\phi}^\times}(\mathbf{x}|\mathbf{c})$，RLHF 接着调整一个可学习的策略 $\pi_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{c})$（通常在 $p_{\boldsymbol{\phi}^\times}(\mathbf{x}|\mathbf{c})$ 上微调），使其倾向于产生更高奖励的响应。同时，策略通过使用 $\mathcal{D}_{\mathrm{KL}}$ 惩罚项进行正则化，以使其尽可能接近参考模型（即

预训练扩散模型 $\pi_{\text{ref}}(\mathbf{x}|\mathbf{c}) := p_{\boldsymbol{\phi}^\times}(\mathbf{x}|\mathbf{c})$)。

$$\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{c}\sim p(\mathbf{c})}\Big[\mathbb{E}_{\mathbf{x}\sim\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{c})}\big[r_{\boldsymbol{\psi}}(\mathbf{c},\mathbf{x})\big] - \beta\mathcal{D}_{\text{KL}}\big(\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{c})\big\|\pi_{\text{ref}}(\cdot|\mathbf{c})\big)\Big], \tag{8.5.4}$$

这使得两种力量变得明确：寻找法官偏好的样本，但同时保持接近预训练的参考。

我们注意到，Equation (8.5.2) 中的奖励目标仅使用标注对，且不要求 $\mathcal{D}$ 由参考模型（即预训练的条件扩散模型）生成。虽然这不是必需的，但从接近预期策略的模型中收集数据对可以减少分布偏移，使学成的奖励在实际使用区域更加可靠。

总之，RLHF 分为两个阶段：首先通过最小化 Equation (8.5.2)(equivalently, the expected binary $\mathcal{D}_{\text{KL}}$ in Equation (8.5.3)) 中的损失来拟合奖励 r^*；然后通过求解 Equation (8.5.4) 来优化策略 π^*。

8.5.3 DPO 框架

强化学习人类反馈的桥梁。 Equation (8.5.4) 中的 KL 正则化策略目标对于每个提示 $\mathbf{c}$ 具有简单的闭式解，给定拟合的奖励 $r := r_{\psi^\times}$ ，以如下基于能量的形式表示 (**peters2010relative**)：

$$\pi^*(\mathbf{x}|\mathbf{c}) = \frac{1}{Z(\mathbf{c})}\pi_{\text{ref}}(\mathbf{x}|\mathbf{c})\exp(r(\mathbf{c},\mathbf{x})/\beta), \tag{8.5.5}$$

其中 $\pi_{\text{ref}}(\mathbf{x}|\mathbf{c}) := p_{\phi^\times}(\mathbf{x}|\mathbf{c})$ 和 $Z(\mathbf{c})$ 是保证 $\int \pi^*(\mathbf{x}|\mathbf{c})\,\mathrm{d}\mathbf{x} = 1$ 成立的配分函数。

当 β 较小时，$\exp(r/\beta)$ 变得更尖锐，因此 π^* 集中在高奖励区域：奖励占主导，策略远离 π_{ref}，多样性降低，训练可能变得不稳定或容易出现奖励欺骗。当 β 较大时，$\exp(r/\beta)$ 变平，使 π^* 更接近 π_{ref}：KL 项占主导，更新较为保守，多样性遵循参考分布，但奖励提升有限。

由于我们的目标是直接微调策略(无需训练分离的奖励模型)，Equation (8.5.5) 使我们能够从任意策略 定义一个 *隐式奖励*。我们将在下文引入：

定义一个由求逆 Equation (8.5.5) 激发的隐式奖励。 Equation (8.5.5) 提出一种立即的反转：对于任意策略 π（其支撑集包含于 π_{ref}），定义

$$r_\pi(\mathbf{c},\mathbf{x}) = \beta\log\frac{\pi(\mathbf{x}|\mathbf{c})}{\pi_{\text{ref}}(\mathbf{x}|\mathbf{c})} + \beta\log Z(\mathbf{c}). \tag{8.5.6}$$

然后，Equation (8.5.5) 在 π 代替 π^* 时成立，即 π 将是 Equation (8.5.4) 针对奖励函数 r_π 的优化器。从这个意义上说，r_π 是一个隐式（策略诱导的）奖励：它仅能确定到依赖于提示的常数 $\beta\log Z(\mathbf{c})$，而在任何成对比较（如 BT 模型中）中该常数会消失。

$$r_\pi(\mathbf{c},\mathbf{x}_w) - r_\pi(\mathbf{c},\mathbf{x}_l) = \beta\Big(\log\frac{\pi(\mathbf{x}_w|\mathbf{c})}{\pi_{\text{ref}}(\mathbf{x}_w|\mathbf{c})} - \log\frac{\pi(\mathbf{x}_l|\mathbf{c})}{\pi_{\text{ref}}(\mathbf{x}_l|\mathbf{c})}\Big).$$

这种抵消正是使常数在偏好学习中无关紧要的原因，并直接导致了基于对数概率差异的 DPO 损失。

DPO 的训练损失。 将隐式奖励 Equation (8.5.6) 代入 Equation (8.5.2) 的 BT 模型，针对在相同提示 $\mathbf{c}$ 下的标注数据对 $(\mathbf{x}_w,\mathbf{x}_l)$。常数 $\log Z(\mathbf{c})$ 在胜者与败者

之间相互抵消，得到关于对数概率差的单一逻辑损失目标：

$$\mathcal{L}_{\text{DPO}}(\boldsymbol{\theta};\pi_{\text{ref}}) = -\mathbb{E}_{(\mathbf{c},\mathbf{x}_w,\mathbf{x}_l)\sim\mathcal{D}}\left[\log\sigma\Big(\beta\big(\log\frac{\pi_{\boldsymbol{\theta}}(\mathbf{x}_w|\mathbf{c})}{\pi_{\text{ref}}(\mathbf{x}_w|\mathbf{c})} - \log\frac{\pi_{\boldsymbol{\theta}}(\mathbf{x}_l|\mathbf{c})}{\pi_{\text{ref}}(\mathbf{x}_l|\mathbf{c})}\big)\Big)\right].$$

用文字表述：DPO 提升了胜者相对于败者的（经过温度缩放的）优势，该优势以相对于参考模型的对数似然提升差异来衡量：

$$-\log\sigma\left(\beta\big[\text{log-ratio difference of } \tfrac{\pi_{\boldsymbol{\theta}}}{\pi_{\text{ref}}} \text{ at } \mathbf{x}_w \text{ vs. } \mathbf{x}_l\big]\right).$$

这通过一个单一且稳定的最大似然风格阶段实现了 RLHF 的目标，而无需训练显式的奖励模型。

8.5.4 扩散-DPO

为什么朴素的 DPO 对扩散模型无效？ 在扩散模型中评估样本似然 $\pi_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{c})$ 需要微分方程求解的瞬时变量变换公式（漂移项的散度）（见 Equation (4.2.7)）[6]，which is computationally intensive. Moreover, differentiating through the entire sampling trajectory can suffer from vanishing or exploding gradients. To avoid these issues, Diffusion-DPO works at the 路径级别。我们以离散时间扩散模型（例如，DDPM）为例进行说明；连续时间扩散模型的情况类似。

路径隐式奖励的定义。 令在反向时间马尔可夫链下的轨迹为 $\mathbf{x}_{0:T} := (\mathbf{x}_T, \ldots, \mathbf{x}_0)$，其条件分布为 $\pi(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})$。其中，$\mathbf{x}_T$ 表示来自先验（最高噪声）的样本，$\mathbf{x}_0$ 是数据空间中的干净输出。由于扩散模型的生成过程沿着完整的降噪路径进行，因此将偏好从最终输出扩展到整个轨迹是自然的。因此，我们为每条轨迹分配一个奖励 $R(\mathbf{c}, \mathbf{x}_{0:T})$，当其仅依赖于 $\mathbf{x}_0$ 时，该奖励退化为终点奖励，但也可以捕捉路径上的累积效应。

我们将 Equation (8.5.4) 中的样本级 KL 替换为路径级 KL，表示为：

$$\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{c}\sim p(\mathbf{c})}\Big[\underbrace{\mathbb{E}_{\mathbf{x}_{0:T}\sim\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{c})}[R(\mathbf{c},\mathbf{x}_{0:T})]}_{\text{reward over paths}} - \beta\mathcal{D}_{\text{KL}}\big(\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{c})\big\|\pi_{\text{ref}}(\cdot|\mathbf{c})\big)\Big],$$

其中 $\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{c})$ 和 $\pi_{\text{ref}}(\cdot|\mathbf{c})$ 是路径的 分布。其目标是最大化反向过程 $\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{c})$ 的奖励，同时匹配原始参考反向过程 $\pi_{\text{ref}}(\cdot|\mathbf{c})$ 的分布。

[6]在离散时间扩散模型（例如 DDPM）中，评估 $\pi_{\boldsymbol{\theta}}(\mathbf{x}_0|\mathbf{c})$ 需要对潜在的反向轨迹 $\mathbf{x}_{1:T}$ 进行边缘化。

对于每个提示 $\mathbf{c}$，优化器具有简单的基于能量的形式

$$\pi^*(\mathbf{x}_{0:T}|\mathbf{c}) = \frac{1}{Z(\mathbf{c})}\pi_{\text{ref}}(\mathbf{x}_{0:T}|\mathbf{c}) \exp\big(R(\mathbf{c}, \mathbf{x}_{0:T})/\beta\big), \tag{8.5.7}$$

其中 $Z(\mathbf{c})$ 为归一化器。对 Equation (8.5.7) 求逆促使我们定义任意策略 π 的一个 隐式路径奖励：

$$R_\pi(\mathbf{c}, \mathbf{x}_{0:T}) := \beta \log \frac{\pi(\mathbf{x}_{0:T}|\mathbf{c})}{\pi_{\text{ref}}(\mathbf{x}_{0:T}|\mathbf{c})} + \beta \log Z(\mathbf{c}),$$

其常数 $\beta \log Z(\mathbf{c})$ 与成对比较无关。

从路径隐式奖励到 DPO。 将 Bradley–Terry 模型应用于在相同提示 $\mathbf{c}$ 下的标注对 $(\mathbf{x}_0^w, \mathbf{x}_0^l)$ 的 路径，并使用标准逻辑回归损失函数：

$$\begin{aligned} \mathcal{L}_{\text{Diff-DPO}}(\boldsymbol{\theta}; \pi_{\text{ref}}) &:= -\mathbb{E}_{(\mathbf{c}, \mathbf{x}_0^w, \mathbf{x}_0^l)\sim\mathcal{D}}\left[\log \sigma\big(\Delta R(\mathbf{c}; \boldsymbol{\theta})\big)\right], \quad \text{where} \\ \Delta R(\mathbf{c}; \boldsymbol{\theta}) &:= \underbrace{\mathbb{E}_{\mathbf{x}_{1:T}^w \sim \pi_{\boldsymbol{\theta}}(\cdot|\mathbf{x}_0^w, \mathbf{c})}\Big[R_{\pi_{\boldsymbol{\theta}}}\big(\mathbf{c}, (\mathbf{x}_0^w, \mathbf{x}_{1:T}^w)\big)\Big]}_{\text{winner path expectation}} \\ &\quad - \underbrace{\mathbb{E}_{\mathbf{x}_{1:T}^l \sim \pi_{\boldsymbol{\theta}}(\cdot|\mathbf{x}_0^l, \mathbf{c})}\Big[R_{\pi_{\boldsymbol{\theta}}}\big(\mathbf{c}, (\mathbf{x}_0^l, \mathbf{x}_{1:T}^l)\big)\Big]}_{\text{loser path expectation}}. \end{aligned} \tag{8.5.8}$$

此处，期望 $\mathbb{E}_{\mathbf{x}_{1:T}\sim\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{x}_0, \mathbf{c})}[\cdot]$ 表示：给定数据集中固定的终点 $\mathbf{x}_0$（例如，胜者 $\mathbf{x}_0^w$），我们对模型诱导的条件路径分布（即反向时间轨迹的后验分布）下潜在的降噪轨迹 $\mathbf{x}_{1:T}$ 取期望，这些轨迹在内核 $\pi_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})$ 的作用下可能生成 $\mathbf{x}_0$。由于这些中间状态是未观测到的，我们对所有此类轨迹的路径奖励取平均。

然而，Equation (8.5.8) 有三个实际原因使其不切实际：

1. **端点条件会导致难以处理的路径后验。** 术语 $\mathbb{E}_{\pi_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}|\mathbf{x}_0, \mathbf{c})}[\cdot]$ 对于必须经过 $\mathbf{x}_0$ 的反向路径进行平均，而采样器运行时 $\mathbf{x}_T \to \cdots \to \mathbf{x}_0$ 则没有此约束。对端点进行条件化会生成一个扩散桥后验，通常没有闭式解且采样成本高昂。

2. **嵌套的、$\boldsymbol{\theta}$ -耦合期望。** 损失 $-\log\sigma(\Delta R(\mathbf{c}; \boldsymbol{\theta}))$ 与

$$\Delta R = \mathbb{E}_{\text{paths}|\mathbf{x}_0^w, \mathbf{c}}[R_{\pi_{\boldsymbol{\theta}}}] - \mathbb{E}_{\text{paths}|\mathbf{x}_0^l, \mathbf{c}}[R_{\pi_{\boldsymbol{\theta}}}]$$

既有路径联合分布，又有被积函数 $R_{\pi_{\boldsymbol{\theta}}}$ 依赖于 $\boldsymbol{\theta}$。因此 $\nabla_{\boldsymbol{\theta}}$ 必须通过采样分布进行求导，导致 REINFORCE/路径式耦合以及高方差梯度。

3. **长链、大和以及昂贵的反向传播**。在 $R_{\pi_{\boldsymbol{\theta}}}(\mathbf{c}, \mathbf{x}_{0:T})$ 中，计算

$$\beta\left[\log \pi_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{c}) - \log \pi_{\text{ref}}(\mathbf{x}_{0:T}|\mathbf{c})\right]$$

需要每步的 $\mathcal{O}(T)$ 对数密度，包含 $T \sim 10^2$ –10^3 ，针对策略 $\pi_{\boldsymbol{\theta}}$ 和参考 π_{ref} ，以及胜者/败者路径。对这些随机链（或桥采样器）进行反向传播在内存和计算上开销巨大且可能不稳定；在每对样本中重复多次，并在所有三元组间重复此过程，使训练超出实际预算。

面向一种易处理的代理模型的 Equation (8.5.8)。 为了使其可计算，我们应用了一个关键的数学洞察。通过利用扩散模型的性质并应用詹森不等式，我们可以优化该损失的一个易处理的上界。这使得问题从评估整个路径的似然转变为评估路径中各个单步转移的期望：

由于 $-\log\sigma(\cdot)$ 是凸函数，Jensen 不等式通过将内部期望移到对数之外，给出了一个上界：

$$\begin{aligned}&\mathcal{L}_{\text{Diff-DPO}}(\boldsymbol{\theta}; \pi_{\text{ref}})\\ &\leq -\mathbb{E}_{(\mathbf{c},\mathbf{x}_0^w,\mathbf{x}_0^l)\sim\mathcal{D}}\mathbb{E}_{\substack{\mathbf{x}_{1:T}^w\sim\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{x}_0^w,\mathbf{c})\\ \mathbf{x}_{1:T}^l\sim\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{x}_0^l,\mathbf{c})}}\left[\log\sigma\Big(\beta\big(R(\mathbf{c},\mathbf{x}_{0:T}^w) - R(\mathbf{c},\mathbf{x}_{0:T}^l)\big)\Big)\right].\end{aligned}$$

利用隐式奖励恒等式 $R_{\pi_{\boldsymbol{\theta}}} = \beta\log\frac{\pi_{\boldsymbol{\theta}}}{\pi_{\text{ref}}} + \beta\log Z(\mathbf{c})$ 以及胜者与败者之间的常数抵消，该界变为

$$\mathcal{L}_{\text{Diff-DPO}}(\boldsymbol{\theta}; \pi_{\text{ref}}) \leq -\mathbb{E}\left[\log\sigma\Big(\beta\big(\log\frac{\pi_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}^w|\mathbf{c})}{\pi_{\text{ref}}(\mathbf{x}_{0:T}^w|\mathbf{c})} - \log\frac{\pi_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}^l|\mathbf{c})}{\pi_{\text{ref}}(\mathbf{x}_{0:T}^l|\mathbf{c})}\big)\Big)\right]. \quad (8.5.9)$$

易处理的代理（分步形式）. 我们现在利用反向过程的马尔可夫性质来分解 $\mathcal{L}_{\text{Diff-DPO}}$ 的上界。这使得我们可以将路径级偏好表示为每一步贡献的和，从而将不可处理的路径损失转化为易处理的单步估计量。所得形式简化为 DSM 风格的均方误差差值。具体而言，对于反向链，

$$\pi_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{c}) = \pi_{\boldsymbol{\theta}}(\mathbf{x}_T|\mathbf{c})\prod_{t=1}^{T}\pi_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}),$$

$$\pi_{\text{ref}}(\mathbf{x}_{0:T}|\mathbf{c}) = \pi_{\text{ref}}(\mathbf{x}_T|\mathbf{c})\prod_{t=1}^{T}\pi_{\text{ref}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}).$$

因此

$$\frac{\pi_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{c})}{\pi_{\text{ref}}(\mathbf{x}_{0:T}|\mathbf{c})} = \frac{\pi_{\boldsymbol{\theta}}(\mathbf{x}_T|\mathbf{c})}{\pi_{\text{ref}}(\mathbf{x}_T|\mathbf{c})} \prod_{t=1}^{T} \frac{\pi_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})}{\pi_{\text{ref}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})}.$$

如果在时间 T 时两个模型的先验相同，$\pi_{\boldsymbol{\theta}}(\mathbf{x}_T|\mathbf{c}) = \pi_{\text{ref}}(\mathbf{x}_T|\mathbf{c})$ ，那么第一个因子等于 1 ，取对数后得到

$$\log \frac{\pi_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{c})}{\pi_{\text{ref}}(\mathbf{x}_{0:T}|\mathbf{c})} = \sum_{t=1}^{T} \log \frac{\pi_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})}{\pi_{\text{ref}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})}.$$

因此，Equation (8.5.9) 中的界可以写成

$$\mathcal{L}_{\text{Diff-DPO}}(\boldsymbol{\theta}; \pi_{\text{ref}}) \leq -\mathbb{E}\left[\log \sigma\Big(\beta \sum_{t=1}^{T} \Delta_t\Big)\right],$$

其中每个步骤的贡献为

$$\Delta_t = \log \frac{\pi_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}^w|\mathbf{x}_t^w, \mathbf{c})}{\pi_{\text{ref}}(\mathbf{x}_{t-1}^w|\mathbf{x}_t^w, \mathbf{c})} - \log \frac{\pi_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}^l|\mathbf{x}_t^l, \mathbf{c})}{\pi_{\text{ref}}(\mathbf{x}_{t-1}^l|\mathbf{x}_t^l, \mathbf{c})}.$$

为了获得一个易处理的估计量，我们应用单步 Jensen 上界：采样 $t \sim \mathcal{U}\{1, \ldots, T\}$ （每对训练样本对应一个时间步）并按 T 缩放。这得到了

$$-\log \sigma\Big(\beta \sum_{t=1}^{T} \Delta_t\Big) \leq \mathbb{E}_t\left[-\log \sigma\big(\beta T \Delta_t\big)\right].$$

因此，最终目标是一个预期的每步代理。

$$\mathcal{L}_{\text{Diff-DPO}}(\boldsymbol{\theta}; \pi_{\text{ref}}) \leq -\mathbb{E}_{\substack{(\mathbf{c}, \mathbf{x}_0^w, \mathbf{x}_0^l) \sim \mathcal{D} \\ t \sim \mathcal{U}\{1, \ldots, T\}}}\left[\log \sigma\big(\beta T \Delta_t\big)\right],$$

这将原始的路径损失降低为一个易处理的单步上界估计量。

对于扩散模型中使用的高斯逆条件（以 $\boldsymbol{\epsilon}$ -prediction 为例），

$$\log \frac{\pi_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})}{\pi_{\text{ref}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})} = \text{const} - \lambda_t\Big(\underbrace{\big\|\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \mathbf{c}) - \boldsymbol{\epsilon}_t\big\|^2}_{\text{policy}} - \underbrace{\big\|\hat{\boldsymbol{\epsilon}}_{\text{ref}}(\mathbf{x}_t, t, \mathbf{c}) - \boldsymbol{\epsilon}_t\big\|^2}_{\text{reference}}\Big),$$

其中 $\lambda_t > 0$ 吸收了噪声调度因子。因此，每个时间步的贡献与在切片 t 处策略与参考之间的 MSE 差异成比例。

为书写简便，对任意 $\mathbf{x}_t$ 定义：

$$\Delta\mathrm{MSE}(\mathbf{x}_t) := \left\|\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \mathbf{c}) - \boldsymbol{\epsilon}\right\|^2 - \left\|\hat{\boldsymbol{\epsilon}}_{\mathrm{ref}}(\mathbf{x}_t, t, \mathbf{c}) - \boldsymbol{\epsilon}\right\|^2.$$

这促使我们采用以下实用的替代方案来表示 $\mathcal{L}_{\text{Diff-DPO}}(\boldsymbol{\theta}; \pi_{\mathrm{ref}})$：

$$\tilde{\mathcal{L}}_{\text{Diff-DPO}}(\boldsymbol{\theta}; \pi_{\mathrm{ref}}) := \mathop{\mathbb{E}}_{\substack{(\mathbf{c}, \mathbf{x}_0^w, \mathbf{x}_0^l) \sim \mathcal{D} \\ t \sim \mathcal{U}\{1, \ldots, T\}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})}} \Big[w(t) \big(\Delta\mathrm{MSE}(\mathbf{x}_t^w) - \Delta\mathrm{MSE}(\mathbf{x}_t^l) \big) \Big],$$

其中 $\mathbf{x}_t^w = \alpha_t \mathbf{x}_0^w + \sigma_t \boldsymbol{\epsilon}$ 和 $\mathbf{x}_t^l = \alpha_t \mathbf{x}_0^l + \sigma_t \boldsymbol{\epsilon}$ 共享相同的噪声 $\boldsymbol{\epsilon}$ 以实现方差减小，且 $w(t) > 0$ 用于收集时间权重（例如 $w(t) \propto \lambda_t$ ）。

直观上，最小化 $\tilde{\mathcal{L}}_{\text{Diff-DPO}}$ 会提高模型在胜者上的预测准确率，相对于参考值而言，同时降低在败者上的预测准确率。由于改进始终是相对于同一时间步的 π_{ref} 来衡量的，因此策略会被引导至类似胜者的降噪轨迹，而远离类似败者的轨迹，同时保持与参考值的锚定关系。

8.6 闭幕词

本章将我们的关注点从基础原理转向了可控生成这一实际挑战。我们基于条件得分的贝叶斯分解，建立了一个统一的引导框架，巧妙地将生成过程分离为无条件方向和导向项。

我们看到了这一原则在多种强大技术中的体现。我们介绍了需要专门训练的方法，例如使用外部分类器的分类器引导（CG），以及更高效的无分类器引导（CFG），后者在一个模型中同时学习条件得分和无条件得分。我们还探讨了灵活的无需训练的引导方法，这些方法通过从任意损失函数定义代理似然，在推理阶段引导预训练模型，从而实现从艺术控制到解决逆问题等应用，且无需重新训练。

除了简单的条件化之外，我们深入探讨了将模型输出与人类偏好对齐这一细微任务。在回顾了标准但复杂的强化学习人类反馈（RLHF）流水线后，我们提出了直接偏好优化（DPO）及其新颖的变体——扩散 DPO，作为更直接且稳定的替代方案。该方法通过直接从偏好数据中推导损失，巧妙地避免了显式奖励模型和强化学习的需求。

通过这些技术，我们已经构建了一个强大的工具集来引导生成过程。然而，一个重大的实际障碍仍未解决：迭代采样过程本身带来的巨大计算成本和延迟。在解决了生成内容的问题后，我们现在转向同样重要的问题：如何加快生成速度。下一章将直接应对这一挑战：

1. 我们将利用采样等价于求解常微分方程这一洞察，探索设计精妙的数值求解器，以大幅减少所需的步骤数量。

2. 我们将研究一系列具有影响力的方法，包括 DDIM、DEIS 以及 DPM-Solver 系列，这些方法通过将采样速度提升几个数量级，极大地提高了扩散模型的实际应用性。

9

高效的采样求解器

扩散模型的生成过程，即从噪声映射到数据样本的过程，其数学本质等价于求解随机微分方程（SDE）或其对应的常微分方程（ODE）。该过程本质上较为缓慢，因为它依赖于数值求解器，通过大量微小的积分步骤来近似求解轨迹（详见 Chapter A 的简要介绍）。因此，加速推理已成为一项核心研究目标。总体而言，现有方法主要分为两类：

- **无需训练的方法**：本章的重点。这些方法开发先进的数值求解器，以在不进行额外训练的情况下提高扩散采样的效率。
- **基于训练的方法**：在 Chapters 10 and 11 中有介绍。这些技术要么将预训练的扩散模型提炼为快速生成器，要么直接学习常微分方程（ODE）的流映射（解），从而仅需少量采样步骤即可完成。

基于 SDE 的采样器（例如 Euler–Maruyama）由于随机性可能产生更多样化的样本，但通常需要更多的步骤 (**xu2023restart**)。在此我们专注于基于 ODE 的生成，其原理可自然推广至 SDE 情景。

9.1 序言

9.1.1 扩散模型的高级求解器

Score SDE 框架 (**song2020score**) 通过严格地将离散时间扩散过程和 ELBO 公式 (**sohl2015deep**; **ho2020denoising**) 与生成式建模的连续时间 SDE/ODE 视角统一起来，建立了关键的基础。这种合一不仅提供了理论上的清晰性，还使得基于数值积分的高效采样算法能够被合理地开发。

具体而言，假设我们有一个预训练的扩散模型 $\mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$（其具有如 Section 6.3 所示的另外三种等价表达式）。在这种情况下，采样过程可被视为求解带有初始条件 $\mathbf{x}(T) \sim p_{\text{prior}}$ 的 PF-ODE，从 $t = T$ 逆时间方向积分至 $t = 0$。

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}(t), t) - \frac{1}{2} g^2(t) \underbrace{\nabla_{\mathbf{x}} \log p_t(\mathbf{x}(t))}_{\approx \mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}(t), t)}.$$

该常微分方程直接与前向随机过程相关。

$$\mathrm{d}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}(t),$$

展示了生成（反向时间）与加噪（正向时间）动力学之间的连续时间关联。

PF-ODE 的精确解可以等价地表示为积分形式：

$$\begin{aligned} \boldsymbol{\Psi}_{T\to 0}(\mathbf{x}(T)) &= \mathbf{x}(T) + \int_T^0 \left[f(\tau)\mathbf{x}(\tau) - \frac{1}{2} g^2(\tau) \nabla_{\mathbf{x}} \log p_\tau(\mathbf{x}(\tau)) \right] \mathrm{d}\tau \\ &\approx \mathbf{x}(T) + \int_T^0 \left[f(\tau)\mathbf{x}(\tau) - \frac{1}{2} g^2(\tau) \mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}(\tau), \tau) \right] \mathrm{d}\tau \\ &=: \widetilde{\boldsymbol{\Psi}}_{T\to 0}(\mathbf{x}(T)). \end{aligned} \tag{9.1.1}$$

此处，$\boldsymbol{\Psi}_{s\to t}(\mathbf{x})$ 表示 *oracle* PF-ODE 的流映射，将时间 s 时的状态 $\mathbf{x}$ 映射到其在时间 t 时的演化状态（见 Equation (4.1.9)）。相比之下，$\widetilde{\boldsymbol{\Psi}}_{s\to t}(\mathbf{x})$ 表示由真实扩散模型 $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ 替换为学成近似 $\mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}, t)$ 后得到的 *empirical* PF-ODE 的流映射。因此，$\widetilde{\boldsymbol{\Psi}}_{s\to t} \approx \boldsymbol{\Psi}_{s\to t}$。

由于 $\widetilde{\boldsymbol{\Psi}}_{s\to t}$ 的积分形式无法以闭式求解，采样必须依赖于*数值求解器*。这些方法通过离散化时间，并将连续积分替换为局部漂移评估的有限求和，从而追踪近似轨迹。基于求解器的积分近似被称为*无需训练*的算法，用于快速扩散采样，因为它们旨在直接从冻结的预训练得分模型 $\mathbf{s}_{\boldsymbol{\phi}^\times}$ 近似 PF-ODE 解，而无需任何

额外的学习过程。

下面我们首先详细说明数值求解器的通用概念，并介绍后续将使用的符号。

连续轨迹的离散近似。 令 $\mathbf{x}_T$ 表示时间 T 时的初始状态，并考虑一个递减的划分

$$T = t_0 > t_1 > \cdots > t_M = 0. \tag{9.1.2}$$

从 $\tilde{\mathbf{x}}_{t_0} = \mathbf{x}_T \sim p_{\text{prior}}$ 开始，求解器生成一个序列 $\{\tilde{\mathbf{x}}_{t_i}\}_{i=0}^{M}$，该序列理想情况下近似于经验 PF-ODE 流 $\widetilde{\boldsymbol{\Psi}}_{T\to t_i}(\mathbf{x}_T)$，而 $\widetilde{\boldsymbol{\Psi}}_{T\to t_i}(\mathbf{x}_T)$ 本身是关于原始映射 $\boldsymbol{\Psi}_{T\to t_i}(\mathbf{x}_T)$ 的代理。每一步数值迭代通过该经验速度场推进状态，最终迭代结果 $\tilde{\mathbf{x}}_{t_M}$ 作为在 $t = 0$ 处干净样本 $\mathbf{x}_0$ 的估计值。

9.1.2 文献中设计求解器的通用框架

zhang2022fast 阐述了设计与扩散模型相关的 PF-ODE 数值求解器的三个实用原则。

一、半线性结构。 尽管 **song2020score** 为一般的漂移 $\mathbf{f}(\mathbf{x}(t), t)$ 奠定了基础，在大多数调度器公式中，漂移以线性形式被具体实现。

$$\mathbf{f}(\mathbf{x}, t) := f(t)\,\mathbf{x}, \quad f : \mathbb{R} \to \mathbb{R},$$

这导致了具有 半线性结构的 PF-ODE：

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \underbrace{f(t)\mathbf{x}(t)}_{\text{linear part}} - \underbrace{\tfrac{1}{2}g^2(t)\mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}(t), t)}_{\text{nonlinear part}}. \tag{9.1.3}$$

这种线性–非线性分解在 $\mathbf{x}$ 中有利于准确率和稳定性，并促使开发专用积分器（见下方 Equation (9.1.6) 附近的讨论）(**hochbruck2005explicit**; **hochbruck2010exponentia**

二、超越得分的参数化。 如 $t \to 0$ 所示，真实得分 $\nabla_{\mathbf{x}} \log p_t(\cdot)$ 可能变化非常迅速（例如，当 p_{data} 集中在低维流形附近时）(**kim2022soft**)。这使得直接尝试逼近得分的神经网络 $\mathbf{s}_{\boldsymbol{\phi}^\times}$ 难以保持准确性。

为了理解这一点，回忆一下预言机关系（见 Equation (6.3.1)）

$$\boldsymbol{\epsilon}^*(\mathbf{x}_t, t) = -\sigma_t \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t),$$

其中 $\boldsymbol{\epsilon}^*(\mathbf{x}_t, t) = \mathbb{E}[\boldsymbol{\epsilon}|\mathbf{x}_t]$ 是 oracle 噪声，(α_t, σ_t) 是扰动核 $\mathbf{x}_t|\mathbf{x}_0 \sim \mathcal{N}(\alpha_t \mathbf{x}_0, \sigma_t^2 I)$ 的均值和标准差，通过 Equation (4.4.2) 与 $f(t), g(t)$ 相关联。根据 L^2 中的正交性性质，

$$\mathbb{E}\,\|\boldsymbol{\epsilon}\|_2^2 = \mathbb{E}\,\|\boldsymbol{\epsilon}^*\|_2^2 + \mathbb{E}\,\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^*\|_2^2 \quad \Rightarrow \quad \mathbb{E}\,\|\boldsymbol{\epsilon}^*\|_2^2 \leq \mathbb{E}\,\|\boldsymbol{\epsilon}\|_2^2 = D.$$

因此，预言噪声预测器始终有界，但得分的增长方式类似于

$$\mathbb{E}\,\|\mathbf{s}^*(\mathbf{x}_t, t)\|_2^2 = \sigma_t^{-2}\,\mathbb{E}\,\|\boldsymbol{\epsilon}^*(\mathbf{x}_t, t)\|_2^2 \leq \frac{D}{\sigma_t^2}.$$

因此，当 $t \to 0$ 时，得分可能以 $1/\sigma_t^2$ 的速率发散，而噪声预测器则保持有界。由于神经网络只能近似光滑增长的函数，得分预测往往数值不稳定且精度较低，这反过来在依赖预训练模型作为漂移项时会损害数值 PF-ODE 求解器的性能。

因此，一种广泛使用的方法是预测噪声 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}$（或其变体，如 $\mathbf{x}$ - 或 $\mathbf{v}$ -预测），它具有稳定的有界性，并与得分之间存在简单的闭式关系：

$$\mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}, t) = -\frac{1}{\sigma_t}\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}, t).$$

将此关系代入 PF-ODE（参见 Equation (6.3.2)）得

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \underbrace{f(t)\mathbf{x}(t)}_{\text{linear part}} + \underbrace{\tfrac{1}{2}\tfrac{g^2(t)}{\sigma_t}\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}(t), t)}_{\text{nonlinear part}}. \tag{9.1.4}$$

这种参数化方法被现代 PF-ODE 求解器普遍采用。

三、半线性 PF-ODEs 的指数积分方法。 对于 Equation (9.1.4) 中的半线性结构，Equation (9.1.6) 中的 *指数积分公式*提供了解的确切表示。为了说明这一点，设 $\mathbf{x}_s$ 表示起始时间 s 的状态，令 $t \in [0, s]$ 为终止时间。[1].

为清晰起见，将 Equation (9.1.4) 的非线性部分写为

$$\mathbf{N}(\mathbf{x}(t), t) := \frac{1}{2}\frac{g^2(t)}{\sigma_t}\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}(t), t).$$

[1]此处，s 为起始时间，t 为终止时间，因此采样以 $s > t$ 为基准向后积分。

常微分方程可进一步写为

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} - \underbrace{f(t)\mathbf{x}(t)}_{\text{linear part}} = \underbrace{\mathbf{N}(\mathbf{x}(t), t)}_{\text{nonlinear part}} . \tag{9.1.5}$$

为了分离线性项，我们引入了指数积分器

$$\mathcal{E}(s \to t) \coloneqq \exp\Big(\int_s^t f(u)\,\mathrm{d}u\Big),$$

并将微分方程两边同乘以其逆 $\mathcal{E}(t \to s)$ 。根据乘法法则，

$$\mathcal{E}^{-1}(s \to t)\left(\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} - f(t)\mathbf{x}(t)\right) = \frac{\mathrm{d}}{\mathrm{d}t}\Big[\,\mathcal{E}^{-1}(s \to t)\mathbf{x}(t)\,\Big].$$

因此，方程变为

$$\frac{\mathrm{d}}{\mathrm{d}t}\Big[\,\mathcal{E}^{-1}(s \to t)\mathbf{x}(t)\,\Big] = \mathcal{E}^{-1}(s \to t)\,\mathbf{N}(\mathbf{x}(t), t).$$

从 s 积分到 t，然后乘回 $\mathcal{E}(s \to t)$ 即得解：

$$\widetilde{\mathbf{\Psi}}_{s\to t}(\mathbf{x}_s) = \underbrace{\mathcal{E}(s \to t)\mathbf{x}_s}_{\text{linear part}} + \frac{1}{2}\int_s^t \frac{g^2(\tau)}{\sigma_\tau}\mathcal{E}(\tau \to t)\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\tau, \tau)\,\mathrm{d}\tau. \tag{9.1.6}$$

我们建议读者参阅 Section A.1.3 以了解推导的完整细节。

为了说明为何在少步采样（大 Δs）时，Equation (9.1.6) 中的指数-积分形式优于 Equation (9.1.4)，我们比较它们的一步更新。利用常数变易法，$\mathcal{E}(s \to s - \Delta s) = e^{-f(s)\Delta s}$ 并将 $\mathbf{N}(\mathbf{x}(\tau), \tau) \approx \mathbf{N}(\mathbf{x}_s, s)$ 在 $\tau \in [s - \Delta s, s]$ 时冻结，Equation (9.1.6) 的指数-欧拉更新为

$$\mathbf{x}_{s-\Delta s}^{\text{Exp-Euler}} = \underbrace{e^{-f(s)\Delta s}\mathbf{x}_s}_{\text{linear part}} + \underbrace{\frac{e^{-f(s)\Delta s} - 1}{f(s)}\,\mathbf{N}(\mathbf{x}_s, s)}_{\text{nonlinear part}}, \tag{9.1.7}$$

其中自然极限为 $\big(e^{-f\Delta s} - 1\big)/f \to -\Delta s$ ，当 $f \to 0$ 时。此处线性因子 $e^{-f(s)\Delta s}$ 被确切计算（无近似）。

相比之下，对 $f(\tau)\mathbf{x}_\tau - \mathbf{N}(\mathbf{x}_\tau, \tau) \approx f(s)\mathbf{x}_s - \mathbf{N}(\mathbf{x}_s, s)$ 用 $\tau \in [s - \Delta s, s]$ 近

似得到 Equation (9.1.4) 的欧拉法步进：

$$\mathbf{x}_{s-\Delta s}^{\text{Euler}} = \mathbf{x}_s - \Delta s\,[\,f(s)\,\mathbf{x}_s + \mathbf{N}(\mathbf{x}_s, s)\,] = \underbrace{(1 - f(s)\Delta s)\,\mathbf{x}_s}_{\text{linear part}} - \underbrace{\Delta s\,\mathbf{N}(\mathbf{x}_s, s)}_{\text{nonlinear part}}. \tag{9.1.8}$$

Equation (9.1.8) 中的线性因子是 Equation (9.1.7) 中指数函数的一阶泰勒近似。

$$e^a = 1 + a + \tfrac{a^2}{2} + \tfrac{a^3}{6} + \cdots, \quad a := -f(s)\Delta s,$$

因此，间隙为 $e^a - (1+a) = \frac{a^2}{2} + \mathcal{O}(a^3)$ 。一旦 $|f(s)|\Delta s$ 不是微小的（即步长 Δs 不够小），欧拉的线性更新 $(1+a)\mathbf{x}_s$ 会以 $a/2$ 阶的相对误差对真实的因子 $e^a\mathbf{x}_s$ 进行错误缩放。这纯粹是由离散化引起的线性失真。指数-欧拉步骤通过应用确切的线性因子避免了这一问题，尤其是在采用大步长时尤为重要。

9.1.3 PF-ODE 数值求解方法

扩散模型的数值求解器大致可分为两类。

时间步进方法。 这类方法将时间区间 $[0, T]$ 离散化，并利用为高效设计的各种数值积分方案来近似 PF-ODE。我们以最具基础性、原理性和广泛应用的几种方法为例进行说明：

降噪扩散隐式模型（DDIM）. DDIM，如 Section 9.2(with its update form already appearing in Section 4.1.4) 所述，是最早期的快速采样器之一，适用于扩散模型。最初从变分视角提出，它引入了一个非马尔可夫前向过程族，其边缘分布与原始扩散过程匹配，从而实现确定性反向过程和灵活的步长跳过。然而，从常微分方程（ODE）视角来看，DDIM 可以更直接地理解：它对应于对指数积分公式 Equation (9.1.6) 应用单次指数-欧拉步骤，即在积分内部将扩散模型项近似为常数，从而得到 Equation (9.1.7) 中的更新公式。

扩散指数积分采样器（DEIS）. DEIS (**zhang2022fast**)，首次在 Section 9.3 中提出，通过应用指数积分法利用了 PF-ODE 的半线性结构。其核心思想是通过积分因子精确处理线性部分，仅对非线性积分项进行近似。与欧拉方法假设指数积分公式内部被积函数为常数不同，DEIS 重用了轨迹上先前估计点的历史信息。具体而言，它对过去评估点进行高阶插值（即 拉格朗日多项式），并用该插

值结果近似下一步的积分值。从几何角度看，这种多项式插值比常数近似更能准确捕捉轨迹的曲率，从而实现更高阶的准确率，并提升大步长下的稳定性。

将过去的评估结果用于锚定下一次更新（使得每一步仅需一次新的模型调用）的方法被称为 *多步方法*。相比之下，*单步方法*（例如 DDIM）仅依赖于最近的状态进行下一步更新。这类方法虽然更简单，但通常需要更高的计算成本才能达到高准确率，因为整体上需要更多的函数评估（或更多的步骤）。

扩散概率模型(DPM)求解器族。 DPM-Solver 族，包括 DPM-Solver (**lu2022dpm**)(Sect DPM-Solver++ (**lu2022dpm2**)(Section 9.5) 以及 DPM-Solver-v3 (**zheng2023dpm**)(Section 基于 PF-ODE 的线性结构，并引入了关键的时间重参数化，即 *半对数信噪比* (SNR)：

$$\lambda_t \coloneqq \frac{1}{2}\log\frac{\alpha_t^2}{\sigma_t^2} = \log\frac{\alpha_t}{\sigma_t}.$$

变量变换将非线性项转化为指数加权积分

$$\int_{\lambda_s}^{\lambda_t} e^{-\lambda}\,\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda)\,\mathrm{d}\lambda,$$

其中 $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}$ 表示在重参数化时间 λ 下的模型（详细内容见 Equation (9.4.4)）。这种表示方法使得积分的高阶近似更加准确。

DPM-Solver 通过在 λ 中使用泰勒展开引入了高阶求解器，针对半对数信噪比重参数化进行了优化，表明少量 NFE 即可生成高质量样本。DPM-Solver++ 将该方法适配到无分类器引导，并采用 $\mathbf{x}$ -预测以提升稳定性。DPM-Solver-v3 进一步通过将参数化选择建模为一个最优化问题，以合理方式最小化局部误差，实现了参数化选择的自动化。

（可选）时间并行方法。 一种互补策略通过并行处理不同时间间隔的计算来加速采样，而不是严格按顺序进行处理。

ParaDiGMs. 在 Section 9.8 中提出，该方法 (**shih2023parallel**) 将常微分方程的解重新表述为一个不动点问题。这一视角使得积分项可以并行计算，从而缓解了传统时间步进求解器的串行瓶颈。重要的是，该方法不仅限于指数积分器形式；它同样适用于具有非线性漂移项的通用 PF-ODEs $\mathbf{f}(\mathbf{x}, t)$。此外，该方法与求解器无关：不动点公式通过用选定时刻模型评估值的加权和替代积分

项，封装了任意时间步进规则，因此可以在并行计算模型评估的同时使用欧拉法、DEIS 或 DPM-Solver 风格的更新。

真实的计算成本（函数求值次数） 在实际应用中，时钟时间开销主要不是由离散化步骤的数量决定，而是由我们调用模型网络的次数决定。我们将这个次数称为*函数评估次数*（*NFE*）。如果采样器每步执行 m 次评估，共进行 N 步，则成本按比例增长为

$$\mathrm{NFE} = m\,N.$$

例如，一阶欧拉或指数欧拉格式具有 $m = 1$ ，而单步 k 阶方法通常需要 $m \geq k$（例如，DPM-Solver 的 k 阶）。多步方法（如 DEIS、DPM-Solver++ 的多步版本）会重用之前的求值结果，因此在短暂的预热阶段后，平均 m 接近 1 。无分类器引导在每一步有效将调用次数翻倍。因此，在实际应用中，“更快”的采样意味着实现更低的 NFE，而不仅仅是减少步骤数。

关于使用 PF-ODE 等价形式的一点说明。 在下面的讨论中，我们将使用 Section 6.3 中的结果，这些结果支持扰动核的等价参数化形式 $(f(t), g(t))$ 与 (α_t, σ_t) 的互换使用，其中 $\mathbf{x}_t|\mathbf{x}_0 \sim \mathcal{N}(\cdot; \alpha_t\mathbf{x}_0, \sigma_t^2\mathbf{I})$ 通过如下关系相互关联

$$f(t) = \frac{\alpha_t'}{\alpha_t}, \quad g^2(t) = \frac{\mathrm{d}}{\mathrm{d}t}(\sigma_t^2) - 2\frac{\alpha_t'}{\alpha_t}\sigma_t^2 = 2\sigma_t\sigma_t' - 2\frac{\alpha_t'}{\alpha_t}\sigma_t^2.$$

在这些关系下，PF-ODE 可以写成几种等价的形式（参见 Equation (6.3.2)）。

9.2 DDIM

在本节中，我们介绍一种加速扩散模型采样过程的开创性方法：降噪扩散隐式模型（Denoising Diffusion Implicit Models, DDIM），该方法也是最广泛使用的基于常微分方程（ODE）的求解器之一。尽管其名称暗示了变分起源，如 Section 6.3.2 中 $(\mathbf{x}, \boldsymbol{\epsilon})$ -预测所示，我们将展示其实际更新规则也可被解释为对 Equation (9.1.6) 中积分的欧拉法直接应用。这种基于常微分方程的视角不仅为 DDIM 提供了严谨的重新诠释，还为设计更灵活高效的快速采样器奠定了基础。

原始变分推导的 DDIM 将在 Section 9.2.3中重新讨论。在 Section 9.2.4中，我们建立了 DDIM 更新规则与条件流匹配之间的清晰对应关系，表明 DDIM 动力学可以被解释为 CFM 所学成的流。

9.2.1 将 DDIM 解释为一个 ODE 求解器

令 $s > t$ 表示两个离散的时间步，其中 s 为更新的起始时间，t 为目标时间。为了近似 Equation (9.1.6) 中的积分，一个自然的选择是在 s（即时间步的起点）固定被积函数，假设

$$\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\tau, \tau) \approx \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s, s), \quad \text{for all } \tau \in [t, s].$$

该假设导出了一个欧拉更新近似（参见 Equation (9.1.7)），从而得到以下更新规则：

$$\tilde{\mathbf{x}}_t = \mathcal{E}(s \to t)\tilde{\mathbf{x}}_s + \left(\frac{1}{2}\int_s^t \frac{g^2(\tau)}{\sigma_\tau}\mathcal{E}(\tau \to t)\,\mathrm{d}\tau\right)\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_s, s), \tag{9.2.1}$$

对于初值点 $\tilde{\mathbf{x}}_s$ 。此时，积分变得解析易处理，得到以下实用且高效的 DDIM 更新公式：

Proposition 9.2.1: DDIM = 欧拉方法（指数欧拉）

在 Equation (9.2.1) 中的更新规则，通过将欧拉方法应用于 Equation (9.1.6) 中的指数积分形式，得到以下 DDIM 更新：

$$\tilde{\mathbf{x}}_t = \frac{\alpha_t}{\alpha_s}\tilde{\mathbf{x}}_s - \alpha_t\left(\frac{\sigma_s}{\alpha_s} - \frac{\sigma_t}{\alpha_t}\right)\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_s, s). \tag{9.2.2}$$

Proof for Proposition.

我们使用 Equation (4.4.2)，即

$$f(t)=\frac{\alpha_t'}{\alpha_t},\quad g^2(t)=\frac{\mathrm{d}}{\mathrm{d}t}(\sigma_t^2)-2\frac{\alpha_t'}{\alpha_t}\sigma_t^2=2\sigma_t\sigma_t'-2\frac{\alpha_t'}{\alpha_t}\sigma_t^2.$$

由此可得

$$\mathcal{E}(s\to t)=e^{\int_s^t f(u)\,\mathrm{d}u}=e^{\log\alpha_u|_{u=s}^{u=t}}=\frac{\alpha_t}{\alpha_s}.$$

因此

$$\begin{aligned}\int_s^t\frac{g^2(\tau)}{2\sigma_\tau}e^{\int_\tau^t f(u)\,\mathrm{d}u}\mathrm{d}\tau&=\int_s^t\frac{g^2(\tau)}{2\sigma_\tau}\frac{\alpha_t}{\alpha_\tau}\mathrm{d}\tau\\&=\alpha_t\int_s^t\frac{1}{2\sigma_\tau\alpha_\tau}\Big(\frac{\mathrm{d}\sigma_\tau^2}{\mathrm{d}\tau}-2\frac{\mathrm{d}\log\alpha_\tau}{\mathrm{d}\tau}\sigma_\tau^2\Big)\mathrm{d}\tau\\&=\alpha_t\int_s^t\frac{\mathrm{d}}{\mathrm{d}\tau}\Big(\frac{\sigma_\tau}{\alpha_\tau}\Big)\mathrm{d}\tau\\&=-\alpha_t\Big(\frac{\sigma_s}{\alpha_s}-\frac{\sigma_t}{\alpha_t}\Big).\end{aligned}$$

这种对应关系表明，DDIM 可以被解释为对指数积分变换后的半线性 PF-ODE 应用的一阶欧拉方法。

9.2.2 DDIM 不同参数化方式背后的直觉

DDIM 是一种广泛使用的加速扩散采样的方法，通常可采用不同的参数化方式（见 Equation (6.3.1)），而不仅仅是 $\boldsymbol{\epsilon}$-预测。在本小节中，我们展示了不同参数化下的重表述，并随后提供对 DDIM 更直观的解释。

DDIM 的不同参数化。 在实际应用中，使用一种标准参数化形式的预训练扩散模型，并在 PF-ODE 的 DDIM 离散化过程中，用相应的预测器替换原始目标。为清晰起见，我们在此列出原始目标版本；可实现的版本则通过相应替换得到。

$$\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}\approx\boldsymbol{\epsilon}^*,\qquad\mathbf{x}_{\boldsymbol{\phi}^\times}\approx\mathbf{x}^*,\qquad\mathbf{s}_{\boldsymbol{\phi}^\times}\approx\mathbf{s}^*,\qquad\mathbf{v}_{\boldsymbol{\phi}^\times}\approx\mathbf{v}^*.$$

Corollary 9.2.1: DDIM in Different Parametrizations

Let $s > t$. Starting from $\tilde{\mathbf{x}}_s \sim p_s$ and ending at time t, the DDIM update in different parametrizations are as:

$$\begin{aligned}
\tilde{\mathbf{x}}_t &= \frac{\alpha_t}{\alpha_s}\tilde{\mathbf{x}}_s + \alpha_t\left(\frac{\sigma_t}{\alpha_t} - \frac{\sigma_s}{\alpha_s}\right)\boldsymbol{\epsilon}^*(\tilde{\mathbf{x}}_s, s) \\
&= \frac{\sigma_t}{\sigma_s}\tilde{\mathbf{x}}_s + \alpha_s\left(\frac{\alpha_t}{\alpha_s} - \frac{\sigma_t}{\sigma_s}\right)\mathbf{x}^*(\tilde{\mathbf{x}}_s, s) \\
&= \frac{\alpha_t}{\alpha_s}\tilde{\mathbf{x}}_s + \sigma_s^2\left(\frac{\alpha_t}{\alpha_s} - \frac{\sigma_t}{\sigma_s}\right)\mathbf{s}^*(\tilde{\mathbf{x}}_s, s) \\
&= \alpha_t \underbrace{\mathbf{x}^*(\tilde{\mathbf{x}}_s, s)}_{\approx \mathbf{x}_{\boldsymbol{\phi}} \times \text{ estimated clean}} + \sigma_t \underbrace{\boldsymbol{\epsilon}^*(\tilde{\mathbf{x}}_s, s)}_{\approx \boldsymbol{\epsilon}_{\boldsymbol{\phi}} \times \text{ estimated noise}}.
\end{aligned} \tag{9.2.3}$$

Equation (9.2.3) 中的最后一个等式清晰地展示了 DDIM：从 $\tilde{\mathbf{x}}_s \sim p_s$ 出发，（估计的）干净部分 $\mathbf{x}^*(\tilde{\mathbf{x}}_s, s)$ 与（估计的）噪声部分 $\boldsymbol{\epsilon}^*(\tilde{\mathbf{x}}_s, s)$ 作为插值端点，以系数 (α_t, σ_t) 重构一个 $\tilde{\mathbf{x}}_t \sim p_t$。

事实上，DDIM 可以被视为对无指数积分器的 $\mathbf{v}$-参数化 PF-ODE 的直接欧拉离散化。根据命题 6.3.2，PF-ODE 也具有如下形式的 $\mathbf{v}$-预测：

$$\frac{\mathrm{d}\mathbf{x}(\tau)}{\mathrm{d}\tau} = \alpha'_\tau \mathbf{x}^*(\mathbf{x}(\tau), \tau) + \sigma'_\tau \boldsymbol{\epsilon}^*(\mathbf{x}(\tau), \tau), \quad \tau \in [t, s].$$

从 $\tilde{\mathbf{x}}_s$ 开始，并对 $[t, s]$ 进行积分，欧拉法在右端点处冻结预测器：

$$\mathbf{x}^*(\mathbf{x}(\tau), \tau) \approx \mathbf{x}^*(\tilde{\mathbf{x}}_s, s), \quad \boldsymbol{\epsilon}^*(\mathbf{x}(\tau), \tau) \approx \boldsymbol{\epsilon}^*(\tilde{\mathbf{x}}_s, s),$$

对所有 $\tau \in [t, s]$ 成立。这给出

$$\begin{aligned}
\tilde{\mathbf{x}}_t &= \tilde{\mathbf{x}}_s + \int_s^t \left(\alpha'_\tau \mathbf{x}^* + \sigma'_\tau \boldsymbol{\epsilon}^*\right) \mathrm{d}\tau \\
&\approx \tilde{\mathbf{x}}_s + (\alpha_t - \alpha_s)\mathbf{x}^*(\tilde{\mathbf{x}}_s, s) + (\sigma_t - \sigma_s)\boldsymbol{\epsilon}^*(\tilde{\mathbf{x}}_s, s) \\
&= \alpha_t \mathbf{x}^*(\tilde{\mathbf{x}}_s, s) + \sigma_t \boldsymbol{\epsilon}^*(\tilde{\mathbf{x}}_s, s),
\end{aligned}$$

其中，最后一个等式直接来自 Equation (6.3.1)。上述推导出的公式与 DDIM 更新中的最终等式完全吻合 (Equation (9.2.3))。详见 Equation (9.2.3) 的图示。

通过速度预测，PF-ODE 中的线性项 $f(t)\mathbf{x}$ 被吸收进目标 $\mathbf{v}^*(\mathbf{x}(t), t) =$

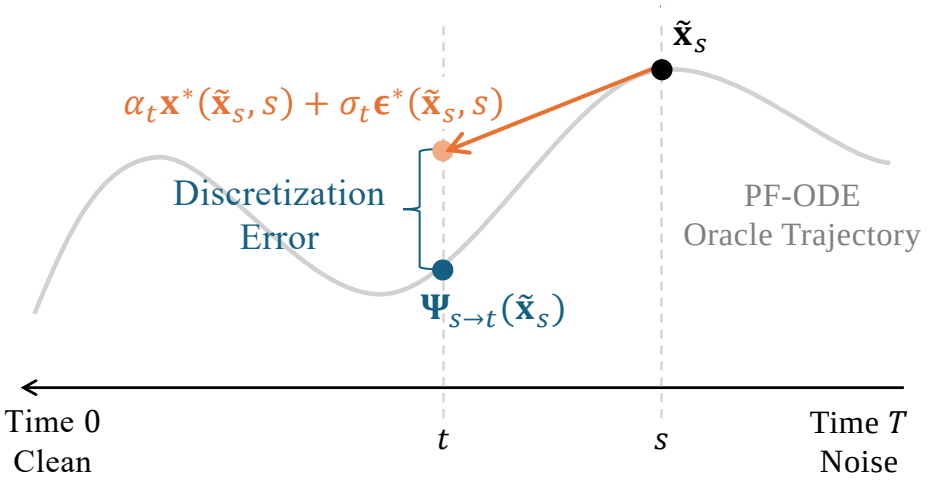


图 9.1: DDIM 作为 PF-ODE 的欧拉离散化的示意图。从时间 s 的状态 $\tilde{\mathbf{x}}_s$ 出发，预言器 PF-ODE 轨迹（灰色曲线）确定性地演化至时间 t 的 $\boldsymbol{\Psi}_{s\to t}(\tilde{\mathbf{x}}_s)$。相比之下，DDIM 更新（橙色）直接将 $\tilde{\mathbf{x}}_s$ 映射到 $\alpha_t \mathbf{x}^*(\tilde{\mathbf{x}}_s, s) + \sigma_t \boldsymbol{\epsilon}^*(\tilde{\mathbf{x}}_s, s)$。此欧拉步长与真实 PF-ODE 轨迹之间的差异引入了离散化误差，以蓝色表示。若 t 与 s 相距较远，该差异可能变得显著，导致生成质量下降。

$\alpha'_t \mathbf{x}_0 + \sigma'_t \boldsymbol{\epsilon}$。根据微积分基本定理，积分 $\int_s^t \alpha'_\tau \, \mathrm{d}\tau$ 和 $\int_s^t \sigma'_\tau \, \mathrm{d}\tau$ 简化为 $(\alpha_t - \alpha_s)$ 和 $(\sigma_t - \sigma_s)$，因此单次欧拉步长即可得到闭式 DDIM 更新：

$$\tilde{\mathbf{x}}_t = \alpha_t \tilde{\mathbf{x}}^*(\tilde{\mathbf{x}}_s, s) + \sigma_t \tilde{\boldsymbol{\epsilon}}^*(\tilde{\mathbf{x}}_s, s).$$

也就是说，使用 $\mathbf{v}$ -预测时，PF-ODE 漂移项中没有可分离的线性项，因此普通的欧拉更新自然与 DDIM 公式一致。相比之下，在 $\boldsymbol{\epsilon}$ -、$\mathbf{x}$ -或 $\mathbf{s}$ -预测参数化下，PF-ODE 漂移项可以分解为一个包含线性项和非线性修正项的半线性形式，这符合 Equation (9.1.5) 中给出的一般模板。此时，朴素的欧拉步长仅能近似线性项，而非精确计算（参见 Equation (9.1.8) 中的论证）。而 DDIM 则对应于一种指数-欧拉（积分因子）步长，能够解析地处理该线性分量。因此，$\mathbf{v}$ -预测带来了最简单且直接的欧拉积分方式，而其他参数化则需要采用指数-欧拉形式才能实现相同的 DDIM 行为。

上述讨论也呼应了 Section 6.3.4 中提出的论点，并得出以下结论：

Observation 9.2.1: (Exponential) Euler and DDIM Updates

Given the same schedulers (α_t, σ_t),

$$\begin{aligned} \mathbf{v}\text{-prediction:} &\quad \text{Euler} = \text{DDIM}, \\ \boldsymbol{\epsilon}\text{-, } \mathbf{x}\text{-, or } \mathbf{s}\text{-prediction:} &\quad \text{exp–Euler} = \text{DDIM} \neq \text{plain Euler}, \end{aligned}$$

where, in the $\boldsymbol{\epsilon}$-, $\mathbf{x}$-, or $\mathbf{s}$-prediction cases, the plain Euler step is not equivalent to DDIM, since the linear term is only approximated and may lead to reduced stability.

不同参数化下的 DDIM 示例。我们通过一个简单的例子来说明，使用基于 Equation (9.2.3) 的 oracle 替换（$\boldsymbol{\epsilon}^*$，$\mathbf{x}^*$，$\nabla_{\mathbf{x}} \log p_t$，和 $\mathbf{v}^*$）。假设前向核 $\alpha_t = 1$ 和 $\sigma_t = t$（**karras2022elucidating**）。DDIM（exp–Euler）更新

$$\tilde{\mathbf{x}}_t = \frac{\alpha_t}{\alpha_s} \tilde{\mathbf{x}}_s - \alpha_t \left(\frac{\sigma_s}{\alpha_s} - \frac{\sigma_t}{\alpha_t} \right) \boldsymbol{\epsilon}^*(\tilde{\mathbf{x}}_s, s)$$

简化为

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_s - (s - t)\, \boldsymbol{\epsilon}^*(\tilde{\mathbf{x}}_s, s).$$

从概念上讲，将时间间隔 $(s - t)$ 乘以原始噪声估计 $\boldsymbol{\epsilon}^*(\tilde{\mathbf{x}}_s, s)$ 从当前样本 $\tilde{\mathbf{x}}_s$ 中减去，会将其推向一个更干净的估计。

使用 $\mathbf{x}$ -预测预言机 $\mathbf{x}^*$，它与噪声预言机相关联，通过

$$\boldsymbol{\epsilon}^*(\tilde{\mathbf{x}}_s, s) = \frac{\tilde{\mathbf{x}}_s - \mathbf{x}^*(\tilde{\mathbf{x}}_s, s)}{s},$$

我们得到

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_s - \frac{s - t}{s} \left(\tilde{\mathbf{x}}_s - \mathbf{x}^*(\tilde{\mathbf{x}}_s, s) \right) = \frac{t}{s} \tilde{\mathbf{x}}_s + \left(1 - \frac{t}{s} \right) \mathbf{x}^*(\tilde{\mathbf{x}}_s, s). \tag{9.2.4}$$

因此，$\tilde{\mathbf{x}}_t$ 是当前样本 $\tilde{\mathbf{x}}_s$ 与 $\mathbf{x}$ -预测 $\mathbf{x}^*(\tilde{\mathbf{x}}_s, s)$ 的凸组合，后者作为干净数据的预言估计。此外，我们可以将其重写为

$$\tilde{\mathbf{x}}_t - \mathbf{x}^* = \frac{t}{s} \left(\tilde{\mathbf{x}}_s - \mathbf{x}^* \right), \quad t < s,$$

这表明降噪残差在每一步都按因子 $t/s \in (0, 1)$ 收缩（因此当 $t < s$ 时不会出现

过冲）。

使用得分预言机，其与噪声预言机相关

$$\boldsymbol{\epsilon}^*(\tilde{\mathbf{x}}_s, s) = -\sigma_s \nabla_{\mathbf{x}} \log p_s(\tilde{\mathbf{x}}_s),$$

DDIM（exp–Euler）更新变为

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_s + (s-t)\, s\, \nabla_{\mathbf{x}} \log p_s(\tilde{\mathbf{x}}_s).$$

这使得 $\tilde{\mathbf{x}}_s$ 沿着得分场（朝向更高似然区域）向上移动，步长与时间间隔 $(s-t)$ 和噪声尺度 s 成比例。

最后，使用速度预言机与 $\mathbf{v}^*(\tilde{\mathbf{x}}_s, s) = -\boldsymbol{\epsilon}^*(\tilde{\mathbf{x}}_s, s)$ ，DDIM 更新可以表示为

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_s + (t-s)\, \mathbf{v}^*(\tilde{\mathbf{x}}_s, s),$$

因此，割线斜率满足有限差分恒等式。

$$\frac{\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_s}{t-s} = \mathbf{v}^*(\tilde{\mathbf{x}}_s, s).$$

直观上，这意味着更新是沿着局部微分方程的漂移方向进行的一次直线步进。

DDIM 的挑战。 然而，一阶欧拉离散化具有全局误差 $\mathcal{O}(h)$ ，因此当最大步长 $h := \max_i |t_i - t_{i-1}|$ 增大时，准确率会下降。为了提高准确率，文献中发展了高阶格式，通过更丰富的局部近似将全局阶次提升至 $\mathcal{O}(h^k)$ $(k \geq 2)$。在合适的步长分配下，这些方法可能以更少的步骤达到目标质量。然而需要注意的是，仅靠高阶并不能保证步骤更少或墙钟时间更低，因为每一步可能需要多次模型评估。在实际应用中，效率的真实衡量标准是函数评估次数 $\mathrm{NFE} = m\,N$ ，“更快”意味着以更小的 NFE 达到所需质量，而不仅仅是步骤更少。

9.2.3 （可选）DDIM 的变分视角

事实上，DDIM 的动机源于从变分视角重新审视 DDPM。在 DDPM 中，反向过程与特定的马尔可夫前向转移核 $p(\mathbf{x}_t|\mathbf{x}_{t-\Delta t})$ 相关联，该核要求采用较小的步长以正确近似多步后验分布。DDIM 通过观察到训练目标仅依赖于边缘扰动 $p_t(\mathbf{x}_t|\mathbf{x}_0)$ ，而与具体的前向转移无关，从而摆脱了这一限制。这一洞见使得能够直接从边缘分布构建反向动态过程，从而在保持边缘一致性的同时跳过中间步

骤。由于转移被定义为对任意 $t < s$，将 $p_s(\mathbf{x}_s|\mathbf{x}_0)$ 映射到 $p_t(\mathbf{x}_t|\mathbf{x}_0)$，因此可以使用更粗略的时间网格，大幅减少更新次数，从而减少模型评估次数，并实现快速的少步采样。

重新审视 DDPM 的变分视角。 在 DDPM 中，训练固定了一族边缘扰动内核 $p_t(\mathbf{x}_t|\mathbf{x}_0)$，并优化仅依赖于这些边缘量的替代目标。然而，在采样时，反向条件是单步前向内核下的贝叶斯后验：

$$p(\mathbf{x}_{t-\Delta t}|\mathbf{x}_t, \mathbf{x}_0) = \frac{p(\mathbf{x}_t|\mathbf{x}_{t-\Delta t})\, p_{t-\Delta t}(\mathbf{x}_{t-\Delta t}|\mathbf{x}_0)}{p_t(\mathbf{x}_t|\mathbf{x}_0)}.$$

这将反向更新与特定的前向转移 $p(\mathbf{x}_t|\mathbf{x}_{t-\Delta t})$ 紧密关联。如果尝试通过增大 Δt 而重复使用相同的单步核来跳过步骤，这将不再匹配真实的多步后验，通常会导致边缘分布性能下降。

原始 DDIM 动机。 DDIM 观察到，训练目标仅约束了边缘分布 $p_t(\mathbf{x}_t|\mathbf{x}_0)$，而未约束中间的逆向转移。因此，对于任意 $t < s$，可以指定一族逆向条件分布 $\pi(\mathbf{x}_t|\mathbf{x}_s, \mathbf{x}_0)$，这些条件分布满足单步边缘一致性[2]：

[2] 如果我们选择“用户自定义”的反向转移核 π 在 Equation (9.2.5) 中与“真实”的条件分布 $\pi(\mathbf{x}_t|\mathbf{x}_s, \mathbf{x}_0) = p(\mathbf{x}_t|\mathbf{x}_s, \mathbf{x}_0)$ 完全相同，那么边际一致性条件

$$\int \pi(\mathbf{x}_t|\mathbf{x}_s, \mathbf{x}_0)\, p_s(\mathbf{x}_s|\mathbf{x}_0)\, \mathrm{d}\mathbf{x}_s = p_t(\mathbf{x}_t|\mathbf{x}_0)$$

仅仅是条件联合分布的全概率定律（也称为塔性质）的结果：

$$p_t(\mathbf{x}_t|\mathbf{x}_0) = \int p(\mathbf{x}_t, \mathbf{x}_s|\mathbf{x}_0)\, \mathrm{d}\mathbf{x}_s = \int p(\mathbf{x}_t|\mathbf{x}_s, \mathbf{x}_0)\, p_s(\mathbf{x}_s|\mathbf{x}_0)\, \mathrm{d}\mathbf{x}_s.$$

或者等价地，通过显式表达贝叶斯后验为

$$p(\mathbf{x}_t|\mathbf{x}_s, \mathbf{x}_0) = \frac{p(\mathbf{x}_s|\mathbf{x}_t, \mathbf{x}_0)\, p_t(\mathbf{x}_t|\mathbf{x}_0)}{p_s(\mathbf{x}_s|\mathbf{x}_0)},$$

然后乘以 $p_s(\mathbf{x}_s|\mathbf{x}_0)$ 并对 $\mathbf{x}_s$ 进行边缘化，我们恢复得到

$$\int p(\mathbf{x}_t|\mathbf{x}_s, \mathbf{x}_0)\, p_s(\mathbf{x}_s|\mathbf{x}_0)\, \mathrm{d}\mathbf{x}_s = p_t(\mathbf{x}_t|\mathbf{x}_0),$$

这正是相同的边际一致性条件。在马尔可夫前向情况下，进一步有 $p(\mathbf{x}_t|\mathbf{x}_s, \mathbf{x}_0) = p(\mathbf{x}_t|\mathbf{x}_s)$，从而简化如下表达式：

$$p_t(\mathbf{x}_t|\mathbf{x}_0) = \int p(\mathbf{x}_t|\mathbf{x}_s)\, p_s(\mathbf{x}_s|\mathbf{x}_0)\, \mathrm{d}\mathbf{x}_s.$$

$$\int \pi(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0)\, p_s(\mathbf{x}_s|\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_s = p_t(\mathbf{x}_t|\mathbf{x}_0). \tag{9.2.5}$$

该构造消除了对前向一步核 $p(\mathbf{x}_t|\mathbf{x}_{t-\Delta t})$ 的依赖，并使粗粒度（跳过）时间步合法化。

离散时间 DDIM 的推导。 考虑一般的前向扰动：

$$p_t(\mathbf{x}_t|\mathbf{x}_0) := \mathcal{N}\big(\mathbf{x}_t; \alpha_t\mathbf{x}_0, \sigma_t^2\mathbf{I}\big),$$

其中 $\mathbf{x}_0 \sim p_{\text{data}}$.

DDIM 不需要反向更新与关联到单步前向核的贝叶斯后验一致。只需 选择一个保持边缘分布的反向条件分布即可。具体而言，对于任意 $t < s$，我们假设高斯族

$$\pi(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0) = \mathcal{N}\big(\mathbf{x}_t;\, a_{t,s}\,\mathbf{x}_0 + b_{t,s}\,\mathbf{x}_s,\; c_{t,s}^2\,\mathbf{I}\big), \tag{9.2.6}$$

其中系数 $(a_{t,s}, b_{t,s}, c_{t,s})$ 由边缘一致性约束 Equation (9.2.5) 确定。由于所有涉及的内核均为高斯内核，对 $\mathbf{x}_s|\mathbf{x}_0 = \alpha_s\mathbf{x}_0 + \sigma_s\boldsymbol{\epsilon}'$ 进行采样后，再从 Equation (9.2.6) 中采样 $\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0$ 可得

$$\begin{aligned}
\mathbf{x}_t &= a_{t,s}\,\mathbf{x}_0 + b_{t,s}\,\mathbf{x}_s + c_{t,s}\,\boldsymbol{\epsilon} \\
&= a_{t,s}\,\mathbf{x}_0 + b_{t,s}\big(\alpha_s\mathbf{x}_0 + \sigma_s\boldsymbol{\epsilon}'\big) + c_{t,s}\,\boldsymbol{\epsilon} \\
&= (a_{t,s} + b_{t,s}\alpha_s)\,\mathbf{x}_0 + \sqrt{b_{t,s}^2\sigma_s^2 + c_{t,s}^2}\;\boldsymbol{\epsilon}'',
\end{aligned} \tag{9.2.7}$$

其中 $\boldsymbol{\epsilon}, \boldsymbol{\epsilon}', \boldsymbol{\epsilon}'' \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 是独立的（高斯和性质）。另一方面，

$$\mathbf{x}_t \sim p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\big(\mathbf{x}_t; \alpha_t\mathbf{x}_0, \sigma_t^2\mathbf{I}\big).$$

令此目标与 Equation (9.2.7) 的均值和方差相等，得到

$$\alpha_t = a_{t,s} + b_{t,s}\alpha_s, \qquad \sigma_t^2 = b_{t,s}^2\sigma_s^2 + c_{t,s}^2.$$

该系统是欠定的，因此我们将 $c_{t,s}$ 视为自由参数，并施加自然约束 $0 \le c_{t,s} \le \sigma_t$

，然后求解 $a_{t,s}, b_{t,s}$ ：

$$b_{t,s} = \frac{\sqrt{\sigma_t^2 - c_{t,s}^2}}{\sigma_s}, \qquad a_{t,s} = \alpha_t - \alpha_s\, b_{t,s}. \tag{9.2.8}$$

此处，我们不失一般性地取 $b_{t,s}$ 的非负根。

将 Equation (9.2.8) 代入 Equation (9.2.6) 得

$$\pi(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0) = \mathcal{N}\Big(\mathbf{x}_t;\ \underbrace{\alpha_t\mathbf{x}_0 + \frac{\sqrt{\sigma_t^2 - c_{t,s}^2}}{\sigma_s}(\mathbf{x}_s - \alpha_s\mathbf{x}_0)}_{\text{mean}},\ c_{t,s}^2\mathbf{I}\Big). \tag{9.2.9}$$

等价地，Equation (9.2.9) 中的均值展开为

$$\Big(\alpha_t - \alpha_s\frac{\sqrt{\sigma_t^2 - c_{t,s}^2}}{\sigma_s}\Big)\mathbf{x}_0 + \Big(\frac{\sqrt{\sigma_t^2 - c_{t,s}^2}}{\sigma_s}\Big)\mathbf{x}_s.$$

Lemma 9.2.2: DDIM Coefficients

Let $\pi(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0)$ be given by Equation (9.2.6). If the marginal-consistency condition Equation (9.2.5) holds, then the coefficients are exactly those in Equation (9.2.8), with $0 \le c_{t,s} \le \sigma_t$.

Remark.

1. 在 DDIM 中，我们通过选择反向核 $\pi(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0)$ 来满足边缘一致性约束，且通常有

   $$\pi(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0) \neq p(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0),$$

   其中 $p(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0)$ 是与特定前向单步核相关的贝叶斯后验。根据贝叶斯法则，

   $$p(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0) \ \propto\ p(\mathbf{x}_s|\mathbf{x}_t)\,p_t(\mathbf{x}_t|\mathbf{x}_0),$$

   该后验在指定 π 或进行训练时无需使用。

2. 仅当方差参数选择与 DDPM 后验方差匹配时（即 Equation (9.2.10) 中的 $\eta = 1$ 情景），才有 $\pi(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0) = p(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0)$；否则 $\pi(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0) \neq p(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0)$。

3. 若不施加马尔可夫约束，通常有 $p(\mathbf{x}_s|\mathbf{x}_t,\mathbf{x}_0) \neq p(\mathbf{x}_s|\mathbf{x}_t)$。等式 $p(\mathbf{x}_s|\mathbf{x}_t,\mathbf{x}_0) = p(\mathbf{x}_s|\mathbf{x}_t)$ 与特定马尔可夫前向模型绑定，而 DDIM 在构建反向过程时并未采用该假设。

前向边缘分布 $\{p_t(\mathbf{x}_t|\mathbf{x}_0)\}_t$ 无法唯一确定反向条件转移。存在无穷多个内核 $\pi(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0)$ 满足 Equation (9.2.5)，其中任意一个均可自由指定。参数 $c_{t,s}$ 索引这一族内核，并控制在每个反向步骤 $s \to t$ 中注入的噪声量。下文将介绍这一类 DDIM 求解器。

DDIM 采样器（步骤 $s \to t$）。 DDIM 采样器基于选定的反向核 $\pi(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0)$ 在 Equation (9.2.9) 中，将 $\mathbf{x}_0$ 替换为来自预训练模型的预测器。使用 $\boldsymbol{\epsilon}$ -预测网络 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}$ （即插即用，无需重新训练），我们设定

$$\mathbf{x}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s,s) := \frac{\mathbf{x}_s - \sigma_s\,\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s,s)}{\alpha_s}, \quad p_{\boldsymbol{\phi}^\times}(\mathbf{x}_t|\mathbf{x}_s) := \pi\big(\mathbf{x}_t \,\big|\, \mathbf{x}_s, \mathbf{x}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s,s)\big).$$

将 $\mathbf{x}_{\boldsymbol{\phi}^\times}$ 代入 Equation (9.2.9) 得到更新

$$\mathbf{x}_t = \frac{\alpha_t}{\alpha_s}\,\mathbf{x}_s + \left(\sqrt{\sigma_t^2 - c_{t,s}^2} - \frac{\alpha_t}{\alpha_s}\sigma_s\right)\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s,s) + c_{t,s}\,\boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0},\mathbf{I}),$$

其中 $c_{t,s} \in [0,\sigma_t]$ 控制随机性。

为便于记号，定义前向因子

$$\alpha_{t|s} := \tfrac{\alpha_t}{\alpha_s}, \qquad \sigma_{t|s}^2 := \sigma_t^2 - \alpha_{t|s}^2\,\sigma_s^2,$$

使得 $p(\mathbf{x}_t|\mathbf{x}_s) = \mathcal{N}(\alpha_{t|s}\mathbf{x}_s, \sigma_{t|s}^2\mathbf{I})$。于是采样器可以表示为

$$\mathbf{x}_t = \alpha_{t|s}\,\mathbf{x}_s + \left(\sqrt{\sigma_t^2 - c_{t,s}^2} - \alpha_{t|s}\sigma_s\right)\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s,s) + c_{t,s}\,\boldsymbol{\epsilon}_t.$$

通过改变 $c_{t,s}$ ，可以获得一组共享同一预训练扩散模型且无需重新训练的采样器：

- **DDPM 步骤（后验方差）**： $c_{t,s} = \frac{\sigma_s}{\sigma_t}\,\sigma_{t|s}$ 使 $\pi(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0)$ 等于由单步前向核诱导的贝叶斯后验 $p(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0)$ 。用其预测器替换 $\mathbf{x}_0$ 得到标准 DDPM

反向更新 $p_{\phi^\times}(\mathbf{x}_t|\mathbf{x}_s)$ ，即具有 $\alpha_t^2+\sigma_t^2=1$ (Equation (2.2.14)) 的马尔可夫 DDPM 步骤。

- **确定性 DDIM ($\eta=0$):** $c_{t,s}=0$ 给出

$$\mathbf{x}_t=\alpha_{t|s}\mathbf{x}_s+\big(\sigma_t-\alpha_{t|s}\sigma_s\big)\boldsymbol{\epsilon}_{\phi^\times}(\mathbf{x}_s,s),$$

这与 ODE 视角下的 DDIM 跳跃相匹配。

- **插值**：定义

$$c_{t,s}=\eta\,\frac{\sigma_s}{\sigma_t}\,\sigma_{t|s},\quad \eta\in[0,1], \tag{9.2.10}$$

使得 η 从随机 DDPM 更新 ($\eta=1$) 平滑地插值到确定性 DDIM 更新 ($\eta=0$)。

9.2.4 DDIM 作为条件流匹配

在本小节中，我们将看到确定性 DDIM 可以被理解为寻找一个条件流映射，该映射将 $p_s(\cdot|\mathbf{x}_0)$ 向前推动到 $p_t(\cdot|\mathbf{x}_0)$。这个条件流的切线与条件流匹配 (CFM) 中使用的条件速度一致。对该条件速度进行边缘化处理得到 PF–ODE 漂移项，其简单的欧拉离散化恢复了 **v**-预测中的边际 DDIM 更新。

我们重新审视 DDIM 的一步条件边缘一致性恒等式 (Equation (9.2.5))

$$\int \pi(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0)p_s(\mathbf{x}_s|\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_s=p_t(\mathbf{x}_t|\mathbf{x}_0),\quad t<s,$$

即，若 $\mathbf{x}_s\sim p_s(\cdot|\mathbf{x}_0)$，则通过所选的反向核向前推进 $\mathbf{x}_s$ 可重现 $p_t(\cdot|\mathbf{x}_0)$。当反向核为确定性时，等价于寻找一个条件映射 $\boldsymbol{\Psi}_{s\to t}(\cdot|\mathbf{x}_0)$，将 $p_s(\cdot|\mathbf{x}_0)$ 向前推进至 $p_t(\cdot|\mathbf{x}_0)$。

$$\pi(\mathbf{x}_t|\mathbf{x}_s,\mathbf{x}_0)=\delta\big(\mathbf{x}_t-\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s|\mathbf{x}_0)\big),\qquad \big(\boldsymbol{\Psi}_{s\to t}(\cdot|\mathbf{x}_0)\big)_{\#}p_s(\cdot|\mathbf{x}_0)=p_t(\cdot|\mathbf{x}_0).$$

在线性-高斯路径 $\mathbf{x}_\tau=\alpha_\tau\mathbf{x}_0+\sigma_\tau\boldsymbol{\epsilon}$ 下，类似于 Equations (9.2.6) and (9.2.7) 中的论证，可导出 条件映射

$$\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s|\mathbf{x}_0)=\frac{\sigma_t}{\sigma_s}\,\mathbf{x}_s+\Big(\alpha_t-\alpha_s\frac{\sigma_t}{\sigma_s}\Big)\mathbf{x}_0,$$

其瞬时的条件速度为

$$\mathbf{v}_t^*(\mathbf{x}|\mathbf{x}_0) = \partial_h\big|_{h=0} \mathbf{\Psi}_{t\to t+h}(\mathbf{x}|\mathbf{x}_0) = \frac{\sigma_t'}{\sigma_t}\,\mathbf{x} + \Big(\alpha_t' - \alpha_t\frac{\sigma_t'}{\sigma_t}\Big)\mathbf{x}_0.$$

我们将 $\mathbf{\Psi}_{s\to t}(\cdot|\mathbf{x}_0)$ 称为 DDIM 条件映射。

使用 $p_t(\mathbf{x}|\mathbf{x}_0)$ ，条件流匹配将时变场拟合到此目标速度，

$$\mathcal{L}_{\mathrm{CFM}}(\boldsymbol{\phi}) = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_t\sim p_t(\cdot|\mathbf{x}_0)}\big\|\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x}_t,t) - \mathbf{v}_t^*(\mathbf{x}_t|\mathbf{x}_0)\big\|^2,$$

因此，CFM 回归目标等于 DDIM 条件图的条件速度。

Observation 9.2.2: Conditional Level

Along the conditional Gaussian path, the DDIM conditional map and the CFM target generate the same conditional flow $\mathbf{\Psi}_{s\to t}(\cdot|\mathbf{x}_0)$.

对给定 $\mathbf{x}_t = \mathbf{x}$ 的 $\mathbf{x}_0$ 的后验分布取条件速度的平均值，得到边缘 PF–ODE drift，

$$\mathbf{v}^*(\mathbf{x},t) = \mathbb{E}\left[\mathbf{v}_t^*(\mathbf{x}|\mathbf{x}_0)|\mathbf{x}_t = \mathbf{x}\right],$$

在线性-高斯调度器下，其形式为可分离的预测器

$$\mathbf{v}^*(\mathbf{x},t) = \alpha_t'\,\mathbf{x}^*(\mathbf{x},t) + \sigma_t'\,\boldsymbol{\epsilon}^*(\mathbf{x},t), \quad \mathbf{x} = \alpha_t\,\mathbf{x}^*(\mathbf{x},t) + \sigma_t\,\boldsymbol{\epsilon}^*(\mathbf{x},t).$$

我们已经看到，这种边缘化 $\mathbf{v}$ –预测的 PF-ODE 的简单欧拉步长恰好是 DDIM 更新（见 Equation (9.2.3) 中的最后一个等式）。

简而言之，DDIM 是 (i) 一种确定性条件传输，其切线等于 CFM 目标；(ii) 在对这一切线进行边缘化后，相当于 PF–ODE 的一个欧拉步长，且该步长与 DDIM 更新一致。

9.3 DEIS

在指数积分公式 (Equation (9.1.6)) 中

$$\int_s^t \frac{g^2(\tau)}{2\,\sigma_\tau}\,\mathcal{E}(\tau \to t)\,\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\tau, \tau)\,\mathrm{d}\tau,$$

唯一未知的是模型输出 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\tau, \tau)$；一旦 (α, σ, g) 确定，调度项和权重 $\mathcal{E}(\tau \to t)$ 便已知。DDIM（欧拉法）通过保持模型输出恒定来近似该积分：

$$\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\tau, \tau) \;\approx\; \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s, s), \quad \tau \in [t, s].$$

然而，这仅是一阶精确的，在模型输出随时间快速变化时可能失效。

一个自然的问题随之产生：*我们能否更好地利用已计算的模型评估结果？*与经典的*多步求解器*类似，我们不再将 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\tau, \tau)$ 视为常数（欧拉法），而是复用之前的输出（锚点）来拟合一条简单的时间曲线。由于权重 $\frac{g^2(\tau)}{2\sigma_\tau}\mathcal{E}(\tau \to t)$ 已知，该拟合曲线的积分便可被精确计算。实际上，未知函数的复杂积分被替换为由历史模型调用定义的近似曲线的精确积分。这正是*扩散指数积分采样器*（DEIS）(**zhang2022fast**) 的核心原理。

对于熟悉经典常微分方程求解器的读者而言，DEIS 可以被看作是在半线性 PF-ODE (Equation (9.1.6)) 的指数积分框架下应用的 Adams–Bashforth 格式 (**iserles2009first**)：线性漂移项通过积分因子确切处理，而剩余的非线性项则通过多步多项式外推法推进。

我们在 Section 9.3.1 中首先介绍如何构造一条经过一组锚点的光滑曲线。在 Section 9.3.2 中，我们进一步将这种插值技术应用于近似 PF-ODE 积分，从而得到 DEIS 算法。最后，在 Section 9.3.3 中，我们证明了当使用常数多项式时，DDIM 作为 DEIS 的特例出现。

9.3.1 多项式外推法

简单曲线的锚点插值。 假设我们知道某个随时间变化的量在最近几个时间点的取值

$$(\tau_0, \mathbf{Y}_0),\ (\tau_1, \mathbf{Y}_1),\ \ldots,\ (\tau_n, \mathbf{Y}_n), \quad \tau_0 < \tau_1 < \cdots < \tau_n,$$

其中每个 $\mathbf{Y}_j$ 可能是向量值。获得一条与这些锚点确切匹配的简单曲线最自然的方法是使用通过这些点的最低次数的多项式。最简便的方法是乘上在其他结点处为零的因子，然后进行归一化，使得在 τ_j 处的值变为 1。小规模的情况直观

易懂：

Example:

$n = 0$ **(Constant):** use the last value,

$$\mathbf{Y}(\tau) \equiv \mathbf{Y}_n.$$

$n = 1$ **(Line):** draw the straight line through the last two anchors,

$$\mathbf{Y}(\tau) = \tfrac{\tau-\tau_n}{\tau_{n-1}-\tau_n}\mathbf{Y}_{n-1} + \tfrac{\tau-\tau_{n-1}}{\tau_n-\tau_{n-1}}\mathbf{Y}_n.$$

$n = 2$ **(Quadratic; Parabola):** pass a quadratic curve through the last three anchors. For example, if the anchors are

$$(\tau_{n-2}, \mathbf{Y}_{n-2}), \quad (\tau_{n-1}, \mathbf{Y}_{n-1}), \quad (\tau_n, \mathbf{Y}_n),$$

the quadratic interpolant is

$$\mathbf{Y}(\tau) = \mathbf{Y}_{n-2}\,\ell_{n-2}(\tau) + \mathbf{Y}_{n-1}\,\ell_{n-1}(\tau) + \mathbf{Y}_n\,\ell_n(\tau),$$

where the Lagrange basis functions are

$$\begin{aligned}
\ell_{n-2}(\tau) &= \tfrac{(\tau-\tau_{n-1})(\tau-\tau_n)}{(\tau_{n-2}-\tau_{n-1})(\tau_{n-2}-\tau_n)},\\
\ell_{n-1}(\tau) &= \tfrac{(\tau-\tau_{n-2})(\tau-\tau_n)}{(\tau_{n-1}-\tau_{n-2})(\tau_{n-1}-\tau_n)},\\
\ell_n(\tau) &= \tfrac{(\tau-\tau_{n-2})(\tau-\tau_{n-1})}{(\tau_n-\tau_{n-2})(\tau_n-\tau_{n-1})}.
\end{aligned}$$

These satisfy the interpolation conditions

$$\ell_j(\tau_k) = \delta_{jk}, \quad \text{for } j,k \in \{n-2, n-1, n\}$$

and $\ell_{n-2}(\tau) + \ell_{n-1}(\tau) + \ell_n(\tau) = 1$ for all τ. This curve not only matches all three anchors but also bends to reflect the local curvature. ■

这些情况都属于一种单一的配方，称为 *Lagrange* 多项式。其思想很简单：我

们通过时间相关的 权重将控制点进行线性组合来构成曲线，

$$\mathbf{Y}(\tau) = \sum_{j=0}^{n} \ell_j(\tau)\, \mathbf{Y}_j, \quad \ell_j(\tau_k) = \delta_{jk}, \quad \sum_{j=0}^{n} \ell_j(\tau) = 1.$$

每个 $\ell_j(\tau)$ 的作用类似于一个 “探照灯”，在其自身的锚点 ($\ell_j(\tau_j) = 1$) 处取值 1 ，而在其他锚点 ($\ell_j(\tau_k) = 0$, $k \neq j$) 处取值 0 。从这个意义上说，拉格朗日插值函数只是锚点的线性组合，其基函数为 $\ell_j(\tau)$ 。

9.3.2 DEIS：PF-ODE 积分的拉格朗日多项式近似

令 $n \geq 0$ 为所选的多项式阶数。在步骤 i ，我们通过利用过去模型输出构建的 n 阶多项式插值，对未知映射 $\tau \mapsto \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\tau, \tau)$ 在 $[t_{i-1}, t_i]$ 上进行近似，并将此近似代入指数积分更新公式 (Equation (9.1.6)) 以得到 $\tilde{\mathbf{x}}_{t_i}$ 。通过拟合一个能够弯曲以捕捉轨迹短期趋势的多项式，该更新直观地更紧密地跟随真实常微分方程解的曲线行为，尤其是在较大的步长下。

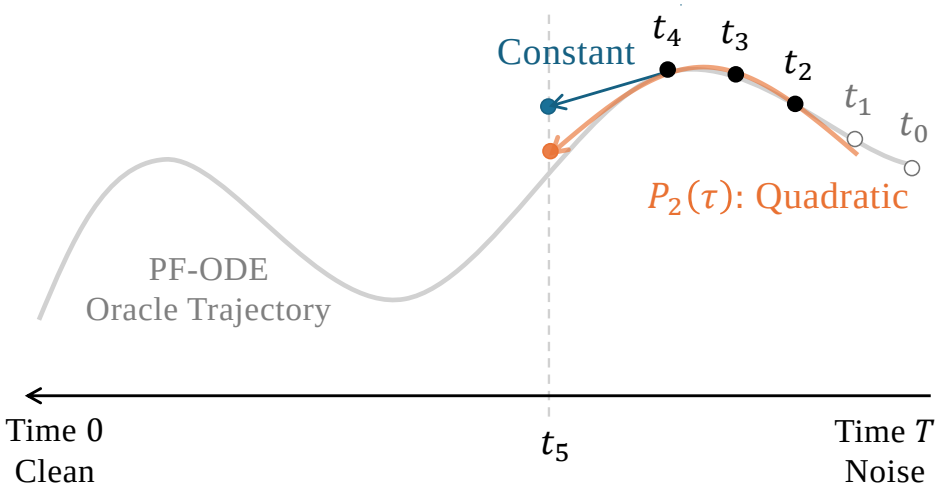


图 9.2: DEIS 作为多步方法的示意图。在三个历史锚点 t_2, t_3, t_4 处，DEIS 通过模型输出构建一条二次曲线，并对其进行解析积分，从而从 t_4 步进到 t_5（外推法）。这种高阶更新相比一阶方法（如 DDIM）能显著降低离散化误差，后者仅使用 t_4 处的值（对积分的常数近似）。

一个次数为 n 的更新需要 $n{+}1$ 个锚点。当它们可用时（足够历史, $i \geq n{+}1$ ），我们使用完整的次数为 n 的方案。在早期步骤中（历史不足, $i \leq n$ ），我们采用最高可行次数 $i{-}1$ 的相同构造，并随着锚点的积累逐步提高次数。下面我们将依次讨论这两种情形。

情形 I：$i = n+1, \dots, M$（充分的历史）。 与其仅依赖最新的估计值 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1}}, t_{i-1})$，DEIS 会将最近的 $n+1$ 次模型评估结果用作锚点，

$$(\tau_j, \mathbf{Y}_j) := \big(t_{i-1-j}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1-j}}, t_{i-1-j})\big), \quad j = 0, \dots, n.$$

作为锚点。将 $\tau \mapsto \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\tau, \tau)$ 视为沿轨迹的时间平滑函数，我们构造了次数为 n 的多项式（拉格朗日插值多项式）

$$P_n(\tau) = \sum_{j=0}^{n} \underbrace{\Big[\prod_{\substack{k=0 \\ k\neq j}}^{n} \frac{\tau - t_{i-1-k}}{t_{i-1-j} - t_{i-1-k}}\Big]}_{=:\ \ell_j^{(i)}(\tau)} \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}\big(\tilde{\mathbf{x}}_{t_{i-1-j}}, t_{i-1-j}\big)$$

这由构造可知，对每个锚点都满足 $P_n(\tau_j) = \mathbf{Y}_j$。

$$P_n\big(\tau_j\big) = \mathbf{Y}_j = \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}\big(\tilde{\mathbf{x}}_{t_{i-1-j}}, t_{i-1-j}\big), \quad j = 0, \dots, n.$$

每个 $\ell_j^{(i)}$ 满足

$$\ell_j^{(i)}\big(t_{i-1-m}\big) = \begin{cases} 1, & m = j, \\ 0, & m \neq j. \end{cases}$$

拉格朗日多项式提供了在新步长上的平滑外推法：

$$\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\tau, \tau) \approx P_n(\tau) = \sum_{j=0}^{n} \ell_j^{(i)}(\tau) \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}\big(\tilde{\mathbf{x}}_{t_{i-1-j}}, t_{i-1-j}\big), \quad \tau \in [t_{i-1}, t_i].$$

然后我们将 $P_r(\tau)$ 代入指数积分公式 (Equation (9.1.6)) 中的 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\tau, \tau)$：

$$\begin{aligned} &\int_{t_{i-1}}^{t_i} \frac{g^2(\tau)}{2\sigma_\tau} \mathcal{E}(\tau \to t_i) \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\tau, \tau)\, \mathrm{d}\tau \\ &\qquad \approx \sum_{j=0}^{r} \underbrace{\int_{t_{i-1}}^{t_i} \frac{g^2(\tau)}{2\sigma_\tau} \mathcal{E}(\tau \to t_i) \ell_j^{(i)}(\tau)\, \mathrm{d}\tau}_{=:\ C_{i,j}} \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1-j}}, t_{i-1-j}). \end{aligned}$$

权重 $C_{i,j}$ 由下式给出

$$C_{i,j} := \frac{1}{2} \int_{t_{i-1}}^{t_i} \frac{g^2(\tau)}{\sigma_\tau} \mathcal{E}(\tau \to t_i) \ell_j^{(i)}(\tau)\, \mathrm{d}\tau,$$

仅依赖于调度 $(\alpha_\tau, \sigma_\tau)$ 和网格 $\{t_i\}$ 。因此，一旦步骤确定，它们就可以以封闭形式确切地预先计算。

使用 $\mathcal{E}(t_{i-1} \to t_i)$ 精确积分线性部分，可得到 *AB-DEIS-r* 更新规则[3]，

$$\tilde{\mathbf{x}}_{t_i} = \mathcal{E}(t_{i-1} \to t_i)\tilde{\mathbf{x}}_{t_{i-1}} + \sum_{j=0}^{r} C_{i,j}\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1-j}}, t_{i-1-j}).$$

在标准光滑性假设下，它产生的局部截断误差为 $r{+}1$ 阶。

情况二：$i = 1, \ldots, n$（历史信息不足）。 在初始步骤中，仅有 i 个先前的点可用。因此我们将其阶数设为 $i{-}1$ 并定义

$$P_{i-1}(\tau) = \sum_{j=0}^{i-1} \ell_j^{(i)}(\tau)\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}\left(\tilde{\mathbf{x}}_{t_{i-1-j}}, t_{i-1-j}\right),$$

其中 $\ell_j^{(i)}$ 是在时间 $\{t_{i-1}, t_{i-2}, \ldots, t_0\}$ 的结点上构建的次数为 $i{-}1$ 的拉格朗日基函数。这匹配了所有可用的锚点，并在 $i \geq n{+}1$ 之后无缝过渡到全历史公式。

这是多步求解器中的标准“热启动”方法。当历史数据较短时，我们拟合数据允许的最高次多项式：使用一个锚点（$i{=}1$），采用次数为 0 （常数）；使用两个锚点（$i{=}2$），采用次数为 1 （线性的）；使用三个锚点（$i{=}3$），采用次数为 2 （二次的）；依此类推，直到达到目标次数 n 。实际上，随着可用历史数据的增加，我们逐步从单步预测过渡到真正的 $(n{+}1)$ 步预测。

Example: Special Cases of Lagrange Polynomials

> **When $r = 0$ (one anchor):**
>
> $$P_0(\tau) = \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1}}, t_{i-1}).$$
>
> This uses only the most recent value, so the approximation is flat in τ. It corresponds to a left-endpoint of the integrand.
>
> **When $r = 1$ (two anchors):** the Lagrange polynomial is a linear map passing

[3]“AB” 指的是经典的 Adams–Bashforth 方法族和指数时间差分多步方法 (**hochbruck2010exponential**)。

through the two pre-specified anchors.

$$P_1(\tau) = \underbrace{\frac{\tau - t_{i-2}}{t_{i-1} - t_{i-2}}}_{\ell_{i-1}(\tau)} \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1}}, t_{i-1}) + \underbrace{\frac{\tau - t_{i-1}}{t_{i-2} - t_{i-1}}}_{\ell_{i-2}(\tau)} \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-2}}, t_{i-2}).$$

Here $\ell_{i-1}(\tau)$ and $\ell_{i-2}(\tau)$ are the Lagrange basis weights. They satisfy the interpolation (nodal) conditions $P_1(t_{i-1}) = \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1}}, t_{i-1})$ and $P_1(t_{i-2}) = \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-2}}, t_{i-2})$, and with $\ell_{i-1}(\tau) + \ell_{i-2}(\tau) = 1$.

AB-DEIS-n 版本更新摘要。 结合两种情况，即充分的历史信息和热启动（历史信息不足），得到 *AB-DEIS-n* 更新[4] 其中 n 为多项式次数（最多使用 n+1 次之前的求值结果）如下：

$$\tilde{\mathbf{x}}_{t_i} = \mathcal{E}(t_{i-1} \to t_i)\, \tilde{\mathbf{x}}_{t_{i-1}} + \sum_{j=0}^{\min\{n,\, i-1\}} C_{i,j}\, \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1-j}},\, t_{i-1-j}),$$

系数

$$C_{i,j} := \frac{1}{2} \int_{t_{i-1}}^{t_i} \frac{g^2(\tau)}{\sigma_\tau}\, \mathcal{E}(\tau \to t_i) \left[\prod_{\substack{k=0 \\ k \neq j}}^{\min\{n,\, i-1\}} \frac{\tau - t_{i-1-k}}{t_{i-1-j} - t_{i-1-k}} \right] d\tau.$$

当 $i \geq n$+1 （充分的历史信息），$\min\{n,\, i-1\} = n$ 且步长在标准光滑性假设下达到局部截断误差 $\mathcal{O}(h^{n+1})$。在热启动（$i \leq n$）期间，$\min\{n,\, i-1\} = i-1$ 且每步阶数为 $\mathcal{O}(h^{\min\{n,\, i-1\}+1})$，逐步提升直至达到完整阶数。

然而，非常大的 n 通常会由于插值病态性、噪声放大以及更严格的稳定性约束而降低性能；较小的次数（例如 $n \in \{1, 2, 3\}$）通常能提供最佳的准确率-稳定性权衡。

正如我们将在下一小节中看到的，特殊情况 n=0 退化为指数欧拉/DDIM。

[4] “AB” 指 Adams–Bashforth 系列的指数时间差分多步方法 (**hochbruck2010exponential**)。

9.3.3 DDIM = AB-DEIS-0

我们观察到当 $n=0$ 时（即常数多项式），系数简化为：

$$C_{i0}=\frac{1}{2}\int_{t_{i-1}}^{t_i}\frac{g^2(\tau)}{\sigma_\tau}\mathcal{E}(\tau\to t_i)\mathrm{d}\tau.$$

代入更新公式得到零阶 AB-DEIS 格式：

$$\begin{aligned}\tilde{\mathbf{x}}_{t_i}&=\mathcal{E}(t_{i-1}\to t_i)\tilde{\mathbf{x}}_{t_{i-1}}+C_{i0}\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1}},t_{i-1})\\&=e^{\int_{t_{i-1}}^{t_i}f(u)\,\mathrm{d}u}\tilde{\mathbf{x}}_{t_{i-1}}+\Big(\int_{t_{i-1}}^{t_i}\frac{g^2(\tau)}{2\sigma_\tau}e^{\int_{\tau}^{t_i}f(u)\,\mathrm{d}u}\mathrm{d}\tau\Big)\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1}},t_{i-1}).\end{aligned}\tag{9.3.1}$$

这是确切的指数-欧拉步长（时间常数 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}$ 在 $[t_{i-1},t_i]$ 内），与确定性 DDIM 更新一致。我们下面正式陈述这一对应关系。

Proposition 9.3.1: DDIM = AB-DEIS-0

Equation (9.3.1) 与 Equation (9.2.2) 中的 DDIM 更新完全相同。

9.4 DPM-Solver

DPM-Solver 系列，包括 DPM-Solver (**lu2022dpm**)、DPM-Solver++ (**lu2022dpm2**) 以及 DPM-Solver-v3 (**zheng2023dpm**)，代表了 PF-ODE 求解器的重大进展。其目标十分简单：在远少于传统步骤的情况下实现相近的样本质量。实际上，这些方法将 DDIM 所需的步数从超过 50 减少到约 10 - 15，使生成过程更加高效。此外，DPM-Solver++ 和 DPM-Solver-v3 专门设计用于处理无分类器引导（CFG）（参见 Section 8.3）以实现条件生成。在本节中，我们首先解释核心的 DPM-Solver (**lu2022dpm**)；其扩展部分见 Section 9.5 和 Section 9.7。

DPM-Solver 的高层次思路。 与 DEIS 类似，DPM-Solver 从 PF-ODE 的线性形式出发，采用 ϵ-预测参数化，使用指数积分器（常数变易法）表示法在 Equation (A.1.2) 中进行计算：

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = \frac{\alpha_t'}{\alpha_t}\mathbf{x}_t - \sigma_t\left(\frac{\alpha_t'}{\alpha_t} - \frac{\sigma_t'}{\sigma_t}\right)\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t, t). \tag{9.4.1}$$

核心思想是通过半对数信噪比对时间进行重参数化，使得指数积分器公式中的非线性项变为一个指数加权积分。这种表示方式在 λ 中允许低成本的泰勒展开，从而自然地导出高阶更新规则。我们很快将提供一个直观的解释，说明为何这种重参数化是有效的。

9.4.1 DPM-Solver 的洞察：通过对数信噪比实现时间重参数化

在半线性结构的基础上，DPM-solver 的一个关键见解是，扩散模型中数值积分的的标准时间参数化 t 并非最优。他们提出改用 半对数信噪比（half-log SNR）来重新参数化时间。

$$\lambda_t \coloneqq \frac{1}{2}\log\frac{\alpha_t^2}{\sigma_t^2} = \log\frac{\alpha_t}{\sigma_t}, \tag{9.4.2}$$

遵循 VDM (**kingma2021variational**) 的对数信噪比参数化。这种变量变换简化了非线性被积函数，从而使得高阶模型估计更加易处理且准确。

PF-ODE 中的变量变换至对数信噪比 我们现在使用半对数信噪比 $\lambda_t := \log(\alpha_t/\sigma_t)$ 重参数化时间。对于常见的噪声调度，λ_t 在 t 上严格递减。在此假设下，其存在

反函数 $t_\lambda(\cdot)$，将 λ 映射到 t，满足

$$t = t_\lambda(\lambda(t)).$$

然后我们将 $\mathbf{x}$ 和 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}$ 的下标从 t 变为 λ。帽子符号 $(\hat{\cdot})$ 表示该量是以 λ 表示的。更精确地，我们定义：

$$\begin{aligned} \hat{\mathbf{x}}_\lambda &:= \mathbf{x}_{t_\lambda(\lambda)}, \\ \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda) &:= \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{t_\lambda(\lambda)}, t_\lambda(\lambda)). \end{aligned} \tag{9.4.3}$$

通过从 t 到 λ_t 的变量变换，PF-ODE 的确切解 $\widetilde{\boldsymbol{\Psi}}_{s\to t}$ 在 Equation (9.4.1) 中变为：

Proposition 9.4.1: 指数权重确切解

给定初始值 $\mathbf{x}_s$ 于时刻 $s > 0$，PF-ODE 在时刻 $t \in [0, s]$ 的确切解 $\widetilde{\boldsymbol{\Psi}}_{s\to t}(\mathbf{x}_s)$ 可重新表述为：

$$\widetilde{\boldsymbol{\Psi}}_{s\to t}(\mathbf{x}_s) = \frac{\alpha_t}{\alpha_s}\mathbf{x}_s - \alpha_t \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda)\, \mathrm{d}\lambda. \tag{9.4.4}$$

Proof for Proposition.

虽然可直接对 Equation (9.4.1) 进行变量变换得到结果，但为清晰完整起见，我们在此提供另一种推导。利用关系式 $g^2(t) = -2\sigma_t^2 \frac{\mathrm{d}\lambda_t}{\mathrm{d}t}$，Equation (9.4.1) 可改写为：

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = \frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}\mathbf{x}_t - \sigma_t \frac{\mathrm{d}\lambda_t}{\mathrm{d}t}\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t, t).$$

应用链式法则：

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = \frac{\mathrm{d}\hat{\mathbf{x}}_\lambda}{\mathrm{d}\lambda}\frac{\mathrm{d}\lambda_t}{\mathrm{d}t} \quad \text{且} \quad \frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t} = \frac{\mathrm{d}\log\alpha_\lambda}{\mathrm{d}\lambda}\frac{\mathrm{d}\lambda_t}{\mathrm{d}t},$$

t 中的 ODE 被变换为 λ 中的 ODE 如下：

$$\begin{aligned}\frac{\mathrm{d}\hat{\mathbf{x}}_\lambda}{\mathrm{d}\lambda} &= \Big(\frac{\mathrm{d}\lambda_t}{\mathrm{d}t}\Big)^{-1}\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}\\ &= \Big(\frac{\mathrm{d}\lambda_t}{\mathrm{d}t}\Big)^{-1}\Big[\frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}\mathbf{x}_t - \sigma_t\frac{\mathrm{d}\lambda_t}{\mathrm{d}t}\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t,t)\Big]\\ &= \Big(\frac{\mathrm{d}\lambda_t}{\mathrm{d}t}\Big)^{-1}\Big[\frac{\mathrm{d}\log\alpha_\lambda}{\mathrm{d}\lambda}\frac{\mathrm{d}\lambda_t}{\mathrm{d}t}\hat{\mathbf{x}}_\lambda - \sigma_\lambda\frac{\mathrm{d}\lambda_t}{\mathrm{d}t}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda,\lambda)\Big]\\ &= \frac{\mathrm{d}\log\alpha_\lambda}{\mathrm{d}\lambda}\hat{\mathbf{x}}_\lambda - \sigma_\lambda\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda,\lambda).\end{aligned}$$

因此，变换后的 ODE 变为 Equation (9.4.5)。随后可对 Equation (9.4.5) 应用相同的“指数积分器（EI）”技术来推导 Equation (9.4.4)。

在 λ –time 中，模型出现在一个指数加权积分内部，

$$\int_{\lambda_s}^{\lambda_t} e^{-\lambda}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda,\lambda)\,\mathrm{d}\lambda,$$

其中，$e^{-\lambda}$ 因子产生闭式系数并平滑被积函数，这正是高阶局部近似所要求的。

等价地，将变量从 t 变为 λ 会将 PF-ODE 变换为如下微分形式（详见前一命题的推导）：

$$\frac{\mathrm{d}\hat{\mathbf{x}}_\lambda}{\mathrm{d}\lambda} = \frac{\alpha'_\lambda}{\alpha_\lambda}\hat{\mathbf{x}}_\lambda - \sigma_\lambda\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda,\lambda). \tag{9.4.5}$$

为什么需要重新参数化时间的直觉？ 对于严格单调的 $\lambda(t)$，一阶变量变换给出

$$\Delta t \approx \frac{\Delta\lambda}{|\lambda'(t)|}.$$

因此，对于固定的 $\Delta\lambda$，当 $|\lambda'(t)|$ 较大时（即 λ 随 t 快速变化时），诱导的 Δt 较小；当 $|\lambda'(t)|$ 较小时，Δt 较大。此重参数化不改变 PF–ODE 的解路径，仅改变其速度：

$$\frac{\mathrm{d}\hat{\mathbf{x}}_\lambda}{\mathrm{d}\lambda} = \frac{1}{\lambda'(t)}\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}.$$

因此，在 $|\lambda'(t)|$ 较大的区域中，λ –domain 导数被 $1/|\lambda'(t)|$ 缩放，通常会使被积函数在均匀的 λ 网格上更平滑，便于近似。（大 $|\lambda'(t)|$ 的精确位置取决于所选的调度方案。）

从概念上讲，当过程接近复杂（数据）分布时，我们可能希望分配更多的时间步。以下是两种简单的调度方案，可说明这种效果：

- $(\alpha_t, \sigma_t) = (1-t, t)$：这对应于 FM 调度器。然后

$$\lambda(t) = \log \frac{1-t}{t}, \quad \lambda'(t) = -\frac{1}{t(1-t)}, \quad \Delta t \approx \Delta\lambda t(1-t).$$

  因此，两端附近的步长很小（$t \to 0, 1$），而中间时段的步长最大。

- $(\alpha_t, \sigma_t) = (1, t)$:这是在 Section D.6 中提出的 EDM 调度器 (**karras2022elucidating**)。如果我们直接将自变量作为噪声水平 $t = \sigma_t$，那么

$$\lambda(t) = \log \frac{1}{t}, \quad \lambda'(t) = -\frac{1}{t}, \quad \Delta t \approx \Delta\lambda t.$$

  λ 中的均匀间距在 t 中是几何的，或等价地在方差中（在小 t /高信噪比时许多小步长，大 t 时步长较粗）。

9.4.2 用泰勒展开估算积分

DEIS 通过拉格朗日插值法对过去评估的被积函数进行拟合。DPM-Solver 则在 λ 处使用局部泰勒展开：其计算成本更低，与 λ 参数化所诱导的光滑性一致，并能得出闭式步长系数。我们将在下文详细介绍。

从 Equation (9.4.4) 出发，从时间 s 的前一点 $\tilde{\mathbf{x}}_s$ 开始，时间 t 处的解 $\tilde{\mathbf{x}}_t$ 由下式给出

$$\tilde{\mathbf{x}}_t = \frac{\alpha_t}{\alpha_s}\tilde{\mathbf{x}}_s - \alpha_t \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda)\, \mathrm{d}\lambda. \tag{9.4.6}$$

因此，我们得出以下形式的积分近似：

$$\int_{\lambda_s}^{\lambda_t} e^{-\lambda} \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda)\, \mathrm{d}\lambda.$$

在区间 $\lambda \in [\lambda_s, \lambda_{t_i}]$ 上，我们对 Equation (9.4.6) 中的被积函数 $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda)$ 以 λ 为变量进行泰勒展开来近似。对于 $n \geq 1$ ，关于 λ_s 的 $(n-1)$ 阶泰勒展开式为

$$\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda) = \sum_{k=0}^{n-1} \frac{(\lambda-\lambda_s)^k}{k!} \hat{\boldsymbol{\epsilon}}^{(k)}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_{\lambda_s}, \lambda_s) + \mathcal{O}((\lambda-\lambda_s)^n),$$

其中，关于 λ 的第 k 阶全导数记为

$$\hat{\boldsymbol{\epsilon}}^{(k)}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda) \coloneqq \frac{\mathrm{d}^k}{\mathrm{d}\lambda^k}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda).$$

将该展开式代入 Equation (9.4.6) 中的积分，可得到一个闭式近似，该近似定义了 n 阶求解器，称为 *DPM-Solver-n*。

从上一步的估计 $\tilde{\mathbf{x}}_s$ 开始，

$$\tilde{\mathbf{x}}_t = \frac{\alpha_t}{\alpha_s}\tilde{\mathbf{x}}_s - \alpha_t \sum_{k=0}^{n-1} \hat{\boldsymbol{\epsilon}}^{(k)}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_{\lambda_s}, \lambda_s)\ C_k + \mathcal{O}(h^{n+1}), \tag{9.4.7}$$

此处，我们记 $h \coloneqq \lambda_t - \lambda_s$ ，并定义：

$$C_k \coloneqq \int_{\lambda_s}^{\lambda_t} e^{-\lambda}\frac{(\lambda - \lambda_s)^k}{k!}\,\mathrm{d}\lambda.$$

C_k 可以通过应用分部积分 k 次解析地预先计算。

我们注意到变量变换 $t \mapsto \lambda$ 用于平滑被积函数并推导系数，而求解器返回的是在 t-网格上的估计值 $\tilde{\mathbf{x}}_t$。

下面，我们以 DPM-Solver-1 为例进行说明。

Example: DPM-Solver-1

Consider $n = 1$ (first order) for demonstration . Starting from the previous estimated point $\tilde{\mathbf{x}}_s$, Equation (9.4.7) simplifies to:

$$\begin{aligned}\tilde{\mathbf{x}}_t &= \frac{\alpha_t}{\alpha_s}\tilde{\mathbf{x}}_s - \alpha_t \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_s, s)\int_{\lambda_s}^{\lambda_t} e^{-\lambda}\,\mathrm{d}\lambda + \mathcal{O}(h^2)\\ &= \frac{\alpha_t}{\alpha_s}\tilde{\mathbf{x}}_s - \sigma_t(e^h - 1)\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_s, s) + \mathcal{O}(h^2).\end{aligned} \tag{9.4.8}$$

The above formula is exactly the DDIM update; we prove the equivalence in Proposition 9.4.2. ∎

DPM-Solver-n 与 $n \geq 2$ 需要计算 k^{th} 导数$\hat{\boldsymbol{\epsilon}}^{(k)}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda)$ ，其中 $k \leq n-1$ 。然而，在实际应用中直接计算高阶导数的计算成本较高。**lu2022dpm** 还提出了这些导数的高效近似方法，将在下一小节中详细说明。

9.4.3 DPM-Solver-n 的实现

DPM-Solver-n 与 $n \geq 2$。 在实际应用中，实现高阶 DPM-Solver-n 需要以下步骤：

- 预先计算系数 C_k ；
- 通过用 k^{th} 导数$\hat{\boldsymbol{\epsilon}}^{(k)}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda)$ 的 $k \leq n-1$ 近似来规避确切的高阶导数计算，这在常微分方程文献中是一个被广泛研究的难题 (**hochbruck2005explicit**; **luan2021efficient**)。一种常见的策略是有限差分近似。

我们现在详细阐述前两点。

预计算 C_k 。 令 s 和 t 分别表示起始时间和终止时间，定义 $h := \lambda_t - \lambda_s$ 。从 $\mathbf{x}_s$ 开始，方程 Equation (9.1.6) 确切解的解析展开式为：

$$\mathbf{x}_t = \frac{\alpha_t}{\alpha_s}\mathbf{x}_s - \sigma_t \sum_{k=0}^{n-1} h^{k+1}\varphi_{k+1}(h)\hat{\boldsymbol{\epsilon}}^{(k)}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_{\lambda_s}, \lambda_s) + \mathcal{O}(h^{n+1}), \tag{9.4.9}$$

其中每个 $\varphi_{k+1}(\cdot)$ 均具有闭式。对于 $k = 0, 1, 2$，它们是：

$$\varphi_1(h) = \frac{e^h - 1}{h}, \qquad \varphi_2(h) = \frac{e^h - h - 1}{h^2}, \qquad \varphi_3(h) = \frac{e^h - \frac{h^2}{2} - h - 1}{h^3}.$$

Example: DPM-Solver-2/3 with Exact Derivatives

For $n = 3$ and discrete time steps with $h := \lambda_t - \lambda_s$, the expansion becomes:

$$\begin{aligned}\mathbf{x}_t = \frac{\alpha_t}{\alpha_s}\mathbf{x}_s &- \sigma_t \left(e^h - 1\right) \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_{\lambda_s}, \lambda_s) \\ &- \sigma_t \left(e^h - h - 1\right) \hat{\boldsymbol{\epsilon}}^{(1)}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_{\lambda_s}, \lambda_s) \\ &- \sigma_t \left(e^h - \frac{h^2}{2} - h - 1\right) \hat{\boldsymbol{\epsilon}}^{(2)}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_{\lambda_s}, \lambda_s) \\ &+ \mathcal{O}(h^4).\end{aligned} \tag{9.4.10}$$

对 $k \leq n-1$ 逼近$\hat{\boldsymbol{\epsilon}}^{(k)}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda)$ 。 对于 $n \geq 2$ ，按照单步常微分方程求解器的标准方法 (**atkinson2009numerical**)，**lu2022dpm** 在 s 与 t 之间引入一个

中间时间步 s^{mid}，利用在 s 与 s^{mid} 处的函数值来近似高阶导数。我们以 $n=2$ 为例进行说明。

令 $\gamma \in (0,1]$ 为一个超参数，用于指定对数信噪比区间 $[\lambda_s, \lambda_t]$ 内的插值点。给定在 s 处的估计值 $\tilde{\mathbf{x}}_s$ ，定义

$$s^{\mathrm{mid}} = t_\lambda\left(\lambda_s + \gamma h\right), \quad \text{where } h := \lambda_t - \lambda_s,$$

中间估计值由下式给出：

$$\mathbf{x}^{\mathrm{mid}} = \frac{\alpha_{s^{\mathrm{mid}}}}{\alpha_s}\tilde{\mathbf{x}}_s - \sigma_{s^{\mathrm{mid}}}\left(e^{\gamma h}-1\right)\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_s, s).$$

这给出了如下二阶近似：

$$\begin{aligned}\tilde{\mathbf{x}}_t = \frac{\alpha_t}{\alpha_s}\tilde{\mathbf{x}}_s - \sigma_t\left(e^{h}-1\right)\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_s, s) \\ - \frac{\sigma_t}{\gamma h}\left(e^{h}-h-1\right)\left(\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}^{\mathrm{mid}}, s^{\mathrm{mid}}) - \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_s, s)\right) + \mathcal{O}(h^3).\end{aligned} \tag{9.4.11}$$

使用 $\gamma = \frac{1}{2}$，Algorithm 5 中的两步更新等价于 Equation (9.4.11)，其局部截断误差为 $\mathcal{O}(h^3)$。

Algorithm 5 DPM-Solver-2 (with $\gamma = \frac{1}{2}$).

Input: initial value $\mathbf{x}_T$, time steps $\{t_i\}_{i=0}^M$, model $\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}$

$\tilde{\mathbf{x}}_{t_0} \leftarrow \mathbf{x}_T$
for $i \leftarrow 1$ to M **do**
  $h_i \leftarrow \lambda_{t_i} - \lambda_{t_{i-1}}$
  $s_i^{\mathrm{mid}} \leftarrow t_\lambda\left(\frac{\lambda_{t_{i-1}} + \lambda_{t_i}}{2}\right)$
  $\mathbf{x}_i^{\mathrm{mid}} \leftarrow \frac{\alpha_{s_i^{\mathrm{mid}}}}{\alpha_{t_{i-1}}}\tilde{\mathbf{x}}_{t_{i-1}} - \sigma_{s_i^{\mathrm{mid}}}\left(e^{\frac{h_i}{2}}-1\right)\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1}}, t_{i-1})$
  $\tilde{\mathbf{x}}_{t_i} \leftarrow \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}}\tilde{\mathbf{x}}_{t_{i-1}} - \sigma_{t_i}\left(e^{h_i}-1\right)\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_i^{\mathrm{mid}}, s_i^{\mathrm{mid}})$
end for
return $\tilde{\mathbf{x}}_{t_M}$

Remark.

> 在 Equation (9.4.11) 中，差商
>
> $$\hat{\boldsymbol{\epsilon}}^{(1)}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_{\lambda_s}, \lambda_s) \approx \frac{\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}^{\mathrm{mid}}, s^{\mathrm{mid}}) - \boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_s, s)}{\gamma h}$$
>
> 近似表示沿轨迹的模型总 λ——导数。该近似精度达到 $\mathcal{O}(h)$，并在 Equation (9.4.11) 中乘以确切的 φ_2 系数 $e^h - h - 1 = \mathcal{O}(h^2)$。因此，所得贡献仅为 $\mathcal{O}(h^3)$，故整体方案对所有 $\gamma \in (0, 1]$ 均能达到二阶精度。每步计算需进行两次模型评估：一次在 $(\tilde{\mathbf{x}}_s, s)$，一次在预测中点 $(\mathbf{x}^{\mathrm{mid}}, s^{\mathrm{mid}})$。插值参数 γ 不影响精度阶数，但会改变误差常数：将 $\gamma = \frac{1}{2}$ 设置为对称格式通常可使常数最小化，因此实践中多采用中点版本。

对于高阶 DPM-Solver-n 与 $n \geq 3$ ，采用类似的方法，利用中间时间步以有限差分方式近似高阶导数。详细方法请参见原始 DPM 论文。

对于熟悉数值常微分方程求解器的读者来说，DPM-Solver 可以被视为一种针对半线性 PF-ODE 的单步 *指数积分器*，并结合了时间变量到（半）对数–信噪比的变换。其二阶和三阶变体为采用每步内少量阶段模型评估的指数龙格-库塔型格式。

实施细节：采样时间步的选择。 为了进行采样，求解器必须首先预先定义一组时间步 $\{t_i\}_{i=0}^{M}$。**lu2022dpm** 建议根据 *log–SNR* 时间中的均匀间距 λ_t 来选择这些步骤，其中

$$\lambda_{t_i} = \lambda_T + \frac{i}{M}(\lambda_0 - \lambda_T), \quad i = 0, \ldots, M.$$

这与早期直接在物理时间变量 t 中使用均匀间距的方法（**ho2020denoising**; **song2020score**）不同。经验上，即使步数很少，DPM-Solver 在使用均匀 λ 间距时也能生成高质量的样本。[5].

从概念上讲，这可以从几何角度理解：局部泰勒近似的准确率取决于动力学在 λ 中演化的平滑程度。因此，λ 中的均匀间距会在整个轨迹上产生近似均匀的局部误差，这使得在信号占主导的区域（高 SNR）t 中采用更精细（更稠密）的步长，而在噪声占主导的区域采用更粗糙（更稀疏）的步长。

尽管推导过程在 λ –空间中进行，且 PF–ODE 在该领域中以方便的半线性形式表述，但预训练模型和噪声调度 (α_t, σ_t) 通常相对于原始时间变量 t 定义。在采样过程中，求解器选择在 λ 中均匀分布的结点以保证数值稳定性，但所有

[5] 另外，自适应步长策略通过结合不同阶数的求解器动态调整时间步长；参见 **lu2022dpm** 的附录 C。

更新方程均以 t 表示。每当需要评估模型或获取调度值时，所选的 λ 结点会被映射回相应的变量，例如物理时间 $t = t_\lambda(\lambda)$ 或方差参数 σ_t ，具体取决于模型的参数化方式（参见，例如，Algorithm 5）。

9.4.4 DDIM = DPM-Solver-1

对于固定的调度 (α_t, σ_t) ，DPM-Solver-1 步骤与确定性 DDIM ($\eta = 0$) 更新一致，与时间参数化方式（物理时间 t 或 log–SNR 时间 λ ）无关；详见下述正式声明。

Proposition 9.4.2: DDIM 即为 DPM-Solver-1

DDIM 的更新规则，如 Equation (9.2.2) 所示，与 DPM-Solver-1 的更新规则（见 Equation (9.4.8)）完全相同。

Proof for Proposition.

根据 λ 的定义，我们有

$$\frac{\sigma_s}{\alpha_s} = e^{-\lambda_s} \quad \text{and} \quad \frac{\sigma_t}{\alpha_t} = e^{-\lambda_t}. \tag{9.4.12}$$

将这些表达式连同 $h = \lambda_t - \lambda_s$ 代入 Equation (9.2.2)，即可还原出 Equation (9.4.8) 中的更新规则，从而完成等价性证明。

上述命题或许可以解释为何 DDIM 在 t-参数化下优于传统的欧拉方法：它通过更合适的 λ-参数化，有效利用了扩散常微分方程的半线性特性。

Remark.

当《得分随机微分方程》论文发表时，龙格-库塔法（RK45）通常被用于求解 Equation (4.2.5) 中的标准概率流常微分方程，但其漂移项的半线性特性尚未得到利用。尽管 DPM-Solver-k（$k \geq 2$）与龙格-库塔方法存在关联，但该方法通过时间重参数化显式利用了这种半线性特性。这解释了为何 DPM-Solver 能以远更少的函数评估次数实现高阶准确率，将通常需要数百步的 DDIM 调度缩减至约 10-15 步，同时保持高质量的样本生成效果。

9.4.5 DPM-Solver-2 与经典 Heun 更新方法的讨论

在 Section 9.2.2 中，我们看到对 PF–ODE 的不同参数化会导致对经典欧拉型更新的不同解释：

$$\mathbf{v}\text{-prediction:} \quad \text{Euler} = \text{DDIM},$$

$$\boldsymbol{\epsilon}\text{-, } \mathbf{x}\text{-, or } \mathbf{s}\text{-prediction:} \quad \text{exp–Euler} = \text{DDIM} \neq \text{plain Euler}.$$

在本小节中，我们通过考察经典 *Heun's method* 与二阶 DPM–Solver 在四种参数化下的类比关系，进一步阐明这种联系。

为了铺垫，我们简要回顾一下海恩法（参见 Section A.1.4）。海恩法是一种二阶求解器，通过预测-校正方案对欧拉法进行改进：它首先使用欧拉法对步长终点进行预测，计算该点的斜率，然后利用起始斜率和预测斜率的平均值进行更新。直观上，它通过沿着区间内的平均斜率前进（梯形面积），实现了远高于简单欧拉法的准确率。

我们在对数信噪比时间 λ 中工作，其中 PF–ODE 可以表示为一个简单的“线性 + 非线性”形式：

$$\frac{\mathrm{d}\hat{\mathbf{x}}(\lambda)}{\mathrm{d}\lambda} = \underbrace{L(\lambda)\hat{\mathbf{x}}(\lambda)}_{\text{linear part}} + \underbrace{\mathbf{N}\big(\hat{\mathbf{x}}(\lambda), \lambda\big)}_{\text{nonlinear part}},$$

其中标量 $L(\lambda)$ 由噪声调度决定，$\mathbf{N}(\cdot, \lambda)$ 收集了非线性部分。该结构自然来源于 Equation (6.3.2)：$\boldsymbol{\epsilon}$ -、$\mathbf{x}$ - 和 $\mathbf{s}$ -预测参数化形式会产生非零的 $L(\lambda)$ ，从而形成半线性形式。相比之下，$\mathbf{v}$ -预测对应于 $L(\lambda) \equiv 0$ （因此 $\mathbf{N} = \mathbf{v}$ ），不包含显式的线性项。

在接下来的讨论中，我们首先回顾不考虑任何半线性结构的普通 Heun 更新，然后引入指数 Heun 更新，该方法专为半线性常微分方程设计，对线性部分进行确切处理，类似于 Equations (9.1.7) and (9.1.8) 中的指数 Euler 步骤。最后，我们在四种参数化下将两种 Heun 更新与 DPM-Solver-2 联系起来，并得出结论：

$$\mathbf{v}\text{-prediction:} \quad \text{Heun} = \text{DPM-Solver-2},$$

$$\boldsymbol{\epsilon}\text{-, } \mathbf{x}\text{-, or } \mathbf{s}\text{-prediction:} \quad \text{exp-Heun} = \text{DPM-Solver-2} \neq \text{plain Heun}.$$

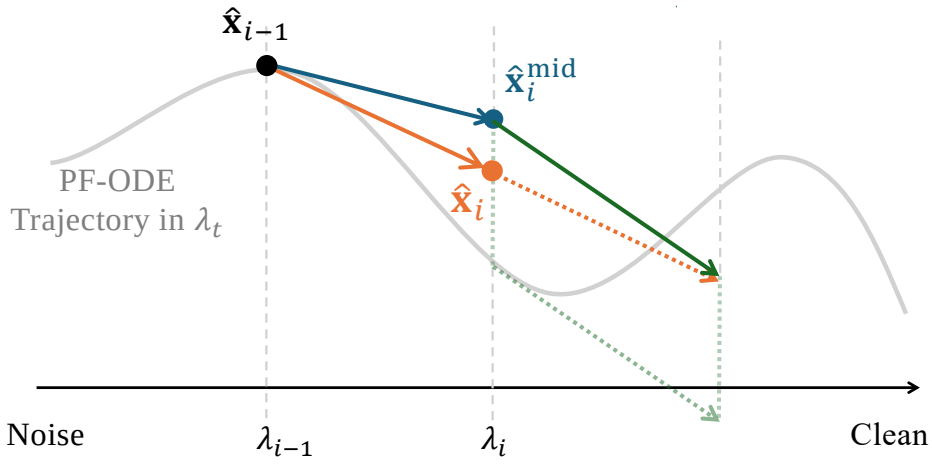


图 9.3: 对数信噪比时间下的普通 Heun 更新。从先前的状态 $\hat{\mathbf{x}}_{i-1}$ 在 λ_{i-1} 处开始，预测器步骤（蓝色箭头）执行显式欧拉移动 $h\mathbf{F}(\hat{\mathbf{x}}_{i-1}, \lambda_{i-1})$ 以获得中间估计 $\hat{\mathbf{x}}_i^{\text{mid}}$。在该预测点处，校正器步骤评估新的斜率 $h\mathbf{F}(\hat{\mathbf{x}}_i^{\text{mid}}, \lambda_i)$（绿色箭头），并通过平行四边形构造结合两个斜率：虚线橙色对角线表示从 $\hat{\mathbf{x}}_{i-1}$ 出发的向量和 $h\Big(\mathbf{F}(\hat{\mathbf{x}}_{i-1}, \lambda_{i-1}) + \mathbf{F}(\hat{\mathbf{x}}_i^{\text{mid}}, \lambda_i)\Big)$，而实线橙色箭头是其半对角线，方向相同但长度减半。此过程实现了在对数信噪比时间下对 PF-ODE 轨迹的普通 Heun 积分。

简单欧拉更新。 记 $\lambda_i := \lambda_{t_i}$，则 $\{\lambda_i\}_{i=0}^M$ 为对数信噪比域中的递增网格，且设 $h := \lambda_i - \lambda_{i-1} > 0$。令 $\hat{\mathbf{x}}_{i-1}$ 表示对数信噪比时间中的前一迭代值。直接应用于完整漂移

$$\mathbf{F}(\hat{\mathbf{x}}, \lambda) := L(\lambda)\hat{\mathbf{x}} + \mathbf{N}(\hat{\mathbf{x}}, \lambda),$$

对数信噪比时间中的普通 Heun 更新公式为

$$\begin{aligned} \text{Predict:} \quad & \hat{\mathbf{x}}_i^{\text{mid}} = \hat{\mathbf{x}}_{i-1} + h\mathbf{F}(\hat{\mathbf{x}}_{i-1}, \lambda_{i-1}), \\ \text{Correct:} \quad & \hat{\mathbf{x}}_i = \hat{\mathbf{x}}_{i-1} + \tfrac{h}{2}\Big(\mathbf{F}(\hat{\mathbf{x}}_{i-1}, \lambda_{i-1}) + \mathbf{F}(\hat{\mathbf{x}}_i^{\text{mid}}, \lambda_i)\Big). \end{aligned} \tag{9.4.13}$$

指数型 Heun 更新（用于半线性 PF-ODE）。 使用 *指数积分器技术*，其思想是分别处理常微分方程的线性和非线性部分。线性项 $L(\lambda)\hat{\mathbf{x}}$ 在步长内被精确积分，而非线性项 $\mathbf{N}(\hat{\mathbf{x}}, \lambda)$ 仅通过在步长内平均其影响来近似。

为了简洁地表达这一点，我们引入以下量

$$\mathcal{E} := \int_{\lambda_{i-1}}^{\lambda_i} L(\tau)\, \mathrm{d}\tau,$$

表示线性系数 $L(\lambda)$ 在区间 $[\lambda_{i-1}, \lambda_i]$ 上的总贡献。利用 $\mathcal{E}$ ，我们定义两个辅助系数 $c_1(\mathcal{E})$ 和 $c_2(\mathcal{E})$ ，以同时处理 $\mathcal{E}$ 非零和为零的情况：

$$c_1(\mathcal{E}) = \begin{cases} \frac{e^{\mathcal{E}}-1}{\mathcal{E}}, & \text{if } \mathcal{E} \neq 0, \\ 1, & \text{if } \mathcal{E} = 0, \end{cases} \qquad c_2(\mathcal{E}) = \begin{cases} \frac{e^{\mathcal{E}}-1-\mathcal{E}}{\mathcal{E}^2}, & \text{if } \mathcal{E} \neq 0, \\ \frac{1}{2}, & \text{if } \mathcal{E} = 0. \end{cases}$$

第二种情况仅确保当线性项消失（$L(\lambda) = 0$）时保持连续性，从而使公式仍然有效，并平滑地简化为如 Equation (9.4.13) 所示的标准 Heun 更新。

利用这些系数，指数-赫恩格式的一个更新步骤可表示为：

$$\begin{aligned} \text{Predict:} \quad & \hat{\mathbf{x}}_i^{\text{mid}} = e^{\mathcal{E}} \hat{\mathbf{x}}_{i-1} + h c_1(\mathcal{E}) \mathbf{N}(\hat{\mathbf{x}}_{i-1}, \lambda_{i-1}), \\ \text{Correct:} \quad & \hat{\mathbf{x}}_i = e^{\mathcal{E}} \hat{\mathbf{x}}_{i-1} + h c_1(\mathcal{E}) \mathbf{N}(\hat{\mathbf{x}}_{i-1}, \lambda_{i-1}) \\ & \qquad + h c_2(\mathcal{E}) \Big(\mathbf{N}(\hat{\mathbf{x}}_i^{\text{mid}}, \lambda_i) - \mathbf{N}(\hat{\mathbf{x}}_{i-1}, \lambda_{i-1}) \Big). \end{aligned} \tag{9.4.14}$$

当 $L(\lambda) \equiv 0$ 时，系数简化为 $c_1 = 1$ 和 $c_2 = \frac{1}{2}$ ，该方法退化为 Equation (9.4.13) 中的普通 Heun 求解器。

当 $L(\lambda) \neq 0$ 时，更新的指数积分形式对线性项进行了确切积分，而普通的 Heun 方法仅提供近似。为了说明这一点，将指数项在小步长 $h = \lambda_i - \lambda_{i-1} > 0$ 下展开。由于

$$\mathcal{E} = \int_{\lambda_{i-1}}^{\lambda_i} L(\tau)\, \mathrm{d}\tau = h L(\lambda_{i-1}) + \mathcal{O}(h^2),$$

我们可以将 $\mathcal{E}$ 视为一个阶为 $\mathcal{O}(h)$ 的小量。泰勒展开式给出：

$$e^{\mathcal{E}} = 1 + \mathcal{E} + \tfrac{\mathcal{E}^2}{2} + \mathcal{O}(\mathcal{E}^3), \; c_1(\mathcal{E}) = 1 + \tfrac{\mathcal{E}}{2} + \tfrac{\mathcal{E}^2}{6} + \mathcal{O}(\mathcal{E}^3), \; c_2(\mathcal{E}) = \tfrac{1}{2} + \tfrac{\mathcal{E}}{6} + \tfrac{\mathcal{E}^2}{24} + \mathcal{O}(\mathcal{E}^3).$$

将这些近似代入 Equation (9.4.14)，并保留至 $\mathcal{E}^2$ 项（即保留至 h^2 阶，因为 $\mathcal{E} = \mathcal{O}(h)$ ），更新式恰好简化为普通的 Heun 形式 (Equation (9.4.13))。两种方法之间的剩余差异仅出现在大小为 $\mathcal{O}(\mathcal{E}^3) = \mathcal{O}(h^3)$ 的高阶项中。直观上，当步长 h 较小时，$\mathcal{E}$ 也较小，因此指数因子会简化为

$$e^{\mathcal{E}} \approx 1 + \mathcal{E}, \quad c_1(\mathcal{E}) \approx 1, \quad c_2(\mathcal{E}) \approx \tfrac{1}{2}.$$

因此，“线性握柄”指数-赫恩更新退化为普通的赫恩步骤。

Heun 更新与 DPM–Solver-2 在四个预测下的关联 我们强调，在 PF-ODE 的 $\boldsymbol{\epsilon}$-预测形式中（参见 Equation (9.4.5)），对数–SNR 时间 λ 中的动力学自然呈现出所需的半线性形式：

$$\frac{\mathrm{d}\hat{\mathbf{x}}_\lambda}{\mathrm{d}\lambda} = \underbrace{\frac{\alpha'_\lambda}{\alpha_\lambda}}_{=:L(\lambda)} \hat{\mathbf{x}}_\lambda + \underbrace{\big(-\sigma_\lambda \hat{\boldsymbol{\epsilon}}_{\phi^\times}(\hat{\mathbf{x}}_\lambda, \lambda)\big)}_{=:\mathbf{N}(\hat{\mathbf{x}}_\lambda, \lambda)}.$$

因此，对于 $\boldsymbol{\epsilon}$ -预测在 log–SNR 时间 λ 中，Equation (9.4.14) 中的指数-Heun 更新与 DPM-Solver-2 确切等价（使用中点参数 $\gamma = \frac{1}{2}$；参见 Algorithm 5）。

同样地，在对数–信噪比时间下的 $\mathbf{x}$ - 和 $\mathbf{s}$ -预测参数化中，它们的 PF-ODE 也具有相同的半线性结构。因此，DPM-Solver-2 在 $\boldsymbol{\epsilon}$ -、$\mathbf{x}$ - 或 $\mathbf{s}$ -预测下与 Equation (9.4.14) 中的指数–赫恩更新方法是 相同的。相反，$\mathbf{v}$ -预测形式自然地消除了线性项，因此其 PF-ODE 不需要指数积分器；在对数–信噪比时间下，普通的赫恩方法已能提供正确的二阶更新。

与 DDIM 中欧拉法与指数欧拉法的情况类似，我们由此得出以下结论：

Observation 9.4.1: Heun and DPM-Solver-2 Updates

Given the PF-ODEs in log-SNR time λ,

$$\begin{aligned} \mathbf{v}\text{-prediction:} &\quad \text{Heun} = \text{DPM-Solver-2}, \\ \boldsymbol{\epsilon}\text{-, } \mathbf{x}\text{-, or } \mathbf{s}\text{-prediction:} &\quad \text{exp-Heun} = \text{DPM-Solver-2} \neq \text{plain Heun}, \end{aligned}$$

where, in the $\boldsymbol{\epsilon}$-, $\mathbf{x}$-, or $\mathbf{s}$-prediction cases, the plain Heun step is not equivalent to DPM-Solver-2, since the linear term is only approximated instead of being integrated exactly.

9.5 DPM-Solver++

9.5.1 从 DPM-Solver 到 DPM-Solver++ 用于指导

高阶求解器可在无需引导的情况下实现更快的采样。然而，扩散模型因其可控且灵活的生成能力而备受青睐，这通常通过引导实现（详见 Chapter 8）。

DPM-Solver++ (**lu2022dpm2**) 指出先前高阶求解器的一个关键局限性：它们存在稳定性问题，在大引导尺度（更强的条件）下可能比 DDIM 更慢。作者将这种不稳定性归因于大引导尺度对输出及其导数的放大作用。由于高阶求解器依赖于高阶导数，因此对这一效应尤为敏感，导致效率和稳定性下降。

9.5.2 DPM-Solver++ 的方法论

为解决上述问题，DPM-Solver++ 提出：

1. 采用 $\mathbf{x}$ -预测参数化方法，而非 $\boldsymbol{\epsilon}$ -预测；

2. 应用阈值化方法（例如，动态阈值化 (**saharia2022photorealistic**)）将预测数据限制在训练数据的范围内（缓解大引导尺度下的训练-测试不匹配问题）。

我们详细说明第一点。回顾 Equation (6.3.1) 可知，数据与噪声的参数化是线性的：

$$\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t, t) = \frac{\mathbf{x}_t - \alpha_t \mathbf{x}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t, t)}{\sigma_t}.$$

利用此关系，DPM-Solver++ 将经验 PF-ODE 的精确解 $\widetilde{\boldsymbol{\Psi}}_{s\to t}(\mathbf{x}_s)$（最初以噪声参数化形式表示，见 Equation (9.4.4)）从任意 $\mathbf{x}_s$ 开始重写为：

$$\widetilde{\boldsymbol{\Psi}}_{s\to t}(\mathbf{x}_s) = \frac{\alpha_t}{\alpha_s}\mathbf{x}_s - \alpha_t \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda)\mathrm{d}\lambda,$$

进入数据参数化

$$\widetilde{\boldsymbol{\Psi}}_{s\to t}(\mathbf{x}_s) = \frac{\sigma_t}{\sigma_s}\mathbf{x}_s + \sigma_t \int_{\lambda_s}^{\lambda_t} e^{\lambda} \hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda)\mathrm{d}\lambda,$$

其中我们采用 Equation (9.4.3) 中的记号，并进一步记：

$$\hat{\mathbf{x}}_\lambda := \mathbf{x}_{t_\lambda(\lambda)},$$
$$\hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda) := \mathbf{x}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{t_\lambda(\lambda)}, t_\lambda(\lambda)).$$

基于 $\mathbf{x}$ -预测，DPM-Solver++ 提供了两种求解器变体：

- **高阶单步求解器**：在 Section 9.5.3 中提出。该方法类似于 DPM-Solver 中的方法，利用高阶泰勒展开来近似积分，但此处以 $\mathbf{x}$ -预测的形式进行表述。更新过程仅使用一个先前的点来估计下一步。
- **多步（两步）求解器**：在 Section 9.5.4 中引入。设计思想与 DEIS（同样是多步方法）类似；然而，DPM-Solver++ 特别地重用前两个点（而 DEIS 允许一般阶数）来估计下一步。每次更新仅需一次新的扩散模型评估。

9.5.3 DPM-Solver++ 单步法通过泰勒展开

遵循与 Section 9.4.3 相似的方法，DPM-Solver++ 在 $\mathbf{x}$ -参数化下推导出高阶求解器。对于 $n \geq 0$ ，将 $\hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}$ 关于 λ 的 n -阶总导数在 λ_{i-1} 处的取值记为

$$\hat{\mathbf{x}}^{(n)}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_{\lambda_{i-1}}, \lambda_{i-1}) := \frac{\mathrm{d}^n}{\mathrm{d}\lambda^n} \hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda)\Big|_{\lambda=\lambda_{i-1}}.$$

给定时间 t_{i-1} 处的先前估计 $\tilde{\mathbf{x}}_{t_{i-1}}$ ，使用在 $\lambda_{t_{i-1}}$ 处的 $(n-1)$ 阶泰勒展开来近似 $\hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda)$ 以得到 $\lambda \in [\lambda_{t_{i-1}}, \lambda_{t_i}]$ （其中 $s = t_{i-1}$ 和 $t = t_i$ ）的近似值，可得 $\widetilde{\boldsymbol{\Psi}}_{s\to t}(\mathbf{x}_s)$ 的如下近似：

$$\tilde{\mathbf{x}}_{t_i} = \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}} \tilde{\mathbf{x}}_{t_{i-1}} + \sigma_{t_i} \sum_{k=0}^{n-1} \underbrace{\hat{\mathbf{x}}^{(k)}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_{\lambda_{i-1}}, \lambda_{i-1})}_{\substack{\text{estimated via} \\ \text{finite difference}}} \underbrace{\int_{\lambda_{t_{i-1}}}^{\lambda_{t_i}} e^\lambda \frac{(\lambda - \lambda_{t_{i-1}})^k}{k!} \mathrm{d}\lambda}_{\text{analytically computable}} + \mathcal{O}(h_i^{n+1}).$$

其中 $h_i := \lambda_{t_i} - \lambda_{t_{i-1}} > 0$。与 Equation (9.4.9) 相同，该积分具有闭式解

$$\int_{\lambda_{t_{i-1}}}^{\lambda_{t_i}} e^\lambda \frac{(\lambda - \lambda_{t_{i-1}})^k}{k!} \mathrm{d}\lambda = e^{\lambda_{t_{i-1}}} h_i^{k+1} \varphi_{k+1}(h_i), \quad \varphi_m(h) := \frac{e^h - \sum_{j=0}^{m-1} \frac{h^j}{j!}}{h^m}.$$

这得到了 DPM-Solver++ 的单步更新（利用一个先前点来预测下一步）。当

$n=1$ 时，它退化为 DDIM 更新。当 $n=2$ 且 $\hat{\mathbf{x}}^{(1)}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_{\lambda_{i-1}}, \lambda_{i-1})$ 通过有限差分近似时，得到 *DPM-Solver++(2S)*，这是一种类似于 Algorithm 5 中的 DPM-Solver-2 的更新方式，但使用的是 $\mathbf{x}$ -预测。DPM-Solver++(2S) 的算法如 Algorithm 6 所示。

Algorithm 6 DPM-Solver++(2S): a midpoint special case.

Input: initial value $\mathbf{x}_T$, time steps $\{t_i\}_{i=0}^M$, data-prediction model $\hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}$

$\tilde{\mathbf{x}}_{t_0} \leftarrow \mathbf{x}_T$; $\lambda_{t_i} \leftarrow \log(\alpha_{t_i}/\sigma_{t_i})$ ▷ log-SNR at the grid

$\hat{\mathbf{x}}_0 \leftarrow \hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_0}, t_0)$ ▷ cache at start

for $i \leftarrow 1$ **to** M **do**

  $h_i \leftarrow \lambda_{t_i} - \lambda_{t_{i-1}}$; $s_i^{\text{mid}} \leftarrow t_\lambda\left(\frac{\lambda_{t_{i-1}}+\lambda_{t_i}}{2}\right)$

  $\mathbf{u}_i \leftarrow \frac{\sigma_{s_i^{\text{mid}}}}{\sigma_{t_{i-1}}}\tilde{\mathbf{x}}_{t_{i-1}} + \alpha_{s_i^{\text{mid}}}\left(1-e^{-h_i/2}\right)\hat{\mathbf{x}}_{i-1}$ ▷ forecast to midpoint

  $\mathbf{D}_i^{\text{mid}} \leftarrow \hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\mathbf{u}_i, s_i^{\text{mid}})$ ▷ one new model call at the midpoint

  $\tilde{\mathbf{x}}_{t_i} \leftarrow \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}}\tilde{\mathbf{x}}_{t_{i-1}} - \alpha_{t_i}\left(e^{-h_i}-1\right)\mathbf{D}_i^{\text{mid}}$

  $\hat{\mathbf{x}}_i \leftarrow \hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_i}, t_i)$ ▷ cache for next step

end for

return $\tilde{\mathbf{x}}_{t_M}$

9.5.4 DPM-Solver++ 多步法通过复用历史信息

高阶单步求解器（显式或隐式地）依赖于模型输出的高阶导数；在强 CFG 条件下，这些导数可能会被强烈放大，从而使更新过程不稳定。DPM-Solver++ 通过在对数信噪比时间 λ 中采用多步（Adams 型）策略缓解了这一问题：它沿着轨迹重用过去数据预测评估的短历史记录，通过有限差分近似所需的导数。这种重用每步仅需一次新的模型调用。与 DEIS 类似，我们将介绍分为：情况 1：无历史记录的热启动（第一步）；情况 2：带有两个历史锚点的后续步骤。

情形 I. DPM-Solver++ 采用一个历史锚点 ($i=1$)。 对于第一步 ($i=1$; 无历史记录)，使用一阶 DPM-风格更新（这与数据预测中的确定性 DDIM 步骤相匹配）。令 $h_1 = \lambda_1 - \lambda_0$ 。

$$\tilde{\mathbf{x}}_{t_1} = \frac{\sigma_{t_1}}{\sigma_{t_0}}\tilde{\mathbf{x}}_{t_0} + \sigma_{t_1}\, e^{\lambda_0}\,(e^{h_1}-1)\hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_0}, t_0)$$

情况 II. DPM-Solver++ 带有两个历史锚点 ($i \geq 2$)。 热启动后，两步多步更新重新使用时间 t_{i-2} 时的估计值与 $\tilde{\mathbf{x}}_{t_{i-2}}$ 时的估计值，以及时间 t_{i-1} 时的估计值与 $\tilde{\mathbf{x}}_{t_{i-1}}$ 时的估计值。在每个步骤 $i \geq 2$ 中，这些提供了两个最新的参考点，等价于 λ –时间：

$$(\lambda_{i-1},\, \hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1}}, t_{i-1})) \quad \text{and} \quad (\lambda_{i-2},\, \hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-2}}, t_{i-2})),$$

使用这些缓存的锚点（无需调用新模型即可形成更新）来计算更新 $\tilde{\mathbf{x}}_{t_i}$。获得 $\tilde{\mathbf{x}}_{t_i}$ 后，我们在 $(\tilde{\mathbf{x}}_{t_i}, t_i)$ 处对模型进行一次评估并缓存 $\hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_i}, t_i)$。此评估在步骤 i 执行，并用作后续步骤 $i{+}1$ 的锚点。即，我们旨在实现每步仅一次调用的更新，同时在大引导下保持稳定，通过离散化确切的 $\mathbf{x}$–预测形式来实现。

$$\widetilde{\boldsymbol{\Psi}}_{s\to t}(\mathbf{x}_s) = \frac{\sigma_t}{\sigma_s}\mathbf{x}_s + \sigma_t \int_{\lambda_s}^{\lambda_t} e^{\lambda} \hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda)\, \mathrm{d}\lambda.$$

在单步 $[\lambda_{i-1}, \lambda_i]$ 内，我们精确处理线性常微分方程部分，并通过将余项积分的被积函数近似为关于 λ 的线性函数（因为有两个锚点）来近似该积分。具体而言，我们近似

$$\lambda \mapsto \hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda)$$

在 $[\lambda_{i-1}, \lambda_i]$ 上由仿射模型

$$\hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_\lambda, \lambda) \approx \mathbf{L}(\lambda) := \mathbf{a}_0 + \mathbf{a}_1(\lambda - \lambda_{i-1}), \quad \lambda \in [\lambda_{i-1}, \lambda_i],$$

其中，$\lambda_i = \lambda_{t_i}$、$h_i = \lambda_i - \lambda_{i-1} > 0$ 以及系数 $\mathbf{a}_0$ 和 $\mathbf{a}_1$ 由通过最近两个锚点的直线唯一确定：

$$\mathbf{a}_0 = \hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1}}, t_{i-1}), \quad \mathbf{a}_1 = \frac{\hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1}}, t_{i-1}) - \hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-2}}, t_{i-2})}{h_{i-1}}.$$

将 $\mathbf{L}(\lambda)$ 代入积分后得到[6]

$$\begin{aligned}\sigma_{t_i}\int_{\lambda_{i-1}}^{\lambda_i} e^{\lambda}\hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_\lambda,\lambda)\,\mathrm{d}\lambda &\approx \sigma_{t_i}\int_{\lambda_{i-1}}^{\lambda_i} e^{\lambda}\mathbf{L}(\lambda)\,\mathrm{d}\lambda\\ &= \left(\sigma_{t_i}\int_{\lambda_{i-1}}^{\lambda_i} e^{\lambda}\,\mathrm{d}\lambda\right)\mathbf{a}_0 + \left(\sigma_{t_i}\int_{\lambda_{i-1}}^{\lambda_i} e^{\lambda}(\lambda-\lambda_{i-1})\,\mathrm{d}\lambda\right)\mathbf{a}_1\\ &= \left(\alpha_{t_i}(1-e^{-h_i})\right)\mathbf{a}_0 + \left(\alpha_{t_i}(h_i-1+e^{-h_i})\right)\mathbf{a}_1\\ &= \alpha_{t_i}(1-e^{-h_i})\Big(\mathbf{a}_0+\beta(h_i)\mathbf{a}_1\Big),\end{aligned}$$

其中 $\beta(h) := \frac{h-1+e^{-h}}{1-e^{-h}}$。到此为止，我们已经得到了 $\tilde{\mathbf{x}}_{t_i}$ 的一个有效估计值，如下所示：

$$\tilde{\mathbf{x}}_{t_i} = \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}}\tilde{\mathbf{x}}_{t_{i-1}} + \alpha_{t_i}\big(1-e^{-h_i}\big)\mathbf{D}_i, \quad \text{with } \mathbf{D}_i = \mathbf{a}_0 + \beta(h_i)\mathbf{a}_1.$$

在实际应用中，我们可以获得一个简化的更新规则，其局部截断误差与上述规则相同（前提是步长比有界）：

$$\tilde{\mathbf{x}}_{t_i} = \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}}\tilde{\mathbf{x}}_{t_{i-1}} + \alpha_{t_i}\big(1-e^{-h_i}\big)\mathbf{D}_i^{\mathrm{sim}}(\tilde{\mathbf{x}}_{t_{i-1}},\tilde{\mathbf{x}}_{t_{i-2}}).$$

此处，我们定义步长比 $r_i = h_i/h_{i-1}$ ，以及

$$\mathbf{D}_i^{\mathrm{sim}}(\tilde{\mathbf{x}}_{t_{i-1}},\tilde{\mathbf{x}}_{t_{i-2}}) := \Big(1+\tfrac{1}{2}r_i\Big)\hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1}},t_{i-1}) - \tfrac{1}{2}r_i\hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-2}},t_{i-2}).$$

在标准光滑性假设下，局部误差为 $\mathcal{O}(h_i^3)$。

为了说明原因，为简化符号表示，我们记作

$$\mathbf{a}_0 = \hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1}},t_{i-1}) =: \hat{\mathbf{x}}_{i-1}, \quad \mathbf{a}_1 = \frac{\hat{\mathbf{x}}_{i-1}-\hat{\mathbf{x}}_{i-2}}{h_{i-1}}.$$

[6]第二个恒等式可由直接的代数运算得出。所需的两个指数矩为

$$\int_{\lambda_{i-1}}^{\lambda_i} e^{\lambda}\,\mathrm{d}\lambda = e^{\lambda_{i-1}}(e^{h_i}-1), \quad \int_{\lambda_{i-1}}^{\lambda_i} e^{\lambda}(\lambda-\lambda_{i-1})\,\mathrm{d}\lambda = e^{\lambda_{i-1}}\left(h_i e^{h_i} - e^{h_i}+1\right).$$

乘以确切形式中的前因子 σ_{t_i} 并利用 $\alpha_t = \sigma_t e^{\lambda_t}$ （因此 $\sigma_{t_i}e^{\lambda_{i-1}} = \alpha_{t_i}e^{-h_i}$ ）可得方便的系数

$$\sigma_{t_i}\int_{\lambda_{i-1}}^{\lambda_i} e^{\lambda}\,\mathrm{d}\lambda = \alpha_{t_i}(1-e^{-h_i}), \quad \sigma_{t_i}\int_{\lambda_{i-1}}^{\lambda_i} e^{\lambda}(\lambda-\lambda_{i-1})\,\mathrm{d}\lambda = \alpha_{t_i}(h_i-1+e^{-h_i}).$$

然后

$$\begin{aligned}\mathbf{D}_i &:= \mathbf{a}_0 + \beta(h_i)\,\mathbf{a}_1 \\ &= \hat{\mathbf{x}}_{i-1} + \frac{\beta(h_i)}{h_{i-1}}\big(\hat{\mathbf{x}}_{i-1} - \hat{\mathbf{x}}_{i-2}\big) \\ &= \Big(1 + \tfrac{r_i}{2}\Big)\hat{\mathbf{x}}_{i-1} - \tfrac{r_i}{2}\hat{\mathbf{x}}_{i-2} + \Big(\tfrac{\beta(h_i)}{h_{i-1}} - \tfrac{r_i}{2}\Big)\big(\hat{\mathbf{x}}_{i-1} - \hat{\mathbf{x}}_{i-2}\big) \\ &= \Big[\Big(1 + \tfrac{1}{2}r_i\Big)\hat{\mathbf{x}}_{i-1} - \tfrac{1}{2}r_i\,\hat{\mathbf{x}}_{i-2}\Big] + \mathcal{O}(h_i^2) \\ &= \mathbf{D}_i^{\mathrm{sim}} + \mathcal{O}(h_i^2)\end{aligned}$$

此处，我们利用在小步长下，$\beta(h)$ 在 $h = 0$ 附近的泰勒展开式

$$\beta(h) = \frac{h}{2} + \mathcal{O}(h^2) \implies \frac{\beta(h_i)}{h_{i-1}} = \frac{h_i}{2h_{i-1}} + \mathcal{O}(h_i^2/h_{i-1}) = \frac{r_i}{2} + \mathcal{O}(h_i^2/h_{i-1}),$$

并且在某些光滑性假设下，$\hat{\mathbf{x}}_{i-1} - \hat{\mathbf{x}}_{i-2} = \mathcal{O}(h_{i-1})$。

Remark.

> 如果 log-SNR 步长是均匀的（每一步的大小 h 相同，因此 $h_i \equiv h$ 且 $r_i = h_i/h_{i-1} = 1$），那么双锚点混合
>
> $$\mathbf{D}_i^{\mathrm{sim}} = \Big(1 + \tfrac{1}{2}r_i\Big)\hat{\mathbf{x}}_{i-1} - \tfrac{1}{2}r_i\,\hat{\mathbf{x}}_{i-2}$$
>
> 就简化为经典常数
>
> $$\mathbf{D}_i^{\mathrm{sim}} = \Big(1 + \tfrac{1}{2}\cdot 1\Big)\hat{\mathbf{x}}_{i-1} - \tfrac{1}{2}\cdot 1\,\hat{\mathbf{x}}_{i-2} = \tfrac{3}{2}\,\hat{\mathbf{x}}_{i-1} - \tfrac{1}{2}\,\hat{\mathbf{x}}_{i-2}.$$
>
> 这些 $(\frac{3}{2}, -\frac{1}{2})$ 恰好是均匀步长下的 Adams-Bashforth 2 权重，即标准的两步线性多步系数。

Algorithm 7 DPM-Solver++(2M).

Input: initial value $\mathbf{x}_T$, time steps $\{t_i\}_{i=0}^{M}$, model $\hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}$

$\tilde{\mathbf{x}}_{t_0} \leftarrow \mathbf{x}_T$; $\quad \lambda_{t_i} \leftarrow \log(\alpha_{t_i}/\sigma_{t_i})$; $\quad h_i \leftarrow \lambda_{t_i} - \lambda_{t_{i-1}}$

$\hat{\mathbf{x}}_0 \leftarrow \hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_0}, t_0)$ ▷ cache at start

Case I. Warm start ($i = 1$) with one anchor (DDIM in x-pred.)

$\tilde{\mathbf{x}}_{t_1} \leftarrow \frac{\sigma_{t_1}}{\sigma_{t_0}} \tilde{\mathbf{x}}_{t_0} - \alpha_{t_1}\big(e^{-h_1} - 1\big)\, \hat{\mathbf{x}}_0$

$\hat{\mathbf{x}}_1 \leftarrow \hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_1}, t_1)$ ▷ One model call & cache

Case II. Using two history cached anchors (multistep)

for $i \leftarrow 2$ **to** M **do**

  $r_i \leftarrow h_i / h_{i-1}$ ▷ step ratio

  $\mathbf{D}_i^{\text{sim}} \leftarrow \Big(1 + \frac{1}{2} r_i\Big)\hat{\mathbf{x}}_{i-1} - \frac{1}{2} r_i\, \hat{\mathbf{x}}_{i-2}$

  $\tilde{\mathbf{x}}_{t_i} \leftarrow \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}} \tilde{\mathbf{x}}_{t_{i-1}} + \alpha_{t_i}\big(1 - e^{-h_i}\big)\, \mathbf{D}_i^{\text{sim}}$

  $\hat{\mathbf{x}}_i \leftarrow \hat{\mathbf{x}}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_i}, t_i)$ ▷ One model call & cache

end for

return $\tilde{\mathbf{x}}_{t_M}$

9.6 PF-ODE 求解器族及其数值类比

在本节中，我们首先将迄今为止介绍的 PF-ODE 求解器（DDIM、DEIS、DPM-Solver、DPM-Solver++）置于经典数值积分方法的背景下。然后，我们进一步深入分析两种具有代表性的高阶求解器——DEIS 和 DPM-Solver++，并比较它们各自的设计。

9.6.1 PF-ODE 求解器族及其经典对应方法

PF-ODE 采样器的多种族系可以通过经典数值分析的视角来理解。一旦线性漂移项通过积分因子处理，每个采样器自然地对应于一种成熟的时间推进方案：欧拉型方法、阿达姆斯-巴什福斯（AB）多步方案或龙格-库塔（RK）单步积分器。我们将在表 9.1 中总结这些对应关系。

表 9.1: PF-ODE 采样器及其数值分析类比。“exp.” 表示对线性项采用积分因子（半线性）处理（参见 Equation (9.1.6)）。AB = Adams-Bashforth，RK = Runge-Kutta。参见 Algorithm 5 了解 *DPM-Solver-2*。

PF-ODE Solver	Type	Classical Numerical Analogue
DDIM	single step	$\mathbf{v}$-prediction: plain Euler; $\boldsymbol{\epsilon}/\mathbf{x}/\mathbf{s}$-prediction: exp. Euler
DEIS	multistep	exp. AB (n^{th}-order)
DPM-Solver-n	single step	exp. RK (n^{th}-order) in log-SNR
DPM-Solver-2	single step	$\mathbf{v}$-prediction: plain Heun in log-SNR (2^{nd}-order); $\boldsymbol{\epsilon}/\mathbf{x}/\mathbf{s}$-prediction: exp. Heun in log-SNR (2^{nd}-order)
DPM-Solver++ 2S	single step	exp. RK (2^{nd}-order)
DPM-Solver++ 2M	multistep	exp. AB (2^{nd}-order)

我们突出展示了 Table 9.1 中两个具有代表性的例子：DDIM 和 DPM–Solver–2 的情况。在固定调度器 (α_t, σ_t) 的情况下，我们强调了 Sections 9.2.2, 9.3.3 and 9.4.4 的示例结果：无论我们使用对数 SNR 时间还是原始物理时间，

$$\mathbf{v}\text{-prediction:} \quad \text{DDIM} = \text{DPM-Solver-1} = \text{DEIS-1} = \text{Euler},$$

$$\boldsymbol{\epsilon}\text{-, } \mathbf{x}\text{-, or } \mathbf{s}\text{-prediction:} \quad \text{DDIM} = \text{DPM-Solver-1} = \text{DEIS-1} = \text{exp Euler}.$$

在 Section 9.4.5 中，我们通过考察四种参数化下 DPM-Solver-2 与经典 Heun 求

解器的关系，扩展了这一类比。

$$\mathbf{v}\text{-prediction:} \quad \text{DPM-Solver-2} = \text{Heun},$$

$$\boldsymbol{\epsilon}\text{-, } \mathbf{x}\text{-, or } \mathbf{s}\text{-prediction:} \quad \text{DPM-Solver-2} = \text{exp-Heun} \neq \text{plain Heun}.$$

DPM-Solver-n 与经典 RK 方法之间更一般的对应关系也可以用同样的方式来理解。

9.6.2 关于 DEIS 与 DPM-Solver++ 的讨论

Aspect	DEIS	DPM++
Core Viewpoint	Exponential–integrator: integrates the linear term exactly; approximates the nonlinear residual by a polynomial over past nodes.	Same integrator idea; formulated in log–SNR time λ with data prediction.
Step type	Multistep only	Single–step (2S) and Multistep (2M)
Polynomial Basis	Lagrange interpolation across past anchors (high–order multistep).	Backward divided differences (Newton/Adams–type) in λ–time for 2M; algebraically spans the same polynomial space as Lagrange, but not presented as a Lagrange fit.
Solvers Order	High–order multistep (general r).	Higher–order single–step methods exist (though 2S is the main focus), and a 2nd-order multistep (2M) scheme is provided; higher–order multistep variants are not covered.
History Use	Uses r+1 past evaluations to build a high–order update.	2S: one intermediate eval (single–step). 2M: reuses two anchors; after warm start, one model call per step.

DEIS 与 DPM-Solver++。 DEIS 和 DPM++ 都是指数积分采样器，它们对线性部分进行确切积分，并通过低阶多项式近似残差积分。在无条件生成中，两者均可在仅 10–20 次 ODE 步骤下实现高保真度。然而，在带有分类引导（CFG）的条件生成中，由于在大引导尺度下具有更好的稳定性，DPM++ 通常更优。我们将在下文总结 DEIS 与 DPM++ 的对比，并进一步讨论。

DEIS. 它是一种多步法，通过在拉格朗日基中对过去结点处的非线性项拟合多项式（通过锚点进行插值）得到。

DPM-Solver++ 。它在对数–信噪比时间中进行数据预测：其单步（2S）变体使用带有一次中间评估的泰勒/指数积分步骤，而其多步（2M）变体通过后向差分重用历史信息，生成相同的插值多项式，但以牛顿（有限差分）基表示。

换句话说，对于相同的插值点和函数值，拉格朗日形式与牛顿形式是同一多项式插值的两种不同坐标系：拉格朗日形式将其表示为函数值乘以基函数的和，而牛顿形式则以系数由差商（易于在多步方案中更新的有限差分比）给出的乘积展开形式表达。DPM++ 论文强调二阶（2S/2M）；原则上，可以使用高阶牛顿基构造更高阶的多步扩展。

9.7 (Optional) DPM-Solver-v3

DPM-Solver 和 DPM-Solver++ 均基于扩散模型的特定参数化形式（$\boldsymbol{\epsilon}$ -/$\mathbf{x}$ -预测）设计其求解器，这种做法缺乏选择参数化的合理依据，且可能并非最优选择。

在本节中，我们引入 DPM-Solver-v3 (**zheng2023dpm**)，它解决了这一问题，并在较少的步数或较大的引导尺度下提升了样本质量。DPM-Solver-v3 可以被视为整个 DPM-Solver 系列洞察的集大成者 (**lu2022dpm**; **lu2022dpm2**)。

DPM-Solver-v3 高级概述 我们继续关注 DPM-Solver (**lu2022dpm**) 的关键原则，即 SRN 中的时间重参数化，如 Equation (9.4.5) 所示：

$$\frac{\mathrm{d}\mathbf{x}_\lambda}{\mathrm{d}\lambda} = \frac{\alpha'_\lambda}{\alpha_\lambda}\mathbf{x}_\lambda - \sigma_\lambda \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda).$$

DPM-Solver-v3 (**zheng2023dpm**) 的核心思想是将三个额外的欠定/自由变量引入 Equation (9.4.5)，从而以新的模型参数化方式等价地重新表述原始的 ODE 解。随后提出一种高效搜索方法，用于在预训练模型上计算这些变量的最优集合，其目标是最小化离散化误差。

9.7.1 洞察 1：调整 Equation (9.4.5) 中的线性项

PF-ODE 是一个刚性常微分方程，在时间演化每个方向上具有显著不同的时间尺度，这使得使用较少的时间步长求解变得复杂。借鉴经典的刚性常微分方程理论 (**hochbruck2010exponential**)，**zheng2023dpm** 提出修改常微分方程的线性部分，以更好地处理这些刚性动力学。我们首先从经典的数值常微分方程理论出发，来说明这种方法的合理性。

动机：来自经典的刚性常微分方程求解器。 我们首先考虑 Equation (9.4.5) 的一种抽象形式：

$$\frac{\mathrm{d}\mathbf{x}_\lambda}{\mathrm{d}\lambda} = \mathbf{v}(\mathbf{x}_\lambda, \lambda),$$

其中 $\mathbf{v}(\mathbf{x}, \lambda)$ 表示向量场。

“Rosenbrock 型指数积分法”是一类为高效求解刚性常微分方程而开发的方法。这些方法的关键特征在于可以选择线性算子 $\mathbf{L}$ ，将其分解向量场为：

$$\mathbf{v}(\mathbf{x}, \lambda) = \mathbf{L}\mathbf{x} + \mathbf{N}(\mathbf{x}, \lambda),$$

其中 $\mathbf{N}(\mathbf{x}, \lambda)$ 表示非线性余项。由此可得以下更新公式，从 $\mathbf{x}_{\lambda_s}$ 出发，应用指数积分技术（如常）：

$$\widetilde{\mathbf{\Psi}}_{\lambda_s \to \lambda_t}(\mathbf{x}_{\lambda_s}) = e^{h\mathbf{L}}\mathbf{x}_{\lambda_s} + \int_{\lambda_s}^{\lambda_t} e^{(\lambda_t - \tau)\mathbf{L}}\mathbf{N}(\mathbf{x}_\tau, \tau)\,\mathrm{d}\tau, \quad \text{with } \Delta h := \lambda_t - \lambda_s.$$

我们观察到线性部分以指数形式存在。通常，通过利用预条件信息来更高效地处理刚性问题，选择 $\mathbf{L}$ 的目标是：(1) 确保方法的稳定性，(2) 提高数值解的收敛速率，以及 (3) 保证 $e^{\Delta\lambda\mathbf{L}}$ 在计算上保持高效。

将上述思想应用于 PF-ODE 中的 Equation (9.4.5)。 我们将引入的概念应用于 Equation (9.4.5)：

$$\frac{\mathrm{d}\mathbf{x}_\lambda}{\mathrm{d}\lambda} = \underbrace{\frac{\alpha'_\lambda}{\alpha_\lambda}\mathbf{x}_\lambda - \sigma_\lambda \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda)}_{\mathbf{v}(\mathbf{x}_\lambda, \lambda)},$$

我们将其重写为：

$$\frac{\mathrm{d}\mathbf{x}_\lambda}{\mathrm{d}\lambda} = \underbrace{\left(\frac{\alpha'_\lambda}{\alpha_\lambda} - \boldsymbol{\ell}_\lambda\right)\mathbf{x}_\lambda}_{\text{linear part}} - \underbrace{\left(\sigma_\lambda \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) - \boldsymbol{\ell}_\lambda \mathbf{x}_\lambda\right)}_{\text{nonlinear part}}. \tag{9.7.1}$$

此处，$\boldsymbol{\ell}_\lambda$ 是一个仅依赖于 λ 的 D 维自由/不确定变量。为便于记号，我们将线性和非线性部分分别记为

$$\begin{aligned} \mathbf{L}(\lambda) &:= \frac{\alpha'_\lambda}{\alpha_\lambda} - \boldsymbol{\ell}_\lambda, \\ \mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) &:= \sigma_\lambda \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) - \boldsymbol{\ell}_\lambda \mathbf{x}_\lambda. \end{aligned} \tag{9.7.2}$$

zheng2023dpm 建议通过求解以下简单的最小二乘问题来选择 $\boldsymbol{\ell}_\lambda$：

$$\boldsymbol{\ell}^*_\lambda = \arg\min_{\boldsymbol{\ell}_\lambda} \mathbb{E}_{\mathbf{x}_\lambda \sim p_\lambda^{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda)} \|\nabla_{\mathbf{x}} \mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda)\|_F^2, \tag{9.7.3}$$

其中 $\|\cdot\|_F$ 表示 Frobenius 范数，$p_\lambda^{\boldsymbol{\phi}^\times}$ 表示在 Equation (9.7.4) 中沿 ODE 轨迹的样本的边缘分布，在 λ 处。

我们注意到 $\boldsymbol{\ell}_\lambda = \boldsymbol{\ell}^*_\lambda$ 可以解析求解。此选择利用了预训练模型的预处理信息，从概念上使 $\mathbf{N}_{\boldsymbol{\phi}^\times}$ 对 $\mathbf{x}$ 中的误差不那么敏感（因为 $\mathbf{N}_{\boldsymbol{\phi}^\times}$ 的利普希茨性，近似于 $\mathbf{x}$ -梯度，被降低），并消除了 $\mathbf{N}_{\boldsymbol{\phi}^\times}$ 的“线性”特性。

9.7.2 洞察 2：在模型参数化中引入自由变量以进一步减小离散化误差

PF-ODE 表现出一种半线性结构(参见 Equation (9.7.1) 和 Equation (9.7.2))。为了便于记号清晰，我们考虑以下（抽象）形式的经验 PF-ODE：

$$\frac{\mathrm{d}\mathbf{x}_\lambda}{\mathrm{d}\lambda} = \mathbf{L}(\lambda)\mathbf{x}_\lambda + \mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda). \tag{9.7.4}$$

动机：理解离散化误差及其最小化策略。 和往常一样，这个经验微分方程在区间 $[\lambda_s, \lambda_t]$ 内的精确解可以使用参数变易公式表示：

$$\widetilde{\boldsymbol{\Psi}}_{\lambda_s\to\lambda_t}(\mathbf{x}_{\lambda_s}) = \mathcal{E}(\lambda_s \to \lambda_t)\mathbf{x}_{\lambda_s} + \int_{\lambda_s}^{\lambda_t} \mathcal{E}(\lambda \to \lambda_s)\mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda)\,\mathrm{d}\lambda, \tag{9.7.5}$$

其中 $\mathcal{E}(s \to t) := e^{-\int_s^t \mathbf{L}(u)\,\mathrm{d}u}$。通过使用估计

$$\mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s}, \lambda_s) \approx \mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) \quad \text{for } \lambda \in [\lambda_s, \lambda_t],$$

我们可以得到一个近似解，如下所示：

$$\tilde{\mathbf{x}}_{\lambda_t} = \mathcal{E}(\lambda_s \to \lambda_t)\mathbf{x}_{\lambda_s} + \int_{\lambda_s}^{\lambda_t} \mathcal{E}(\lambda \to \lambda_s)\mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s}, \lambda_s)\,\mathrm{d}\lambda. \tag{9.7.6}$$

将 Equation (9.7.5) 与 Equation (9.7.6) 相减，并将 $\mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s}, \lambda_s)$ 展开为泰勒级数如下：

$$\mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s}, \lambda_s) = \mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) + (\lambda_s - \lambda)\mathbf{N}^{(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) + \mathcal{O}((\lambda_s - \lambda)^2),$$

我们可以量化一阶离散化误差：

$$\tilde{\mathbf{x}}_{\lambda_t} - \widetilde{\boldsymbol{\Psi}}_{\lambda_s\to\lambda_t}(\mathbf{x}_{\lambda_s}) = \int_{\lambda_s}^{\lambda_t} \mathcal{E}(\lambda \to \lambda_s)(\lambda_s - \lambda)\mathbf{N}^{(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda)\,\mathrm{d}\lambda + \mathcal{O}(h^3),$$

其中 $h := \lambda_t - \lambda_s$。这一观察表明，离散化误差依赖于 $\mathbf{N}^{(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda)$。

为了减小这种误差，**zheng2023dpm** 提出将 Equation (9.7.5) 重写为一种使用新参数化 $\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda)$ 的等价形式，使得误差项保持类似的结构：

$$\tilde{\mathbf{x}}_{\lambda_t} - \widetilde{\boldsymbol{\Psi}}_{\lambda_s\to\lambda_t}(\mathbf{x}_{\lambda_s}) = \int_{\lambda_s}^{\lambda_t} \mathcal{E}(\lambda \to \lambda_s)(\lambda_s - \lambda)\mathbf{N}^{\text{new},(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda)\,\mathrm{d}\lambda + \mathcal{O}(h^3).$$

此外，λ-导数的 $\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda)$ 满足：

$$\mathbf{N}^{\text{new},(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) \propto \mathbf{N}^{(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) - (\mathbf{a}_\lambda \mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) + \mathbf{b}_\lambda)\,. \tag{9.7.7}$$

此处，$\mathbf{a}_\lambda$ 和 $\mathbf{b}_\lambda$ 是自由/未定变量。

因此，目标是确定使离散化误差最小的最优 $\mathbf{a}_\lambda$ 和 $\mathbf{b}_\lambda$ 值，从而减小 $\mathbf{N}^{\text{new},(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda)$ 。这可以通过求解以下最小二乘最优化问题来实现：

$$(\mathbf{a}^*_\lambda, \mathbf{b}^*_\lambda) = \arg\min_{\mathbf{a}_\lambda, \mathbf{b}_\lambda} \mathbb{E}_{\mathbf{x}_\lambda \sim p^{\boldsymbol{\phi}^\times}_\lambda(\mathbf{x}_\lambda)} \left[\| \mathbf{N}^{(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) - (\mathbf{a}_\lambda \mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) + \mathbf{b}_\lambda) \|_2^2 \right]. \tag{9.7.8}$$

值得注意的是，Equation (9.7.8) 可以根据预训练的扩散模型得到解析解，该解可以预先计算。

实现最小化离散化误差的策略。 我们首先考虑 $\mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda)$ 的一个线性变换版本，定义为：

$$\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) := e^{-\int_{\lambda_s}^{\lambda} \mathbf{a}_u \, \mathrm{d}u} \mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) - \int_{\lambda_s}^{\lambda} e^{-\int_{\lambda_s}^{r} \mathbf{a}_u \, \mathrm{d}u} \mathbf{b}_r \, \mathrm{d}r. \tag{9.7.9}$$

然后我们可以很容易地计算其由下式给出的 λ -导数：

$$\mathbf{N}^{\text{new},(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) = e^{-\int_{\lambda_s}^{\lambda} \mathbf{a}_u \, \mathrm{d}u} \Big[\mathbf{N}^{(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) - \big(\mathbf{a}_\lambda \mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) + \mathbf{b}_\lambda\big) \Big], \tag{9.7.10}$$

这正是 Equation (9.7.7) 所示的形式。

利用这一点，$\mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda)$ 可以重写为：

$$\begin{aligned}
\mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) &= e^{\int_{\lambda_s}^{\lambda} \mathbf{a}_u \, \mathrm{d}u} \Big[\underbrace{e^{-\int_{\lambda_s}^{\lambda} \mathbf{a}_u \, \mathrm{d}u} \mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) - \int_{\lambda_s}^{\lambda} e^{-\int_{\lambda_s}^{r} \mathbf{a}_u \, \mathrm{d}u} \mathbf{b}_r \, \mathrm{d}r}_{\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda)} \\
&\qquad + \int_{\lambda_s}^{\lambda} e^{-\int_{\lambda_s}^{r} \mathbf{a}_u \, \mathrm{d}u} \mathbf{b}_r \, \mathrm{d}r \Big] \\
&= e^{\int_{\lambda_s}^{\lambda} \mathbf{a}_u \, \mathrm{d}u} \Big[\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) + \int_{\lambda_s}^{\lambda} e^{-\int_{\lambda_s}^{r} \mathbf{a}_u \, \mathrm{d}u} \mathbf{b}_r \, \mathrm{d}r \Big]
\end{aligned}$$

通过这种重新表述，我们可以将 Equation (9.7.5) 和 Equation (9.7.6) 重写

为：

$$\widetilde{\mathbf{\Psi}}_{\lambda_s\to\lambda_t}(\mathbf{x}_{\lambda_s}) = \mathcal{E}(\lambda_s\to\lambda_t)\mathbf{x}_{\lambda_s} + \int_{\lambda_s}^{\lambda_t}\mathcal{E}(\lambda\to\lambda_s)e^{\int_{\lambda_s}^{\lambda}\mathbf{a}_u\,\mathrm{d}u} \left[\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda,\lambda) + \int_{\lambda_s}^{\lambda} e^{-\int_{\lambda_s}^{r}\mathbf{a}_u\,\mathrm{d}u}\mathbf{b}_r\,\mathrm{d}r\right]\mathrm{d}\lambda \tag{9.7.11}$$

$$\tilde{\mathbf{x}}_{\lambda_t} = \mathcal{E}(\lambda_s\to\lambda_t)\mathbf{x}_{\lambda_s} + \int_{\lambda_s}^{\lambda_t}\mathcal{E}(\lambda\to\lambda_s)e^{\int_{\lambda_s}^{\lambda}\mathbf{a}_u\,\mathrm{d}u} \left[\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s) + \int_{\lambda_s}^{\lambda} e^{-\int_{\lambda_s}^{r}\mathbf{a}_u\,\mathrm{d}u}\mathbf{b}_r\,\mathrm{d}r\right]\mathrm{d}\lambda \tag{9.7.12}$$

将这两个方程相减并应用泰勒展开：

$$\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s) = \mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda,\lambda) + (\lambda_s-\lambda)\mathbf{N}^{\text{new},(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda,\lambda) + \mathcal{O}\big((\lambda_s-\lambda)^2\big),$$

我们到达了：

$$\begin{aligned}
&\tilde{\mathbf{x}}_{\lambda_t} - \widetilde{\mathbf{\Psi}}_{\lambda_s\to\lambda_t}(\mathbf{x}_{\lambda_s}) \\
&= \int_{\lambda_s}^{\lambda_t}\mathcal{E}(\lambda\to\lambda_s)(\lambda_s-\lambda)e^{\int_{\lambda_s}^{\lambda}\mathbf{a}_u\,\mathrm{d}u}\mathbf{N}^{\text{new},(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda,\lambda)\,\mathrm{d}\lambda + \mathcal{O}(h^3) \\
&= \int_{\lambda_s}^{\lambda_t}\mathcal{E}(\lambda\to\lambda_s)(\lambda_s-\lambda)\Big[\mathbf{N}^{(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda,\lambda) - \big(\mathbf{a}_\lambda\mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda,\lambda) + \mathbf{b}_\lambda\big)\Big]\,\mathrm{d}\lambda + \mathcal{O}(h^3)
\end{aligned}$$

此处，最后一个等式源于 Equation (9.7.10)，这是我们的设计核心，它消去了因子 $e^{\int_{\lambda_s}^{\lambda}\mathbf{a}_u\,\mathrm{d}u}$。

因此，通过求解 Equation (9.7.8)，我们可以确定最优系数 $(\mathbf{a}^*_\lambda, \mathbf{b}^*_\lambda)$，有效最小化离散化误差。

9.7.3 结合两种见解。

我们现在对迄今为止的讨论进行总结。

理论中的步骤。 对于任意 λ，我们首先通过求解 Equation (9.7.3) 中的最小二乘问题，解析地计算 ℓ^*_λ：

$$\ell^*_\lambda = \arg\min_{\ell_\lambda}\mathbb{E}_{\mathbf{x}_\lambda\sim p^{\boldsymbol{\phi}^\times}_\lambda(\mathbf{x}_\lambda)}\|\nabla_{\mathbf{x}}\mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda,\lambda)\|^2_F,$$

其中 $\mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda)$ 在 Equation (9.7.2) 中定义。接下来，我们通过在 Equation (9.7.8) 中求解最小二乘问题来解析计算 $(\mathbf{a}_\lambda^*, \mathbf{b}_\lambda^*)$，此时 $\boldsymbol{\ell}_\lambda = \boldsymbol{\ell}_\lambda^*$ 固定：

$$(\mathbf{a}_\lambda^*, \mathbf{b}_\lambda^*) = \arg\min_{\mathbf{a}_\lambda, \mathbf{b}_\lambda} \mathbb{E}_{\mathbf{x}_\lambda \sim p_\lambda^{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda)} \left[\| \mathbf{N}_{\boldsymbol{\phi}^\times}^{(1)}(\mathbf{x}_\lambda, \lambda) - (\mathbf{a}_\lambda \mathbf{N}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda, \lambda) + \mathbf{b}_\lambda) \|_2^2 \right].$$

因此，如 Equation (9.7.12)(with $\boldsymbol{\ell}_\lambda$, $\mathbf{a}_\lambda$, and $\mathbf{b}_\lambda$ replaced by $\boldsymbol{\ell}_\lambda^*$, $\mathbf{a}_\lambda^*$, and $\mathbf{b}_\lambda^*$ in Equation (9.7.9)) 所定义，所得的 $\tilde{\mathbf{x}}_{\lambda_t}$ 作为 $\widetilde{\boldsymbol{\Psi}}_{\lambda_s \to \lambda_t}(\mathbf{x}_{\lambda_s})$ 的期望估计值。

实施注意事项。 尽管 $\boldsymbol{\ell}_\lambda^*$ 、$\mathbf{a}_\lambda^*$ 和 $\mathbf{b}_\lambda^*$ 具有涉及预训练扩散模型 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}$ 的雅克比-向量积的解析解（详见 **zheng2023dpm** 的附录 C.1.1），但它们的计算需要对 $p_\lambda^{\boldsymbol{\phi}^\times}$ 取期望。

在实际应用中，这些量通过蒙特卡罗（MCMC）方法进行估计。具体而言，通过应用一种交替求解器（例如，200 步的 DPM-Solver++ (**lu2022dpm2**)）对 Equation (9.4.5) 进行处理，生成一批数据点 $\mathbf{x}_\lambda \sim p_\lambda^{\boldsymbol{\phi}^\times}$（大约 1K-4K 个样本），随后解析计算与 $\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}$ 相关的各项。重要的是，这些统计量均可预先计算，从而确保在应用 DPM-Solver-v3 时，可避免与这些计算相关的计算开销。

9.7.4 高阶 DPM-Solver-v3

预先计算的统计量 $\boldsymbol{\ell}_\lambda^*$ 、$\mathbf{a}_\lambda^*$ 和 $\mathbf{b}_\lambda^*$ ，由分析一阶离散化误差得到，也可用于构建高阶求解器（参见 Section 9.7.5 以获取进一步解释）。

为了得到对 Equation (9.7.11) 的 $(n+1)$ 阶近似，我们利用对 $\mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new}}(\mathbf{x}_\lambda, \lambda)$ 关于 λ 在 λ_s 处的 n 阶泰勒展开，忽略 $\mathcal{O}((\lambda_s - \lambda)^{(n+1)})$ 阶的项。这使得我们能够近似 $\mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new}}(\mathbf{x}_\lambda, \lambda)$ 对于 $\lambda \in [\lambda_s, \lambda_t]$：

$$\mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new}}(\mathbf{x}_\lambda, \lambda) \approx \mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new}}(\mathbf{x}_{\lambda_s}, \lambda_s) + \sum_{k=1}^{n} \frac{(\lambda - \lambda_s)^k}{k!} \mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new},(k)}(\mathbf{x}_{\lambda_s}, \lambda_s),$$

其中，该近似导致了对 Equation (9.7.11) 的估计解。

$$\begin{aligned}\tilde{\mathbf{x}}_{\lambda_t} &= \mathcal{E}(\lambda_s \to \lambda_t)\mathbf{x}_{\lambda_s} + \int_{\lambda_s}^{\lambda_t} \underbrace{\mathcal{E}(\lambda \to \lambda_s) e^{\int_{\lambda_s}^{\lambda} \mathbf{a}_u^* du}}_{=:\mathbf{E}(\lambda_s \to \lambda)} \\ &\quad \cdot \left[\sum_{k=0}^{n} \frac{(\lambda - \lambda_s)^k}{k!} \mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new},(k)}(\mathbf{x}_{\lambda_s}, \lambda_s) + \underbrace{\int_{\lambda_s}^{\lambda} e^{-\int_{\lambda_s}^{r} \mathbf{a}_u^* du} \mathbf{b}_r^* dr}_{=:\mathbf{B}(\lambda_s \to \lambda)} \right] \mathrm{d}\lambda. \\ &= \mathcal{E}(\lambda_s \to \lambda_t)\mathbf{x}_{\lambda_s} + \int_{\lambda_s}^{\lambda_t} \mathbf{E}(\lambda_s \to \lambda)\mathbf{B}(\lambda_s \to \lambda)\, \mathrm{d}\lambda \\ &\quad + \sum_{k=0}^{n} \mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new},(k)}(\mathbf{x}_{\lambda_s}, \lambda_s) \int_{\lambda_s}^{\lambda_t} \mathbf{E}(\lambda_s \to \lambda) \frac{(\lambda - \lambda_s)^k}{k!}\, \mathrm{d}\lambda.\end{aligned} \tag{9.7.13}$$

在类似于 Section 9.4.3 中高阶 DPM 推导的方式下，对于 $(n+1)$ 阶近似，我们利用前 $n+1$ 步的 $\mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new},(k)}(\mathbf{x}_\lambda, \lambda)$ 有限差分 $\lambda_{i_n}, \ldots, \lambda_{i_1}, \lambda_{i_0} := \lambda_s$ 来估计每个 $\mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new},(k)}(\mathbf{x}_{\lambda_s}, \lambda_s)$。该方法旨在匹配泰勒展开中的系数。

下面，我们通过一个例子来演示该方法：$n = 2$ 。

Example: Estimating Higher-Order Derivatives ($n = 2$ Case).

When $n = 2$, we aim to estimate the derivatives $\mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new},(k)}(\mathbf{x}_\lambda, \lambda)$ with $k = 1, 2$ with previous 3 timesteps $\lambda_{i_2}, \lambda_{i_1}, \lambda_s$.

Linear System for Approximated Derivatives. Let $\delta_k = \lambda_{i_k} - \lambda_s$ $(k = 1, 2)$. We expand $\mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new}}(\mathbf{x}_\lambda, \lambda)$ around λ_s using a Taylor series. Evaluating at $\lambda = \lambda_{i_1}$ and $\lambda = \lambda_{i_2}$, and rearranging to isolate the derivative terms, we obtain:

$$\begin{aligned}&\mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new}}(\mathbf{x}_{\lambda_{i_1}}, \lambda_{i_1}) - \mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new}}(\mathbf{x}_{\lambda_s}, \lambda_s) \\ &\qquad \approx \delta_1 \mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new},(1)}(\mathbf{x}_{\lambda_s}, \lambda_s) + \frac{\delta_1^2}{2!} \mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new},(2)}(\mathbf{x}_{\lambda_s}, \lambda_s), \\ &\mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new}}(\mathbf{x}_{\lambda_{i_2}}, \lambda_{i_2}) - \mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new}}(\mathbf{x}_{\lambda_s}, \lambda_s) \\ &\qquad \approx \delta_2 \mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new},(1)}(\mathbf{x}_{\lambda_s}, \lambda_s) + \frac{\delta_2^2}{2!} \mathbf{N}_{\boldsymbol{\phi}^\times}^{\text{new},(2)}(\mathbf{x}_{\lambda_s}, \lambda_s).\end{aligned}$$

Here, higher-order terms $\mathcal{O}(\delta_1^3)$ and $\mathcal{O}(\delta_2^3)$ are neglected, respectively. This

forms the linear system:

$$\begin{bmatrix} \delta_1 & \delta_1^2 \\ \delta_2 & \delta_2^2 \end{bmatrix} \begin{bmatrix} \frac{\widetilde{\mathbf{N}}^{\text{new},(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)}{1!} \\ \frac{\widetilde{\mathbf{N}}^{\text{new},(2)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)}{2!} \end{bmatrix} = \begin{bmatrix} \mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_{i_1}},\lambda_{i_1}) - \mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s) \\ \mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_{i_2}},\lambda_{i_2}) - \mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s) \end{bmatrix}$$

with the approximated derivatives $\widetilde{\mathbf{N}}^{\text{new},(k)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)$ to be solved; hence,

$$\widetilde{\mathbf{N}}^{\text{new},(k)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s) \approx \mathbf{N}^{\text{new},(k)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s).$$

Solving for the Approximated Derivatives $\widetilde{\mathbf{N}}^{\text{new},(k)}_{\boldsymbol{\phi}^\times}$. Let:

$$\mathbf{R}_2 = \begin{bmatrix} \delta_1 & \delta_1^2 \\ \delta_2 & \delta_2^2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda},\lambda_{i_2}) - \mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda},\lambda_s) \\ \mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda},\lambda_{i_1}) - \mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda},\lambda_s) \end{bmatrix}.$$

The approximated derivatives are:

$$\begin{bmatrix} \frac{\widetilde{\mathbf{N}}^{\text{new},(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)}{1!} \\ \frac{\widetilde{\mathbf{N}}^{\text{new},(2)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)}{2!} \end{bmatrix} = \mathbf{R}_2^{-1}\mathbf{b},$$

which can be computed explicitly.

在示例的精神基础上，我们可以很容易地将其推广到一般 n 情况下 $k \leq n$ 的 k 阶导数。

$$\underbrace{\begin{bmatrix} \delta_1 & \delta_1^2 & \cdots & \delta_1^n \\ \delta_2 & \delta_2^2 & \cdots & \delta_2^n \\ \vdots & \vdots & \ddots & \vdots \\ \delta_n & \delta_n^2 & \cdots & \delta_n^n \end{bmatrix}}_{\mathbf{R}_n} \begin{bmatrix} \frac{\widetilde{\mathbf{N}}^{\text{new},(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)}{1!} \\ \frac{\widetilde{\mathbf{N}}^{\text{new},(2)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)}{2!} \\ \vdots \\ \frac{\widetilde{\mathbf{N}}^{\text{new},(n)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)}{n!} \end{bmatrix} = \begin{bmatrix} \mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_{i_1}},\lambda_{i_1}) - \mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s) \\ \mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_{i_2}},\lambda_{i_2}) - \mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s) \\ \vdots \\ \mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_{i_n}},\lambda_{i_n}) - \mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s) \end{bmatrix}. \tag{9.7.14}$$

通过求逆范德蒙矩阵 $\mathbf{R}_n$ ，我们可以解析地求解近似导数 $\widetilde{\mathbf{N}}^{\text{new},(k)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)$ for $k \leq n$ 。因此，通过将 $\mathbf{N}^{\text{new},(k)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)$ 代入 Equation (9.7.13) 中的 $\widetilde{\mathbf{N}}^{\text{new},(k)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)$ ，我们得到一个近似解，我们仍将其记为 $\tilde{\mathbf{x}}_{\lambda_t}$ ：

$$\tilde{\mathbf{x}}_{\lambda_t} = \mathcal{E}(\lambda_s \to \lambda_t)\mathbf{x}_{\lambda_s} + \int_{\lambda_s}^{\lambda_t} \mathbf{E}(\lambda_s \to \lambda)\mathbf{B}(\lambda_s \to \lambda)\,\mathrm{d}\lambda + \sum_{k=0}^{n} \widetilde{\mathbf{N}}_{\boldsymbol{\phi}^\times}^{\mathrm{new},(k)}(\mathbf{x}_{\lambda_s}, \lambda_s) \int_{\lambda_s}^{\lambda_t} \mathbf{E}(\lambda_s \to \lambda)\frac{(\lambda - \lambda_s)^k}{k!}\,\mathrm{d}\lambda. \tag{9.7.15}$$

9.7.5 DPM-Solver-v3 的更多解读

减小一阶离散化误差有助于高阶求解器。 $\mathbf{a}_\lambda^*$ 和 $\mathbf{b}_\lambda^*$ 是通过减小一阶离散化误差得到的；然而，理论上它们也可以用于控制高阶求解器中的误差。该结果总结如下命题。

Proposition 9.7.1: 降低一阶离散化误差有助于高阶求解器

从相同初始条件 $\mathbf{x}_{\lambda_s}$ 出发，令近似解 $\tilde{\mathbf{x}}_{\lambda_t}$ 按 Equation (9.7.15) 定义，确切解 $\widetilde{\boldsymbol{\Psi}}_{\lambda_s \to \lambda_t}(\mathbf{x}_{\lambda_s})$ 按 Equation (9.7.11) 定义。则离散化误差由下式给出：

$$\begin{aligned}\tilde{\mathbf{x}}_{\lambda_t} - \widetilde{\boldsymbol{\Psi}}_{\lambda_s \to \lambda_t}(\mathbf{x}_{\lambda_s}) = &\int_{\lambda_s}^{\lambda_t} \Big(\int_{\lambda_s}^{\lambda} \mathbf{N}_{\boldsymbol{\phi}^\times}^{\mathrm{new},(1)}(\mathbf{x}_u, u)\,\mathrm{d}u \Big) \mathbf{E}(\lambda_s \to \lambda)\,\mathrm{d}\lambda \\ &+ \sum_{k=1}^{n} \left(\sum_{j=1}^{n} (\mathbf{R}_n^{-1})_{kj} \int_{\lambda_s}^{\lambda_{i_j}} \mathbf{N}_{\boldsymbol{\phi}^\times}^{\mathrm{new},(1)}(\mathbf{x}_\lambda, \lambda)\,\mathrm{d}\lambda \right) \\ &\cdot \int_{\lambda_s}^{\lambda_t} \mathbf{E}(\lambda_s \to \lambda)\frac{(\lambda - \lambda_s)^k}{k!}\,\mathrm{d}\lambda.\end{aligned}$$

Proof for Proposition.

$\widetilde{\boldsymbol{\Psi}}_{\lambda_s \to \lambda_t}(\mathbf{x}_{\lambda_s})$ 可改写为如下表达式：

$$\widetilde{\boldsymbol{\Psi}}_{\lambda_s \to \lambda_t}(\mathbf{x}_{\lambda_s}) = \mathcal{E}(\lambda_s \to \lambda_t)\mathbf{x}_{\lambda_s} + \int_{\lambda_s}^{\lambda_t} \mathbf{E}(\lambda_s \to \lambda)\mathbf{B}(\lambda_s \to \lambda)\,\mathrm{d}\lambda + \int_{\lambda_s}^{\lambda_t} \mathbf{N}_{\boldsymbol{\phi}^\times}^{\mathrm{new}}(\mathbf{x}_\lambda, \lambda)\mathbf{E}(\lambda_s \to \lambda)\,\mathrm{d}\lambda.$$

将 Equation (9.7.15) 减去 $\widetilde{\boldsymbol{\Psi}}_{\lambda_s\to\lambda_t}(\mathbf{x}_{\lambda_s})$：

$$\begin{aligned}\tilde{\mathbf{x}}_{\lambda_t}-\widetilde{\boldsymbol{\Psi}}_{\lambda_s\to\lambda_t}(\mathbf{x}_{\lambda_s})=&\int_{\lambda_s}^{\lambda_t}\Big(\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda,\lambda)-\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)\Big)\mathbf{E}(\lambda_s\to\lambda)\,\mathrm{d}\lambda\\&+\sum_{k=1}^{n}\widetilde{\mathbf{N}}^{\text{new},(k)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)\int_{\lambda_s}^{\lambda_t}\mathbf{E}(\lambda_s\to\lambda)\frac{(\lambda-\lambda_s)^k}{k!}\,\mathrm{d}\lambda.\end{aligned}$$

通过求逆矩阵 $\mathbf{R}_n$ 中的 Equation (9.7.14)，对于任意 $k=1,\dots,n$ 的解由下式给出：

$$\widetilde{\mathbf{N}}^{\text{new},(k)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)=\sum_{j=1}^{n}(\mathbf{R}_n^{-1})_{kj}\Big(\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_{i_j}},\lambda_{i_j})-\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)\Big).$$

应用微积分基本定理可得结果：

$$\begin{aligned}\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda,\lambda)-\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)&=\int_{\lambda_s}^{\lambda}\mathbf{N}^{\text{new},(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_u,u)\,\mathrm{d}u\\\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_{i_j}},\lambda_{i_j})-\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_{\lambda_s},\lambda_s)&=\int_{\lambda_s}^{\lambda_{i_j}}\mathbf{N}^{\text{new},(1)}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda,\lambda)\,\mathrm{d}\lambda.\end{aligned}$$

由上述命题和 Equation (9.7.10) 可知，在足够光滑的假设下，控制 $\left\|\mathbf{N}^{\text{new},(1)}_{\boldsymbol{\phi}^\times}\right\|_2$ 可以减少 $\left\|\tilde{\mathbf{x}}_{\lambda_t}-\widetilde{\boldsymbol{\Psi}}_{\lambda_s\to\lambda_t}(\mathbf{x}_{\lambda_s})\right\|_2$。

广义参数化的表达能力 $\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}$。利用 Equation (9.7.2) 和 Equation (9.7.9)，我们可以将 $\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}$ 重写为以下形式：

$$\begin{aligned}\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda,\lambda)=\sigma_\lambda e^{-\int_{\lambda_s}^{\lambda}\mathbf{a}_u\,\mathrm{d}u}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda,\lambda)-\ell_\lambda e^{-\int_{\lambda_s}^{\lambda}\mathbf{a}_u\,\mathrm{d}u}\mathbf{x}_\lambda\\-\int_{\lambda_s}^{\lambda}e^{-\int_{\lambda_s}^{r}\mathbf{a}_u\,\mathrm{d}u}\mathbf{b}_r\,\mathrm{d}r,\end{aligned}\tag{9.7.16}$$

其概念上具有以下形式：

$$\mathbf{T}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda,\lambda):=\boldsymbol{\alpha}(\lambda)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda,\lambda)+\boldsymbol{\beta}(\lambda)\mathbf{x}_\lambda+\boldsymbol{\gamma}(\lambda).\tag{9.7.17}$$

实际上，对于固定的 λ，$\boldsymbol{\Psi}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda,\lambda)$ 可以通过依赖于 λ_s 的线性变换用 $\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\lambda,\lambda)$ 表示（详见 (**zheng2023dpm**) 附录 I.1）。

9.7.6 DPM-Solver-v3 与其他方法的连接

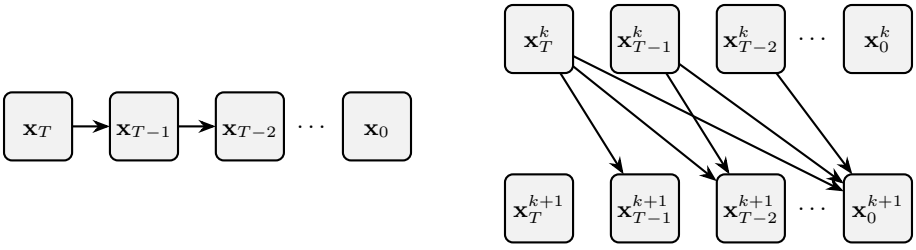
DPM-Solver-v3 的 $\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}$ 是一种通用参数化。 通过将 DPM-Solver-v3 与之前的 ODE 公式及其对应的 $\boldsymbol{\epsilon}$ -/$\mathbf{x}$ -预测进行比较，我们可以很容易地识别出，它们是我们方法的特例，只需将 $\boldsymbol{\ell}_\lambda$ 、$\mathbf{a}_\lambda$ 和 $\mathbf{b}_\lambda$ 设为特定值即可：

- $\boldsymbol{\epsilon}$ -预测: $(\boldsymbol{\ell}_\lambda, \mathbf{a}_\lambda, \mathbf{b}_\lambda) = (\mathbf{0}_D, -\mathbf{1}_D, \mathbf{0}_D)$
- $\mathbf{x}$ -预测: $(\boldsymbol{\ell}_\lambda, \mathbf{a}_\lambda, \mathbf{b}_\lambda) = (\mathbf{1}_D, \mathbf{0}_D, \mathbf{0}_D)$

一阶离散化作为改进的 DDIM。 由于 $\mathbf{N}^{\text{new}}_{\boldsymbol{\phi}^\times}$ 既不表示噪声也不表示数据参数化，而是旨在最小化一阶离散化误差的改进参数化方法，因此 Equation (9.7.12) 中的一阶 DPM-Solver-v3 更新与 Equation (9.2.2) 中的 DDIM 更新有所不同：

$$\mathbf{x}_{t_{i-1}\to t_i} = \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}}\tilde{\mathbf{x}}_{t_{i-1}} - \alpha_{t_i}\left(\frac{\sigma_{t_{i-1}}}{\alpha_{t_{i-1}}} - \frac{\sigma_{t_i}}{\alpha_{t_i}}\right)\boldsymbol{\epsilon}_{\boldsymbol{\phi}^\times}(\tilde{\mathbf{x}}_{t_{i-1}}, t_{i-1}).$$

9.8 (Optional) ParaDiGMs



(a) Sequential sampling by time-stepping estimation in generation process.

(b) Picard iterations with skip dependencies.

图 9.4: 两个计算图的比较。左：传统的时间步进常微分方程求解，其中解在时间上依次传播。右：皮卡迭代，通过利用前一次迭代的结果同时更新所有时间结点，实现并行计算，从而避免了时间步进的严格顺序性。

9.8.1 从时间推进到并行时间求解器

在前面的部分中，我们重点讨论了时间步进方法，该方法通过从先验时间 T 逐步演化到任意时间 $t \in [0, T]$ 来估计轨迹。

$$\mathbf{v}_{\boldsymbol{\phi}^\times}(\mathbf{x}, t) := \mathbf{f}(\mathbf{x}, t) - \frac{1}{2} g^2(t) \mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}, t)$$

表示来自预训练扩散模型的经验 PF–ODE 漂移。从 T 到任意中间时间 t 的确切演化为：

$$\widetilde{\boldsymbol{\Psi}}_{T \to t}(\mathbf{x}(T)) = \mathbf{x}(T) + \int_T^t \mathbf{v}_{\boldsymbol{\phi}^\times}(\mathbf{x}(\tau), \tau)\, \mathrm{d}\tau, \quad \mathbf{x}(T) \sim p_{\text{prior}}. \tag{9.8.1}$$

时间推进格式通过基于先前时间步的离散更新来近似该积分。

在本节中，我们转向时间并行方法，以 *ParaDiGMS* 为例，该方法基于经典的 Picard 迭代，实现时间维度上的并行积分。ParaDiGMS 的核心思想是用计算资源换取更快的仿真速度。

9.8.2 ParaDiGMS 方法论

从轨迹到皮卡德迭代作为不动点更新。 Equation (9.8.1) 中的积分表达式可以被理解为一个将整个轨迹 $\{\mathbf{y}(\tau)\}_{\tau \in [0,T]}$ 映射到新轨迹的映射。形式上，对于任

意候选轨迹 $\{\mathbf{y}(\tau)\}_{\tau\in[0,T]}$，我们通过 $\mathcal{L}$ 定义算子 $\mathcal{L}$。

$$(\mathcal{L}[\mathbf{y}(\cdot)])(t) = \mathbf{y}(T) + \int_T^t \mathbf{v}_{\boldsymbol{\phi}^\times}\big(\mathbf{y}(\tau),\tau\big)\,\mathrm{d}\tau, \quad t\in[0,T].$$

也就是说，$\mathcal{L}$ 将终点 $\mathbf{y}(T)$ 沿路径向后积分预定的速度场 $\mathbf{v}_{\boldsymbol{\phi}^\times}$，从而将其回溯至过去。

一个真实的解轨迹 $\mathbf{x}^*(\cdot)$ 是指在该映射下保持不变的轨迹。换句话说，$\mathbf{x}^*(\cdot)$ 是 $\mathcal{L}$ 的一个 *不动点*：

$$\mathbf{x}^*(t) = \mathcal{L}[\mathbf{x}^*(\cdot)](t) \quad\Longleftrightarrow\quad \mathbf{x}^*(t) = \mathbf{x}^*(T) + \int_T^t \mathbf{v}_{\boldsymbol{\phi}^\times}\big(\mathbf{x}^*(\tau),\tau\big)\,\mathrm{d}\tau.$$

这种重新表述将问题从逐步求解常微分方程转变为寻找与算子 $\mathcal{L}$ 一致的轨迹。

在上述算子视角的基础上，一旦我们得到了从轨迹到轨迹的映射 $\mathcal{L}$，寻找其不动点的一种自然方法是通过连续置换（Picard 迭代）：反复应用 $\mathcal{L}$，并在计算积分时使用前一次迭代得到的轨迹。更精确地说，从任意初始路径 $\mathbf{x}^{(0)}(\cdot)$（实际中，常取一个常数路径 $\mathbf{x}^{(0)}(t)\equiv\mathbf{x}^{(0)}(T)$，其中 $\mathbf{x}^{(0)}(T)\sim p_{\text{prior}}$ 为固定值）出发，更新公式为

$$\begin{aligned}\mathbf{x}^{(k+1)}(t) :=& \mathcal{L}^{(k)}[\mathbf{x}^{(0)}(\cdot)](t)\\ =& \mathbf{x}^{(k)}(T) + \int_T^t \mathbf{v}_{\boldsymbol{\phi}^\times}\big(\mathbf{x}^{(k)}(\tau),\tau\big)\,\mathrm{d}\tau, \quad k=0,1,2,\ldots\end{aligned} \tag{9.8.2}$$

该公式保持了正确的时间 T 锚定：迭代始终从先验抽取的状态 $\mathbf{x}^{(k)}(T)$ 开始，然后随着时间从 T 递减到 t 而累积漂移。

在 T 到 0 网格上的离散佩卡德。 要将 Equation (9.8.2) 转化为一个实用的算法，我们在 $[0,T]$ 上放置一个均匀递减的网格，通过选择步数 N，设定 $\Delta t := T/M$，并定义

$$t_j := T - j\Delta t, \quad j=0,1,\ldots,M,$$

所以 $t_0 = T$ 和 $t_M = 0$。将采样的迭代记为

$$\mathbf{x}_j^{(k)} := \mathbf{x}^{(k)}(t_j).$$

由于网格在时间上反向运行，从 T 到 t_j 的积分具有负方向。通过在划分 $\{[t_{i+1}, t_i]\}_{i=0}^{j-1}$ 上使用左端点进行近似得到

$$\int_T^{t_j} \mathbf{v}_{\phi^\times}(\mathbf{x}^{(k)}(\tau), \tau)\, \mathrm{d}\tau \approx -\Delta t \sum_{i=0}^{j-1} \mathbf{v}_{\phi^\times}\left(\mathbf{x}_{t_i}^{(k)}, t_i\right),$$

由于每个小积分在 $[t_{i+1}, t_i]$ 处的值等于 $-\int_{t_i}^{t_{i+1}} \cdot\, \mathrm{d}\tau$。将此近似代入 Equation (9.8.2) 得到离散 Picard 更新

$$\mathbf{x}_j^{(k+1)} = \mathbf{x}_0^{(k)} - \Delta t \underbrace{\sum_{i=0}^{j-1} \mathbf{v}_{\phi^\times}\left(\mathbf{x}_i^{(k)}, t_i\right)}_{\substack{\text{cumulative sum} \\ \text{of drifts}}}, \quad j = 1, \ldots, M. \tag{9.8.3}$$

该方案简单且适合并行处理：每个漂移评估 $\mathbf{v}_{\phi^\times}\left(\mathbf{x}_i^{(k)}, t_i\right)$ 仅依赖于同一结点 t_i 上的前一个迭代值，因此网格上所有 $i = 0, \ldots, j-1$ 的评估可以独立计算。随后通过累加求和恢复积分，该操作可串行执行，也可通过并行的 *前缀和（扫描/滑动窗口）* 完成。

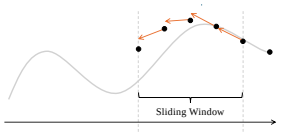


图 9.5: 在大小为 $p = 4$ 的批量窗口上并行计算 $\mathbf{x}_{\ell:\ell+p}^{(k)}$ 的漂移

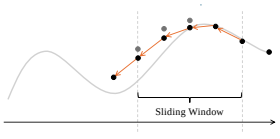


图 9.6: 使用窗口内点的累积漂移更新 $\mathbf{x}_{\ell:\ell+p}^{(k+1)}$ 的值

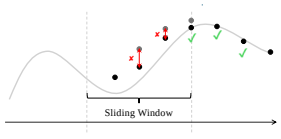


图 9.7: 根据误差 $\|\mathbf{x}_i^{(k+1)} - \mathbf{x}_i^{(k)}\|^2$ 确定窗口应前移的距离。

滑动窗口与并行评估。 离散 Picard 更新 Equation (9.8.3) 将每个 $\mathbf{x}_j^{(k+1)}$ 表示为左锚定值 $\mathbf{x}_0^{(k)}$ 减去漂移的累积和。为了限制内存使用并利用并行硬件，将相同的思想局部应用于短的滑动索引块会更加方便。

固定窗口长度 p 和左端索引 ℓ；此时窗口覆盖 $j = \ell, \ldots, \ell + p$，其中 $t_\ell > t_{\ell+1} > \cdots > t_{\ell+p}$。在迭代 k 期间：

步骤 1. 在窗口上并行评估漂移。 使用前一次迭代结果，以并行方式计算，

$$\mathbf{v}_{\phi^\times}\left(\mathbf{x}_{\ell+i}^{(k)}, t_{\ell+i}\right), \quad i = 0, 1, \ldots, p-1.$$

这些是 p 局部增量，用于从左边缘 t_ℓ 推进穿过窗口。

步骤 2. 左端锚定的累积更新。 通过在 $j=\ell$ 处锚定并在子区间上累积漂移来形成加窗更新：

$$\mathbf{x}_{\ell+j+1}^{(k+1)} = \mathbf{x}_{\ell}^{(k)} - \Delta t \sum_{i=0}^{j} \mathbf{v}_{\phi^{\times}}\big(\mathbf{x}_{\ell+i}^{(k)}, t_{\ell+i}\big), \quad j=0,1,\dots,p-1. \tag{9.8.4}$$

这正是 Equation (9.8.3) 在窗口上的限制，负号反映了时间方向的减小。内部求和是在窗口化漂移上的前缀和（扫描），因此所有部分和都可以在并行硬件上高效生成。

步骤 3. 进度控制与窗口推进。 在形成当前窗口的左锚定累积更新后（步骤 2），我们现在决定窗口滑动的距离。我们通过逐点 Picard 变化来衡量局部收敛性

$$\texttt{error}_j := \big\|\mathbf{x}_{\ell+j}^{(k+1)} - \mathbf{x}_{\ell+j}^{(k)}\big\|^2, \quad j=1,\dots,p-1,$$

并与预设的容差 $\texttt{tol}_{\ell+j}$ 进行比较。也就是说，$\texttt{error}_j$ 表示在上一次 Picard 迭代中，结点 $\ell+j$ 处的迭代值变化了多少。如果该数值较小，表明两个连续近似值在该结点处具有局部一致性，从而说明在该结点处不动点迭代已实现局部收敛。如果数值较大，则说明该结点尚未稳定，需要更多的 Picard 平滑处理。

步幅被选为窗口中第一个不满足此测试的索引（如果都没有失败，则为整个窗口长度 p）：

$$\texttt{stride} := \min\Big(\{\, j \ge 1 :\ \texttt{error}_j > \texttt{tol}_{\ell+j} \,\} \ \cup\ \{p\}\Big).$$

然后我们通过设置 $\ell \leftarrow \ell + \texttt{stride}$ 来滑动窗口。换句话说：我们接受从左边缘开始直到（但不包括）第一个未收敛的结点的所有结点；如果所有结点均已收敛，我们则接受整个窗口。然后我们根据这些被接受的结点数量 $\ell \leftarrow \ell + \texttt{stride}$ 滑动窗口，并继续此过程。这最多推进一个窗口长度 p，绝不会跳过任何未满足容差的结点。如果滑动将超出网格末端 M，我们将窗口截断为 $p \leftarrow \min\{p,\, M-\ell\}$ 并继续。

当窗口向前移动时，会暴露新的时间结点，这些结点尚无数值。为在这些结点上启动皮卡德迭代，我们只需从窗口的左边界复制数值，并将其用作初始猜测。这种“常数外推法”成本低且稳定，后续更新将对其进行修正。如需更高精度，也可用基于过去点的线性或多项式外推法来替代。

至此，ParaDiGMS 的过程已全部完成。我们将该算法总结在 Algorithm 8 中。

Algorithm 8 ParaDiGMS with Sliding Windows

Input: Drift $\mathbf{v}_{\boldsymbol{\phi}^\times}(\mathbf{x}, t)$; $\{t_j\}_{j=0}^{M}$; window length p; $\{\texttt{tol}_j\}_{j=1}^{M}$
Output: Approximate trajectory $\{\mathbf{x}_j^{(k)}\}_{j=0}^{M}$ with $\mathbf{x}_M^{(k)}$ at $t = 0$

$k \leftarrow 0$, $\ell \leftarrow 0$
Sample $\mathbf{x}_0^{(0)} \sim p_{\text{prior}}$; set $\mathbf{x}_j^{(0)} \leftarrow \mathbf{x}_0^{(0)}$ for $j = 1, \ldots, \min(p, M)$
▷ constant extrapolation
while $\ell < M$ **do**
  $J \leftarrow \min(p, M - \ell)$ ▷ current window length
  Step 1: Parallel
  For $i = 0, \ldots, J-1$: $g_i \leftarrow \mathbf{v}_{\boldsymbol{\phi}^\times}\left(\mathbf{x}_{\ell+i}^{(k)}, t_{\ell+i}\right)$
  ▷ drifts from previous iterate (Picard freezing)
  Compute prefix sums $S_j \leftarrow \sum_{i=0}^{j} g_i$ for $j = 0, \ldots, J-1$
  ▷ scan over windowed drifts
  Step 2: Cumulative Updates
  For $j = 0, \ldots, J-1$: $\mathbf{x}_{\ell+j+1}^{(k+1)} \leftarrow \mathbf{x}_{\ell}^{(k)} - \Delta t\, S_j$
  ▷ left-anchored update; cf. Equation (9.8.4)
  Step 3: Progress Control and Window Advance
  For $j = 1, \ldots, J-1$: $\texttt{error}_j \leftarrow \left\|\mathbf{x}_{\ell+j}^{(k+1)} - \mathbf{x}_{\ell+j}^{(k)}\right\|^2$
  ▷ pointwise Picard change
  $\texttt{stride} \leftarrow \min\Big(\{\, j \in \{1, \ldots, J-1\} : \texttt{error}_j > \texttt{tol}_{\ell+j} \,\} \cup \{J\}\Big)$
  Initialize New Nodes
  For $r = 1, \ldots, \texttt{stride}$: $\mathbf{x}_{\ell+J+r}^{(k+1)} \leftarrow \mathbf{x}_{\ell+J}^{(k+1)}$
  ▷ constant extrapolation into newly exposed indices
  $\ell \leftarrow \ell + \texttt{stride}$; $k \leftarrow k + 1$
end while
return $\{\mathbf{x}_j^{(k)}\}_{j=0}^{M}$

9.8.3 与时间步进求解器的关系

滑动窗口大小的选择。为了将滑动窗口方案置于上下文中，首先注意最小窗口大小时发生的情况。当 $p = 1$ 时，窗口仅包含一个步骤，因此 Equation (9.8.4) 简化为 PF–ODE 的一阶时间推进更新。如果采用相同的微分方程表达方式（例如数据预测或噪声预测），并选择与 DDIM 相同的离散时间步长调度，则该方法退化为 DDIM。

增大 p 会扩大并行度（每个窗口推进的结点数更多），而不会改变总的步数 N。因此，样本质量仍然由基础离散化（网格/参数化选择以及每步公式）和每个窗口上的 Picard 收敛性决定，我们通过局部容差来监控这一收敛性。

与高阶求解器（如 DPM）的兼容性。 滑动窗口 Picard 结构控制的是增量的计算方式（并行计算并通过扫描累加），而不是定义这些增量的局部公式。因此，可以将左端点规则替换为任意一致的高阶求积方法，而无需改变并行布局。例如，Equation (9.8.4) 的梯形变体如下

$$\mathbf{x}_{\ell+j+1}^{(k+1)} = \mathbf{x}_{\ell}^{(k)} - \Delta t \Big[\tfrac{1}{2}\mathbf{v}_{\boldsymbol{\phi}\times}\big(\mathbf{x}_{\ell}^{(k)}, t_{\ell}\big) + \sum_{i=1}^{j-1} \mathbf{v}_{\boldsymbol{\phi}\times}\big(\mathbf{x}_{\ell+i}^{(k)}, t_{\ell+i}\big) + \tfrac{1}{2}\mathbf{v}_{\boldsymbol{\phi}\times}\big(\mathbf{x}_{\ell+j}^{(k)}, t_{\ell+j}\big) \Big],$$

其中所有漂移仍来自前一次 Picard 迭代，因此每个结点的计算保持独立，内部求和仍为前缀和。

同样地，DPM 求解器族（例如，基于对数信噪比时间的 DPM–Solver++ 2M）所使用的多步或指数积分更新可以通过将每个窗口增量替换为先前模型评估的相应高阶线性组合（$\mathbf{x}$-或 $\boldsymbol{\epsilon}$-预测，带有预计算系数）来插入。扫描过程则像之前一样精确地累积这些加权增量。简而言之：并行方案与求解器（离散化）的选择无关，用于近似积分。准确率来自基础求解器；窗口前缀和仅使其更快。

9.9 闭幕词

本章探讨了扩散模型面临的最显著实际限制之一：其缓慢的迭代采样过程。我们研究了一类无需训练的强大解决方案，通过利用微分方程的丰富数值方法领域来加速生成。核心策略是更高效地求解定义从噪声到数据的确定性生成轨迹的 PF-ODE：

1. 我们从基础的 DDIM 开始，它可被理解为一种一阶指数欧拉方法。
2. 随后，我们转向了像 DEIS 这样的高阶多步方法，这些方法通过利用过去评估的历史信息来提高准确率。
3. 最后，我们研究了高效性能的 DPM-Solver 系列，该系列通过引入关键的 log-SNR 时间重参数化，实现了卓越的性能。

通过这些复杂的求解器，高质量生成所需的函数评估次数（NFEs）已从数百或数千次大幅减少至仅 10 到 20 次，使扩散模型变得更具实用性。

然而，这些无需训练的方法本质上仍然是迭代的。它们逐步逼近一条连续路径。这引发了一个自然且雄心勃勃的问题：*我们能否仅通过一个或极少数几个离散步骤实现高质量生成？*

本专著的最后部分 D 将通过基于训练的加速来探讨这一问题。我们将研究两种主要策略：

1. 首先，在 Chapter 10 中，我们将考察基于蒸馏的方法，其中快速的 *学生* 生成器在远少于步骤的情况下被训练以复现一个缓慢的预训练 *教师* 扩散模型的输出。
2. 接着，在 Chapter 11 中，我们将进一步探讨从零开始学习快速、少步生成器的方法，例如一致性训练（Consistency Training），该方法定义了一种独立的训练原则，无需依赖任何预训练模型。

从改进求解器转向学习解映射本身，代表了高效生成式建模的前沿，旨在结合扩散模型的质量与单步生成器的速度。

Part D

面向快速扩散生成模型的学习

10

基于蒸馏的快速采样方法

本章介绍了基于训练的方法，这些方法通过教会新的生成器仅用一步或几步即可生成样本，从而加速扩散模型的采样过程。其核心思想称为蒸馏，即让快速的学生模型从缓慢的预训练扩散模型（教师）采样器中学习。尽管教师可能需要数百步，但学生模型仅用几步即可达到相当的质量。[1]. 与基于求解器的加速方法不同，后者通过改进数值积分方案来提升性能，蒸馏则直接训练生成器以采取高效的捷径。我们重点介绍两种主要范式：*分布层面蒸馏*，该方法跳过完整轨迹的模拟，而是将学生模型的输出分布与教师模型对齐；以及 *流映射层面蒸馏*，该方法训练学生模型以更快、更紧凑的方式复现教师模型的采样路径。

[1] 此处的蒸馏指的是减少采样步骤的数量，而非缩小模型大小。

10.1 序言

扩散模型的一个核心瓶颈是其采样速度较慢。

如通过 Tweedie 公式 (Section 6.3.1) 所示，扩散模型可以被解释为一个 "$\mathbf{x}$-预测" 模型，$\mathbf{x}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t, t)$，其训练目标是从噪声输入 $\mathbf{x}_t$ 中恢复期望的干净数据，噪声水平为 t:

$$\mathbf{x}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t, t) \approx \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t],$$

其中期望是关于 $p(\mathbf{x}_0|\mathbf{x}_t)$ 取的，代表与 $\mathbf{x}_t$ 对应的所有合理干净数据。一个自然的想法是使用 $\mathbf{x}_{\boldsymbol{\phi}^\times}(\mathbf{x}_t, t)$ 进行单步生成。然而，由于该降噪器对许多可能的结果进行了平均，预测结果变得过于平滑，仅使用少量降噪步骤进行生成会导致模糊且质量较低的样本。

另一方面，如 Section 4.2.2所述，扩散采样通过一系列迭代步骤沿着常微分方程（ODE）或随机微分方程（SDE）轨迹进行。这能够生成高保真度的样本，但所需的大量步骤使得该过程本质上较慢。减少 NFE（即采样步骤数与模型调用次数的乘积）虽然能加快生成速度，但不可避免地会降低保真度。每个求解器步骤引入的积分误差阶数为 $\mathcal{O}(h^n)$，其中 n 为求解器阶数，$h = \max_i |t_i - t_{i-1}|$ 为步长。步骤越少意味着更大的时间增量 h，从而导致累积采样误差增加，使轨迹准确性下降。这在扩散采样中形成了质量与效率之间的根本权衡。

为克服这一瓶颈，一个主要的研究方向是*蒸馏*，该方法假设可以访问一个训练良好的扩散模型（即*教师*），并训练一个生成器（即*学生*）通过单次前向传播或少数几步计算来复现其行为。这将教师的多次采样步骤压缩为一个快速过程，在保持高样本保真度的同时，有效绕过了缓慢的迭代求解器。

下面我们介绍蒸馏的两个视角：*分布层级蒸馏*和 *流图层级蒸馏*[2].

10.1.1 分布级蒸馏

基于分布的蒸馏的目标是训练一个单步生成器 $\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})$，该生成器将噪声 $\mathbf{z} \sim p_{\text{prior}}$ 映射到样本 $\hat{\mathbf{x}} = \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})$，从而诱导出一个近似于目标数据分布 $p_{\text{data}}(\mathbf{x})$ 的分布 $p_{\boldsymbol{\theta}}(\hat{\mathbf{x}})$。这通常通过最小化统计散度来实现

$$\min_{\boldsymbol{\theta}} \mathcal{D}(p_{\boldsymbol{\theta}}(\hat{\mathbf{x}}), p_{\text{data}}(\hat{\mathbf{x}})),$$

[2]从时间顺序上看，以 *知识蒸馏* (KD) (**luhman2021knowledge**) 和 *渐进式蒸馏* (PD) (**ho2020denoising**) 为代表的流图级蒸馏于 2021 年提出，早于 2023 年左右出现的分布级蒸馏方法家族。然而，为了更流畅地阐述并为下一章做好衔接，我们首先介绍分布级蒸馏。

其中 $\mathcal{D}$ 表示合适的散度度量，例如 KL。

在实际应用中，基于分布的方法将生成器的分布与预训练扩散模型产生的经验分布 $p_{\boldsymbol{\phi}^\times}(\mathbf{x})$ 进行对齐：

$$\min_{\boldsymbol{\theta}} \mathcal{D}(p_{\boldsymbol{\theta}}(\hat{\mathbf{x}}), p_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}))\,,$$

其中，$p_{\boldsymbol{\phi}^\times}$ 作为 p_{data} 的代理。这些方法并非显式地评估该散度，而是近似其梯度，该梯度可直接从预训练的教师模型中计算得到。这使得学生模型能够在无需完整散度评估的情况下，将其分布与教师模型对齐。

该公式通过分布对齐，将扩散模型的多步生成过程提炼为单步模型。我们在 Section 10.2 中详细阐述了这一方法。

10.1.2 流图层级蒸馏

我们考虑 PF-ODE，其可表示为任意预测模型（见 Equation (6.3.1)）：

$$\frac{\mathrm{d}\mathbf{x}(\tau)}{\mathrm{d}\tau} = f(\tau)\mathbf{x}(\tau) - \tfrac{1}{2}g^2(\tau)\nabla_{\mathbf{x}} \log p_\tau(\mathbf{x}(\tau)) =: \mathbf{v}^*(\mathbf{x}(\tau), \tau). \tag{10.1.1}$$

其解映射从时间 s 的 $\mathbf{x}_s$ 出发，反向演化至时间 $t \le s$，记为 $\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s)$；即，

$$\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s) \coloneqq \mathbf{x}_s + \int_s^t \mathbf{v}^*(\mathbf{x}(\tau), \tau)\,\mathrm{d}\tau, \tag{10.1.2}$$

其中，积分求解了 PF-ODE。直观上，$\boldsymbol{\Psi}_{s\to t}$ 将时间 s 时的噪声 $\mathbf{x}_s$ 传输至时间 t 时的较低噪声状态（最终到达时间 $t = 0$ 时的数据）。

从扩散模型中进行采样对应于计算 $\boldsymbol{\Psi}_{T\to 0}(\mathbf{x}_T)$ 对 $\mathbf{x}_T \sim p_{\text{prior}}$。通常，该积分通过利用速度场 $\mathbf{v}$ 的迭代数值求解器来近似（见 Chapter 9），但需要大量步骤（例如，即使在 DPM-Solver 中也至少需要 10 步），使得采样速度慢于经典的单步生成模型（如 GAN）。这引发了一个自然的问题：

Question 10.1.1

我们能否直接学习解映射 $\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s)$？

特别是，学习一个满足 $\boldsymbol{\Psi}_{T\to 0}(\mathbf{x}_T)$ 且 $\mathbf{x}_T \sim p_{\text{prior}}$ 的映射，能够实现单步生成。

轨迹蒸馏 轨迹蒸馏旨在训练一个神经生成器，使其在实例层面近似解映射。由于 PF-ODE 积分通常无法得到闭式解，因此在训练过程中必须进行数值近似。

为形式化表述，我们引入通用求解器记号

$$\texttt{Solver}_{s\to t}(\mathbf{x}_s;\,\boldsymbol{\phi}^{\times}) \quad \text{or simply} \quad \texttt{Solver}_{s\to t}(\mathbf{x}_s), \tag{10.1.3}$$

表示从 s 到 t 对经验 PF-ODE 进行数值积分，起始于 $\mathbf{x}_s$ ，使用教师参数 $\boldsymbol{\phi}^{\times}$（在上下文明确时可省略）。

早期方法：直接知识蒸馏。 为了实现少步甚至单步生成，一种直接的方法是训练一个生成器 $\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_T, T, 0)$ 来模仿数值求解器在整个轨迹上计算的输出：

$$\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_T, T, 0) \approx \texttt{Solver}_{T\to 0}(\mathbf{x}_T), \qquad \mathbf{x}_T \sim p_{\text{prior}}.$$

这一思想构成了最早的轨迹蒸馏方法之一——知识蒸馏 (**luhman2021knowledge**)，该方法使用回归损失

$$\mathcal{L}_{\text{KD}}(\boldsymbol{\theta}) := \mathbb{E}_{\mathbf{x}_T\sim p_{\text{prior}}} \left\|\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_T, T, 0) - \texttt{Solver}_{T\to 0}(\mathbf{x}_T)\right\|_2^2.$$

尽管该方法能从预训练教师模型中获得直接监督，但无法利用原始训练数据中的强监督信息。此外，如果在训练环中调用微分方程求解，计算成本较高，因为每次参数更新都需要求解微分方程以形成目标。最后，由于生成器仅学习从 T 到 0 的全局映射，可能在从中间状态引导生成过程时失去可控性。因此，Chapter 8 中引入的大多数可控生成技术无法直接应用。

渐进蒸馏前言。 渐进蒸馏(PD) (**salimans2021progressive**) 使用来自 Teacher 片段的 *局部*监督，训练一个时间条件的 Student 。设 $t_0 = T > t_1 > \cdots > t_N = 0$ 为固定的时序网格。Teacher 为 $k = 0, \ldots, N-1$ 提供时间步进映射 $\texttt{Teacher}_{t_k\to t_{k+1}}$ 。

而不是仅监督单跳 $T \to 0$ ，PD 训练 Student 两步跳跃映射以匹配两个连续的 Teacher 步：

$$\texttt{Student}_{t_k\to t_{k+2}} \;\approx\; \texttt{Teacher}_{t_{k+1}\to t_{k+2}} \circ \texttt{Teacher}_{t_k\to t_{k+1}},$$

对于 $k = 0, 2, 4, \ldots$。匹配是通过简单的回归损失（例如，均方误差）来完成的。

在局部配对片段上训练后，Student 再也不遵循原始网格的每个时间间隔。

而是每隔一个时间点前进一次。

$$t_0 \to t_2 \to t_4 \to \cdots \to t_N,$$

这意味着每个 `Student` 步长实际上覆盖了两个连续的 `Teacher` 步长。因此，`Student` 仅使用 $N/2$ 次转移就完成了相同的总时间跨度 $[0, T]$。

在此阶段之后，训练好的 `Student` 代替 `Teacher` 作为新的参考模型。随后在更粗的网格上重复整个过程，时间步长加倍 ($N \to N/2 \to N/4 \to \cdots$)，逐步将轨迹压缩为越来越少的时间步，直至达到所需的推理步数。这种迭代减半过程在保持全局时间范围的同时，持续压缩生成过程的时间分辨率。

流图学习的统一视角。 各种方法，包括知识蒸馏（KD）和预测蒸馏（PD），都可以用统一的损失框架来表示：

$$\mathcal{L}_{\text{oracle}}(\boldsymbol{\theta}) := \mathbb{E}_{s,t}\mathbb{E}_{\mathbf{x}_s \sim p_s}\left[w(s,t)d\big(\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t), \boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s)\big)\right], \tag{10.1.4}$$

其中 $\boldsymbol{\Psi}_{s\to t}$ 为预言流映射，$w(s,t) \geq 0$ 指定了不同时间对 (s,t) 的权重，$d(\cdot,\cdot)$ 为差异度量，例如 $d(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_2^2$ 或 $d(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_1$，而 p_s 表示时间 s 处的前向噪声边缘分布。由于 $\boldsymbol{\Psi}_{s\to t}$ 无法以闭式表达，必须依赖近似方法，通常通过预训练的扩散模型（教师模型）或另一个易处理的替代模型来实现。

KD 以一个简单的实例出现，Equation (10.1.4)。选择一个退化的权重 $w(s,t) = \delta(s-T)\,\delta(t-0)$ 并使用先验分布 $p_T = p_{\text{prior}}$ [3], oracle 损失 $\mathcal{L}_{\text{oracle}}(\boldsymbol{\theta})$ 简化为：

$$\mathbb{E}_{\mathbf{x}_T \sim p_T}\left\|\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_T, T, 0) - \boldsymbol{\Psi}_{T\to 0}(\mathbf{x}_T)\right\|_2^2 \approx \mathcal{L}_{\text{KD}}(\boldsymbol{\theta}),$$

通过 $\texttt{Solver}_{T\to 0} \approx \boldsymbol{\Psi}_{T\to 0}$。这种表述的另一种视角见 Section D.5。

PD 同样符合此模板，但其并非仅通过单一极端时间对 $(T, 0)$ 进行监督，而是使用了许多邻近的时间对，并施加一个简单的*局部一致性*规则：一个短步接另一个短步应与直接的两步移动相匹配。我们将在 Equation (10.3.3) 中回到这一点。

在实际应用中，主要挑战在于，预言流映射 $\boldsymbol{\Psi}_{s\to t}$ 通常没有闭式表达式，使得直接监督不可行。已开发出一系列方法来高效地近似这一目标，但其成功往往取决于教师模型的质量。我们将在 Equation (10.1.4) 中回到 Chapter 11，提出一种系统性的框架，用于从零开始的训练方法，从而在学习环中消除教师模型的

[3]这个假设在 T 足够大时成立，或者在适当的噪声调度 (α_t, σ_t) 下也成立。

影响。

10.2 Distribution-Based Distillation

已有若干研究在不同名称下同时探索了基于分布的蒸馏方法，包括分布匹配蒸馏（DMD）（**yin2024one**; **yin2024improved**）、变分得分蒸馏（VSD）（**pooledreamfusion**; **wang2023prolificdreamer**; **luo2023diff**; **lu2024simplifying**）和得分恒等蒸馏（SiD）（**zhou2024score**）。尽管存在技术差异，它们均遵循相同的原则：训练一个生成器，使其前向加噪边缘分布与教师模型的分布相匹配。我们以 VSD 作为代表性方法进行研究，因为其他方法也遵循类似的原理。

10.2.1 VSD 作为代表性方法的制定

正向过程。 令 $\{p_t\}_{t\in[0,T]}$ 表示由前向扩散过程诱导的边缘密度。

$$\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

初始分布为 $p_0 = p_{\text{data}}$。相反，令 $p_0^{\boldsymbol{\theta}}$ 表示由确定性单步生成器 $\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})$ 从潜变量 $\mathbf{z} \sim p_{\text{prior}}(\mathbf{z})$ 生成的合成样本的分布。定义 $\{p_t^{\boldsymbol{\theta}}\}_{t\in[0,T]}$ 为对 $p_0^{\boldsymbol{\theta}}$ 应用相同前向扩散过程后得到的边缘密度，即

$$\mathbf{x}_t^{\boldsymbol{\theta}} := \alpha_t \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z}) + \sigma_t \boldsymbol{\epsilon}, \tag{10.2.1}$$

其中 $\mathbf{z} \sim p_{\text{prior}}$ 和 $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 。因此，p_t 和 $p_t^{\boldsymbol{\theta}}$ 共享相同的高斯扩散核 $p_t(\mathbf{x}_t|\mathbf{x}_0)$ ，但在起始分布上有所不同（p_{data} 与 $p_0^{\boldsymbol{\theta}}$ 为一步合成样本的分布）。

训练目标与梯度。 文献通常采用 KL 散度来匹配分布 p_t 和 $p_t^{\boldsymbol{\theta}}$ ，通常是通过最小化

$$\begin{aligned}\mathcal{L}_{\text{VSD}}(\boldsymbol{\theta}) &:= \mathbb{E}_t \left[\omega(t) \mathcal{D}_{\text{KL}}(p_t^{\boldsymbol{\theta}} \,\|\, p_t)\right] \\ &= \mathbb{E}_{t,\mathbf{z},\boldsymbol{\epsilon}} \left[\omega(t) \left(\log p_t^{\boldsymbol{\theta}}(\mathbf{x}_t^{\boldsymbol{\theta}}) - \log p_t(\mathbf{x}_t^{\boldsymbol{\theta}})\right)\right],\end{aligned}$$

其中 $\omega(t)$ 为一个与时间相关的权重函数。我们将在 Section 10.2.3 中讨论为何 KL 散度在分布级蒸馏中扮演着特殊角色。

如（**wang2023prolificdreamer**）所示，当 $p_0^{\boldsymbol{\theta}^*} = p_{\text{data}}$ 时达到最优，表明生成器的分布与数据分布一致，且训练目标可作为学习数据分布的有效损失。

然而，基于密度的目标函数缺乏高效的训练机制。幸运的是，通过对 $\boldsymbol{\theta}$ 求梯度，我们得到了 Equation (10.2.2) 中的表达式，这在以下命题中进行了总结。

为简化符号表示，我们将 $\hat{\mathbf{x}}_t := \mathbf{x}_t^{\boldsymbol{\theta}}$ 定义为如 Equation (10.2.1) 所示。

Proposition 10.2.1: $\boldsymbol{\theta}$ - $\mathcal{L}_{\text{VSD}}$ 的梯度

我们有

$$\begin{aligned}&\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{VSD}}(\boldsymbol{\theta})\\=&\mathbb{E}_{t,\mathbf{z},\boldsymbol{\epsilon}}\left[\omega(t)\alpha_t\left(\nabla_{\mathbf{x}}\log p_t^{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t)-\nabla_{\mathbf{x}}\log p_t(\hat{\mathbf{x}}_t)\right)\cdot\partial_{\boldsymbol{\theta}}\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})\right].\end{aligned}\tag{10.2.2}$$

Proof for Proposition.

推导过程应用链式法则：

$$\begin{aligned}&\nabla_{\boldsymbol{\theta}}\mathbb{E}_t\left[\mathcal{D}_{\text{KL}}(p_t^{\boldsymbol{\theta}}\|p_t)\right]\\&=\mathbb{E}_{t,\mathbf{z},\boldsymbol{\epsilon}}\left[\partial_{\boldsymbol{\theta}}\left(\log p_t^{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t)-\log p_t(\hat{\mathbf{x}}_t)\right)\right]\\&=\mathbb{E}_{t,\mathbf{z},\boldsymbol{\epsilon}}\left[\underbrace{\partial_{\boldsymbol{\theta}}\log p_t^{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t)}_{\text{first}}+(\nabla_{\mathbf{x}}\log p_t^{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t))^{\top}\partial_{\boldsymbol{\theta}}\hat{\mathbf{x}}_t-(\nabla_{\mathbf{x}}\log p_t(\hat{\mathbf{x}}_t))^{\top}\partial_{\boldsymbol{\theta}}\hat{\mathbf{x}}_t\right].\end{aligned}$$

根据得分函数恒等式，第一项消去：

$$\mathbb{E}_{\hat{\mathbf{x}}_t\sim p_t^{\boldsymbol{\theta}}}\left[\partial_{\boldsymbol{\theta}}\log p_t^{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t)\right]=\int\partial_{\boldsymbol{\theta}}p_t^{\boldsymbol{\theta}}(\mathbf{x})\,\mathrm{d}\mathbf{x}=\partial_{\boldsymbol{\theta}}\int p_t^{\boldsymbol{\theta}}(\mathbf{x})\,\mathrm{d}\mathbf{x}=\partial_{\boldsymbol{\theta}}(1)=0.$$

通过重参数化 $\hat{\mathbf{x}}_t=\alpha_t\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})+\sigma_t\boldsymbol{\epsilon}$ 可得 $\partial_{\boldsymbol{\theta}}\hat{\mathbf{x}}_t=\alpha_t\partial_{\boldsymbol{\theta}}\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})$ ，因此

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{VSD}}(\boldsymbol{\theta})=\mathbb{E}_{t,\mathbf{z},\boldsymbol{\epsilon}}\left[\omega(t)\alpha_t\big(\nabla_{\mathbf{x}}\log p_t^{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t)-\nabla_{\mathbf{x}}\log p_t(\hat{\mathbf{x}}_t)\big)^{\top}\partial_{\boldsymbol{\theta}}\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})\right].$$

由此得证 Equation (10.2.2)。详见 Section D.5。

我们观察到，当对 $\boldsymbol{\theta}$ 求梯度时，评分函数自然地出现。因此，我们需要对一阶生成器的评分 $\nabla_{\mathbf{x}}\log p_t^{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t)$ 以及数据分布的评分 $\nabla_{\mathbf{x}}\log p_t(\hat{\mathbf{x}}_t)$ 进行近似，这将在下一小节中详细说明。

10.2.2 VSD 训练流水线

现有工作 (**yin2024one**; **yin2024improved**; **pooledreamfusion**; **wang2023prolificdr luo2023diff**; **lu2024simplifying**) 通常通过双层最优化方法解决此问题：在 $\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})$ 生成的样本上训练一个新的扩散模型以逼近 $\nabla_{\mathbf{x}}\log p_t^{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t)$ ，并利用预训

练的扩散模型作为难以处理的先验评分函数 $\nabla_{\mathbf{x}} \log p_t(\hat{\mathbf{x}}_t)$ 在合成样本 $\hat{\mathbf{x}}_t$ 上的代理 *(即教师的评分)*。更精确地说，训练过程通过两个阶段交替进行：

- **得分估计阶段**。固定 $\boldsymbol{\theta}$ 。令 $\hat{\mathbf{x}}_0 = \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})$ 和 $\hat{\mathbf{x}}_t = \alpha_t \hat{\mathbf{x}}_0 + \sigma_t \boldsymbol{\epsilon}$ 满足 $\mathbf{z} \sim p_{\text{prior}}$ ，$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 。使用已知的高斯扩散核 $p_t(\mathbf{x}_t|\mathbf{x}_0)$ 通过 DSM 训练 $s_{\boldsymbol{\zeta}}$ ：

$$\mathcal{L}_{\text{DSM}}(\boldsymbol{\zeta}; \boldsymbol{\theta}) = \mathbb{E}_{t,\mathbf{z},\boldsymbol{\epsilon}} \Big\| \mathbf{s}_{\boldsymbol{\zeta}}(\hat{\mathbf{x}}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\hat{\mathbf{x}}_t|\hat{\mathbf{x}}_0) \Big\|^2,$$

  这在最优解处（对于固定的 $\boldsymbol{\theta}$ ）给出 $\mathbf{s}_{\boldsymbol{\zeta}}(\cdot, t) \approx \nabla_{\mathbf{x}} \log p_t^{\boldsymbol{\theta}}(\cdot)$。

- **生成器更新阶段**。在 $s_{\boldsymbol{\zeta}}$ 冻结（停止梯度）的情况下，通过使用 Equation (10.2.2) 中的梯度来更新 $\boldsymbol{\theta}$，将各个得分项替换为相应的代理项：

$$\mathbf{s}_{\boldsymbol{\zeta}}(\hat{\mathbf{x}}_t, t) \approx \nabla_{\mathbf{x}} \log p_t^{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t), \text{ and } \mathbf{s}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_t, t) \approx \nabla_{\mathbf{x}} \log p_t(\hat{\mathbf{x}}_t) \quad \text{(teacher)}.$$

  Equation (10.2.2) 然后近似变为：

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{VSD}}(\boldsymbol{\theta}) \approx \mathbb{E}_{t,\mathbf{z},\boldsymbol{\epsilon}} \Big[\omega(t) \alpha_t \big(\mathbf{s}_{\boldsymbol{\zeta}}(\hat{\mathbf{x}}_t, t) - \mathbf{s}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_t, t) \big)^\top \partial_{\boldsymbol{\theta}} \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z}) \Big].$$

这两个阶段重复进行，直到对所有 t ，在 $p_t^{\boldsymbol{\theta}}$ 的支撑集上，$\mathbf{s}_{\boldsymbol{\zeta}}(\cdot, t) \approx \mathbf{s}_{\boldsymbol{\phi}^\times}(\cdot, t)$ 为零，从而使 Equation (10.2.2) 中的插件梯度消失。在收敛状态下，我们有 $p_t^{\boldsymbol{\theta}} \approx p_t^{\boldsymbol{\phi}^\times}$（教师的边缘分布）对所有 $t > 0$ 成立。由于前向加噪算子（高斯卷积）对于任意固定的 $t > 0$ 都是单射的，因此可得 $p_0^{\boldsymbol{\theta}} \approx p_0^{\boldsymbol{\phi}^\times}$ （教师的 $t = 0$ 分布）。因此，学成的一步生成器 $\mathbf{G}_{\boldsymbol{\theta}}$ 在 $t = 0$ 处与教师的分布相匹配；当教师能很好地近似 p_{data} 时，这进一步意味着 $p_0^{\boldsymbol{\theta}} \approx p_{\text{data}}$ 。

10.2.3 附加讨论：散度选择与 VSD 应用

超越 KL 散度：我们能否使用一般的散度？ 原则上，可以将 VSD 中的前向 KL 项 $\mathcal{D}_{\text{KL}}(p_t^{\boldsymbol{\theta}} \| p_t)$ 替换为更一般的散度族，例如 f -散度（见 Equation (1.1.4)）：

$$\mathcal{D}_f(p_t^{\boldsymbol{\theta}} \| p_t) = \int p_t(\mathbf{x}) f\left(\frac{p_t^{\boldsymbol{\theta}}(\mathbf{x})}{p_t(\mathbf{x})} \right) \mathrm{d}\mathbf{x}.$$

然而，梯度 $\nabla_{\boldsymbol{\theta}} \mathcal{D}_f(p_t^{\boldsymbol{\theta}} \| p_t)$ 依赖于 密度比

$$r_t(\mathbf{x}) = \frac{p_t^{\boldsymbol{\theta}}(\mathbf{x})}{p_t(\mathbf{x})},$$

通过 $f'(r_t)$，这对于一个 隐式学生生成器来说是不易处理的。这里的“学生”被称为 隐式，因为它可以通过随机映射 $\hat{\mathbf{x}}_t = \alpha_t \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z}) + \sigma_t \boldsymbol{\epsilon}$ 生成样本 $\hat{\mathbf{x}}_t$，但其诱导密度 $p_t^{\boldsymbol{\theta}}(\mathbf{x})$ 并未提供闭式表达式或似然。因此，在此情景下，计算 $\mathcal{D}_f$ 的泛函导数需要对 $r_t(\mathbf{x})$ 或其对数梯度进行逐点访问，而这两种方式均无法在此设置中评估。一种常见的解决方法是引入一个辅助评论员或判别器，通过 f 散度的变分公式来近似密度比，如 f -GAN (**nowozin2016f**) 所示，尽管这会引入额外的网络和嵌套的极小极大最优化。

相比之下，对于前向 KL，路径梯度简洁地简化为得分差形式（Equation (10.2.2)）：

$$\nabla_{\boldsymbol{\theta}} \mathcal{D}_{\mathrm{KL}}(p_t^{\boldsymbol{\theta}} \| p_t) = \mathbb{E}\Big[\big(\nabla_{\mathbf{x}} \log p_t^{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t) - \nabla_{\mathbf{x}} \log p_t(\hat{\mathbf{x}}_t) \big)^{\top} \partial_{\boldsymbol{\theta}} \hat{\mathbf{x}}_t \Big].$$

该结构实现了仅基于得分的易处理更新。教师端预训练的扩散模型已提供 $\nabla_{\mathbf{x}} \log p_t(\cdot)$，因此我们可以直接复用，而无需学习辅助的密度比估计量。该公式产生了一个非对抗性的训练目标，该目标保持完全可微且计算高效。

仅使用 2D 预训练扩散模型进行 3D 生成的视觉-结构分解。 VSD (**wang2023prolificdream** 连同其早期的特殊情形 SDS (**pooledreamfusion**)（其中生成器为由 $\boldsymbol{\theta}$ 参数化的 Dirac 分布），最初是为无 3D 与 2D 数据配对监督（即无真实 3D 标签）的 3D 场景而提出的。令 $\boldsymbol{\theta} \in \mathbb{R}^d$ 表示 3D 场景的参数，令 $\mathbf{R}(\boldsymbol{\theta})$ 为一个可微分渲染器，用于生成图像 $\hat{\mathbf{x}}_0 := \mathbf{R}(\boldsymbol{\theta})$。前向加噪过程定义为

$$\hat{\mathbf{x}}_t = \alpha_t \, \mathbf{R}(\boldsymbol{\theta}) + \sigma_t \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

一个预训练的 2D（图像）扩散教师模型提供得分

$$\mathbf{s}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_t, t | \mathbf{c}) \approx \nabla_{\hat{\mathbf{x}}_t} \log p_t(\hat{\mathbf{x}}_t | \mathbf{c}),$$

可选择地基于文本 $\mathbf{c}$ 进行条件设定。目标是在每个 t 处将噪声渲染的分布与教师模型的边缘分布对齐。一种最小化表述是在渲染分布下的得分对齐（VSD）目标：

$$\mathcal{L}_{\mathrm{VSD}}^{\mathrm{3D}}(\boldsymbol{\theta}) := \mathbb{E}_{t,\boldsymbol{\epsilon}}\Big[\omega(t) \, \big\| \mathbf{s}_{\boldsymbol{\zeta}}(\hat{\mathbf{x}}_t, t) - \mathbf{s}_{\boldsymbol{\phi}^\times}(\hat{\mathbf{x}}_t, t | \mathbf{c}) \big\|_2^2 \Big], \quad \hat{\mathbf{x}}_t = \alpha_t \mathbf{R}(\boldsymbol{\theta}) + \sigma_t \boldsymbol{\epsilon},$$

通过渲染器将图像空间得分引导转换为 3D 参数。在更新 $\boldsymbol{\theta}$ 时，将两个得分均视为相对于 $\hat{\mathbf{x}}_t$ 的停止梯度，可得到

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{VSD}}^{\mathrm{3D}}(\boldsymbol{\theta}) = \mathbb{E}_{t,\boldsymbol{\epsilon}}\left[\omega(t)\, \alpha_t \left(\mathbf{s}_{\boldsymbol{\zeta}} - \mathbf{s}_{\boldsymbol{\phi}^{\times}}\right)^{\top} \frac{\partial \mathbf{R}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta})\right].$$

当学生得分 $\mathbf{s}_{\boldsymbol{\zeta}}$ 被抑制(Dirac 生成器)时,公式简化为 SDS (**pooledreamfusion**)。在实际操作中，最优化严格按照 Section 10.2.2 所描述的方式交替进行：首先更新噪声渲染下的学生得分，然后通过两个得分的停止梯度更新 $\boldsymbol{\theta}$。为简洁起见，此处省略了进一步的数学细节。

10.3 Progressive Distillation

渐进式蒸馏（Progressive Distillation, PD）（**salimans2021progressive**）由两个过程组成，共同使扩散模型能够更高效地学习 PF-ODE 轨迹。其核心思想是在保持与教师轨迹一致性的前提下，逐步减少高质量采样所需的积分步数。

- **蒸馏操作**：基于预训练的教师模型（最初为扩散模型），将一个确定性采样器（例如 DDIM）蒸馏到一个学生模型中，该模型仅使用一半的采样步数即可复现相同的轨迹。
- **渐进式操作**：迭代重复此蒸馏过程，每次将步骤数减半，直到学生模型能够在较小的固定预算内（通常为 1 –4 步）生成高质量样本。

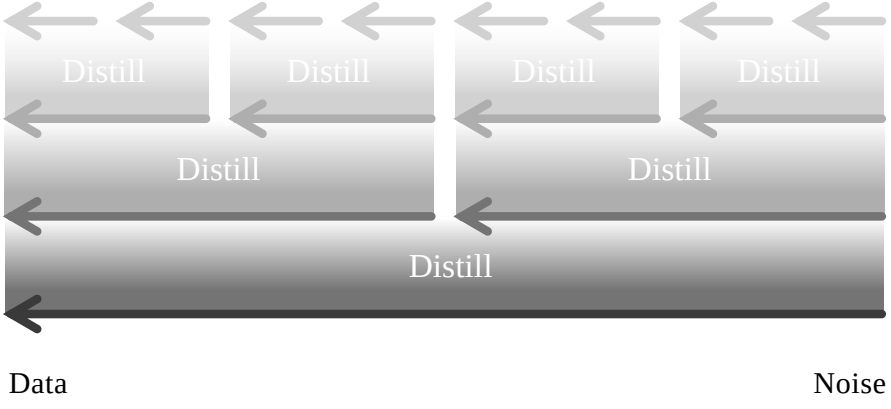


图 10.1: 渐进蒸馏（PD）示意图。在每一轮中，学生模型被训练为使单步能够复现两个相邻教师步骤的效果。该过程将 N 个教师步骤蒸馏为 $N/2$ 个学生步骤，重复此过程可逐步将轨迹长度减半，直至达到所需的步骤数。箭头表示多步教师转移如何被压缩为更少的学生步骤，从数据到噪声的方向进行。

我们首先介绍 Section 10.3.1 中 PD 的蒸馏操作，然后在 Section 10.3.2 中总结整个训练流水线。Section 10.3.4 提供了 CFG 指导的扩展。

10.3.1 PD 中的蒸馏操作

在本节中，我们将 DDIM 固定在 $\mathbf{x}$ -预测参数化作为时间步进规则，并仍用 $\texttt{Solver}_{s\to t}$ 表示将当前教师模型的 $\mathbf{x}$ -去噪器代入 DDIM 后得到的确定性映射。

在第一轮 PD（教师 = 预训练扩散模型）中，这等价于通过 DDIM 积分扩散 PF-ODE；在后续轮次（教师 = 之前的学生）中，$\texttt{Solver}_{s\to t}$ 仅仅是当前教师所诱导的 DDIM 转移，而非原始的扩散 PF-ODE。

蒸馏步骤如下：从一个噪声输入 $\mathbf{x}_s$（干净数据的扰动版本，$\mathbf{x}_s = \alpha_s \mathbf{x}_0 + \sigma_s \boldsymbol{\epsilon}$）开始，学生模型被训练以预测目标 $\tilde{\mathbf{x}}$，从而使得单个学生步骤 $s \to t$ 能够复现教师模型的两个连续步骤 $s \to u \to t$。令 $\mathbf{x}_{\boldsymbol{\phi}^\times}(\mathbf{x}, \tau)$ 表示本轮中教师模型的 $\mathbf{x}$ -预测去噪器。连续应用两次教师诱导的 DDIM 转移得到

$$\tilde{\mathbf{x}}_u := \texttt{Solver}_{s \to u}(\mathbf{x}_s;\, \mathbf{x}_{\boldsymbol{\phi}^\times}), \qquad \tilde{\mathbf{x}}_t := \texttt{Solver}_{u \to t}(\tilde{\mathbf{x}}_u;\, \mathbf{x}_{\boldsymbol{\phi}^\times}).$$

此处，我们使用 Equation (10.1.3) 的记号来表示将 $\mathbf{x}_{\boldsymbol{\phi}^\times}$ 代入 DDIM 后，从 s 到 t（起始于 $\mathbf{x}_s$）所诱导的确定性转移映射。

Question 10.3.1

什么是伪清洁 $\tilde{\mathbf{x}}$ 在时间 s 时的状态，使得求解器在直接步进 $s \to t$ 时产生的输出 $\tilde{\mathbf{x}}_t$ 与通过 $s \to u \to t$ 时相同？具体而言，确定满足以下条件的 $\tilde{\mathbf{x}}$：

$$\tilde{\mathbf{x}}_t = \texttt{Solver}_{s \to t}(\mathbf{x}_s; \tilde{\mathbf{x}}).$$

一旦获得了 $\tilde{\mathbf{x}}$ 的闭式表达式，我们便训练一个学生模型 $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s, s)$（此处也是一个 $\mathbf{x}$ -预测模型）来通过最小化 $\tilde{\mathbf{x}}$ 近似“一步两步”目标

$$\min_{\boldsymbol{\theta}} \mathbb{E}_s \mathbb{E}_{\mathbf{x}_s \sim p_s} \left[w(\lambda_s) \left\| \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s, s) - \tilde{\mathbf{x}} \right\|_2^2 \right]. \tag{10.3.1}$$

在下面，我们展示通过初等代数，DDIM 规则可以闭式地得到 $\tilde{\mathbf{x}}$（注意该结果在离散时间和连续时间下均成立）：

Lemma 10.3.1: Two-Steps-in-One Target $\tilde{\mathbf{x}}$ of DDIM

Starting from an initial condition $\mathbf{x}_s$, if the solver is taken as DDIM, then the "two-step-in-one" target $\tilde{\mathbf{x}}$ can be computed as

$$\tilde{\mathbf{x}} = \frac{\sigma_s}{\alpha_t\sigma_s - \alpha_s\sigma_t}\tilde{\mathbf{x}}_t - \frac{\sigma_t}{\alpha_t\sigma_s - \alpha_s\sigma_t}\mathbf{x}_s.$$

Here, $\tilde{\mathbf{x}}_t$ is obtained by applying DDIM (in Equation (9.2.3)) twice, from $s \to u \to t$:

$$\begin{aligned} s \to u: \quad & \tilde{\mathbf{x}}_u = \frac{\sigma_u}{\sigma_s}\mathbf{x}_s + \alpha_s\left(\frac{\alpha_u}{\alpha_s} - \frac{\sigma_u}{\sigma_s}\right)\mathbf{x}_{\phi^\times}(\mathbf{x}_s, s) \\ u \to t: \quad & \tilde{\mathbf{x}}_t = \frac{\sigma_t}{\sigma_u}\tilde{\mathbf{x}}_u + \alpha_u\left(\frac{\alpha_t}{\alpha_u} - \frac{\sigma_t}{\sigma_u}\right)\mathbf{x}_{\phi^\times}(\tilde{\mathbf{x}}_u, u). \end{aligned}$$

Proof for Lemma.

$\tilde{\mathbf{x}}_t$ must be matched with the one-step DDIM from s to t, $\tilde{\mathbf{x}}'_t$, expressed as:

$$s \to t: \quad \tilde{\mathbf{x}}'_t = \frac{\sigma_t}{\sigma_s}\mathbf{x}_s + \alpha_s\left(\frac{\alpha_t}{\alpha_s} - \frac{\sigma_t}{\sigma_s}\right)\tilde{\mathbf{x}}.$$

By equating $\tilde{\mathbf{x}}'_t$ and $\tilde{\mathbf{x}}_t$, we can solve for $\tilde{\mathbf{x}}$ in terms of $\tilde{\mathbf{x}}_t$, s, and t:

$$\begin{aligned} & \tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}'_t \\ \Longleftrightarrow\ & \tilde{\mathbf{x}}_t = \frac{\sigma_t}{\sigma_s}\mathbf{x}_s + \alpha_s\left(\frac{\alpha_t}{\alpha_s} - \frac{\sigma_t}{\sigma_s}\right)\tilde{\mathbf{x}} \\ \Longleftrightarrow\ & \tilde{\mathbf{x}} = \frac{\tilde{\mathbf{x}}_t - \frac{\sigma_t}{\sigma_s}\mathbf{x}_s}{\alpha_s\left(\frac{\alpha_t}{\alpha_s} - \frac{\sigma_t}{\sigma_s}\right)} \\ \Longleftrightarrow\ & \tilde{\mathbf{x}} = \frac{\sigma_s}{\alpha_t\sigma_s - \alpha_s\sigma_t}\tilde{\mathbf{x}}_t - \frac{\sigma_t}{\alpha_t\sigma_s - \alpha_s\sigma_t}\mathbf{x}_s. \end{aligned} \tag{10.3.2}$$

利用此公式，PD 在时间 s 计算出伪清洁目标，其单步 DDIM 步骤 $s \to t$ 可精确落在两步输出 $\tilde{\mathbf{x}}_t$ 上。

实用的离散时间网格与损失。 在实际应用中，我们固定一个递减的网格 $t_0 = T > t_1 > \cdots > t_N = 0$，为简洁起见，记作 $s := t_k$ 、$u := t_{k+1}$ 、$t := t_{k+2}$ 。教师提供一步映射 $\texttt{Teacher}_{t_k \to t_{k+1}}$ ，学生则学习一个两步跳跃映射，使其与教师

的复合映射相匹配：

$$\texttt{Student}_{t_k\to t_{k+2}} \approx \texttt{Teacher}_{t_{k+1}\to t_{k+2}} \circ \texttt{Teacher}_{t_k\to t_{k+1}}.$$

我们对三元组 $(s,u,t)=(t_k,t_{k+1},t_{k+2})$ 进行采样，其中包含 $k\in\{0,\dots,N-2\}$。目标函数 Equation (10.3.1) 变为

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{k\sim\mathcal{U}[\![0,N-2]\!]}\, \mathbb{E}_{\mathbf{x}_{t_k}\sim p_{t_k}}\Big[w(\lambda_{t_k})\,\big\|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_{t_k},t_k)-\tilde{\mathbf{x}}^{(k)}\big\|_2^2\Big],$$

其中，教师两步目标 $\tilde{\mathbf{x}}^{(k)}$ 通过引理 10.3.1 计算得到。若网格是均匀的，可写为 $t_k=T(1-k/N)$，使得

$$s=T\Big(1-\frac{k}{N}\Big),\quad u=T\Big(1-\frac{k+1}{N}\Big),\quad t=T\Big(1-\frac{k+2}{N}\Big),$$

对应于大小为 $\Delta s=T/N$ 的均匀间隔时间步。

10.3.2 PD 的完整训练流水线及其采样

通过在局部配对片段上进行 Equation (10.3.1) 训练后，Student 不再遵循原始网格的每个间隔。相反，每个学成步长覆盖两个连续的 Teacher 步长，因此 Student 在每隔一个时间点上推进，

$$t_0\to t_2\to t_4\to\cdots\to t_N,$$

因此，仅使用 $N/2$ 次转移便遍历了相同的时域 $[0,T]$。在此阶段之后，训练好的 Student 代替 Teacher 成为新的去噪模型。然后在更粗的网格上重复此过程（时间步长加倍），从而得到如下的演进过程

$$N\ \to\ N/2\ \to\ N/4\ \to\ \cdots,$$

直到达到所需的推理步数。在每次迭代中，新的 Student 从更新后的 Teacher 初始化。这种迭代式减半在保持全局时间范围的同时，逐步压缩生成过程的时间分辨率。

采样。 推理时，使用（DDIM）求解器并以当前 Student 作为去噪器，采样器在训练所诱导的粗网格上推进。第一轮后，采取“跳 2”跳跃（$t_0\to t_2\to\cdots\to t_N$），

下一轮采取“跳 4”（$t_0 \to t_4 \to \cdots \to t_N$），依此类推，每次迭代将采样步骤数减半，同时保持起始和结束时间不变。

10.3.3 附加讨论：局部半群匹配与泛化求解器的可能性

渐进蒸馏作为局部半群匹配。 在统一的目标 Equation (10.1.4) 中，难以处理的预言机目标 $\boldsymbol{\Psi}_{s\to 0}$ 被一个由教师诱导的替代目标所取代，该替代目标利用了微分方程流的 *半群性质*（详见后文 Equation (11.2.1)）：从 s 到 t 的演化应等价于从 s 到任意中间点 u，然后再从 u 到 t。

$$\boldsymbol{\Psi}_{s\to t} = \boldsymbol{\Psi}_{u\to t} \circ \boldsymbol{\Psi}_{s\to u}.$$

PD 通过训练学生模型的单步映射以匹配教师模型两个相邻单步片段的组合，从而在本地实现这一目标。

$$\mathbb{E}_s \mathbb{E}_{\mathbf{x}_s\sim p_s} \Big\| \underbrace{\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, s-2\Delta s)}_{\text{student one-step}} - \underbrace{\texttt{Solver}_{s-\Delta s\to s-2\Delta s}\big(\texttt{Solver}_{s\to s-\Delta s}(\mathbf{x}_s)\big)}_{\text{teacher two-step composition}} \Big\|_2^2. \tag{10.3.3}$$

最小化 Equation (10.3.3) 会在一个短的递减网格上实例化半群恒等式（取 $s > u > t$，其中 $u = s - \Delta s$ 且 $t = s - 2\Delta s$）：

$$\begin{aligned}\boldsymbol{\Psi}_{s\to s-2\Delta s} &= \boldsymbol{\Psi}_{s-\Delta s\to s-2\Delta s} \circ \boldsymbol{\Psi}_{s\to s-\Delta s} \\ &\approx \texttt{Solver}_{s-\Delta s\to s-2\Delta s} \circ \texttt{Solver}_{s\to s-\Delta s},\end{aligned}$$

因此，训练仅需要较短的教师片段，而无需从时间 s 到 0 的完整推演。

为了与公式 Equation (10.1.4) 中的几步去噪器视角相连接，将学生的几步映射定义为学成跳跃的复合：

$$\underbrace{\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, 0)}_{\text{few-step denoiser}} = \big(\mathbf{G}_{\boldsymbol{\theta}}(\cdot, 2\Delta s, 0) \circ \cdots \circ \mathbf{G}_{\boldsymbol{\theta}}(\cdot, s, s-2\Delta s)\big)(\mathbf{x}_s).$$

从概念上讲，Equation (10.3.3) 为全局回归提供了一种高效的局部代理。

$$\mathbb{E}_{s,\mathbf{x}_s} \Big\| \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, 0) \; - \; (\texttt{Solver})^{\circ}_{s\to 0}(\mathbf{x}_s) \Big\|_2^2,$$

其中 $(\texttt{Solver})^{\circ}_{s\to 0}$ 表示从 s 到 0 在步长为 Δs 的网格上的教师完整组合，作为

$\mathbf{\Psi}_{s\to 0}$ 的代理。

我们可以使用其他求解器吗？ 在上述 PD 介绍中，我们专注于以 $\mathbf{x}$ -预测参数化形式的 DDIM 作为具体的 PF–ODE 采样器。基于网格二分的局部半群匹配在确定性状态到状态映射层面是求解器无关的，并可扩展至 Chapter 9 中的标准参数化转换后的时间步进方法（$\mathbf{x}$, $\boldsymbol{\epsilon}$, $\mathbf{v}$, 得分）。然而，这里的闭式伪目标依赖于一个 *单步、显式更新*，其一步映射在回归目标上是 *仿射的*（如 DDIM 以及应用于 PF–ODE 的显式单步方案，例如指数–欧拉或显式龙格-库塔）。对于需要历史记录或内部求解的 *多步或 隐式求解器*，应直接匹配相应的转移映射（参见 Equation (10.3.3)），并提供必要的历史记录或预热起点；此时通常不存在类似的闭式反演。

如果采样器是随机的，则对每个样本固定噪声序列以获得确定性转移 $\texttt{Teacher}_{s\to t}^{(\omega)}$（其中 ω 为固定的噪声种子）。在这种情况下，PD 退化为一个固定的转移映射；闭式伪目标通常需要单步显式仿射更新；否则，采用如 Equation (10.3.3) 所示的直接匹配方法。

10.3.4 带引导的帕金森病

meng2023distillation 提出了一种用于蒸馏无分类器引导（CFG）扩散模型的两阶段流水线：（1）将引导信息蒸馏到一个单一网络中，该网络以引导权重作为输入；（2）应用渐进式蒸馏（PD）来减少采样步骤。他们在像素空间和潜在空间（例如，Stable Diffusion）中均进行了验证。

第一阶段蒸馏：蒸馏指导。 令 $\mathbf{x}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s, s, \mathbf{c})$ 表示在 “$\mathbf{x}$ -预测” 参数化下的（预训练）条件扩散模型输出（即，一个干净的估计值），在时间 s 和条件 $\mathbf{c}$ 下；该条件也可以为空，$\mathbf{c} = \emptyset$ （无条件分支）。在 Equation (8.3.3) 中的 ω -加权 CFG 组合可表示为

$$\mathbf{x}^{\omega}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s, s, \mathbf{c}) := (1+\omega)\,\mathbf{x}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s, s, \mathbf{c}) - \omega\,\mathbf{x}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s, s, \emptyset), \tag{10.3.4}$$

其中 $\omega \sim p_\omega(\omega)$ 为某种上下文无关语法权重分布 p_ω，通常为 $p_\omega(\omega) = \mathcal{U}[\omega_{\min}, \omega_{\max}]$。

第一阶段引入了一个新模型 $\mathbf{x}_{\boldsymbol{\theta}_1}(\mathbf{x}_s, s, \mathbf{c}, \omega)$，该模型直接以 ω 作为输入，并通过监督回归学习来复现 CFG 输出 $\mathbf{x}^{\omega}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s, s, \mathbf{c})$：

$$\min_{\boldsymbol{\theta}_1} \mathbb{E}_{\omega\sim p_\omega, s, \mathbf{x}\sim p_{\text{data}}, \mathbf{x}_s\sim p(\mathbf{x}_s|\mathbf{x})} \lambda(s) \big\| \mathbf{x}_{\boldsymbol{\theta}_1}(\mathbf{x}_s, s, \mathbf{c}, \omega) - \mathbf{x}^{\omega}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s, s, \mathbf{c}) \big\|_2^2.$$

此处 $\lambda(s)$ 为一种标准的与调度相关的权重；在每次迭代中对 ω 采样，可使单个网络在任意引导强度下模拟 CFG。

第二阶段蒸馏：PD。 第一阶段模型 $\mathbf{x}_{\boldsymbol{\theta}_1}(\mathbf{x}_s, s, \mathbf{c}, \omega)$ 作为 PD 中的教师模型，被逐步蒸馏为采样步骤更少的学生模型 $\mathbf{x}_{\boldsymbol{\theta}_2}(\mathbf{x}_s, s, \mathbf{c}, \omega)$ ，遵循 Section 10.3.2。在每次迭代中，步骤数减半（例如，$N \to N/2 \to N/4 \to \cdots$）。

10.4 闭幕词

本章介绍了基于训练的加速的第一个主要范式。在耗尽了通过数值求解器实现的无需训练的改进方法后，我们将重点转向了一种新策略：训练一个快速的学生生成器，使其学习复制一个缓慢的预训练教师扩散模型的行为。

我们探讨了两种主要的蒸馏理念。首先，在基于分布的蒸馏中，以变分评分蒸馏（VSD）等方法为代表，学生模型的输出分布被训练以匹配教师模型的分布。这是通过在不同噪声水平下对齐两者的评分函数来实现的，提供了一个稳定且非对抗性的目标。其次，在流映射蒸馏中，我们看到像渐进式蒸馏（PD）这样的方法如何训练学生模型直接逼近教师模型的解轨迹。PD 采用迭代方法，每轮将采样步骤数量减半，证明是一种强大而实用的方法，可将一个长周期的迭代过程压缩为仅几步。

这些蒸馏技术成功弥合了迭代扩散模型的高样本质量与单步生成器的推理速度之间的差距，为高效、高保真的合成提供了一条极具吸引力的路径。

然而，依赖预训练的教师模型引入了两阶段流水线：首先训练一个速度较慢但性能强大的教师模型，然后将其知识蒸馏到快速的学生模型中。这提出了生成式建模研究最前沿的一个根本性问题：我们能否完全跳过教师模型？

是否可能设计一种独立的训练原则，直接从数据中学习这些快速、少步骤的生成器？本书的最后一章将探讨这一问题。

1. 我们将探索前沿方法，例如一致性模型，该模型学习从微分方程轨迹上的任意一点到其终点的映射关系。

2. 我们将深入探讨一致性模型的泛化概念，这类模型学习在单步内将微分方程轨迹上的任意一点映射到另一点。

从改进求解器或提炼解决方案转向学习解决方案映射本身，标志着向一类既具有理论基础又设计高效的新一代生成式模型迈出了重要一步。

11

从零开始学习快速生成器

> 真理永远存在于简单之中，而不在于事物的繁多与混乱。
>
> 艾萨克 · 牛顿

在 Chapter 10 中我们看到，扩散模型中的慢速迭代采样器可以通过蒸馏压缩为少量步骤的生成器。从工程角度来看，两阶段流水线是实用的，因为它们将复杂的生成训练任务分解为清晰且独立的目标。第一阶段学习数据分布，第二阶段加速采样或提升质量。这种分离使得每个阶段可以独立优化，从而使整个系统更易于管理、更加稳定和可靠。

然而，在本章中，重点转向了推动深度生成建模发展的核心问题：

Question 11.0.1

我们能否设计一种独立的生成原则，使其能够稳定高效地进行训练，实现快速采样，并允许用户轻松引导或控制生成内容？

在本章中，我们沿着这一方向展开讨论，提出一种替代方法：在不依赖预训练模型的情况下，训练多步扩散生成器。我们的重点是 *流映射模型*，该模型学习一种直接变换，通过近似 PF-ODE 的理想流映射，将样本沿时间轴进行迁移。这种表述提供了一种合理的方法，将概率质量从先验分布 p_{prior} 传输到数据分布 p_{data} ，同时保持前向扩散过程在每个中间时刻指定的边缘分布 p_t 。

11.1 序言

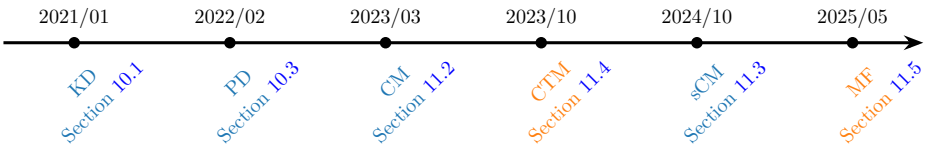


图 11.1: 流图模型的时间线。 我们用蓝色 表示特殊情况$\boldsymbol{\Psi}_{s\to 0}$，用橙色 表示一般地图$\boldsymbol{\Psi}_{s\to t}$。

流图模型的动机。 在 Chapter 10 中，我们展示了如何通过从预训练的扩散模型中蒸馏知识来获得少步生成器，从而估计在 oracle 流映射损失 $\mathcal{L}_{\text{oracle}}(\boldsymbol{\theta})$（参见 Equation (10.1.4)）中的不可访问回归目标。该方法有效且实用：可以设计一个两阶段流水线以提高鲁棒性，并且在数据和计算效率方面通常仍具有竞争力。

在本章中，我们将关注深度生成建模核心的一个更广泛挑战：*我们能否建立一种独立的生成原则，实现稳定、可扩展且高效的训练，快速的采样与生成，并能轻松地根据用户意图进行引导，而无需依赖预训练模型？设计这种独立的原则是生成式建模的核心所在。*

扩散模型提供了一种有用的设计原则：从一个连续时间的前向过程开始，该过程逐步将数据转换为简单的先验（噪声）作为参考，并将建模任务定义为学习反向时间传输，以恢复该过程并匹配所需的边缘分布。这种时间依赖的表述也使得在中间步骤中更容易引导生成过程，相较于 one-shot 生成映射更为便捷。针对受扩散启发的方法，这引出了一个问题：

Question 11.1.1

> *我们能否在没有预训练模型的情况下，仅通过一个网络 $\mathbf{G}_{\boldsymbol{\theta}}(\cdot, s, t)$（一种 流映射模型）来学习流映射 $\boldsymbol{\Psi}_{s\to t}(\cdot)$，同时保持高保真度的生成？*

本章提出了实现这一目标的方法，围绕单一目标展开，该目标同样支撑蒸馏，并为流映射表述提供了一个统一的视角 (**boffi2024flow**; **hu2025cmt**):

$$\mathcal{L}_{\text{oracle}}(\boldsymbol{\theta}) := \mathbb{E}_{s,t}\,\mathbb{E}_{\mathbf{x}_s\sim p_s}\Big[w(s,t)d\big(\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s,s,t),\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s)\big)\Big]. \tag{10.1.4}$$

此处 s,t 从某种时间分布（例如均匀分布）中采样，$w(s,t)\geq 0$ 为时间对 (s,t) 分配权重，而 $d(\cdot,\cdot)$ 是一种差异度量，例如平方 ℓ_2 范数。最优流映射 $\boldsymbol{\Psi}_{s\to t}$

表示理想的变换，它将时间 s 的状态 $\mathbf{x}_s$ 直接传输至时间 t ：

$$\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s) = \mathbf{x}_s + \int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u,$$

其中，神谕漂移量表示为

$$\mathbf{v}^*(\mathbf{x}_u, u) = \mathbb{E}\left[\alpha'_u\mathbf{x}_0 + \sigma'_u\boldsymbol{\epsilon}|\mathbf{x}_u\right],$$

同时也可以采用等价的参数化形式（见 Chapter 6），常见的选择包括 $\mathbf{x}$ 预测和 $\mathbf{v}$ 预测形式。

在预言损失的最优解处，学成的模型能确切地恢复真实的流映射。

$$\mathbf{G}^*(\mathbf{x}_s, s, t) = \boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s), \quad \text{for all } s, t, \text{ and } \mathbf{x}_s \sim p_s.$$

由于流映射 $\boldsymbol{\Psi}_{s\to t}$ 无法以闭式表达，因此必须对其进行近似。一种选择，如 Chapter 10 所讨论的，是依赖于预训练的扩散模型。另一种选择，正如本章将要展示的，是可以引入新的、更易处理的替代模型。为清晰起见，现有方法可大致根据训练过程在循环中是否调用教师模型进行分类：*蒸馏*，其显式调用教师模型，以及 *从零开始训练*，通过构建自包含的替代模型来避免调用教师模型。

基于这一原则性目标，我们现在转向系统性方法来学习流图模型，旨在开发既实用又能更准确反映真实数据分布且计算高效的生成方法。我们首先对此范式进行高层次的介绍。

特殊流图：一致性函数。 *一致性模型* (**song2023consistency**) 是流映射学习早期的开创性方法之一。它们学习一个几步去噪器 $\mathbf{f}_{\boldsymbol{\theta}}(\cdot, s)$，以近似流映射到原点的特殊情况：

$$\boldsymbol{\Psi}_{s\to 0}(\cdot), \qquad s \in (0, T].$$

其核心思想是，每个含噪样本 $\mathbf{x}_s$ 应在轨迹末端被映射回干净的数据点 $\mathbf{x}_0$。形式上，CM 家族 (**song2023consistency**; **songimproved**; **geng2024consistency**; **lu2024simplifying**) 的理想训练目标为

$$\mathcal{L}_{\text{oracle-CM}}(\boldsymbol{\theta}) := \mathbb{E}_s\mathbb{E}_{\mathbf{x}_s\sim p_s}\left[w(s)d\left(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s, s), \boldsymbol{\Psi}_{s\to 0}(\mathbf{x}_s)\right)\right]. \tag{11.1.0}$$

然而，在实际应用中，预言机 $\boldsymbol{\Psi}_{s\to 0}(\mathbf{x}_s)$ 无法获得。因此，它被一个*停止梯*

度目标 $\mathbf{f}_{\boldsymbol{\theta}^-}$ 所替代，该目标取自同一条轨迹上稍早的一个步骤 $\boldsymbol{\Psi}_{s\to s-\Delta s}(\mathbf{x}_s)$：

$$\boldsymbol{\Psi}_{s\to 0}(\mathbf{x}_s) \approx \mathbf{f}_{\boldsymbol{\theta}^-}\left(\boldsymbol{\Psi}_{s\to s-\Delta s}(\mathbf{x}_s),\, s-\Delta s\right), \quad \Delta s > 0,$$

其中，$\boldsymbol{\Psi}_{s\to s-\Delta s}(\mathbf{x}_s)$ 本身也必须被近似。两种实用的策略可用：(i) 蒸馏，依赖于一个预训练的扩散模型；(ii) 从零开始训练，使用无教师指导的一点估计。

通用流程图。 两种代表性方法是 一致性轨迹模型（CTM）和 平均流（MF）。

一致性轨迹模型。 一致性轨迹模型（CTM）(**kim2023consistency**) 是首篇学习任意起始和结束时间下的通用流映射 $\boldsymbol{\Psi}_{s\to t}$ 的工作，可被视为 Equation (10.1.4) 统一目标下的一个具体实例。CTM 采用了一种受欧拉启发的参数化方法，通过将最优流映射表示为

$$\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s) := \mathbf{x}_s + \int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u = \frac{t}{s}\mathbf{x}_s + \frac{s-t}{s}\underbrace{\left[\mathbf{x}_s + \frac{s}{s-t}\int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u\right]}_{\approx\, \mathbf{g}_{\boldsymbol{\theta}}},$$

这激发了神经参数化

$$\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) := \frac{t}{s}\,\mathbf{x}_s + \frac{s-t}{s}\,\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t),$$

其中 $\mathbf{g}_{\boldsymbol{\theta}}$ 是一个经过训练的神经网络，使得 $\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s) \approx \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t)$ 。

由于无法访问预言机 $\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s)$，CTM 会针对在中间时间 u 处评估的 停止梯度目标进行训练：

$$\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s) \approx \mathbf{G}_{\boldsymbol{\theta}^-}\left(\boldsymbol{\Psi}_{s\to u}(\mathbf{x}_s),\, u,\, t\right), \qquad u \in [t, s],$$

其中，中间状态 $\boldsymbol{\Psi}_{s\to u}(\mathbf{x}_s)$ 以两种方式之一进行近似：(i) 蒸馏，即对预训练的扩散教师模型应用几步求解器；或 (ii) 从零开始训练，通过 $\mathbf{G}_{\boldsymbol{\theta}}$ 参数化直接构建自诱导教师模型。

平均流。 平均流（MF）(**geng2025mean**) 通过在区间 $[t, s]$（其中 $t \le s$）上对平均漂移进行建模，扩展了流匹配的思想：

$$\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) \approx \mathbf{h}^*(\mathbf{x}_s, s, t) := \frac{1}{t-s}\int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u,$$

同时与 Equation (10.1.4) 对齐。对恒等式求导

$$(t-s)\,\mathbf{h}^*(\mathbf{x}_s,s,t)=\int_s^t \mathbf{v}^*(\mathbf{x}_u,u)\,\mathrm{d}u$$

关于 s 的结果得到一个自指关系，该关系激发了 MF 目标

$$\mathcal{L}_{\mathrm{MF}}(\boldsymbol{\theta}):=\mathbb{E}_s\,\mathbb{E}_{\mathbf{x}_s\sim p_s}\Big[w(s)\,\big\|\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_s,s,t)-\mathbf{h}^{\mathrm{tgt}}_{\boldsymbol{\theta}^-}(\mathbf{x}_s,s,t)\big\|_2^2\Big],$$

带停止梯度的目标

$$\mathbf{h}^{\mathrm{tgt}}_{\boldsymbol{\theta}^-}(\mathbf{x}_s,s,t):=\mathbf{v}^*(\mathbf{x}_s,s)-(s-t)\,(\mathbf{v}^*(\mathbf{x}_s,s)\,\partial_{\mathbf{x}}\mathbf{h}_{\boldsymbol{\theta}^-}+\partial_s\mathbf{h}_{\boldsymbol{\theta}^-})\,.$$

在实际应用中，预言速度 $\mathbf{v}^*(\mathbf{x}_s,s)$ 也必须被近似。两种常用策略是：(i) 蒸馏，利用一个通过流匹配训练的预训练扩散模型；或 (ii) 从零开始训练，使用由前向破坏过程 $\mathbf{x}_s=\alpha_s\mathbf{x}_0+\sigma_s\boldsymbol{\epsilon}$ 导出的单点条件速度 $\alpha_s'\,\mathbf{x}_0+\sigma_s'\,\boldsymbol{\epsilon}$ 。

CTM 与 MF 之间的关系。 CTM 和 MF 近似相同的路径积分，但对其不同的代理进行参数化：

$$\begin{aligned}\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s)&:=\mathbf{x}_s+\int_s^t \mathbf{v}^*(\mathbf{x}_u,u)\,\mathrm{d}u\\&=\frac{t}{s}\,\mathbf{x}_s+\frac{s-t}{s}\underbrace{\Big[\mathbf{x}_s+\frac{s}{s-t}\int_s^t \mathbf{v}^*(\mathbf{x}_u,u)\,\mathrm{d}u\Big]}_{\approx\,\mathbf{g}_{\boldsymbol{\theta}}}\\&=\mathbf{x}_s+(t-s)\underbrace{\Big[\frac{1}{t-s}\int_s^t \mathbf{v}^*(\mathbf{x}_u,u)\,\mathrm{d}u\Big]}_{\approx\,\mathbf{h}_{\boldsymbol{\theta}}}.\end{aligned}$$

换句话说，CTM 通过 $\mathbf{g}_{\boldsymbol{\theta}}$ 学习一个 *斜率位移*，而 MF 学习的是 *平均漂移* $\mathbf{h}_{\boldsymbol{\theta}}$；两者都是近似定义 $\boldsymbol{\Psi}_{s\to t}$ 的同一积分的一致方法。

接下来会发生什么？ 我们从 CM 家族开始，其重点在于特定的流映射 $\boldsymbol{\Psi}_{s\to 0}$。这一部分涵盖了其在离散时间下的原点 Section 11.2 以及连续时间下的扩展 Section 11.3。随后，我们转向一般的流映射，并对两个关键代表 CTM 和 MF 进行详细讨论。它们的参数化、训练策略以及实际近似分别在 Section 11.4 和 Section 11.5 中给出。

我们注意到，文献 Section D.6中提出的*阐明扩散模型*（EDM）为设计 $\mathbf{x}$-预

测模型的网络参数化提供了系统的指导方针，并表现出强劲的实验性能。尽管本节可视为可选内容，但 EDM 的表述为 CM 风格的模型提供了一个有价值的理论基础。

为了便于后续阐述的清晰性，我们并未严格遵循这些方法出现的时间顺序。相反，我们按照概念上的关系来组织讨论。尽管如此，为了承认原创性并尊重时间顺序，我们在 Figure 11.1中提供了历史时间线。

11.2 特殊流图：离散时间的一致性模型

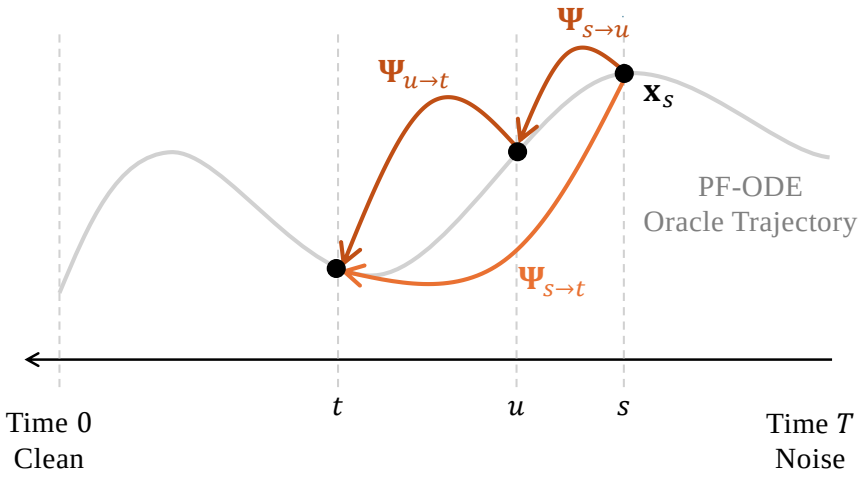


图 11.2: 流映射半群性质示意图。该性质表明，从 s 转移到 u，然后从 u 转移到 t，等价于直接从 s 转移到 t。

流图的一个重要原理：半群性质。 一致性模型（在 Sections 11.2 and 11.3 中提出）及其推广形式——一致性轨迹模型（Section 11.4），通过利用流映射的一个关键数学结构来定义其回归目标。该结构是基本的 *半群性质*：

$$\boldsymbol{\Psi}_{u\to t}\circ\boldsymbol{\Psi}_{s\to u}=\boldsymbol{\Psi}_{s\to t},\quad \boldsymbol{\Psi}_{s\to s}=\mathbf{I},\quad \text{for all } s,u,t\in[0,T]. \tag{11.2.1}$$

直观上，这意味着如果我们先通过 $\boldsymbol{\Psi}_{s\to u}$ 从 s 演化到 u，然后再通过 $\boldsymbol{\Psi}_{u\to t}$ 从 u 演化到 t，最终得到的状态与直接从 s 演化到 t 的状态完全相同。这只不过是常微分方程求解的基本原理。[1]: 一旦流的起始点被确定，其未来的演化就完全被决定，沿着一条明确的路径进行。无论我们以一个长步骤跟随这条路径，还是将其划分为更小的区间，我们仍然沿着相同的轨迹前进，并到达相同的最终状态。

为了进一步理解半群性质，考虑 PF-ODE 的解轨迹 $\{\mathbf{x}(s)\}_{s\in[0,T]}$

$$\frac{\mathrm{d}\mathbf{x}(\tau)}{\mathrm{d}\tau}=\mathbf{v}^*(\mathbf{x}(\tau),\tau),$$

在固定初始条件 $\mathbf{x}(T)$ 于时间 T 时，向后求解时间。若将终端时间固定在 $t=0$

[1] 半群性质由常微分方程初值问题的唯一性定理得出（参见 Chapter A）。

，则相应的流映射可更简洁地表示为

$$\mathbf{f}^*(\cdot, s) := \mathbf{\Psi}_{s\to 0}(\cdot),$$

这被称为一致性函数。根据构造，该函数直接继承了 Equation (11.2.1) 与 $t = 0$ 的半群恒等式的几个基本性质：

(i) **全局一致性**：轨迹上的每个点都映射到同一个干净的终点，

$$\mathbf{f}^*(\mathbf{x}(s), s) = \mathbf{x}(0), \quad \text{for all } \ s \in [0, T].$$

这是因为

$$\begin{aligned}\mathbf{f}^*(\mathbf{x}(s), s) &= \mathbf{\Psi}_{s\to 0}\big(\mathbf{\Psi}_{0\to s}(\mathbf{x}(0))\big) = \big(\mathbf{\Psi}_{s\to 0} \circ \mathbf{\Psi}_{0\to s}\big)(\mathbf{x}(0)) \\ &= \mathbf{\Psi}_{0\to 0}(\mathbf{x}(0)) = \mathbf{x}(0).\end{aligned}$$

(ii) **自洽性**：同一轨迹上的任意两点必须产生相同的输出，

$$\mathbf{f}^*(\mathbf{x}(s), s) = \mathbf{f}^*(\mathbf{x}(u), u), \quad \text{for all } \ s, u \in [0, T]. \tag{11.2.2}$$

这是半群恒等式的直接重新解释：$\Psi_{s\to 0} \circ \Psi_{0\to s} = \Psi_{u\to 0} \circ \Psi_{0\to u}$。

(iii) **局部一致性**：一致性函数在沿轨迹评估时关于 s 不变，

$$\frac{\mathrm{d}}{\mathrm{d}s}\, \mathbf{f}^*(\mathbf{x}(s), s) = 0, \qquad \mathbf{f}^*(\mathbf{x}(0), 0) = \mathbf{x}(0). \tag{11.2.3}$$

这由全局一致性得出，全局一致性表明 $\mathbf{f}^*(\mathbf{x}(s), s)$ 在轨迹上不随 s 变化。

这三个性质都是等价的。每个性质都表明，在任意解轨迹 $s \mapsto \mathbf{x}(s)$ 上，从流到原点/一致性映射 $\mathbf{f}^*(\mathbf{x}(s), s) = \mathbf{\Psi}_{s\to 0}(\mathbf{x}(s))$ 产生的终点 $\mathbf{x}(0)$ 与起始时间无关。

一致性模型的目标。 一个 CM 旨在训练一个神经网络 $\mathbf{f}_{\boldsymbol{\theta}}\colon \mathbb{R}^D \times [0, T] \to \mathbb{R}^D$ 来近似特殊的流映射 $\mathbf{\Psi}_{s\to 0}$ ，即一致性函数[2]. 其核心思想是在 PF-ODE 的多条

[2]一个常微分方程的相容性函数的概念可推广至随机微分方程的函数 $\mathbf{f}(\mathbf{x}, t)$，使得 $\mathbf{f}(\mathbf{x}_t, t)$ 关于随机微分方程的自然滤波是（局部）鞅，即

$$\mathbb{E}[\mathbf{f}(\mathbf{x}_t, t)|\mathbf{x}_s] = \mathbf{f}(\mathbf{x}_s, s), \quad \text{对所有} \quad t \geq s.$$

该推广由 (**daras2023consistent**; **lai2023fp**) 提出/观察到，其理论联系在 (**lai2023equivalence**) 中进行了总结。

轨迹上施加半群性质，确保同一数据点的不同噪声版本都能一致地映射回相同的干净原点（更准确地说，这对应于 Equation (11.2.1) 中的特殊情况 $t = 0$ 和 $u = s - \Delta s$）。

然而，实现这一目标有多种方式。选择取决于是否已有预训练的扩散模型，以及训练是在离散时间还是连续时间框架下进行。我们首先在 Table 11.1 中总结这些变体，并在 Figure 11.3 中说明它们的目标。随后的章节 (Sections 11.2 and 11.3) 逐步展开每种方法的细节。

表 11.1: 一致性模型的训练目标

	Distillation	**From Scratch**
Discrete-time	Equation (11.2.4)	Equation (11.2.6)
Continuous-time	Equation (11.3.5)	Equation (11.3.6)

11.2.1 离散时间近似用于学习一致性函数

原则上，一致性函数可以通过最小化预言机损失 Equation (11.1.0) 来学成。

$$\mathcal{L}_{\text{oracle-CM}}(\boldsymbol{\theta}) := \mathbb{E}_s \mathbb{E}_{\mathbf{x}_s \sim p_s} \left[w(s) d\left(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s, s), \boldsymbol{\Psi}_{s \to 0}(\mathbf{x}_s) \right) \right].$$

该目标强制将每个噪声样本 $\mathbf{x}_s$ 映射回其干净的端点 $\boldsymbol{\Psi}_{s \to 0}(\mathbf{x}_s)$。

挑战在于，实际中无法获得预言机映射 $\boldsymbol{\Psi}_{s \to 0}(\mathbf{x}_s)$。为克服这一问题，**song2023consistenc** 利用 *半群性质*：任意含噪声状态及其沿相同 PF-ODE 轨迹的连续步骤必须映射到相同的清洁终点。具体而言，预言机目标被替换为从轨迹上稍早一点位置获取的 *停止梯度*目标：

$$\begin{aligned} \boldsymbol{\Psi}_{s \to 0}(\mathbf{x}_s) &= \boldsymbol{\Psi}_{s - \Delta s \to 0}\left(\boldsymbol{\Psi}_{s \to s - \Delta s}(\mathbf{x}_s) \right) \\ &\approx \mathbf{f}_{\boldsymbol{\theta}^-}\left(\boldsymbol{\Psi}_{s \to s - \Delta s}(\mathbf{x}_s),\ s - \Delta s \right), \qquad \Delta s > 0, \end{aligned}$$

其中 $\boldsymbol{\theta}^-$ 为在停止梯度操作符下的参数。另一个难点在于，中间状态 $\boldsymbol{\Psi}_{s \to s - \Delta s}(\mathbf{x}_s)$ 也不存在闭式解，必须对其进行近似。已提出了两种实际可行的模式：

采用预训练扩散模型（一致性蒸馏）。 假设我们能够访问一个预训练的扩散模型。一致性蒸馏（CD）利用教师模型通过仅模拟单步反向 ODE 来近似中间状态 $\boldsymbol{\Psi}_{s \to s - \Delta s}(\mathbf{x}_s)$。

$$\boldsymbol{\Psi}_{s \to s - \Delta s}(\mathbf{x}_s) \approx \texttt{Solver}_{s \to s - \Delta s}(\mathbf{x}_s).$$

更具体地说，一个预训练的扩散模型提供了评分函数的估计值 $\mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s, s) \approx \nabla_{\mathbf{x}_s} \log p_s(\mathbf{x}_s)$ 。利用这一点，可以从 $\mathbf{x}_s$ 执行一步 DDIM 更新，以获得在 $s' = s - \Delta s$ 状态的近似值：

$$\begin{aligned}\boldsymbol{\Psi}_{s\to s-\Delta s}(\mathbf{x}_s) &\approx \frac{\alpha_{s'}}{\alpha_s}\,\mathbf{x}_s + \sigma_s^2\left(\frac{\alpha_{s'}}{\alpha_s} - \frac{\sigma_{s'}}{\sigma_s}\right)\nabla_{\mathbf{x}_s}\log p_s(\mathbf{x}_s)\\ &\approx \frac{\alpha_{s'}}{\alpha_s}\,\mathbf{x}_s + \sigma_s^2\left(\frac{\alpha_{s'}}{\alpha_s} - \frac{\sigma_{s'}}{\sigma_s}\right)\mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s, s)\\ &:= \tilde{\mathbf{x}}_{s'}^{\boldsymbol{\phi}^\times}.\end{aligned}$$

将此构造与停止梯度目标相结合，可得到一个实用的 离散时间代理目标函数，用于近似最优损失 $\mathcal{L}_{\text{oracle-CM}}(\boldsymbol{\theta})$。形式上，在划分 $0 = s_1 < s_2 < \cdots < s_N = T$ 上，CD 训练目标由下式给出

$$\mathcal{L}_{\text{CD}}^N(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \boldsymbol{\phi}^\times) := \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, i}\Big[\,\omega(s_i)\, d\big(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_{s_{i+1}}, s_{i+1}),\, \mathbf{f}_{\boldsymbol{\theta}^-}(\tilde{\mathbf{x}}_{s_i}^{\boldsymbol{\phi}^\times}, s_i)\big)\Big]. \quad (11.2.4)$$

其中，$\omega(\cdot)$ 为时变权重，$d(\cdot,\cdot)$ 为距离度量，$\boldsymbol{\theta}^-$ 为停止梯度参数，可防止退化到平凡解（例如，恒定预测）。

无预训练扩散模型（一致性训练）。 当没有可用的预训练扩散模型时，仍然可以使用简单的单点近似直接估计最优得分 $\nabla_{\mathbf{x}} \log p_s(\mathbf{x}_s)$（尽管方差较高）。回顾一下，其具有条件期望形式：

$$\begin{aligned}\nabla_{\mathbf{x}_s}\log p_s(\mathbf{x}_s) &= \mathbb{E}_{\mathbf{x}_0\sim p(\mathbf{x}_0|\mathbf{x}_s)}\left[\nabla_{\mathbf{x}_s}\log p(\mathbf{x}_s|\mathbf{x}_0)\right]\\ &= \mathbb{E}_{\mathbf{x}_0\sim p(\mathbf{x}_0|\mathbf{x}_s)}\left[-\frac{\mathbf{x}_s - \alpha_s\mathbf{x}_0}{\sigma_s^2}\right].\end{aligned}$$

上述恒等式暗示了一个简单的单样本估计量。如果通过 $\mathbf{x}_s$ 从配对样本 $(\mathbf{x}_0, \boldsymbol{\epsilon})$ 获得 $\mathbf{x}_s = \alpha_s\mathbf{x}_0 + \sigma_s\boldsymbol{\epsilon}$ ，则

$$\widehat{\nabla_{\mathbf{x}}\log p_s}(\mathbf{x}_s) := -\frac{\boldsymbol{\epsilon}}{\sigma_s} = -\frac{\mathbf{x}_s - \alpha_s\mathbf{x}_0}{\sigma_s^2}$$

作为在 $\mathbf{x}_s$ 处得分的 无偏估计量（相对于 $p(\mathbf{x}_0|\mathbf{x}_s)$ 条件无偏）。它恰好对应于降噪分数匹配中用作回归目标的 条件得分。

将此估计值代入从 s 到 $s' = s - \Delta s$ 的 DDIM 一步更新公式（见 Equation (9.2.3)）可得

$$\begin{aligned}
\boldsymbol{\Psi}_{s\to s'}(\mathbf{x}_s) &\approx \frac{\alpha_{s'}}{\alpha_s}\,\mathbf{x}_s + \sigma_s^2\left(\frac{\alpha_{s'}}{\alpha_s} - \frac{\sigma_{s'}}{\sigma_s}\right)\nabla_{\mathbf{x}_s}\log p_s(\mathbf{x}_s) \quad \text{(DDIM)} \\
&\approx \frac{\alpha_{s'}}{\alpha_s}\,\mathbf{x}_s + \sigma_s^2\left(\frac{\alpha_{s'}}{\alpha_s} - \frac{\sigma_{s'}}{\sigma_s}\right)\widehat{\nabla_{\mathbf{x}_s}\log p_s(\mathbf{x}_s)} \quad \text{(1-pt score)} \\
&= \frac{\alpha_{s'}}{\alpha_s}\,\mathbf{x}_s - \left(\frac{\alpha_{s'}}{\alpha_s} - \frac{\sigma_{s'}}{\sigma_s}\right)(\mathbf{x}_s - \alpha_s\mathbf{x}_0) \\
&= \alpha_{s'}\mathbf{x}_0 + \sigma_{s'}\boldsymbol{\epsilon},
\end{aligned} \tag{11.2.5}$$

在哪里[3] $\mathbf{x}_0$ 是相同的数据样本，$\boldsymbol{\epsilon}$ 是用于构建 $\mathbf{x}_s$ 的相同高斯噪声。

这导致了一个无教师的离散时间代理目标 $\mathcal{L}_{\text{oracle-CM}}$，其表达式为

$$\mathcal{L}_{\text{CT}}^N(\boldsymbol{\theta}, \boldsymbol{\theta}^-) := \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon},i}\left[\omega(s_i)\, d\left(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_{s_{i+1}}, s_{i+1}),\, \mathbf{f}_{\boldsymbol{\theta}^-}(\mathbf{x}_{s_i}, s_i)\right)\right], \tag{11.2.6}$$

使用 $\mathbf{x}_{s_i} = \alpha_{s_i}\mathbf{x}_0 + \sigma_{s_i}\boldsymbol{\epsilon}$ 和 $\mathbf{x}_{s_{i+1}} = \alpha_{s_{i+1}}\mathbf{x}_0 + \sigma_{s_{i+1}}\boldsymbol{\epsilon}$。

直接使用 $\alpha_{s'}\mathbf{x}_0 + \sigma_{s'}\boldsymbol{\epsilon}$ 作为 $\boldsymbol{\Psi}_{s\to s'}(\mathbf{x}_s)$ 的近似而不引入期望会导致高方差[4]. 然而，请回想降噪分数匹配中的类似情况（见 Section 6.1），其中单个条件分数样本作为训练目标，但在损失中对 $\mathbf{x}_0, \boldsymbol{\epsilon}$ 进行平均后即变为无偏。根据同样的推理，$\mathcal{L}_{\text{CT}}^N$ 中对 $\mathbf{x}_0$ 和 $\boldsymbol{\epsilon}$ 的期望会抵消这种采样噪声，从而得到一个无偏的损失级近似。以下定理形式化了该点估计的期望层面合理性。

[3] 最后一个恒等式可直接由前向破坏过程 $\mathbf{x}_s = \alpha_s\mathbf{x}_0 + \sigma_s\boldsymbol{\epsilon}$ 通过初等代数得出。

[4] 单点（条件）得分估计量 $\widehat{\nabla_{\mathbf{x}}\log p_s(\mathbf{x}_s)}$ 可以看作是一个单样本蒙特卡罗估计量，在给定 $\mathbf{x}_s$ 时具有条件无偏性：对该估计量在（通常难以处理的）干净后验 $p(\cdot|\mathbf{x}_s)$ 上进行平均，即可恢复真实得分，即

$$\nabla_{\mathbf{x}_s}\log p_s(\mathbf{x}_s) = \mathbb{E}_{\mathbf{x}_0\sim p(\cdot|\mathbf{x}_s)}\left[\widehat{\nabla_{\mathbf{x}}\log p_s(\mathbf{x}_s)}\right].$$

Theorem 11.2.1: CM-CT 误差等价性 $\mathcal{O}(\Delta s^2)$

令 $s' := s - \Delta s$ ，并定义

$$\mathcal{L}_{\mathrm{CM}}(\boldsymbol{\theta}, \boldsymbol{\theta}^-) := \mathbb{E}_{s,\mathbf{x}_0,\boldsymbol{\epsilon}}\big[w(s)\, d\big(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s, s), \mathbf{f}_{\boldsymbol{\theta}^-}(\mathbf{x}_{s'}^{\mathrm{DDIM}}, s')\big)\big],$$
$$\mathcal{L}_{\mathrm{CT}}(\boldsymbol{\theta}, \boldsymbol{\theta}^-) := \mathbb{E}_{s,\mathbf{x}_0,\boldsymbol{\epsilon}}\big[w(s) d\big(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s, s), \mathbf{f}_{\boldsymbol{\theta}^-}(\mathbf{x}_{s'}, s')\big)\big],$$

其中

$$\mathbf{x}_{s'}^{\mathrm{DDIM}} := \frac{\alpha_{s'}}{\alpha_s}\mathbf{x}_s + \sigma_s^2\left(\frac{\alpha_{s'}}{\alpha_s} - \frac{\sigma_{s'}}{\sigma_s}\right)\nabla_{\mathbf{x}_s}\log p_s(\mathbf{x}_s)$$

为理想 DDIM 更新。$\mathbf{x}_s = \alpha_s\mathbf{x}_0 + \sigma_s\boldsymbol{\epsilon}$ 与 $\mathbf{x}_{s'} = \alpha_{s'}\mathbf{x}_0 + \sigma_{s'}\boldsymbol{\epsilon}$ 共享相同的参数对 $(\mathbf{x}_0, \boldsymbol{\epsilon})$ ，其中 $\mathbf{x}_0 \sim p_{\mathrm{data}}$ 且 $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 。则有

$$\mathcal{L}_{\mathrm{CM}}(\boldsymbol{\theta}, \boldsymbol{\theta}^-) = \mathcal{L}_{\mathrm{CT}}(\boldsymbol{\theta}, \boldsymbol{\theta}^-) + \mathcal{O}(\Delta s^2).$$

Proof for Theorem.

首先注意到，采用理想得分的 DDIM 更新等于条件期望：

$$\mathbf{x}_{s'}^{\mathrm{DDIM}} = \mathbb{E}[\mathbf{x}_{s'}|\mathbf{x}_s],$$

该结论也可通过对 $p(\cdot|\mathbf{x}_s)$ 取期望从 Equation (11.2.5) 得到验证。接下来对

$$d(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s, s), \mathbf{f}_{\boldsymbol{\theta}^-}(\cdot, s'))$$

在 $\mathbf{x}_{s'}^{\mathrm{DDIM}} = \mathbb{E}[\mathbf{x}_{s'}|\mathbf{x}_s]$ 处进行泰勒展开。由于内层期望取自满足 $\mathbb{E}[\mathbf{x}_{s'} - \mathbf{x}_{s'}^{\mathrm{DDIM}}|\mathbf{x}_s] = 0$ 的 $\mathbf{x}_{s'}|\mathbf{x}_s$ ，泰勒展开的线性项将消失。这表明通过将条件分布重参数化为含 $\mathbf{x}_{s'} = \alpha_{s'}\mathbf{x}_0 + \sigma_{s'}\boldsymbol{\epsilon}$ 的 $\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}|\mathbf{x}_s}[\cdot]$ ，采用单点得分的 DDIM 更新能够对相同 $(\mathbf{x}_0, \boldsymbol{\epsilon})$ 实现精确的路径级 $\mathbf{x}_{s'}$ 复原，从而保持内层期望不变。剩余项为二次项 $\mathcal{O}(\Delta s^2)$ ，故得 $\mathcal{L}_{\mathrm{CT}} = \mathcal{L}_{\mathrm{CM}} + \mathcal{O}(\Delta s^2)$ 。详细推导见 Section D.5。

总之，CD 利用教师模型进行初始化和引导，这通常能稳定最优化过程并降低方差。相比之下，一致性训练（CT）无需预训练模型，因此可完全从零开始训练。尽管存在这一差异，CT 仍可作为一个完全独立的生成式模型使用。

实际考虑因素。 在实际应用中，**song2023consistency** 采用 **karras2022elucidating**(see Section D.6) 的 EDM 公式化方法，结合前向扰动核

$$\mathbf{x}_s = \mathbf{x}_0 + s\boldsymbol{\epsilon},$$

并使用其中提出的神经网络参数化方法 (cf. Equation (D.6.1)):

$$\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, s) = c_{\text{skip}}(s)\mathbf{x} + c_{\text{out}}(s)\,\mathbf{F}_{\boldsymbol{\theta}}\left(c_{\text{in}}(s)\,\mathbf{x},\, c_{\text{noise}}(s)\right),$$

其中 $\mathbf{F}_{\boldsymbol{\theta}}$ 是一个神经网络，且系数满足 Equation (D.6.5)。这种参数化具有重要的边界性质

$$\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, 0) = \mathbf{x},$$

这在时间零强制一致性，并确保当不存在噪声时，网络输出与输入相匹配。

11.2.2 一致性模型采样

一旦一致性模型 $\mathbf{f}_{\boldsymbol{\theta}^\times}$ 训练完成，无论是在连续时间还是离散时间下，都可以在单步或几步内生成样本。该算法总结于 Algorithm 9。

一步生成。 给定从先验分布中采样的初始潜在变量 $\hat{\mathbf{x}}_T$ （实际中为 $\mathcal{N}(\mathbf{0}, T^2\mathbf{I})$），可以通过一次函数求值生成干净样本：

$$\mathbf{f}_{\boldsymbol{\theta}^\times}(\hat{\mathbf{x}}_T, T).$$

多步生成。 采用预先选定的时间步长

$$T > \tau_1 > \tau_2 > \cdots > \tau_{M-1} = 0,$$

从初始噪声 $\hat{\mathbf{x}}_T$ 开始，在较早的时间点通过一致性模型交替进行噪声注入和大幅干净跳跃，逐步优化样本：

$$\hat{\mathbf{x}}_T \xrightarrow[\text{get a clean}]{\text{long jump}} \mathbf{f}_{\boldsymbol{\theta}^\times}(\hat{\mathbf{x}}_T, T) \xrightarrow[\text{to level } \tau_1]{\text{add noise}} \hat{\mathbf{x}}_{\tau_1} \xrightarrow[\text{get a clean}]{\text{long jump}} \mathbf{f}_{\boldsymbol{\theta}^\times}(\hat{\mathbf{x}}_{\tau_1}, \tau_1) \xrightarrow[\text{to level } \tau_2]{\text{add noise}} \cdots.$$

Algorithm 9 CM's Sampling with One-Step or Multi-Step Generation

Input: Consistency model $\mathbf{f}_{\boldsymbol{\theta}\times}(\cdot,\cdot)$, sequence of time points $T > \tau_1 > \tau_2 > \cdots > \tau_{M-1} = 0$, initial noise $\hat{\mathbf{x}}_T$

if one-step **then**
  $\mathbf{x} \leftarrow \mathbf{f}_{\boldsymbol{\theta}\times}(\hat{\mathbf{x}}_T, T)$
else
  $\mathbf{x} \leftarrow \mathbf{f}_{\boldsymbol{\theta}\times}(\hat{\mathbf{x}}_T, T)$
  for $m = 1$ **to** $M - 1$ **do**
    Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\hat{\mathbf{x}}_{\tau_m} \leftarrow \alpha_{\tau_m}\mathbf{x} + \sigma_{\tau_m}\boldsymbol{\epsilon}$
    $\mathbf{x} \leftarrow \mathbf{f}_{\boldsymbol{\theta}\times}(\hat{\mathbf{x}}_{\tau_m}, \tau_m)$
  end for
end if

Output: $\mathbf{x}$

11.3 特殊流图：连续时间中的一致性模型

我们现在超越一致性模型的离散时间情景，转而考虑连续时间视角。与固定时间网格并仅在这些采样点上训练的离散方法不同，连续形式将流映射视为对所有时间都定义的。这种转变消除了对任意离散化的依赖，更合理地契合了潜在动力学。它还有助于减少由离散化积分自然产生的近似误差，并确保一致性在整个过程中被全局强制执行，而不仅仅是在选定的步骤上。

11.3.1 连续时间一致性模型

为了激发连续时间表述的动机，我们首先重新考察 Equation (11.2.3)，它描述了可以取时间导数的条件。利用链式法则，我们得到

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}^*(\mathbf{x}(s),s)=0 \\ \iff &\left(\nabla_{\mathbf{x}}\mathbf{f}^*\right)(\mathbf{x}(s),s)\cdot \underbrace{\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{x}(s)}_{\text{ODE velocity}} + \left(\frac{\partial}{\partial s}\mathbf{f}^*\right)(\mathbf{x}(s),s)=0, \end{aligned} \tag{11.3.1}$$

其中轨迹 $\mathbf{x}(s)$ 遵循 PF-ODE

$$\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{x}(s)=\mathbf{v}^*(\mathbf{x}(s),s).$$

该关系表明，一致性函数 $\mathbf{f}^*$ 沿常微分方程的任意解轨迹保持不变。在实际应用中，速度场 $\mathbf{v}^*$ 可以通过预训练的扩散模型（当此类模型可用时）或直接采用单点近似方法（如 $\alpha_s'\mathbf{x}_0+\sigma_s'\boldsymbol{\epsilon}$ 所示）进行估计，具体说明见 Section 11.2。

Equation (11.3.1) 提出了一种在连续时间中设计训练目标的自然方法。一种方法是通过最小化残差来施加该条件，其方式类似于物理信息神经网络 (PINNs) (**raissi2018deep**; **boffi2024flow**)：

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{s,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s,s)\right\|_2^2\right].$$

然而，在实际应用中，**song2023consistency**; **lu2024simplifying** 观察到另一种公式在训练时表现更好。他们不直接强制微分条件，而是考虑离散近似的

连续时间极限，如 $\Delta s \to 0$ 所示：

$$\mathcal{L}_{\mathrm{CM}}^{\Delta s}(\boldsymbol{\theta}, \boldsymbol{\theta}^{-}) := \mathbb{E}\left[\omega(s)\left\|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s, s) - \mathbf{f}_{\boldsymbol{\theta}^{-}}\left(\boldsymbol{\Psi}_{s \to s-\Delta s}(\mathbf{x}_s), s - \Delta s\right)\right\|_2^2\right]. \tag{11.3.2}$$

令 $\Delta s \to 0$ 的极限 Equation (11.3.2) 等价于让时间步数 $N \to \infty$ in Equations (11.2.4) and (11.2.6) 趋于无穷。

我们将这一关键思想总结如下命题。

Proposition 11.3.1: 连续时间一致性训练

以下收敛结果成立：

$$\lim_{\Delta s \to 0} \frac{1}{\Delta s} \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{CM}}^{\Delta s}(\boldsymbol{\theta}, \boldsymbol{\theta}^{-}) = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{CM}}^{\infty}(\boldsymbol{\theta}, \boldsymbol{\theta}^{-}).$$

其中，

$$\mathcal{L}_{\mathrm{CM}}^{\infty}(\boldsymbol{\theta}, \boldsymbol{\theta}^{-}) := \mathbb{E}_{s, \mathbf{x}_0, \boldsymbol{\epsilon}}\left[2\omega(t)\mathbf{f}_{\boldsymbol{\theta}}^{\top}(\mathbf{x}_s, s) \cdot \frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s, s)\right],$$

以及全微分恒等式，

$$\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s, s) = \partial_s \mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s, s) + \left(\partial_{\mathbf{x}} \mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s, s)\right)\mathbf{v}^{*}(\mathbf{x}_s, s). \tag{11.3.3}$$

Proof for Proposition.

对停止梯度目标在 $(\mathbf{x}_s, s)$ 处进行一阶泰勒展开表明，损失函数 $\mathcal{L}_{\mathrm{CM}}^{\Delta s}$ 在 $\mathcal{O}(\Delta s^2)$ 精度内，其行为类似于学生更新 $\nabla_{\boldsymbol{\theta}} \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s, s)$ 与切线变化 $\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s, s)$ 的内积。因此，缩放后的梯度满足

$$\lim_{\Delta s \to 0} \frac{1}{\Delta s} \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{CM}}^{\Delta s} = \nabla_{\boldsymbol{\theta}} \mathbb{E}\left[\tilde{\omega}(s)\mathbf{f}_{\boldsymbol{\theta}}^{\top}(\mathbf{x}_s, s) \cdot \frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s, s)\right],$$

即为所证恒等式。证明过程详见 Section D.5。 ∎

上述结果写在梯度算子 $\nabla_{\boldsymbol{\theta}}$ 之下，使得涉及 $\boldsymbol{\theta}^{-}$ 的项消失，因为 $\boldsymbol{\theta}^{-}$ 在停止梯度下被视为常数。请注意，$\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s, s)$ 表示沿预言机轨迹的全导数，而非简单的偏时间导数。

总之，连续时间一致性模型可以通过最小化以下目标函数进行训练（忽略因子 2 ）：

$$\mathcal{L}_{\mathrm{CM}}^{\infty}(\boldsymbol{\theta},\boldsymbol{\theta}^{-}) := \mathbb{E}_{s,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\omega(s)\mathbf{f}_{\boldsymbol{\theta}}^{\top}(\mathbf{x}_s,s)\cdot\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)\right]. \tag{11.3.4}$$

11.3.2 训练连续时间一致性模型

类似于 Section 11.2.1 中讨论的离散时间情况，我们现在澄清 Equation (11.3.4) 中切线项的实际近似，该近似涉及不可访问的预言机速度 $\mathbf{v}^{*}$：

$$\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s) = \partial_s\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s) + \left(\partial_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)\right)\mathbf{v}^{*}(\mathbf{x}_s,s).$$

训练完连续时间 CM 后，采样遵循与离散时间情况相同的步骤（Section 11.2.2）。

连续时间一致性蒸馏。 如果存在一个预训练的扩散模型，使得 $\mathbf{v}_{\boldsymbol{\phi}^{\times}} \approx \mathbf{v}^{*}$，则在 Equation (11.3.3) 中的切向量 $\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)$ 可以通过代理模型近似表示为

$$\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s) \approx \partial_s\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s) + \left(\partial_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)\right)\mathbf{v}_{\boldsymbol{\phi}^{\times}}(\mathbf{x}_s,s). \tag{11.3.5}$$

我们将得到的目标记为 $\mathcal{L}_{\mathrm{CM}}^{\infty}(\boldsymbol{\theta},\boldsymbol{\theta}^{-};\boldsymbol{\phi}^{\times})$。相应地，命题 11.3.1 可以重述为

$$\lim_{N\to\infty} N\,\nabla_{\boldsymbol{\theta}}\,\mathcal{L}_{\mathrm{CD}}^{N}(\boldsymbol{\theta},\boldsymbol{\theta}^{-};\boldsymbol{\phi}^{\times}) = \nabla_{\boldsymbol{\theta}}\,\mathcal{L}_{\mathrm{CD}}^{\infty}(\boldsymbol{\theta},\boldsymbol{\theta}^{-};\boldsymbol{\phi}^{\times}).$$

连续时间一致性训练（从零开始）。 另一方面，如果无法获得预训练的扩散模型，则可以使用单点条件估计 $\alpha'_s\mathbf{x}_0 + \sigma'_s\boldsymbol{\epsilon}$ 来近似原始速度 $\mathbf{v}^{*}$。在这种情况下，Equation (11.3.3) 中的切向量 $\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)$ 被替代为代理向量

$$\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s) \approx \partial_s\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s) + \left(\partial_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)\right)\left(\alpha'_s\mathbf{x}_0 + \sigma'_s\boldsymbol{\epsilon}\right). \tag{11.3.6}$$

我们将得到的目标记为 $\mathcal{L}_{\mathrm{CT}}^{\infty}(\boldsymbol{\theta},\boldsymbol{\theta}^{-})$，这对应于从零开始训练的情景。相应地，命题 11.3.1 可以重新表述为

$$\lim_{N\to\infty} N\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{CT}}^{N}(\boldsymbol{\theta},\boldsymbol{\theta}^{-}) = \nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{CT}}^{\infty}(\boldsymbol{\theta},\boldsymbol{\theta}^{-}).$$

到目前为止，我们已经介绍了列表 Table 11.1 中列出的所有基本方法，以实现一致性函数 $\boldsymbol{\Psi}_{s\to 0}$ 的学习。为了提供更清晰的概览，Figure 11.3 总结了用

于训练一致性函数的不同损失函数之间的关系。该图还表明了每种方法是否依赖于预训练的扩散模型，并区分了连续时间和离散时间的目标。

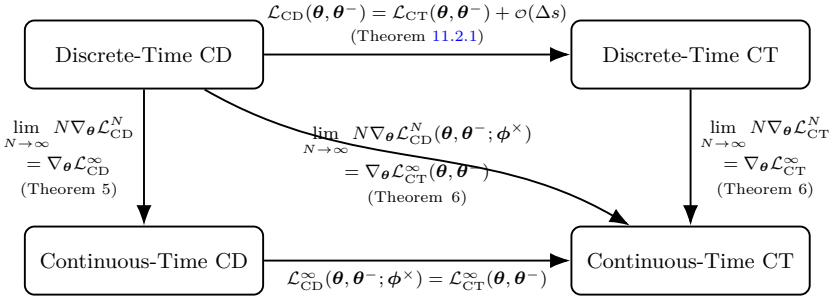


图 11.3: 图示在 ℓ_2 距离度量下，离散/连续时间 CD 与 CT 之间的关系：$d(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_2^2$。标记的定理遵循 (**song2023consistency**) 中的标签。每当定理涉及 CT 时，我们假设得分为完美：$\mathbf{s}_{\boldsymbol{\phi}^\times}(\mathbf{x},t) \equiv \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$。$\mathcal{L}^\infty_{\mathrm{CT}}$ 定义于 Equation (11.3.4)，而 $\mathcal{L}^\infty_{\mathrm{CD}}$ 定义于 Equation (11.3.5)。

然而，切向量 $\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^-}$ 通常在训练过程中引起不稳定性。在接下来的可选章节中，我们介绍了来自 *Simplifying, Stabilizing and Scaling Continuous Time Consistency Models* (sCM) (**lu2024simplifying**) 的技术，以缓解这些问题。

11.3.3 （可选）连续时间一致性训练的实际考虑

我们的研究兴趣在于从零开始的训练场景，因为这能够产生一个不依赖外部预训练扩散模型的独立生成式模型。因此，我们重点讨论连续时间情况。

然而，在实际应用中，直接使用 Equation (11.3.4) 进行训练通常不稳定，因为项 $\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^-}$ 可能表现出较大的或无界的时变导数，导致最优化过程中出现梯度爆炸。为克服这一问题，通常需要采用合适的参数化方法和稳定化策略 (**geng2025consistency**; **lu2024simplifying**)。如 Section 6.2.2 所总结，影响训练稳定性的主要因子包括 *扩散过程*、*参数化选择*、*时间加权函数* 和 *时间采样分布*，这些因素在连续时间的 CM 中也应被仔细设计并解耦。

扩散过程 与其使用标准的扩散参数化 $\mathbf{x}_s = \alpha_s \mathbf{x}_0 + \sigma_s \boldsymbol{\epsilon}$ 与 $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$，**lu2024simplifying** 采用三角函数调度。该调度虽然在数学上等价于原始形式 (如 Equation (6.3.4) 所示)，但提供了更清晰的结构以及训练目标中更好的分离性，有助于提升训练过程中的稳定性。[5]. In addition, they incorporate the standard deviation σ_{d} of the data distribution p_{data}, in line with EDM' s design in

[5]直观上，三角函数及其导数都是有界的，这有助于防止形如 $\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^-}$ 的项出现尺度爆炸。后续将进行详细讨论。

Section D.6.1:

$$\mathbf{x}_s := \cos(s)\mathbf{x}_0 + \sin(s)\mathbf{z}, \quad \text{where } \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma_{\mathrm{d}}^2\mathbf{I}). \tag{11.3.7}$$

该公式形式是完全通用的。对于任意形如 $\mathbf{x}_s = \alpha_s\mathbf{x}_0 + \sigma_s\boldsymbol{\epsilon}$ 且满足 $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 的扩散过程，我们都可以等价地表示为：

$$\mathbf{x}_s = \alpha_s\mathbf{x}_0 + \frac{\sigma_s}{\sigma_{\mathrm{d}}} \cdot (\sigma_{\mathrm{d}}\boldsymbol{\epsilon}),$$

通过定义 $\mathbf{z} := \sigma_{\mathrm{d}}\boldsymbol{\epsilon}$、$\alpha'_s := \alpha_s$ 和 $\sigma'_s := \frac{\sigma_s}{\sigma_{\mathrm{d}}}$。变换后的变量对 (α'_s, σ'_s) 可以使用 Equation (6.3.5) 中描述的规范化方法映射到三角形式 $(\cos(s), \sin(s))$。

参数化。 通过考虑 Section D.6.1 中 EDM 的类似原理，**lu2024simplifying** 提出了如下与 Equation (D.6.1) 相似的神经网络参数化方法：

$$\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, s) := c_{\mathrm{skip}}(s)\mathbf{x} + c_{\mathrm{out}}(s)\mathbf{F}_{\boldsymbol{\theta}}\left(c_{\mathrm{in}}(s)\mathbf{x}, c_{\mathrm{noise}}(s)\right).$$

此处，$c_{\mathrm{skip}}(s)$、$c_{\mathrm{out}}(s)$ 和 $c_{\mathrm{in}}(s)$ 可以使用 Section D.6.1(see Appendix B of **lu2024simplifying** for detailed derivations) 中提出的相同准则推导得出，其表达式为

$$c_{\mathrm{skip}}(s) = \cos(s), \quad c_{\mathrm{out}}(s) = -\sigma_{\mathrm{d}}\sin(s), \quad c_{\mathrm{in}}(s) \equiv \frac{1}{\sigma_{\mathrm{d}}}.$$

这与默认选择 $c_{\mathrm{noise}}(s) = s$ 一起考虑，其中 $\partial_s c_{\mathrm{noise}}(s)$ 被限制以确保训练稳定性，这一点将在 Equation (11.3.10) 附近讨论。这导致在三角函数调度下的以下参数化：

$$\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, s) = \cos(s)\mathbf{x} - \sin(s)\sigma_{\mathrm{d}}\mathbf{F}_{\boldsymbol{\theta}}\left(\frac{\mathbf{x}}{\sigma_{\mathrm{d}}}, c_{\mathrm{noise}}(s)\right). \tag{11.3.8}$$

我们注意到，这种参数化方法还确保了神经网络对所有 $\mathbf{x}$ 始终满足边界条件 $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, 0) \equiv \mathbf{x}$ ，这是相容性函数的一个重要性质。

稳定切线训练的技术。 在三角函数调度和 Equation (11.3.8) 中描述的网络参数化下，Equation (11.3.4) 中的损失梯度变为

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{CT}}^{\infty}(\boldsymbol{\theta},\boldsymbol{\theta}^{-})=\nabla_{\boldsymbol{\theta}}\mathbb{E}_{s,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[-\omega(s)\sigma_{\mathrm{d}}\sin(s)\mathbf{F}_{\boldsymbol{\theta}}^{\top}\left(\frac{\mathbf{x}_s}{\sigma_{\mathrm{d}}},s\right)\cdot\frac{\mathrm{d}\mathbf{f}_{\boldsymbol{\theta}^{-}}}{\mathrm{d}s}(\mathbf{x}_s,s)\right]. \tag{11.3.9}$$

理论上，使用公式 Equation (11.3.9) 中的梯度更新进行训练可能足以学习到一个一致性函数。然而，**lu2024simplifying**在实践中经验性地观察到，由于正切函数的行为，训练过程可能会变得不稳定，具体表现为

$$\underbrace{\frac{\mathrm{d}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)}{\mathrm{d}s}}_{\textbf{A.}}=$$

$$-\cos(s)\left(\sigma_{\mathrm{d}}\nabla_{\mathbf{x}_s}\mathbf{F}_{\boldsymbol{\theta}^{-}}\left(\frac{\mathbf{x}_s}{\sigma_{\mathrm{d}}},s\right)-\frac{\mathrm{d}\mathbf{x}_s}{\mathrm{d}s}\right)-\sin(s)\left(\mathbf{x}_s+\sigma_{\mathrm{d}}\frac{\mathrm{d}\mathbf{F}_{\boldsymbol{\theta}^{-}}}{\mathrm{d}s}\left(\frac{\mathbf{x}_s}{\sigma_{\mathrm{d}}},c_{\mathrm{noise}}(s)\right)\right).$$

特别是，在该项中观察到不稳定性。

$$\underbrace{\sin(s)\frac{\mathrm{d}\mathbf{F}_{\boldsymbol{\theta}^{-}}}{\mathrm{d}s}\left(\frac{\mathbf{x}_s}{\sigma_{\mathrm{d}}},c_{\mathrm{noise}}(s)\right)}_{\textbf{B.}}=\sin(s)\nabla_{\mathbf{x}_s}\mathbf{F}_{\boldsymbol{\theta}^{-}}\frac{\mathrm{d}\mathbf{x}_s}{\mathrm{d}s}+\sin(s)\partial_s\mathbf{F}_{\boldsymbol{\theta}^{-}}.$$

更具体地说，不稳定性源于该组件

$$\sin(s)\partial_s\mathbf{F}_{\boldsymbol{\theta}^{-}}=\sin(s)\underbrace{\frac{\partial c_{\mathrm{noise}}(s)}{\partial s}\cdot\frac{\partial\mathrm{emb}(c_{\mathrm{noise}})}{\partial c_{\mathrm{noise}}}}_{\textbf{C.}}\cdot\frac{\partial\mathbf{F}_{\boldsymbol{\theta}^{-}}}{\partial\mathrm{emb}(c_{\mathrm{noise}})}. \tag{11.3.10}$$

在此，我们遵循扩散模型（DM）和条件模型（CM）文献中的常见做法，对时间变量 $c_{\mathrm{noise}}(s)$ 应用位置嵌入或傅里叶嵌入，记为 $\mathrm{emb}(\cdot)$：

$$s\mapsto c_{\mathrm{noise}}(s)\mapsto\mathrm{emb}(c_{\mathrm{noise}}(s))\mapsto\mathbf{F}_{\boldsymbol{\theta}^{-}}\left(\frac{\mathbf{x}_s}{\sigma_{\mathrm{d}}},\mathrm{emb}(c_{\mathrm{noise}}(s))\right).$$

因此，引入了一些额外的实证技术来缓解不稳定性：

- **A. 切线规范化**。通过将 $\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\theta^{-}}$ 替换为 $\frac{\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\theta^{-}}}{\left\|\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\theta^{-}}\right\|_2+c}$ 来显式地对切线函数进行规范化，其中 $c>0$ 是经验设定的常数。或者，将切线值限制在 $[-1,1]$ 范围内也可以有效控制其方差。

- **B. 切线预热**。由于项 $\sin(s)(\mathbf{x}_s+\sigma_{\mathrm{d}}\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{F}_{\boldsymbol{\theta}^{-}})$ 可能引起不稳定性，可以采用可选技术，将系数 $\sin(s)$ 替换为 $r\cdot\sin(s)$，其中 r 在前几次训练迭代中从

0 线性增加到 1。

- **C. 时间嵌入**。鉴于公式 Equation (11.3.10) 中的导数链，**lu2024simplifying** 选择了一个较小的模参数来控制导数 $\frac{\partial \text{emb}(c_{\text{noise}})}{\partial c_{\text{noise}}}$ 。出于类似的原因，选择了 $c_{\text{noise}}(s) = s$ ，其中 $\partial_s c_{\text{noise}}(s) = 1$ —一个有界常数。

在此基础上，为了改善规范化（稳定性）和高效地基于 JVP 计算 $\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\theta^-}$，通常还需要架构上的改进，但这超出了我们的讨论范围。

时间权重函数 时间权重函数 $\omega(s)$ 的手动设计可能导致性能不佳。为解决此问题，借鉴 EDM-2 (**karras2024analyzing**) 的类似方法，**lu2024simplifying** 学习一个自适应权重函数 $\omega_{\boldsymbol{\varphi}}(s)$ ，以平衡不同时间点的训练损失方差 s （参见 Equation (11.3.11) 以了解期望结果）。

进一步地，我们观察到 Equation (11.3.9) 中的目标函数具有以下形式

$$\mathbb{E}_{s,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\mathbf{F}_{\boldsymbol{\theta}}^{\top}\mathbf{y}\right], \quad \text{with} \quad \mathbf{y} = -\omega(s)\sigma_{\mathrm{d}}\sin(s)\frac{\mathrm{d}\mathbf{f}_{\boldsymbol{\theta}^-}}{\mathrm{d}s}.$$

由于 $\mathbf{y}$ 是一个与 $\boldsymbol{\theta}$ 无关的向量，因此 Equation (11.3.9) 等价于

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{s,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\mathbf{F}_{\boldsymbol{\theta}}^{\top}\mathbf{y}\right] = \frac{1}{2}\nabla_{\boldsymbol{\theta}}\mathbb{E}_{s,\mathbf{x}_0,\boldsymbol{\epsilon}}\left\|\mathbf{F}_{\boldsymbol{\theta}} - \mathbf{F}_{\boldsymbol{\theta}^-} + \mathbf{y}\right\|_2^2.$$

基于此观察，**lu2024simplifying** 进一步提出训练一个自适应权重网络 $\omega_{\boldsymbol{\varphi}}(s)$ 来估计损失范数，公式化为以下最小化问题：

$$\min_{\boldsymbol{\varphi}}\mathbb{E}_{s,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{e^{\omega_{\boldsymbol{\varphi}}(s)}}{D}\left\|\mathbf{F}_{\boldsymbol{\theta}} - \mathbf{F}_{\boldsymbol{\theta}^-} + \mathbf{y}\right\|_2^2 - \omega_{\boldsymbol{\varphi}}(s)\right].$$

为了理解自适应权重的影响，注意到最优解 $\omega^*(s)$ （通过对上述目标关于 $\omega_{\boldsymbol{\varphi}}$ 取偏导数得到）满足

$$\mathbb{E}_{s,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{e^{\omega^*(s)}}{D}\left\|\mathbf{F}_{\boldsymbol{\theta}} - \mathbf{F}_{\boldsymbol{\theta}^-} + \mathbf{y}\right\|_2^2\right] = 1. \tag{11.3.11}$$

也就是说，在再缩放之后，不同 s 之间的期望（权重）损失保持一致。因此，自适应权重有效地降低了训练损失在不同时间步之间的方差，从而实现了更加均衡和稳定的训练。

Algorithm 10 Training of Continuous-time Consistency Models (sCM)

Input: dataset $\mathcal{D}$ with std. σ_{d}, pre-trained DM $\mathbf{F}_{\text{pretrain}}$ with parameter $\boldsymbol{\theta}_{\text{pretrain}}$, model $\mathbf{F}_{\boldsymbol{\theta}}$, weighting $\omega_{\boldsymbol{\varphi}}$, learning rate η, proposal $(P_{\text{mean}}, P_{\text{std}})$, constant c, warmup iteration H

Init: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_{\text{pretrain}}$, Iters $\leftarrow 0$

Repeat

  $\mathbf{x}_0 \sim \mathcal{D}$, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma_{\mathrm{d}}^2\mathbf{I})$, $\tau \sim \mathcal{N}(P_{\text{mean}}, P_{\text{std}}^2)$, $s \leftarrow \arctan\left(\frac{e^{\tau}}{\sigma_{\mathrm{d}}}\right)$

  $\mathbf{x}_s \leftarrow \cos(s)\mathbf{x}_0 + \sin(s)\mathbf{z}$

  if consistency training **then**

    $\frac{\mathrm{d}\mathbf{x}_s}{\mathrm{d}s} \leftarrow \cos(s)\mathbf{z} - \sin(s)\mathbf{x}_0$

  else

    $\frac{\mathrm{d}\mathbf{x}_s}{\mathrm{d}s} \leftarrow \sigma_{\mathrm{d}}\mathbf{F}_{\text{pretrain}}\left(\frac{\mathbf{x}_s}{\sigma_{\mathrm{d}}}, s\right)$

  end if

  $r \leftarrow \min\left(1, \frac{\text{Iters}}{H}\right)$ ▷ Tangent warmup

  $\mathbf{w} \leftarrow -\cos^2(s)(\sigma_{\mathrm{d}}\mathbf{F}_{\boldsymbol{\theta}}^{-} - \frac{\mathrm{d}\mathbf{x}_s}{\mathrm{d}s}) - r\cos(s)\sin(s)\left(\mathbf{x}_s + \sigma_{\mathrm{d}}\frac{\mathrm{d}\mathbf{F}_{\boldsymbol{\theta}^-}}{\mathrm{d}s}\right)$

  $\mathbf{w} \leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|+c}$ ▷ Tangent normalization

  $\mathcal{L}_{\text{sCM}}(\boldsymbol{\theta}, \boldsymbol{\varphi}) \leftarrow \frac{e^{\omega_{\boldsymbol{\varphi}}(s)}}{D}\left\|\mathbf{F}_{\boldsymbol{\theta}}\left(\frac{\mathbf{x}_s}{\sigma_{\mathrm{d}}}, s\right) - \mathbf{F}_{\boldsymbol{\theta}^-}\left(\frac{\mathbf{x}_s}{\sigma_{\mathrm{d}}}, s\right) - \mathbf{w}\right\|_2^2 - \omega_{\boldsymbol{\varphi}}(s)$ ▷ Adaptive weighting

  $(\boldsymbol{\theta}, \boldsymbol{\varphi}) \leftarrow (\boldsymbol{\theta}, \boldsymbol{\varphi}) - \eta\nabla_{\boldsymbol{\theta},\boldsymbol{\varphi}}\mathcal{L}_{\text{sCM}}(\boldsymbol{\theta}, \boldsymbol{\varphi})$

  Iters $\leftarrow$ Iters $+ 1$

until convergence

时间采样分布 **lu2024simplifying** 从对数正态提议分布 (**karras2022elucidating**) 中采样 $\tan(s)$，即

$$e^{\sigma_{\mathrm{d}}\tan(s)} \sim \mathcal{N}(\cdot; P_{\text{mean}}, P_{\text{std}}^2). \tag{11.3.12}$$

其中，P_{mean} 和 P_{std} 是两个超参数。

训练目标概述 综上所述，上述讨论的最终训练损失表示为：

$$\mathcal{L}_{\text{sCM}}(\boldsymbol{\theta}, \boldsymbol{\varphi}) :=$$
$$\mathbb{E}_{s,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{e^{\omega_{\boldsymbol{\varphi}}(s)}}{D}\left\|\mathbf{F}_{\boldsymbol{\theta}}\left(\frac{\mathbf{x}_s}{\sigma_{\mathrm{d}}}, s\right) - \mathbf{F}_{\boldsymbol{\theta}^-}\left(\frac{\mathbf{x}_s}{\sigma_{\mathrm{d}}}, s\right) - \cos(s)\frac{\mathrm{d}\mathbf{f}_{\boldsymbol{\theta}^-}}{\mathrm{d}s}(\mathbf{x}_s, s)\right\|_2^2 - \omega_{\boldsymbol{\varphi}}(s)\right].$$

此处，s 根据 Equation (11.3.12) 采样，$\mathbf{x}_s$ 通过 Equation (11.3.7) 计算得出。使用该损失训练的模型称为 *sCM*，其训练过程总结于 Algorithm 10。

11.4 General Flow Map: Consistency Trajectory Model

一致性轨迹模型（CTM）(**kim2023consistency**) 是最早学习 通用流动映射 $\boldsymbol{\Psi}_{s\to t}$ 的方法之一。

实际应用中的 CTM 设置。 与 CM 族类似，CTM 最初遵循 EDM (**karras2022elucidating**)(
的公式，采用 $\mathbf{x}$ -预测形式的 PF-ODE，结合噪声调度 $\alpha_t = 1$ 和 $\sigma_t = t$。在此设置下，PF-ODE 变为

$$\frac{\mathrm{d}\mathbf{x}(\tau)}{\mathrm{d}\tau} = \frac{\mathbf{x}(\tau) - \mathbb{E}[\mathbf{x}|\mathbf{x}(\tau)]}{\tau}.$$

从时间 s 的 $\mathbf{x}_s$ 出发并演化到较晚的时间 $t \le s$ ，相应的流映射（解）可等价地表示为

$$\mathbf{x}_s + \int_s^t \frac{\mathbf{x}_\tau - \mathbb{E}[\mathbf{x}|\mathbf{x}_\tau]}{\tau}\,\mathrm{d}\tau.$$

CTM 采用了一种受欧拉法启发的参数化方法：对 PF-ODE 应用单步欧拉求解器（等价于 DDIM；见 Equation (9.2.4)），可得

$$\mathbf{x}_{s\to t}^{\text{Euler}} = \mathbf{x}_s - (s-t)\,\frac{\mathbf{x}_s - \mathbb{E}[\mathbf{x}|\mathbf{x}_s]}{s} = \frac{t}{s}\,\mathbf{x}_s + \Big(1 - \frac{t}{s}\Big)\mathbb{E}[\mathbf{x}|\mathbf{x}_s],$$

其中 $\mathbf{x}_{s\to t}^{\text{Euler}}$ 近似于在时间 t 时给定时间 s 状态 $\mathbf{x}_s$ 的解。

虽然 EDM 设置提供了一个简单的示例，CTM 允许定义更广泛的噪声调度，由任意线性高斯前向核 (α_t, σ_t) 描述，并将 PF-ODE 表示为 $\mathbf{v}$-预测形式：

$$\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s) = \mathbf{x}_s + \int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u.$$

在接下来的讨论中，我们将重点放在这一通用表述上。

11.4.1 CTM 参数化用于灵活的转移学习

在上述 PF-ODE 的单步欧拉求解器之后，CTM 将预言流映射 $\boldsymbol{\Psi}_{s\to t}$ 重写为输入 $\mathbf{x}_s$ 与残差函数 $\mathbf{g}^*$ 的凸组合：

$$\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s) := \mathbf{x}_s + \int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u = \frac{t}{s}\,\mathbf{x}_s + \frac{s-t}{s}\underbrace{\left[\mathbf{x}_s + \frac{s}{s-t}\int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u\right]}_{=:\,\mathbf{g}^*}.$$

其中残差项 $\mathbf{g}^*$ 定义为

$$\mathbf{g}^*(\mathbf{x}_s, s, t) := \mathbf{x}_s + \frac{s}{s-t}\int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u. \tag{11.4.1}$$

这促使了神经参数化的发展

$$\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) := \frac{t}{s}\,\mathbf{x}_s + \frac{s-t}{s}\,\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t), \tag{11.4.2}$$

其中 $\mathbf{g}_{\boldsymbol{\theta}}$ 是一个旨在 $\mathbf{g}_{\boldsymbol{\theta}} \approx \mathbf{g}^*$ 的神经网络，因此 $\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t)$ 被训练以逼近最优流映射，

$$\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) \approx \boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s).$$

因此，CTM 自然地融入了 Equation (10.1.4) 的通用一致性映射框架中，该框架将学成的映射与真值流映射对齐。

此外，此公式自然满足初始条件

$$\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, s) = \mathbf{x}_s,$$

在训练过程中无需任何显式的强制执行。

CTM 参数化方法的优势。 当取极限使 t 趋近于 s（即起始时间与结束时间相同）时，$\mathbf{g}^*$ 的一个关键特性变得明显：

Proposition 11.4.1: $\mathbf{g}^*$ 的性质

(i) **扩散模型恢复**:

$$\mathbf{g}^*(\mathbf{x}_s, s, s) = \lim_{t\to s} \mathbf{g}^*(\mathbf{x}_s, s, t) = \mathbf{x}_s - s\mathbf{v}^*(\mathbf{x}_s, s).$$

(ii) **积分表示**:

$$\mathbf{g}^*(\mathbf{x}_s, s, t) = \mathbf{x}_s - s\mathbf{v}^*(\mathbf{x}_s, s) + \mathcal{O}(|t-s|).$$

Proof for Proposition.

根据 $\mathbf{g}^*$ 的定义可得

$$\lim_{s\to t} \mathbf{g}^*(\mathbf{x}_s, s, t) = \mathbf{x}_t - s\lim_{s\to t} \frac{1}{t-s}\int_s^t \mathbf{v}^*(\mathbf{x}_\tau, \tau)\,\mathrm{d}\tau = \mathbf{x}_s - s\mathbf{v}^*(\mathbf{x}_s, s).$$

由此证得第一等式。对于第二个结论，由泰勒展开可得

$$\begin{aligned}\mathbf{g}^*(\mathbf{x}_s, s, t) &= \mathbf{x}_s - \frac{s}{s-t}\int_t^s \mathbf{v}^*(\mathbf{x}_\tau, \tau)\,\mathrm{d}\tau \\ &= \mathbf{x}_s - \frac{s}{s-t}\Big[(s-t)\mathbf{v}^*(\mathbf{x}_s, s) + \mathcal{O}((t-s)^2)\Big] \\ &= \mathbf{x}_s - s\mathbf{v}^*(\mathbf{x}_s, s) + \mathcal{O}(|t-s|).\end{aligned}$$

由此命题，我们可以得出结论

1. 估计 $\mathbf{g}^*$ 可以近似得到不仅包括有限 s 到 t 转移（对于 $s \le t$ ），还包括由瞬时速度 $\mathbf{v}^*$ 表征的 *无穷小* s 到 s 转移。

2. $\mathbf{g}^*(\mathbf{x}_s, s, t)$ 被解释为泰勒展开的残差项与 $\mathbf{v}^*$ 相加的结果。

因此，通过利用 Equation (11.4.2) 中的 CTM 参数化，学习 $\mathbf{G}_{\boldsymbol{\theta}} \approx \boldsymbol{\Psi}_{s\to t}$ （或等价地，$\mathbf{g}_{\boldsymbol{\theta}} \approx \mathbf{g}^*$ ）不仅能够通过 $\mathbf{G}_{\boldsymbol{\theta}}$ 实现长跳变能力，还能通过 $\mathbf{g}_{\boldsymbol{\theta}}$ 恢复扩散模型的速度（或等价地，评分函数/去噪器）。因此，这种参数化至关重要：通过学习 $\mathbf{g}^*$ ，CTM 在单一框架下统一了扩散模型与一致性模型（特殊流映射）的优势。

在接下来的两个小节中，我们首先介绍 CTM 的一致性损失 (Section 11.4.2)，

该损失支持蒸馏和从零开始训练，并通过强制实现半群性质来达到 $\mathbf{G}_{\boldsymbol{\theta}}(\cdot, s, t) \approx \boldsymbol{\Psi}_{s\to t}(\cdot, s, t)$。随后，我们描述了由 Equation (11.4.2) 中的参数化自然产生的辅助损失 (Section 11.4.3)，包括扩散模型损失和 GAN 损失，这些损失显著提升了 CTM 的性能。

11.4.2 CTM 中的一致性损失

CTM 旨在逼近最优解映射

$$\mathbf{G}_{\boldsymbol{\theta}}(\cdot, s, t) \approx \boldsymbol{\Psi}_{s\to t}(\cdot, s, t),$$

对于任意 $s \geq t$。由于通常无法以闭式表达获得预言机 $\boldsymbol{\Psi}_{s\to t}$，CTM 通过强制执行 *半群性质* (Equation (11.2.1)) 构建一个可行的回归目标：对于任意 $s \geq u \geq t$，

$$\boldsymbol{\Psi}_{u\to t} \circ \boldsymbol{\Psi}_{s\to u} = \boldsymbol{\Psi}_{s\to t}.$$

根据是否可用预训练的扩散模型，流场 $\boldsymbol{\Psi}_{s\to t}$ 可以通过不同方式近似。本文中，我们假设 $s \geq u \geq t \in [0, T]$ 。

通过蒸馏进行训练。 假设可以访问一个生成 $\mathbf{v}_{\boldsymbol{\phi}^\times}(\mathbf{x}_s, s) \approx \mathbf{v}^*(\mathbf{x}_s, s)$ 的预训练扩散模型。那么 PF-ODE 可通过经验动力学近似表示为

$$\frac{\mathrm{d}\mathbf{x}(\tau)}{\mathrm{d}\tau} = \mathbf{v}_{\boldsymbol{\phi}^\times}(\mathbf{x}_\tau, \tau). \tag{11.4.3}$$

CTM 训练 $\mathbf{G}_{\boldsymbol{\theta}}$ 以匹配应用于该经验微分方程的数值求解器 $\texttt{Solver}_{s\to t}(\mathbf{x}_s; \boldsymbol{\phi}^\times)$，后者作为原始模型的可计算代理：

$$\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) \approx \texttt{Solver}_{s\to t}(\mathbf{x}_s; \boldsymbol{\phi}^\times) \approx \boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s, s, t).$$

在强教师模型的帮助下，求解器可以恢复 $\boldsymbol{\Psi}_{s\to t}$ 至离散化误差范围内，因此最优学生模型能紧密匹配真实值（见 (**kim2023consistency**)，命题 3 和 4)。

然而，在训练环中对整个区间 $[t, s]$ 进行求解在 s 与 t 相距较远时可能代价较高。为了提高效率并提供更平滑的信号，CTM 引入了 *软一致性匹配*，以实现半群性质。如 Figure 11.4 所示，CTM 在时间 t 比较两个预测：直接的学生输出 $\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t)$ ，以及一条混合的教师-学生路径，该路径先将教师从 s 推进到一

个随机的 $u \sim \mathcal{U}[t, s)$ ，然后让学生从 u 跳跃到 t ：

$$\mathbf{G}_{\boldsymbol{\theta}^-}\big(\texttt{Solver}_{s\to u}(\mathbf{x}_s;\boldsymbol{\phi}^\times),\, u,\, t\big).$$

该学生被训练以匹配这一综合预测：

$$\underbrace{\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t)}_{\approx \boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s)} \approx \underbrace{\mathbf{G}_{\boldsymbol{\theta}^-}\big(\texttt{Solver}_{s\to u}(\mathbf{x}_s;\boldsymbol{\phi}^\times), u, t\big)}_{\approx \boldsymbol{\Psi}_{u\to t}(\boldsymbol{\Psi}_{s\to u}(\mathbf{x}_s))}, \tag{11.4.4}$$

其中 $\boldsymbol{\theta}^-$ 是 $\mathbf{G}_{\boldsymbol{\theta}}$ 的梯度停止副本。

通过变化 u ，CTM 在全局监督和局部监督之间进行插值：

- **全局一致性** ($u = s$)：学生在完整区间 (t, s) 上模仿教师，接收最具信息量的教师信号。
- **局部一致性** ($u = s - \Delta s$)：学生从靠近 s 的短教师步骤中学习；当 $s = 0$ 时，这简化为一致性蒸馏。

为了在对齐轨迹的同时增强样本质量，两个预测均通过停止梯度的学生模型映射到时间 0 ，并在特征空间度量 d 中进行比较：

$$\begin{aligned}
\mathbf{x}_{\text{est}}(\mathbf{x}_s, s, t) &:= \mathbf{G}_{\boldsymbol{\theta}^-}\big(\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t),\, t,\, 0\big),\\
\mathbf{x}_{\text{target}}(\mathbf{x}_s, s, u, t) &:= \mathbf{G}_{\boldsymbol{\theta}^-}\big(\mathbf{G}_{\boldsymbol{\theta}^-}(\texttt{Solver}_{s\to u}(\mathbf{x}_s;\boldsymbol{\phi}^\times), u, t),\, t,\, 0\big).
\end{aligned}$$

CTM 一致性损失为

$$\mathcal{L}_{\text{consist}}(\boldsymbol{\theta};\boldsymbol{\phi}^\times) := \mathbb{E}_{s\in[0,T]}\mathbb{E}_{t\in[0,s]}\mathbb{E}_{u\in[t,s)}\mathbb{E}_{\mathbf{x}_0}\mathbb{E}_{\mathbf{x}_s|\mathbf{x}_0}\Big[d\big(\mathbf{x}_{\text{est}}, \mathbf{x}_{\text{target}}\big)\Big], \tag{11.4.5}$$

这鼓励学生在保持生成质量的同时，使经验性 PF-ODE 解相匹配。

从零开始训练。 利用 CTM 的特殊参数化（命题 11.4.1(i)），

$$\mathbf{g}^*(\mathbf{x}_\tau, \tau, \tau) = \mathbf{x}_\tau - \tau\,\mathbf{v}^*(\mathbf{x}_\tau, \tau) \implies \mathbf{v}^*(\mathbf{x}_\tau, \tau) = \frac{\mathbf{x}_\tau - \mathbf{g}^*(\mathbf{x}_\tau, \tau, \tau)}{\tau}.$$

因此，我们可以将 oracle 残差函数 $\mathbf{g}^*(\cdot,\tau,\tau)$ 替换为 CTM 自身对 $\tau\in[0,T]$ 的估计 $\mathbf{g}_{\boldsymbol{\theta}^-}(\cdot,\tau,\tau)$，从而得到一个自诱导的经验 PF-ODE：

$$\frac{\mathrm{d}\mathbf{x}(\tau)}{\mathrm{d}\tau}=\frac{\mathbf{x}(\tau)-\mathbf{g}_{\boldsymbol{\theta}^-}\left(\mathbf{x}(\tau),\tau,\tau\right)}{\tau}. \tag{11.4.6}$$

然后我们通过求解该常微分方程并训练学生模型以匹配求解器的输出，来近似原始解映射。

$$\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s,s,t)\approx \texttt{Solver}_{s\to t}(\mathbf{x}_s;\boldsymbol{\theta}^-)\approx \boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s,s,t).$$

如在蒸馏情形 Equation (11.4.4) 中，当 $[t,s]$ 和 s 远离时，对 t 的完整积分可能代价较高。因此，CTM 强制实施半群性质以获得更短的监督路径：

$$\underbrace{\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s,s,t)}_{\approx\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s)}\approx \underbrace{\mathbf{G}_{\boldsymbol{\theta}^-}\big(\texttt{Solver}_{s\to u}(\mathbf{x}_s;\boldsymbol{\theta}^-),\,u,\,t\big)}_{\approx\boldsymbol{\Psi}_{u\to t}(\boldsymbol{\Psi}_{s\to u}(\mathbf{x}_s))},$$

其中 $u\sim\mathcal{U}[t,s)$ 和 $\boldsymbol{\theta}^-$ 是学生网络的停止梯度副本。与蒸馏唯一的区别在于，外部教师 $\mathbf{v}_{\boldsymbol{\phi}^\times}$ 被自诱导教师 $\mathbf{g}_{\boldsymbol{\theta}^-}$ 所替代。

为了将轨迹匹配与样本质量相结合，将两个预测均通过停止梯度的学生模型映射到时间 0，并在特征空间中进行比较。未经任何预训练模型处理的目标是

$$\hat{\mathbf{x}}_{\text{target}}:=\mathbf{G}_{\boldsymbol{\theta}^-}\big(\mathbf{G}_{\boldsymbol{\theta}^-}(\texttt{Solver}_{s\to u}(\mathbf{x}_s;\boldsymbol{\theta}^-),\,u,\,t),\,t,\,0\big),$$

这取代了 $\mathbf{x}_{\text{target}}$ 中的 Equation (11.4.5)，导致：

$$\mathcal{L}_{\text{consist}}(\boldsymbol{\theta};\boldsymbol{\theta}^-):=\mathbb{E}_{s\in[0,T]}\mathbb{E}_{t\in[0,s]}\mathbb{E}_{u\in[t,s)}\mathbb{E}_{\mathbf{x}_0}\mathbb{E}_{\mathbf{x}_s|\mathbf{x}_0}\Big[d\big(\mathbf{x}_{\text{est}},\hat{\mathbf{x}}_{\text{target}}\big)\Big], \tag{11.4.7}$$

从概念上讲，这是 CTM 内部的自蒸馏：模型提供自身短时 horizon 的教师信号，而学生学习完整的转移过程。

11.4.3 CTM 中的辅助损失

(自) 蒸馏可能表现不如教师模型，因为它仅优化由教师生成的目标，缺乏来自真实数据的直接监督。相比之下，CTM 可以自然地引入数据驱动的正则化项，

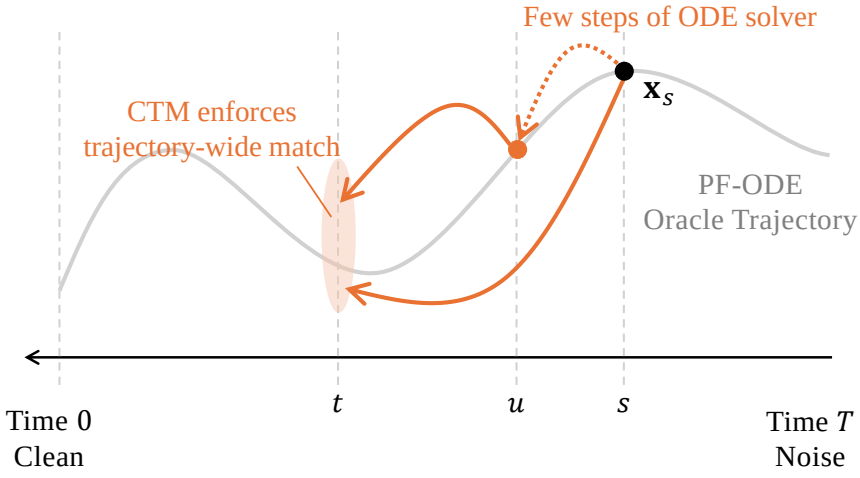


图 11.4: CTM 的半群性质示意图。对于任意 $s \geq u \geq t$，CTM 强制满足 $\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) \approx \mathbf{G}_{\boldsymbol{\theta}^-}\big(\texttt{Solver}_{s\to u}(\mathbf{x}_s), u, t\big)$，即，一个短时间求解器片段 $s \to u$ 之后接上一次 CTM "跳跃" 至 t，等价于直接的 CTM 映射 $s \to t$。该求解器可以是一个预训练的扩散模型，也可以是 CTM 自诱导的教师模型。

例如通过在目标中加入降噪分数匹配和对抗(GAN)项 (**goodfellow2014generative**)，以更好地学习流映射。

扩散损失的自然融合。 扩散-模型损失 (更准确地说，条件流匹配损失；见 Equation (5.2.9)) 自然地融入了 CTM，并提供了一个固定的回归目标，从而有助于流映射模型的学习。为了理解这一点，请注意我们有

$$\mathbf{v}^*(\mathbf{x}_s, s) = \frac{\mathbf{x}_s - \mathbf{g}^*(\mathbf{x}_s, s, s)}{s}, \qquad \mathbf{g}^*(\mathbf{x}_s, s, s) \approx \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, s).$$

这自然地通过 $\mathbf{g}_{\boldsymbol{\theta}}$ 诱导出一个速度参数化：

$$\mathbf{v}_{\boldsymbol{\theta}}(\mathbf{x}_s, s) := \frac{1}{s}\big(\mathbf{x}_s - \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, s)\big).$$

使用线性高斯路径

$$\mathbf{x}_s = \alpha_s \mathbf{x}_0 + \sigma_s \boldsymbol{\epsilon}, \qquad \mathbf{x}_0 \sim p_{\text{data}}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}),$$

扩散模型的损失可以表示为

$$\mathcal{L}_{\text{DM}}(\boldsymbol{\theta}) := \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, s}\Big[w(s)\big\|\mathbf{v}_{\boldsymbol{\theta}}(\mathbf{x}_s, s) - (\alpha_s' \mathbf{x}_0 + \sigma_s' \boldsymbol{\epsilon})\big\|_2^2\Big]. \tag{11.4.8}$$

$\mathcal{L}_{\text{DM}}$ 在 t 接近 s 时通过显式监督轨迹上的小跳跃来提高准确率。在此区域，Equation (11.4.2) 中的因子 $1-\frac{t}{s}$ 趋近于零，这可能削弱梯度并减慢学习；$\mathcal{L}_{\text{DM}}$ 提供更强的局部信号并稳定训练。

从概念上讲，Equations (11.4.5) and (11.4.7) 强制轨迹匹配（零阶），而 Equation (11.4.8) 强制斜率匹配（一阶）。

（可选）生成对抗网络损失。 虽然一致性和扩散模型损失提供了强大的回归信号，但它们可能导致输出过于平滑。因此，CTM 可选择性地添加对抗项，通过将生成器分布与数据分布对齐，以鼓励生成更清晰、更真实的样本。给定一个判别器 $D_{\boldsymbol{\zeta}}$ ，用于区分真实样本 $\mathbf{x}_0 \sim p_{\text{data}}$ 与生成样本 $\mathbf{x}_{\text{est}}(\mathbf{x}_s, s, t)$ ，目标是

$$\mathcal{L}_{\text{GAN}}(\boldsymbol{\theta}, \boldsymbol{\zeta}) := \\ \mathbb{E}_{\mathbf{x}_0}\big[\log D_{\boldsymbol{\zeta}}(\mathbf{x}_0)\big] + \mathbb{E}_{s\in[0,T]}\mathbb{E}_{t\in[0,s]}\mathbb{E}_{\mathbf{x}_0}\mathbb{E}_{\mathbf{x}_s|\mathbf{x}_0}\big[\log(1 - D_{\boldsymbol{\zeta}}(\mathbf{x}_{\text{est}}(\mathbf{x}_s, s, t)))\big],$$

where $D_{\boldsymbol{\zeta}}$ is maximized and $\mathbf{G}_{\boldsymbol{\theta}}$ is minimized. Intuitively, the discriminator acts as an adaptive perceptual 距离 that encourages realistic detail. Theoretically, the GAN term drives 分布匹配 (Jensen–Shannon divergence) between p_{data} and the 模型分布 induced by $\mathbf{G}_{\boldsymbol{\theta}}$ (**goodfellow2014generative**), which can raise fidelity beyond the teacher.

总体 CTM 目标。 总之，CTM 将（自）蒸馏、扩散和生成对抗网络损失统一到一个训练框架中：

$$\mathcal{L}_{\text{CTM}}(\boldsymbol{\theta}, \boldsymbol{\zeta}) := \mathcal{L}_{\text{consist}}(\boldsymbol{\theta}; \boldsymbol{\phi}^{\times}/\boldsymbol{\theta}^{-}) + \lambda_{\text{DM}}\mathcal{L}_{\text{DM}}(\boldsymbol{\theta}) + \lambda_{\text{GAN}}\mathcal{L}_{\text{GAN}}(\boldsymbol{\theta}, \boldsymbol{\zeta}),$$

其中，教师模型可以是外部预训练模型 $\boldsymbol{\phi}^{\times}$ 或自诱导教师模型 $\boldsymbol{\theta}^{-}$。回归风格的组件 $\mathcal{L}_{\text{consist}}$ 和 $\mathcal{L}_{\text{DM}}$ 充当强正则化项，而可选的 GAN 项在不牺牲稳定性 (**kim2024pagoda**) 的前提下提升了细粒度细节。

11.4.4 基于 CTM 的灵活采样

CTM 学习任意 $s > t$ 的通用流映射 $\boldsymbol{\Psi}_{s\to t}$，这意味着它支持任意时间到任意时间的转移。这一特性使得采样策略更加灵活。例如，CTM 提出了 γ 采样，

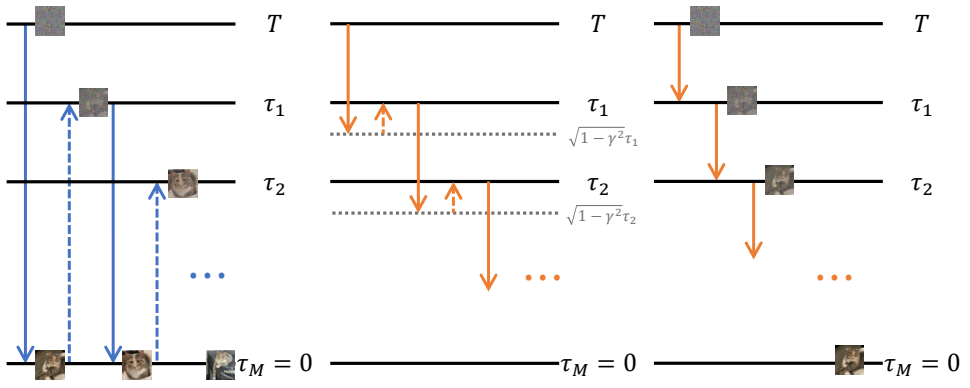


图 11.5: 不同 γ 值下的 γ -采样示意图。该过程在通过网络评估进行降噪与反向添加噪声之间交替进行，$(\tau_n \xrightarrow{\text{Denoise}} \sqrt{1-\gamma^2}\tau_{n+1} \xrightarrow{\text{Noisify}} \tau_{n+1})_{n=0}^{M-1}$ 。最左侧的面板展示了 $\gamma = 1$ ，对应完全随机的情况。最右侧的面板显示了 $\gamma = 0$ ，对应完全确定性的情况。中间的面板描绘了介于两者之间的 $\gamma \in (0, 1)$ 值，实现了这两种极端情况之间的插值。

其中超参数 γ 控制生成过程中的随机性。此外，CTM 可以复用为扩散模型开发的标准推理技术，如基于 ODE 的求解器和确切似然计算。

在下文中，我们固定一个用于采样 $T = \tau_0 > \tau_1 > \tau_2 > \cdots > \tau_M = 0$ 的离散时间网格。

Algorithm 11 CTM's γ-sampling

Input: Trained CTM $\mathbf{G}_{\boldsymbol{\theta}^\times}$, $\gamma \in [0, 1]$, $T = \tau_0 > \tau_1 > \tau_2 > \cdots > \tau_M = 0$.

1: Start from $\mathbf{x}_{\tau_0} \sim p_{\text{prior}} = \mathcal{N}(\mathbf{0}, T^2\mathbf{I})$
2: **for** $n = 0$ to $M - 1$ **do**
3: $\quad \tilde{\tau}_{n+1} \leftarrow \sqrt{1-\gamma^2}\tau_{n+1}$
4: $\quad$ Denoise $\mathbf{x}_{\tilde{\tau}_{n+1}} \leftarrow \mathbf{G}_{\boldsymbol{\theta}^\times}(\mathbf{x}_{\tau_n}, \tau_n, \tilde{\tau}_{n+1})$
5: $\quad$ Diffuse $\mathbf{x}_{\tau_{n+1}} \leftarrow \mathbf{x}_{\tilde{\tau}_{n+1}} + \gamma\tau_{n+1}\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
6: **end for**

Output: $\mathbf{x}_{\tau_M}$

γ 采样方法。 CTM 的 γ-采样引入了一类统一的采样器，其自然地源于学习一个通用流映射模型。它涵盖了先前的方法，例如 CM 的多步采样（见 Algorithm 9）以及类似常微分方程求解器的时间步进式采样。参数 γ 直接控制生成过程中语义变化的程度，使得 γ 采样成为一种灵活且任务感知的策略，适用于多种下游应用。

- Figure 11.5-(左)：当 $\gamma = 1$ 时，它与 CM 中引入的多步采样一致（即一种特殊的流映射 $\boldsymbol{\Psi}_{s\to 0}$），该方法完全随机，且在步骤数变化时会导致语义上的变化。

- Figure 11.5-(右)：当 $\gamma = 0$ 时，它退化为完全确定性的时间步进，用于估计 PF-ODE 的解轨迹。γ 采样与 $\gamma = 0$ 之间的关键区别在于，CTM 避免了数值求解器的离散化误差。

- Figure 11.5-(中间): 当 $0 < \gamma < 1$ 时，γ-采样通过在采样过程中注入受控量的随机性，在两个极端之间进行插值。

我们强调，只有当模型学习了通用流映射 $\boldsymbol{\Psi}_{s\to t}$ 时，才能实现具有 $\gamma \in (0, 1]$ 功能的采样器。

γ 采样分析。 CTM 在实验中观察到，当采样步数 $M \geq 4$ 时，CM 的多步采样质量会下降。为了阐明这一现象，CTM 分析了其潜在成因：当 $\gamma \neq 0$ 时，每次神经“跳跃”都会引入微小的偏差，而这些偏差在模型迭代地将状态映射回时间零点的过程中不断累积。这种误差累积解释了为何长距离的多步运行表现不佳。我们在下述命题中形式化了这一思想。

Proposition 11.4.2: (非正式) 两步 γ -采样

令 $\tau \in (0, T)$ 和 $\gamma \in [0, 1]$。令 $p_{\boldsymbol{\theta}^*,2}$ 表示从 γ -采样器获得的密度，该采样器采用最优 CTM，遵循转移序列 $T \to \sqrt{1-\gamma^2}\tau \to \tau \to 0$，从 p_{prior} 开始。则有

$$\mathcal{D}_{\text{TV}}\left(p_{\text{data}}, p_{\boldsymbol{\theta}^*,2}\right) = \mathcal{O}\left(\sqrt{T - \sqrt{1-\gamma^2}\tau + \tau}\right).$$

此处，$\mathcal{D}_{\text{TV}}$ 表示分布间的总变差（参见 Equation (1.1.4)）。

Proof for Proposition.

关于采样步数为 M 的一般情形，请参阅 **kim2023consistency** 的定理 8。

上述定理的启示可总结如下：

- **当** $\gamma = 1$ （对应于 CM 的多步采样）**：**该方法在每一步 n 上执行从 τ_n 到

0 的迭代长距离转移。这导致误差累积达到阶次为

$$\mathcal{O}\left(\sqrt{T+\tau_1+\cdots+\tau_M}\right).$$

- **当** $\gamma = 0$ (对应于 CTM 的确定性多步采样)**:** 消除了转移之间的时序重叠。这避免了误差累积，得到了更紧的 $\mathcal{O}(\sqrt{T})$ 界。经验上，采用 $\gamma = 0$ 的 CTM 在采样速度和样本质量之间提供了有利的权衡：增加采样步数可提升生成质量，而不会引入不稳定性。

CTM 支持扩散推理。 由于 CTM 通过 $\mathbf{g}_{\boldsymbol{\theta}}$ 直接学习评分函数（或去噪器），得益于其在 Equation (11.4.2) 中的参数化，它与最初为扩散模型开发的推理技术兼容。例如，可以通过使用 $\mathbf{g}_{\boldsymbol{\theta}}(\cdot, s, s)$ 计算确切的似然（Section 4.2.2），或应用 DDIM 或 DPM（Chapter 9）等高级采样器进行生成。

11.5 General Flow Map: Mean Flow

正如扩散模型允许多种等价的参数化和训练目标，一般的流映射 $\mathbf{\Psi}_{s\to t}$ 也可以通过多种合理的方式进行学习。在本节中，我们介绍 *均值流*（MF）(**geng2025mean**)，它是通用流映射族 $\mathbf{\Psi}_{s\to t}$ 的后期代表之一，展示了如何有效学习此类映射的一种替代但原理严谨的视角。

11.5.1 平均流的建模与训练

与基于 EDM 框架的 CM 和 CTM 不同，MF 基于流匹配公式（$\alpha_t = 1 - t$ 和 $\sigma_t = t$ 对应 $t \in [0, 1]$ ）。MF 并非直接参数化流映射，而是学习区间 $[t, s]$ 上的 *平均漂移*（其中 $t < s$ ）。

$$\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) \approx \mathbf{h}^*(\mathbf{x}_s, s, t) := \frac{1}{t-s}\int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u.$$

相应的预言机损失是

$$\mathbb{E}_{t<s}\mathbb{E}_{\mathbf{x}_s\sim p_s}\Big[w(s)\|\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) - \mathbf{h}^*(\mathbf{x}_s, s, t)\|_2^2\Big]. \tag{11.5.1}$$

特别是，当 $s \to t$ 时，损失函数简化为流匹配损失：

$$\mathbb{E}_t\mathbb{E}_{\mathbf{x}_t\sim p_t}\Big[w(t)\|\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, t) - \mathbf{v}^*(\mathbf{x}_t, t)\|_2^2\Big], \tag{11.5.2}$$

学习瞬时速度。我们将在稍后 Section 11.5.3 中看到，MF 保持与 Equation (10.1.4) 中一般目标的一致性，但其是从一个不同（尽管等价）的角度来逼近该目标的。由于在一般情况下，预言回归目标 $\mathbf{h}^*(\mathbf{x}_s, s, t)$ 并不存在闭式解，因此 MF 通过利用微分得到的一个恒等式来构建一个代理目标。

$$(t-s)\,\mathbf{h}^*(\mathbf{x}_s, s, t) = \int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\ \mathrm{d}u$$

关于 s 。这得到

$$\begin{aligned}\mathbf{h}^*(\mathbf{x}_s, s, t) &= \mathbf{v}^*(\mathbf{x}_s, s) - (s-t)\tfrac{\mathrm{d}}{\mathrm{d}s}\mathbf{h}^*(\mathbf{x}_s, s, t)\\ &= \mathbf{v}^*(\mathbf{x}_s, s) - (s-t)\Big[(\partial_{\mathbf{x}}\mathbf{h}^*)(\mathbf{x}_s, s, t)\mathbf{v}^*(\mathbf{x}_s, s) + \partial_s\mathbf{h}^*(\mathbf{x}_s, s, t)\Big],\end{aligned}$$

其中第二行应用了链式法则结合

$$\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{x}_s = \mathbf{v}^*(\mathbf{x}_s, s).$$

受此恒等式启发，MF 用一种停止梯度的替代方法取代了难以处理的虚设模型，从而得到实际的训练目标。

$$\mathcal{L}_{\mathrm{MF}}(\boldsymbol{\theta}) := \mathbb{E}_{t<s}\mathbb{E}_{\mathbf{x}_s\sim p_s}\Big[w(s)\,\|\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) - \mathbf{h}^{\mathrm{tgt}}_{\boldsymbol{\theta}^-}(\mathbf{x}_s, s, t)\|_2^2\Big], \qquad (11.5.3)$$

其中，回归目标定义为

$$\mathbf{h}^{\mathrm{tgt}}_{\boldsymbol{\theta}^-}(\mathbf{x}_s, s, t) := \mathbf{v}^*(\mathbf{x}_s, s) - (s-t)\Big[\underbrace{(\partial_{\mathbf{x}}\mathbf{h}_{\boldsymbol{\theta}^-})(\mathbf{x}_s, s, t)\mathbf{v}^*(\mathbf{x}_s, s) + \partial_s\mathbf{h}_{\boldsymbol{\theta}^-}(\mathbf{x}_s, s, t)}_{\mathrm{JVP}}\Big].$$

在实际应用中，预言机速度 $\mathbf{v}^*$ 无法以闭式形式计算，必须进行近似。两种常用策略可供选择：依赖于预训练的扩散模型（蒸馏），或直接从数据构建估计量（从零开始训练）。无论选择哪种方式，最终都需要计算目标网络 $\mathbf{h}_{\boldsymbol{\theta}^-}$ 的 雅克比-向量积（JVP）：

$$[\partial_{\mathbf{x}}\mathbf{h}_{\boldsymbol{\theta}^-}, \partial_s\mathbf{h}_{\boldsymbol{\theta}^-}, \partial_t\mathbf{h}_{\boldsymbol{\theta}^-}]^\top \cdot [\mathbf{v}^*, 1, 0]$$

蒸馏 使用具有流匹配主干的预训练扩散模型，$\mathbf{v}_{\boldsymbol{\phi}^\times} \approx \mathbf{v}^*$.

从零开始训练。 使用一点条件速度 $\alpha'_s\mathbf{x}_0 + \sigma'_s\boldsymbol{\epsilon}$ ，该速度通过前向噪声注入 $\mathbf{x}_s = \alpha_s\mathbf{x}_0 + \sigma_s\boldsymbol{\epsilon}$ 与 $\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0}, \mathbf{I})$ 获得。这在配对 $(\mathbf{x}_0, \boldsymbol{\epsilon})$ 处评估时，给出了在层级 s 下瞬时漂移的无偏单样本估计。

11.5.2 平均流采样

一旦一个 MF $\mathbf{h}_{\boldsymbol{\theta}^\times}$ 被训练，它自然会恢复流映射的代理。对于任意起点 $\mathbf{x}_s$ ，从 s 到 t 的映射（近似地）由

$$\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s) = \mathbf{x}_s + (t-s)\,\mathbf{h}^*(\mathbf{x}_s, s, t) \approx \mathbf{x}_s + (t-s)\,\mathbf{h}_{\boldsymbol{\theta}^\times}(\mathbf{x}_s, s, t).$$

这使得一步采样和多步采样成为可能。例如，绘制 $\mathbf{x}_T \sim p_{\text{prior}}$ ，即可实现干净样本的一步生成。

$$\mathbf{x}_0 \leftarrow \mathbf{x}_T + T\,\mathbf{h}_{\boldsymbol{\theta}^-}(\mathbf{x}_T, T, 0).$$

另外，可以通过准备一个时间网格并以与 CTM 中相同的时间步进方式依次应用映射来执行多步生成。由于 MF 学习了一个通用的流映射，它同样支持如 CTM 中的 γ-采样，其中可控的超参数 γ 将随机性注入到采样过程中。

11.5.3 CTM 与 MF 的等价性

乍看之下，CTM 和 MF 似乎毫无关联。事实上，两者仅仅是同一预言流映射 $\boldsymbol{\Psi}_{s\to t}$ 的不同参数化形式，其训练损失（CTM 的一致性损失与 Equation (11.5.1)）仅在时间加权上有所不同 (**hu2025cmt**)。

参数化的关联关系。 两种方法均在相同的通用框架下运行，但以不同的方式表示学成的函数。流场图可等价地表示为

$$\begin{aligned}
\boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s) &= \mathbf{x}_s + \int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u \\
&= \frac{t}{s}\mathbf{x}_s + \frac{s-t}{s}\underbrace{\left[\mathbf{x}_s + \frac{s}{s-t}\int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\;\mathrm{d}u\right]}_{\approx\,\mathbf{g}_{\boldsymbol{\theta}}} \\
&= \mathbf{x}_s + (t-s)\underbrace{\left[\frac{1}{t-s}\int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\;\mathrm{d}u\right]}_{\approx\,\mathbf{h}_{\boldsymbol{\theta}}}.
\end{aligned}$$

此处，第一个是流映射的定义，第二个形式通过 $\mathbf{g}_{\boldsymbol{\theta}}$ 突出了 CTM 参数化（见 Equations (11.4.1) and (11.4.2)），而最后一个则通过 $\mathbf{h}_{\boldsymbol{\theta}}$ 突出了 MF 参数化。

训练损失的关系 在上述对预言机流映射 $\boldsymbol{\Psi}_{s\to t}$ 关于 CTM 参数化的重新解释下

$$\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) \approx \mathbf{g}^*(\mathbf{x}_s, s, t) := \mathbf{x}_s + \frac{s}{s-t}\int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u$$

以及 MF 参数化

$$\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) \approx \mathbf{h}^*(\mathbf{x}_s, s, t) := \frac{1}{t-s}\int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u,$$

我们现在证明，CTM 和 MF 的训练损失实际上是等价的。

考虑关系

$$\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) := \mathbf{x}_s - s\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t),$$

以 $d(\mathbf{x}, \mathbf{y}) := \|\mathbf{x} - \mathbf{y}\|^2$ 为例。将其代入 Equation (10.1.4) 并将 $\mathbf{G}_{\boldsymbol{\theta}}$ 视为 CTM 的流映射参数化 (Equation (11.4.2)) 可得

$$\begin{aligned}
& d\big(\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t), \boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s)\big) \\
=\ & \|\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) - \boldsymbol{\Psi}_{s\to t}(\mathbf{x}_s)\|^2 \\
=\ & \left\| \left(\frac{t}{s}\mathbf{x}_s + \frac{s-t}{s}\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t)\right) - \left(\frac{t}{s}\mathbf{x}_s + \frac{s-t}{s}\left[\mathbf{x}_s + \frac{s}{s-t}\int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u\right]\right)\right\|^2 \\
=\ & \left(\frac{s-t}{s}\right)^2 \left\| \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) - \left(\mathbf{x}_s + \frac{s}{s-t}\int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u\right)\right\|^2 \qquad (11.5.4) \\
=\ & \left(\frac{s-t}{s}\right)^2 \left\| (\mathbf{x}_s - s\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t)) - \left(\mathbf{x}_s + \frac{s}{s-t}\int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u\right)\right\|^2 \\
=\ & \left(\frac{s-t}{s}\right)^2 \left\| (\mathbf{x}_s - s\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t)) - \left(\mathbf{x}_s + \frac{s}{s-t}\int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u\right)\right\|^2 \\
=\ & (s-t)^2 \left\| \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) - \left(\frac{1}{t-s}\int_s^t \mathbf{v}^*(\mathbf{x}_u, u)\,\mathrm{d}u\right)\right\|^2 \qquad (11.5.5)
\end{aligned}$$

因此，

$$\frac{1}{s^2}\left\| \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) - \mathbf{g}^*(\mathbf{x}_s, s, t)\right\|^2 = \left\| \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) - \mathbf{h}^*(\mathbf{x}_s, s, t)\right\|^2.$$

因此，CTM 和 MF 损失在权重函数的意义下本质上是等价的。此外，在这两种情况下设定 $t = 0$ 会恢复 CM 情景 ($\boldsymbol{\Psi}_{s\to 0}$)，其中每个状态都直接映射到干净数据。

实际应用中的辅助损失。 在 CTM 中，训练通过 Equation (11.4.7) 的一致性损失与自定义的扩散模型损失 Equation (11.4.8) 联合进行。MF 也采用了类似的策略。如 Equation (11.5.2) 所示，当 $s \to t$ 时，MF 损失退化为标准的流匹配目标。在实际应用中，MF 通过 $s \neq t$ 控制配对的比例，其中包含 $s = t$；因此，整体最优化成为 Equation (11.5.3) 中的 MF 目标与 Equation (11.5.2) 中的流匹配目标的混合。

两种参数化方法均能实现从扩散模型训练到流场映射学习的平滑转移，其

中扩散模型训练通过固定的回归目标学习瞬时速度，而流场映射学习则采用带停止梯度的伪回归目标。

CTM 和 MF 参数化均支持灵活推理。 CTM ($\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t)$) 和 MF ($\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t)$) 均旨在近似潜在流映射 $\boldsymbol{\Psi}_{s\to t}$ ：

$$\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) \approx \boldsymbol{\Psi}_{s\to t}, \quad \text{and} \quad \mathbf{x}_s + (t-s)\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, t) \approx \boldsymbol{\Psi}_{s\to t}.$$

由于两个模型均学习任意两个时间步之间的显式映射，它们自然支持 CTM 的 γ-采样，并且与最初为扩散模型开发的推理技术保持兼容，例如引导 (Chapter 8)、确切似然计算 (Equation (4.2.7)) 以及使用高阶求解器的加速采样 (Chapter 9)。这种兼容性源于其参数化在无穷小极限 $t \to s$ 下能够恢复瞬时扩散漂移。

$$\mathbf{g}^*(\mathbf{x}_s, s, s) = \mathbf{x}_s - \mathbf{v}^*(\mathbf{x}_s, s), \quad \text{and} \quad \mathbf{h}^*(\mathbf{x}_s, s, s) = \mathbf{v}^*(\mathbf{x}_s, s).$$

该性质并不为专门的流图公式 $\boldsymbol{\Psi}_{s\to 0}$ 所共享，例如 CM 系列中的公式。因此，CTM 和 MF 均可被视为灵活且通用的流图公式，能够将基于扩散的推理泛化为直接的时间到时间映射。

结论。 CTM 与 MF 之间的这种等价性类似于扩散模型 (Section 6.3) 中的情况，其中不同的参数化最终描述了相同的潜在目标。原则上，这些表述在数学上是完全相同的。然而，在实际应用中，由于损失权重、网络设计或最优化动力学等因素，它们的行为可能会有所不同，这可能导致在特定条件下某一方法的表现优于另一方法。

这一观点表明，CTM 和 MF 并非唯一可能：其他形式的流映射参数化同样可能实现高效且稳定的训练，从而为新型独立生成式模型开辟道路。探索这些替代方案将进一步丰富扩散模型及其流映射扩展的体系，最终推动少步生成能力的边界。

11.6 闭幕词

本章最后部分使我们的探索回到了起点，最终形成了一种生成式建模的新范式：从零开始学习快速、少步骤的生成器。我们超越了改进数值求解器或蒸馏预训练模型的方法，专注于设计独立的训练原则，这些原则在理论上严谨且天生高效。

这里提出的核心创新是直接学习潜在概率流微分方程的流映射（$\mathbf{\Psi}_{s\to t}$）。使这一方法在无教师的情况下仍具有可处理性的关键在于利用了微分方程流的基本半群性质。该性质表明，长轨迹可以分解为较短的片段，从而为训练提供了强大的自监督信号。

我们从一致性模型（CMs）开始，这类模型通过学习一种特殊的流映射，将任意噪声状态回传至其干净的原点（$\mathbf{\Psi}_{s\to 0}$）。随后，一致性轨迹模型（CTM）和均值流（MF）将这一思想推广，学习了适用于所有满足 $s \geq t$ 的 s,t 的完整、任意时间到任意时间的流映射 $\mathbf{\Psi}_{s\to t}$。尽管它们在参数化方式上看似不同，但我们表明，这些方法本质上是近似同一路径积分的等效方式，而该路径积分定义了流映射。

这些流图模型体现了本专著中所发展的原理的有力综合。它们继承了得分 SDE 框架的连续时间基础和流匹配的确定性传输视角，但重新制定了训练目标，使其自洽且高效。

通过直接学习解映射，这些独立模型成功结合了迭代扩散过程的高样本质量与单步生成器的推理速度。它们解决了保真度与速度之间的根本权衡问题，标志着生成式建模的一个重要里程碑。这一成就并非终点，而是强大、高效且可控的生成式人工智能设计新篇章的开端。

> 重要的是不要停止提问。好奇心本身就有存在的理由。
>
> 阿尔伯特 · 爱因斯坦

Appendices

A

微分方程速成课

微分方程（DEs）是建模动态系统的基本工具，可大致分为 *常微分方程*（ODEs）、*随机微分方程*（SDEs）和 *偏微分方程*（PDEs）。

常微分方程（ODEs）描述了系统状态如何根据确切规则随时间演化，使得已知初始状态能够确切地确定未来的演化路径。随机微分方程（SDEs）在此基础上引入了随机性，用于建模噪声或不确定性对系统行为的影响，使结果具有概率性而非确定性。偏微分方程（PDEs）则解释了依赖于多个变量（如时间和空间）的函数如何共同演化，能够刻画诸如热传导、波传播，以及*随机系统中概率密度的时间演化*（提示：福克-普朗克方程）等现象。这类微分方程构成了理解系统在确定性和随机影响下随时间和空间演化的基础语言。

在本章中，我们提供微分方程的基本预备知识。

A.1 常微分方程基础

本节介绍了常微分方程的基本理论，重点阐述了在给定初值条件下解的唯一性。同时，还介绍了使用数值求解器求解常微分方程的实际方法。

A.1.1 常微分方程的直观理解

确定性过程被称为 *常微分方程*（ODE）。在多变量情况下，我们考虑如下形式的系统：

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{v}(\mathbf{x}(t), t), \tag{A.1.1}$$

其中 $\mathbf{x}(t) \in \mathbb{R}^D$ 是一个向量值函数，表示系统在时间 t 的状态，$\mathbf{v} : \mathbb{R}^D \times \mathbb{R} \to \mathbb{R}^D$ 是一个向量场，用于指定空间和时间每一点处变化的方向和大小。

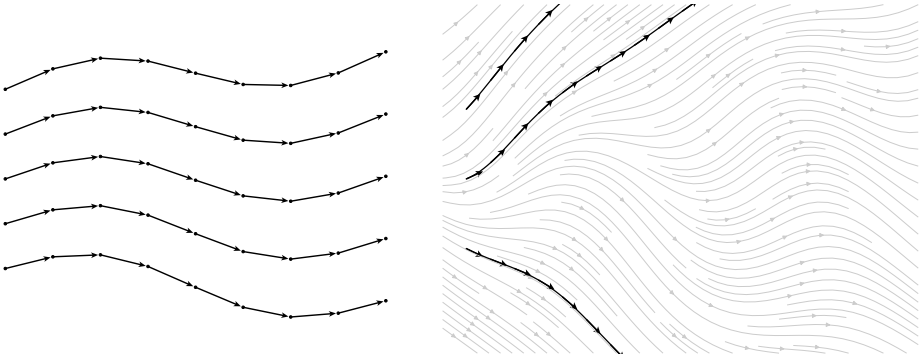


图 A.1: 常微分方程示例。速度场 $\mathbf{v}(\mathbf{x}, t)$ 在每一点上赋予一个漂移向量。解轨迹 $\mathbf{x}(t)$ 是一条切线始终与局部漂移匹配的路径。左图展示了逐步求解器更新（点和箭头）对路径的近似，而右图显示了与速度场一致流动的确切轨迹（黑色）。若未指定初始状态 $\mathbf{x}(0)$，则存在无穷多条瞬时变化与同一速度场匹配的轨迹。然而，一旦 $\mathbf{x}(0)$ 确定，该常微分方程即确定了一条唯一的路径 $\mathbf{x}(t)$，其流动遵循漂移方向。

常微分方程求解的高层次直觉。 为了建立直觉，可以想象向量场 $\mathbf{v}(\mathbf{x}, t)$ 为一个动态的箭头景观，它告诉你在任意时刻 t，点 $\mathbf{x}$ 应该如何移动。求解微分方程意味着在该场中描绘一条曲线 $\mathbf{x}(t)$，使得该曲线在任意点处的切线（即瞬时速度）与 $\mathbf{v}(\mathbf{x}(t), t)$ 所给出的向量相一致。

- **向量场视角**：函数 $\mathbf{v}(\mathbf{x}, t)$ 定义了事物应该如何运动：它提供了运动或变化的局部“指令”。
- **轨迹视角**：解 $\mathbf{x}(t)$ 是一个粒子在每一时刻都遵循向量场 $\mathbf{v}$ 所设定规则时所走过的路径。

因此，求解常微分方程就像是将一个粒子放入流场中，观察它随时间的运动轨迹。

A.1.2 常微分方程的存在性与唯一性

到目前为止，我们已经看到，求解常微分方程意味着找到一条路径，该路径在每一点上都遵循向量场给出的方向。直观上，这就像追踪一个粒子在由速度定义的流动中运动时所经过的轨迹。

但这个图像引出了一个重要的问题：

Question A.1.1

如果我们选取一个起点，能否确定确实存在一条遵循这些方向的路径？如果存在这样的路径，它是否唯一，还是粒子可能会突然跳到另一条轨迹上？

回答这些问题至关重要，因为这告诉我们系统的状态是否能从其初始位置可靠地预测。*存在性与唯一性定理*给出了向量场的条件，确保从任意给定的初始点出发，恰好存在一条路径。这保证了解的行为具有一致性，是常微分方程理论的基石。

局部（时间上）存在性与唯一性定理。 下面我们陈述该定理的一个局部版本，该版本断言：对于给定的初始条件，在初始时刻的邻域内存在唯一解。

Theorem A.1.1: 局部存在性与唯一性

设 $\mathbf{v}(\mathbf{x},t)$ 关于 $\mathbf{x}$ 和 t 在领域 $D \subseteq \mathbb{R}^D \times \mathbb{R}$ 内连续。若 $\mathbf{v}$ 关于 $\mathbf{x}$ 满足利普希茨条件：

$$\|\mathbf{v}(\mathbf{x}_1,t) - \mathbf{v}(\mathbf{x}_2,t)\| \leq L\|\mathbf{x}_1 - \mathbf{x}_2\| \quad \forall (\mathbf{x}_1,t),(\mathbf{x}_2,t) \in D,$$

其中 $L > 0$ 为常数，则对任意初始条件 $\mathbf{x}(t_0) = \mathbf{x}_0$ ，存在唯一解 $\mathbf{x}(t)$ 在区间 $[t_0 - \delta, t_0 + \delta]$ 上满足 Equation (A.1.1) 。

Proof for Theorem.

（证明概要）存在唯一性定理可通过皮卡-林德勒夫迭代法构造性证明。该方法生成函数序列 $\{\mathbf{x}_n(t)\}$ 收敛于解 $\mathbf{x}(t)$ 。迭代定义为：

$$\mathbf{x}_{n+1}(t) = \mathbf{x}_0 + \int_{t_0}^{t} \mathbf{v}(\mathbf{x}_n(s), s)\, \mathrm{d}s.$$

- 从初始猜测 $\mathbf{x}_0(t) = \mathbf{x}_0$ 开始
- 使用积分形式迭代优化 $\mathbf{x}_n(t)$
- 根据利普希茨条件，应用压缩映射定理可保证收敛性

该证明的本质基于 Picard–Lindelöf 迭代方法，其核心思想同样被 Section 9.8用于加速扩散模型的采样过程。

全局（时间上）存在性与唯一性定理。 虽然局部存在唯一性定理保证了解在小时间区间内的存在性，但“全局（在时间上）存在唯一性定理”在附加正则性条件下将这一结果推广到了整个区间 $[t_0, T]$。此类中的一个著名结果是 *Carathéodory* 定理，该定理在两个关键假设下确保了常微分方程解的全局存在性和唯一性：状态变量的局部利普希茨连续性以及线性增长界。

(i) **在 $\mathbf{x}$ 中的局部利普希茨条件**：存在一个在 $[0, T]$ 上可积的函数 $\mathrm{Lip}(t)$，使得对所有 $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^D$，

$$\|\mathbf{v}(\mathbf{x}_1, t) - \mathbf{v}(\mathbf{x}_2, t)\| \leq \mathrm{Lip}(t)\|\mathbf{x}_1 - \mathbf{x}_2\|.$$

(ii) **线性增长条件**：存在一个在 $[0, T]$ 上可积的函数 $M(t)$，使得对所有 $\mathbf{x} \in \mathbb{R}^D$，

$$\|\mathbf{v}(\mathbf{x}, t)\| \leq M(t)(1 + \|\mathbf{x}\|).$$

我们建议读者参阅 (**reid1971ordinary**) 以获得该定理假设条件、形式化声明及详细证明的全面讨论。

Remark.

为将上述定理应用于扩散模型中的概率流常微分方程（参见 Equation (4.1.7)），可能需要对评分函数 $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ 附加假设条件，例如条件（i）与（ii）。对于不关注技术细节的读者，这些假设无需进一步论证即可合理接受。

总之，当给定一个由时变速度场定义的常微分方程的初始条件时，粒子流的轨迹是唯一确定的。

唯一性蕴含解的不相交性 常微分方程解的唯一性由局部存在性与唯一性定理保证，这蕴含了一个基本性质：从不同初值出发的两条不同解轨迹不可能相互交叉。这反映了常微分方程的确定性特征，确保每个状态沿唯一路径演化。以下推论正式表述了这一结果。

Corollary A.1.1: Non-Intersection of Solutions

Consider two solutions $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$ to the ODE

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{v}\left(\mathbf{x}(t), t\right), \quad t \in [0, T].$$

Suppose they have distinct initial values $\mathbf{x}_1(0) \neq \mathbf{x}_2(0)$. Then, these solutions do not intersect on $[0, T]$, i.e.,

$$\mathbf{x}_1(t) \neq \mathbf{x}_2(t) \quad \text{for all } t \in [0, T].$$

Proof for Corollary.

Assume, for the sake of contradiction, that there exists some $t^* \in (0, T]$ such that

$$\mathbf{x}_1(t^*) = \mathbf{x}_2(t^*).$$

Define the first time at which the two solutions meet as

$$t_0 := \inf\{t \in [0, T] | \mathbf{x}_1(t) = \mathbf{x}_2(t)\}.$$

Since $\mathbf{x}_1(0) \neq \mathbf{x}_2(0)$ and t^* is contained in this set, it follows that $t_0 > 0$. By continuity of $\mathbf{x}_1$ and $\mathbf{x}_2$, we have

$$\mathbf{x}_1(t_0) = \mathbf{x}_2(t_0).$$

Consider the initial value problem

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{v}(\mathbf{x}(t), t), \quad \mathbf{x}(t_0) = \mathbf{x}_1(t_0).$$

By the uniqueness theorem for ODEs, both $\mathbf{x}_1$ and $\mathbf{x}_2$ must coincide on the interval $[t_0, T]$. Applying uniqueness backward in time similarly implies that

the two solutions coincide on $[0, t_0]$. Therefore, the solutions satisfy

$$\mathbf{x}_1(t) = \mathbf{x}_2(t) \quad \text{for all } t \in [0, T],$$

which contradicts the assumption that $\mathbf{x}_1(0) \neq \mathbf{x}_2(0)$. Hence, we conclude that

$$\mathbf{x}_1(t) \neq \mathbf{x}_2(t) \quad \text{for all } t \in [0, T].$$

通过保证解路径不相交，该定理为流映射模型提供了隐含但至关重要的支持（见 Chapters 10 and 11）。

A.1.3 指数积分因子

即使由一般时变速度 $\mathbf{v}$ 确定的常微分方程（ODE）不具有闭式解，但在某些特殊情况下，我们仍可对其进行解析求解，或将它的表达形式简化为更具结构性的形式。

一个说明性的例子。 考虑以下线性标量常微分方程：

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = L(t)\mathbf{x}(t),$$

其中 $L(t) \in \mathbb{R}$ 是一个连续函数。该方程可以解析求解，其解是众所周知的（对于任意 s 和 t ）：

$$\mathbf{x}(t) = \mathbf{x}(s) \cdot \exp\left(\int_s^t L(\tau)\ \mathrm{d}\tau\right).$$

该公式展示了解如何根据一个指数因子演化，该因子累积了时变系数 $L(t)$ 的影响。这促使我们使用 *指数积分因子*：

$$\mathcal{E}(s \to t) := \exp\left(\int_s^t L(\tau)\,\mathrm{d}\tau\right), \tag{A.1.2}$$

尤其是在动态过程同时包含线性和非线性分量的更一般情景中。

半线性常微分方程与指数积分因子。 我们现在考虑一类更广泛的常微分方程，称为*半线性常微分方程*。这类方程将动力学分解为一个关于状态变量的线性部

分和一个非线性余项：

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = L(t)\mathbf{x}(t) + \mathbf{N}(\mathbf{x}(t), t), \tag{A.1.3}$$

其中 $\mathbf{x}(t) \in \mathbb{R}^D$ 为状态向量，$L(t)$ 为标量值连续函数，$\mathbf{N} : \mathbb{R}^D \times [0, T] \to \mathbb{R}^D$ 为非线性向量场。这种半线性结构在许多物理和工程系统中自然出现。特别是，它也出现在扩散模型的概率流常微分方程公式中（见 Equation (4.1.7)）。识别这一结构可使指数积分因子得以应用，这不仅简化了分析，还提高了数值稳定性。具体而言，该技术在快速扩散常微分方程求解器的设计中起着核心作用（见 Chapter 9）。

步骤 1：通过积分因子分离非线性项。 观察到我们可以通过从 Equation (A.1.3) 中的半线性常微分方程减去线性漂移项来分离非线性部分：

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} - L(t)\mathbf{x}(t) = \mathbf{N}(\mathbf{x}(t), t).$$

为了吸收线性项，我们将等式两边乘以逆积分因子：

$$\mathcal{E}^{-1}(s \to t) = \exp\left(-\int_s^t L(\tau)\,\mathrm{d}\tau\right).$$

现在对左侧应用乘法法则：

$$\mathcal{E}^{-1}(s \to t)\left(\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} - L(t)\mathbf{x}(t)\right) = \frac{\mathrm{d}}{\mathrm{d}t}\left[\mathcal{E}^{-1}(s \to t)\mathbf{x}(t)\right].$$

因此，方程变为：

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\mathcal{E}^{-1}(s \to t)\mathbf{x}(t)\right] = \mathcal{E}^{-1}(s \to t)\mathbf{N}(\mathbf{x}(t), t).$$

这种变换通过分离非线性分量简化了原方程，使我们能够在变换后的坐标系中完全专注于非线性动力学。

步骤 2：对时间积分。 我们现在从 s 积分到 t：

$$\int_s^t \frac{\mathrm{d}}{\mathrm{d}\tau}\left[\mathcal{E}^{-1}(s \to \tau)\mathbf{x}(\tau)\right]\mathrm{d}\tau = \int_s^t \mathcal{E}^{-1}(s \to \tau)\mathbf{N}(\mathbf{x}(\tau), \tau)\,\mathrm{d}\tau.$$

左边仅仅是变换后变量在 t 和 s 处取值的差：

$$\mathcal{E}^{-1}(s \to t)\mathbf{x}(t) - \mathbf{x}(s).$$

因此，我们得到：

$$\mathcal{E}^{-1}(s \to t)\mathbf{x}(t) = \mathbf{x}(s) + \int_s^t \mathcal{E}^{-1}(s \to \tau)\mathbf{N}(\mathbf{x}(\tau), \tau)\, \mathrm{d}\tau.$$

步骤 3：求解 $\mathbf{x}(t)$。 两边同乘指数流 $\mathcal{E}(s \to t)$，得到解：

$$\mathbf{x}(t) = \underbrace{\mathcal{E}(s \to t)\mathbf{x}(s)}_{\text{linear part}} + \underbrace{\int_s^t \mathcal{E}(\tau \to t)\mathbf{N}(\mathbf{x}(\tau), \tau)\, \mathrm{d}\tau}_{\text{nonlinear part}}. \tag{A.1.4}$$

解自然地分离为线性和非线性部分。指数积分器通过确切地解析求解线性部分，仅对非线性残差进行离散化来利用这一结构。这确保了步长由非线性动力学决定，而非可能很大的线性系数决定，从而即使步数较少也能得到稳定且准确的更新（参见指数欧拉更新 Equation (9.1.7) 与普通欧拉更新 Equation (9.1.8) 的比较）。

A.1.4 常微分方程的数值求解器

我们考虑微分方程 Equation (A.1.1) 与初始条件 $\mathbf{x}(0)$ 。求解该微分方程需要找到一个连续轨迹 $\mathbf{x}(t)$ ，使其对所有 $t \in [0, T]$ 都满足该方程。理想情况下，期望获得闭式解，但在实际中这很少能实现。

一个有用的视角是将常微分方程改写为积分形式：

$$\mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \mathbf{v}(\mathbf{x}(\tau), \tau)\, \mathrm{d}\tau, \tag{A.1.5}$$

这表示解为初始状态加上速度随时间的累积效应。然而，由于 $\mathbf{v}$ 的非线性和时变特性，积分通常难以处理，导致无法获得闭式解。

在这些情况下，我们转向*数值方法*，它们将时间离散化并迭代近似 $\mathbf{x}(t)$。常见的方法包括欧拉法、龙格-库塔法以及用于刚性系统的专用积分器。这些方法逐步模拟系统，提供真实轨迹的实用近似。

Remark.

> 当 $\mathbf{v}$ 采用 Equation (A.1.3) 中的半线性形式时，其解允许包含指数积分因子 (Equation (A.1.4)) 的积分表示，从而分离线性和非线性分量。这种结构使得数值求解器能专注于近似非线性项，降低计算复杂度并催生了定制化算法（参见 Chapter 9）。

关键概念。 数值求解器通过离散化时间并利用常微分方程的斜率 $\mathbf{v}$ 来估计状态，从而近似常微分方程的连续动态。这包括：

- **离散化**：将时间领域划分为离散步骤 $t_0, t_1, \ldots, t_n$ 。
- **步长**：区间 $\Delta t_i = t_{i+1} - t_i$ 称为步长。
- **近似**：每一步的解都是数值估计的；准确率取决于步长和所用的方法。
- **误差控制**：离散化和近似产生的误差被监控和控制。

数值求解器的高级分类。 常微分方程求解器大致可分为：

- **时间步进方法**：这些方法逐步推进解，例如显式/隐式欧拉法、龙格-库塔法。
- **时间并行方法**：这些方法利用并行性同时计算不同时间区间内的解，适用于大规模问题。

常用的数值求解器。 其中，欧拉法、赫恩法和龙格-库塔法为*单步方法*，因为每次更新仅使用当前状态 $(t_n, \mathbf{x}_n)$。相比之下，*多步方法*（如阿达姆斯-巴什福斯法或阿达姆斯-莫尔顿法）在计算 $\mathbf{x}_{n+1}$ 时不仅使用当前状态 $\mathbf{x}_n$，还使用多个先前的状态值 $\mathbf{x}_{n-1}, \mathbf{x}_{n-2}, \ldots$。这类方法通过复用历史信息（历史锚点）来节省计算量，而非在当前步骤内重新评估所有内容。虽然此处不讨论此类方法，但相关的算法（例如阿达姆斯-巴什福斯法，见 Sections 9.3 and 9.5）同样利用了多个先前状态。

另一方面，Picard 迭代具有不同的性质：它作为一种理论上的不动点构造，其思想将在 Section 9.8 中再次探讨。

欧拉法。 欧拉法是最简单的时间步进格式：

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{v}(\mathbf{x}_n, t_n),$$

其中 h 是步长。它具有一阶准确率：局部误差 $\mathcal{O}(h^2)$ ，全局误差 $\mathcal{O}(h)$ 。虽然实现简单，但为了稳定性和准确率，需要较小的 h 。

赫恩方法（改进欧拉法）。 赫恩方法是一种二阶预测-校正格式：

$$\begin{aligned}\text{Predict:}\quad & \mathbf{x}_{\text{pred}} = \mathbf{x}_n + h\mathbf{v}(\mathbf{x}_n, t_n),\\ \text{Correct:}\quad & \mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}\big(\mathbf{v}(\mathbf{x}_n, t_n) + \mathbf{v}(\mathbf{x}_{\text{pred}}, t_n + h)\big).\end{aligned}$$

它实现了局部误差 $\mathcal{O}(h^3)$ 和全局误差 $\mathcal{O}(h^2)$ 。**karras2022elucidating** 倡导在扩散模型中使用 Heun 方法求解常微分方程，尽管更高阶的方法如 DPM-Solvers（参见 Sections 9.4 and 9.5）通常能取得更好的性能。

龙格-库塔方法。 龙格-库塔（RK）方法通过使用中间斜率的加权平均来泛化欧拉法。四阶方法（RK4）是一种标准选择：

$$\begin{aligned}\mathbf{k}_1 &= \mathbf{v}(\mathbf{x}_n, t_n),\\ \mathbf{k}_2 &= \mathbf{v}(\mathbf{x}_n + \tfrac{h}{2}\mathbf{k}_1, t_n + \tfrac{h}{2}),\\ \mathbf{k}_3 &= \mathbf{v}(\mathbf{x}_n + \tfrac{h}{2}\mathbf{k}_2, t_n + \tfrac{h}{2}),\\ \mathbf{k}_4 &= \mathbf{v}(\mathbf{x}_n + h\mathbf{k}_3, t_n + h),\\ \mathbf{x}_{n+1} &= \mathbf{x}_n + \tfrac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4).\end{aligned}$$

RK4 在准确率和成本之间取得了平衡，因此被广泛使用。DPM-Solver 基于类似的思想，实现了针对扩散模型的高阶准确积分，利用了其线性结构（详见 (**lu2022dpm**) 的附录 B.6 中的对比）。

皮卡迭代。 皮卡迭代通过以下方式对解进行逐次近似：

$$\mathbf{x}^{(k+1)}(t) = \mathbf{x}(0) + \int_0^t \mathbf{v}\big(\mathbf{x}^{(k)}(s), s\big)\,\mathrm{d}s,$$

从一个初始猜测函数 $\mathbf{x}^{(0)}(t)$ 开始，其中 $\mathbf{x}^{(0)}(0) = \mathbf{x}(0)$ 。尽管在理论上具有基础性，但皮卡迭代由于对初始猜测的强烈依赖，通常收敛较慢。此外，每次迭代都需要计算时间上的积分，这可能带来较大的计算开销。

常微分方程的前向与反向求解。 到目前[F]止,我們已經考慮了在 Equation (A.1.1) 時間向前求解常微分方程，從初始條件 $\mathbf{x}(0)$ 演化到後續時間 $t > 0$ 。

相反，反向时间积分通过从终值条件 $\mathbf{x}(T)$ 向更早的时间 $t < T$ 步进来计算解。将时间重新参数化为 $T - t$ 会将常微分方程变换为：

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = -\mathbf{v}\big(\mathbf{x}(t), T - t\big), \qquad \mathbf{x}(0) = \mathbf{x}(T).$$

反向时间积分采用与正向时间积分相同的方法，但在递减的时间网格上进行。使用欧拉法和步长 $h > 0$ ，从 $t_0 = T$ 开始，初始值为 $\mathbf{x}_0 = \mathbf{x}(T)$ ，更新公式为

$$t_{n+1} = t_n - h, \quad \mathbf{x}_{n+1} = \mathbf{x}_n - h\,\mathbf{v}(\mathbf{x}_n, t_n).$$

必须注意确保数值稳定性，尤其是对于刚性问题（即状态向量的某些分量演化速度远快于其他分量，需要非常小的时间步才能实现稳定积分），这在扩散模型的 PF-ODE 采样中很常见。

虽然对于常微分方程（ODEs）而言，时间反演在理论上是直接的，因为只需对时间进行重参数化即可，这得益于 $\mathbf{x}(0)$ 与 $\mathbf{x}(T)$ 之间的双射映射；但对于随机微分方程（SDEs）则不成立。其固有的随机性使得直接的时间反演成为不可能，这一点我们将在下一节中进一步阐述。

A.2 随机微分方程基础

随机微分方程（SDEs）是常微分方程（ODEs）的扩展，引入了随机性，为受不确定性影响的系统建模提供了数学框架。本章介绍了随机微分方程，从常微分方程的离散化开始，延伸到随机微分方程的离散化，最终讨论了一般随机微分方程，包括伊藤微积分和伊藤公式。

A.2.1 从常微分方程到随机微分方程：一种直观入门

让我们从描述状态变量 $\mathbf{x}(t) \in \mathbb{R}^D$ 的确定性演化的一个常微分方程开始：

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}(t), t), \quad \mathbf{x}(0) = \mathbf{x}_0. \tag{A.2.1}$$

此处，$\mathbf{f} : \mathbb{R}^D \times [0, T] \to \mathbb{R}^D$ 是一个随时间变化的速度场，支配着 $\mathbf{x}(t)$ 的动力学行为。该常微分方程的解是一条光滑轨迹 $t \mapsto \mathbf{x}(t)$ ，其完全由初始条件 $\mathbf{x}_0$ 决定。

离散化视角。 为了建立直观认识，考虑对公式 Equation (A.2.1) 在小时间步 Δt 下的欧拉离散化：

$$\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \mathbf{f}(\mathbf{x}_t, t)\Delta t.$$

当 $\Delta t \to 0$ 时，此近似更加精确，在对 $\mathbf{f}$ 施加标准正则性条件的情况下，收敛到常微分方程的确切解。

引入随机性：从常微分方程到随机微分方程。 在许多现实系统中，对动力学的完全掌握是不切实际的。噪声、不确定性或未建模的相互作用可能会影响系统的演化。为了引入这种随机性，我们通过加入一个随机项来扩展常微分方程：

$$\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \mathbf{f}(\mathbf{x}_t, t)\Delta t + g(t)\sqrt{\Delta t} \cdot \boldsymbol{\epsilon}_t, \tag{A.2.2}$$

在哪里

- $g : [0, T] \to \mathbb{R}$ 是一个扩散系数（可能依赖于状态和时间，但此处假设仅依赖于时间），
- $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$ 是独立同分布的标准高斯向量。

此修正后的更新规则不仅反映了确定性漂移，还包含了由 $\sqrt{\Delta t}$ 缩放的随机扰动。缩放确保了在极限 $\Delta t \to 0$ 下，随机扰动保持有限。重要的是，这种表述在 $\Delta t \to 0$ 时产生一个 *连续时间随机过程*，从而引导我们进入随机微分方程（SDE）的框架。

随机微分方程。 形式上，离散更新 Equation (A.2.2) 在 $\Delta t \to 0$ 时的极限定义了随机微分方程（SDE）：

$$\mathrm{d}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}(t). \tag{A.2.3}$$

此处，$\mathbf{w}(t) \in \mathbb{R}^D$ 为一个 *维纳过程*（标准布朗运动），是一种连续时间随机过程，其特征为：

- **初始状态**： $\mathbf{w}(0) = 0$ 几乎必然；
- **独立增量**：对于 $0 \le s < t$ ，增量 $\mathbf{w}(t) - \mathbf{w}(s)$ 与过去相互独立；
- **高斯增量**：

$$\mathbf{w}(t) - \mathbf{w}(s) \sim \mathcal{N}\big(\mathbf{0}, (t-s)\mathbf{I}_D\big) \tag{A.2.4}$$

- **连续性**：样本路径 $t \mapsto \mathbf{w}(t)$ 几乎必然连续但处处不可微。

此外，记号

$$\mathrm{d}\mathbf{w}(t) := \mathbf{w}(t + \mathrm{d}t) - \mathbf{w}(t)$$

常用于表示维纳过程的无穷小增量。

虽然具有启发性，但这种记号是启发式的，不应被解释为经典的微分（例如黎曼或勒贝格意义下的微分），因为布朗路径几乎必然处处不可微。相反，它作为一种形式化的简写，用来表达高斯增量性质：

$$\mathrm{d}\mathbf{w}(t) \sim \mathcal{N}(0, \mathrm{d}t\,\mathrm{I}_D),$$

这意味着，在长度为 $\mathrm{d}t$ 的无穷小时间间隔内，维纳过程的增量表现得像一个零均值且协方差为 $\mathrm{d}t\,\mathrm{I}_D$ 的高斯随机变量。

A.2.2 Equation (A.2.3) 的進一步🄵明

方程 Equation (A.2.3) 中的随机微分方程应理解为其 *积分形式*：

$$\mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \mathbf{f}(\mathbf{x}(s), s)\,\mathrm{d}s + \int_0^t g(s)\,\mathrm{d}\mathbf{w}(s), \tag{A.2.5}$$

以 *Itô 意义*解释。其中，第一项是一个经典的（黎曼或勒贝格）积分，表示累积的确定性漂移，而第二项是一个 *Itô 随机积分*，它相对于维纳过程 $\mathbf{w}(t)$ 进行积分。我们不提供 Itô 积分的完整严格构造，但给出以下直观理解。

伊藤积分的直觉理解。 伊藤积分可被视为离散和的极限（在概率意义下）：

$$\sum_i g(t_i)\big(\mathbf{w}(t_{i+1}) - \mathbf{w}(t_i)\big),$$

其中被积函数 $g(t)$ 在每个子区间的 *左端点* t_i 处求值。这种左端点求值至关重要，它将伊藤积分与经典积分区分开来，后者通常使用中点或其他求值规则。

由于布朗路径连续但几乎必然处处不可微，经典积分不再适用。伊藤积分能够处理这种不规则性，捕捉随机波动随时间累积的效果。

微分符号的使用。 像 $\mathrm{d}\mathbf{x}(t)$ 、$\mathrm{d}t$ 和 $\mathrm{d}\mathbf{w}(t)$ 这样的表达式并不是经典的微分，而是表示相应过程的无穷小增量的形式记号。尽管这些记号具有启发性，但由于其在将随机微分方程（SDEs）以类似于常微分方程（ODEs）的方式表达时的便利性，以及在伊藤微积分中促进形式运算的能力，它们被广泛使用。

Itô 微积分在扩散模型中的应用将在 Chapter C中解释。

与常微分方程的比较。 在常微分方程中，例如

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}(t), t),$$

积分形式

$$\mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \mathbf{f}(\mathbf{x}(\tau), \tau)\,\mathrm{d}\tau$$

由*微积分基本定理*所支持，该定理保证了可微函数可以从其导数中恢复。

相比之下，在如 Equation (A.2.3) 所示的随机微分方程中，由于布朗运动不具备可微性，且随机积分不遵循经典的链式法则，因此该定理没有直接类比。相

反，伊藤微积分引入了替代工具（例如伊藤引理）来分析和处理随机动力学。

因此，尽管随机微分方程的微分形式简洁且直观，但要严格理解它们，必须通过使用伊藤积分的积分形式来解释。

A.2.3 随机微分方程的数值求解器。

与常微分方程类似，公式 Equation (A.2.3) 中的随机微分方程存在唯一解[1] 若 $\mathbf{f}(\cdot, t)$ 和 $g(\cdot)$ 满足某些光滑性条件：$\mathbf{f}(\cdot, t)$ 在 $\mathbf{x}$ 上为 Lipschitz 且具有线性增长性，而 $g(\cdot)$ 为平方可积。

对于一般的随机微分方程（SDEs）如 Equation (A.2.3) 所示，闭式解通常不可得，因此需要采用数值方法。一种常用的方法是 *Euler–Maruyama 方法*，该方法将常微分方程（ODEs）的 Euler 方法进行泛化，实际上我们已在 Equation (A.2.2) 中见过它。该方法在时间步 Δt 内对漂移项 $\mathbf{f}(\mathbf{x}(t), t)$ 进行近似，并利用均值为 $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$、方差为 $\sqrt{\Delta t}\,\boldsymbol{\epsilon}_t$ 的高斯增量来模拟随机噪声 $g(t)\,\mathrm{d}\mathbf{w}(t)$。

稍后，在 Section C.1.5 中，我们将看到线性随机微分方程（SDE）存在闭式解。

[1] 解是以强意义给出的，这意味着 $\mathbf{x}(t)$ 以给定的布朗运动 $\mathbf{w}(t)$ 在固定概率空间上满足随机微分方程的积分形式（见 Equation (A.2.5)）。此处省略详细的技术定義。

B

密度演化：从变量变换到福克-普朗克方程

理解概率密度在变换下的演化过程，在概率论和生成式建模中具有基础性意义。特别是，扩散模型旨在构建生成过程，其诱导的概率路径能够逆转预先定义的前向过程。这一演化过程由连续性方程控制，或在随机情况下由福克-普朗克方程控制。

尽管这些名称听起来可能不太熟悉或令人望而生畏，但实际上它们是微积分中变量替换公式的连续时间类比。在 Section B.1 中，通过逐步展示一系列变量替换公式，从确定性双射出发，最终过渡到随机微分方程，从而引出这些概念。这一渐进过程自然地连接了离散映射与连续时间流动力学。参见 Figure B.1 以了解这一统一框架的概览。

在 Section B.2 中，我们提供了连续性方程的物理和直观解释，强调了其与动力系统中密度守恒的关系。

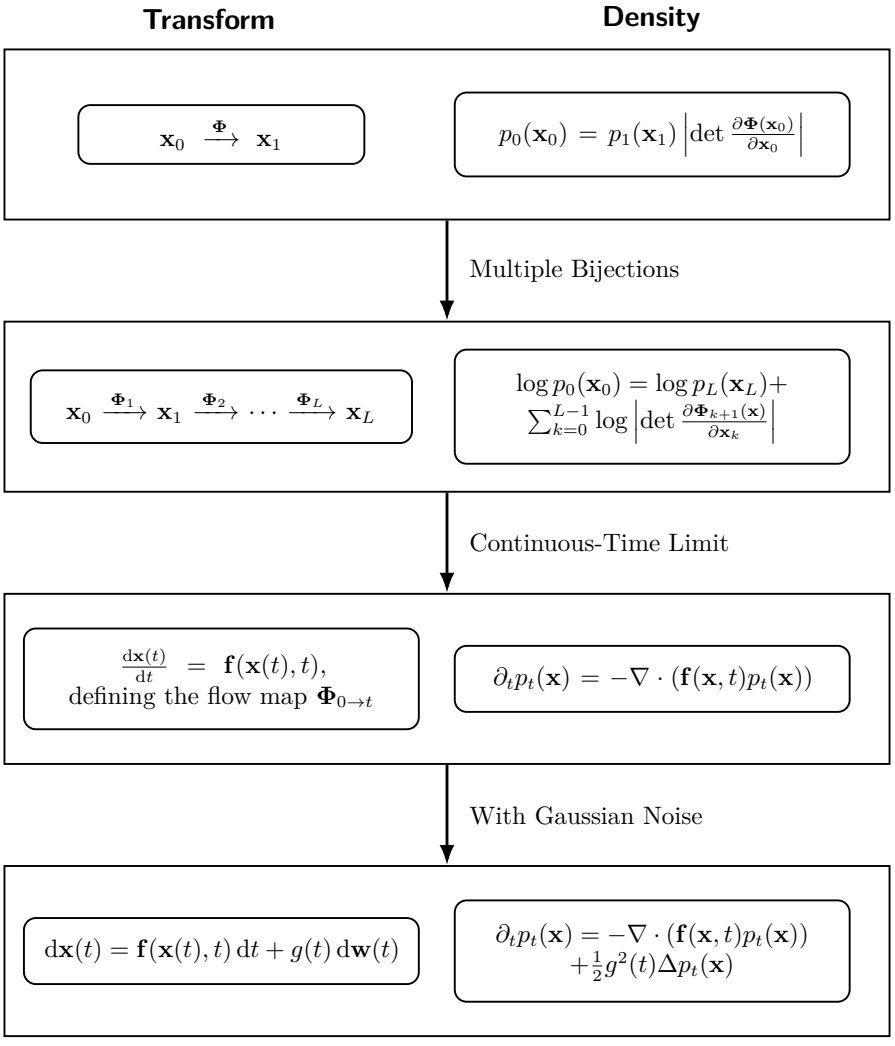


图 B.1: 一个统一的变量变换公式。从上到下：(1) 一个双射及其；(2) 多个双射的复合；(3) 由常微分方程控制的连续时间确定性流及其对应的连续性方程；(4) 由随机微分方程建模的随机流及其对应的福克-普朗克方程。

B.1 变量变换公式：从确定性映射到随机流

在本节中，我们通过类比微积分中的经典变量变换公式，旨在阐明连续性方程和福克-普朗克方程。我们从熟悉的单变量情形出发，将其推广到多变量情形以及概率密度 (Section B.1.1)，然后进一步推广到双射映射的复合情形，其连续时间极限即为连续性方程 (Section B.1.2)。最后，通过引入随机噪声来纳入随机

性，自然地将连续性方程推广为福克-普朗克方程 (Section B.1.3)。

B.1.1 确定性映射的变量替换公式

我们根据一个确定性映射移动粒子，并研究其分布（密度）如何演化。关键原理是概率质量的守恒，这基于微积分和概率论中的一个基本结果：变量变换公式。该公式描述了在光滑双射映射下，积分（因此概率密度）如何变换。为了建立直观理解，我们首先考虑单步更新，然后将讨论扩展到序列变换。

单次更新。 考虑通过施加一个向量场（类似于力）$\mathbf{\Psi} : \mathbb{R}^D \to \mathbb{R}^D$ 作用一个单元时间所诱导的单个更新规则。从初始粒子状态 $\mathbf{x}_0$ 出发，其下一个状态由下式给出

$$\mathbf{x}_1 = \mathbf{\Psi}(\mathbf{x}_0).$$

潜在模式（一种密度）及其运动方式。 如果初始状态遵循由密度 p_0 描述的潜在“规律/模式”（即 $\mathbf{x}_0 \sim p_0$ ），那么应用 $\mathbf{\Psi}$ 会为 $\mathbf{x}_1$ 产生一个新的密度 p_1 （即 $\mathbf{x}_1 \sim p_1$ ）。假设 $\mathbf{\Psi}$ 是一个光滑双射，p_1 可通过标准 变量变换公式从 p_0 得到：

$$p_1(\mathbf{x}_1) = p_0(\mathbf{\Psi}^{-1}(\mathbf{x}_1)) \cdot \left| \det \left(\frac{\partial \mathbf{\Psi}^{-1}}{\partial \mathbf{x}_1} \right) \right|. \tag{B.1.1}$$

此处 $\frac{\partial \mathbf{\Psi}}{\partial \mathbf{x}}$ 为 $\mathbf{\Psi}$ 的雅可比矩阵，记为 $\partial_{\mathbf{x}} \mathbf{\Psi}$。等价地，在原始坐标下，

$$p_0(\mathbf{x}_0) \;=\; p_1\big(\mathbf{\Psi}(\mathbf{x}_0)\big)\, \big|\det \partial_{\mathbf{x}} \mathbf{\Psi}(\mathbf{x}_0)\big|.$$

用文字描述，$\mathbf{\Psi}$ 将密度 p_0 重塑为 p_1。因子 $\left|\det \partial_{\mathbf{x}} \mathbf{\Psi}\right|$ 表示体积的局部变化；由于概率质量守恒，密度通过其逆来补偿。

作为一个简单的情况，如果 $\mathbf{\Psi}$ 是线性的且矩阵 $\mathbf{A}$ 可逆（即 $\mathbf{x}_1 = \mathbf{A}\mathbf{x}_0$ ），那么

$$p_1(\mathbf{x}_1) \;=\; p_0(\mathbf{A}^{-1}\mathbf{x}_1)\, \big|\det \mathbf{A}^{-1}\big|.$$

从示意图上看，我们可以将其理解为：

$$\begin{array}{lccc} \textbf{Sample:} & \mathbf{x}_0 & \xrightarrow{\mathbf{\Psi}} & \mathbf{x}_1 \\ \textbf{Density:} & p_{\mathbf{x}_0}(\mathbf{x}_0) & \xrightarrow{\mathbf{\Psi}} & p_{\mathbf{x}_1}(\mathbf{x}_1) \end{array}$$

为什么 Equation (B.1.1) 是变量替换公式？ 这直接来自于微积分中的熟知法则。

单变量情形。 设 $y = \Psi(x)$ 光滑且可求逆。将 y 上的积分用 x 表示，得到

$$\int g(y)\,\mathrm{d}y = \int g(\Psi(x)) \cdot |\Psi'(x)|\,\mathrm{d}x,$$

其中 $|\Psi'(x)|$ 用于补偿区间拉伸或压缩，确保面积不变。

多变量情形。 对于 $\mathbf{\Psi} : \mathbb{R}^D \to \mathbb{R}^D$ 且 $\mathbf{y} = \mathbf{\Psi}(\mathbf{x})$ ，

$$\int g(\mathbf{y})\,\mathrm{d}\mathbf{y} = \int g(\mathbf{\Psi}(\mathbf{x}))\,\big|\det(\partial_{\mathbf{x}}\mathbf{\Psi})\big|\,\mathrm{d}\mathbf{x},$$

因此，无穷小体积的变换为

$$\mathrm{d}\mathbf{y} = \big|\det(\partial_{\mathbf{x}}\mathbf{\Psi})\big|\,\mathrm{d}\mathbf{x}.$$

由此，可得公式 Equation (B.1.1) 中的密度公式：

$$\begin{aligned} p_{\mathbf{y}}(\mathbf{y}) &= \int_{\mathbb{R}^D} \delta(\mathbf{y} - \mathbf{\Psi}(\mathbf{x}))\, p_{\mathbf{x}}(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &= p_{\mathbf{x}}\big(\mathbf{\Psi}^{-1}(\mathbf{y})\big)\,\left|\det\left(\frac{\partial \mathbf{\Psi}^{-1}}{\partial \mathbf{y}}\right)\right|. \end{aligned}$$

组合多个双射。 我们现在按顺序应用几次更新。令 $\mathbf{x}_k = \mathbf{\Psi}_k(\mathbf{x}_{k-1})$ 为 $k = 1, \ldots, L$ ；即，

$$\mathbf{x}_0 \xrightarrow{\mathbf{\Psi}_1} \mathbf{x}_1 \xrightarrow{\mathbf{\Psi}_2} \cdots \xrightarrow{\mathbf{\Psi}_L} \mathbf{x}_L,$$

其中每个 $\mathbf{\Psi}_k : \mathbb{R}^D \to \mathbb{R}^D$ 都是一个光滑双射。如果初始状态服从密度 p_0 （即 $\mathbf{x}_0 \sim p_0$ ），那么更新序列诱导出密度 $p_1, \ldots, p_L$ ，其中 $\mathbf{x}_1, \ldots, \mathbf{x}_L$ 。

由于每一步的概率质量都守恒，密度的演化遵循

$$p_k(\mathbf{x}_k) \;=\; p_{k-1}(\mathbf{x}_{k-1})\,\Big|\det \partial_{\mathbf{x}_{k-1}} \mathbf{\Psi}_k(\mathbf{x}_{k-1})\Big|^{-1}, \quad k = 1, \ldots, L.$$

通过递归，$\mathbf{x}_L$ 处的最终密度为

$$p_{\mathbf{x}_L}(\mathbf{x}_L) = p_{\mathbf{x}_0}(\mathbf{x}_0) \cdot \prod_{k=1}^{L} \left| \det \left(\frac{\partial \mathbf{\Psi}_k}{\partial \mathbf{x}_{k-1}} \right) \right|^{-1}. \tag{B.1.2}$$

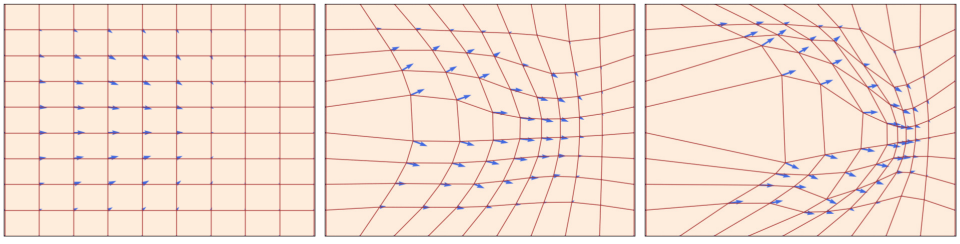
等价地，以对数密度形式：

$$\log p_{\mathbf{x}_L}(\mathbf{x}_L) = \log p_{\mathbf{x}_0}(\mathbf{x}_0) - \sum_{k=1}^{L} \log \left| \det \left(\frac{\partial \mathbf{\Psi}_k}{\partial \mathbf{x}_{k-1}} \right) \right|.$$

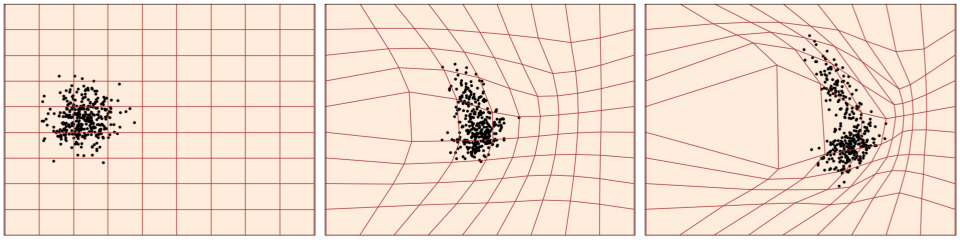
这个表达式反映了每个变换 $\mathbf{\Psi}_k$ 如何通过雅克比行列式来拉伸或压缩体积。沿着变换路径累积的这些局部体积变化决定了复合映射下的最终概率密度。

Equation (B.1.2) 作为归一化流的潜在核心原则（参见 Section 5.1.2）。

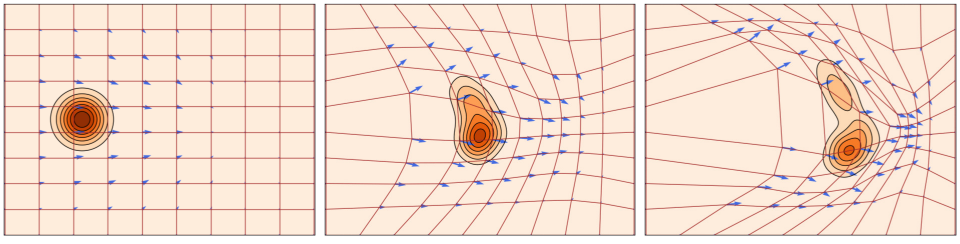
B.1.2 连续时间极限：连续性方程



(a) 向量场图示。箭头表示会拖动粒子在空间中移动的力，从而相应地扭曲潜在的网格。



(b) 粒子云动力学。预定义的向量场（被解释为力）产生流动，将粒子从其初始状态传输。



(c) 密度演化。随着粒子被向量场输运，密度等值线随之变形，反映了流动如何重塑潜在分布。

图 B.2: 向量场下粒子与密度演化的示意图。每一列显示连续的时间快照（从左到右）。这些示意图改编自 **lipman2024flow**，经作者许可使用。

我们现在从离散更新转向连续描述。假设粒子运动由一个随时间变化的速度场 $\mathbf{f} : \mathbb{R}^D \times [0, T] \to \mathbb{R}^D$ 驱动。设想通过无穷多个微小的双射更新来演化一个粒子 $\mathbf{x}_0 \sim p_0$。在每个长度为 $\Delta t > 0$ 的步骤 t 中，更新为

$$\mathbf{x}_{t+\Delta t} = \boldsymbol{\Psi}(\mathbf{x}_t) := \mathbf{x}_t + \Delta t\, \mathbf{f}(\mathbf{x}_t, t).$$

当 $\Delta t \to 0$ 时，这些更新的组合收敛到由速度场 $\mathbf{f} : \mathbb{R}^D \times [0, T] \to \mathbb{R}^D$ 所控制的连续流：

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}(t), t), \qquad \mathbf{x}(0) = \mathbf{x}_0 \sim p_0. \tag{B.1.3}$$

在适当的光滑性假设下（见 Chapter A），该常微分方程对每个初值条件都存在唯一解，这定义了一个确定性流映射 $\mathbf{\Psi}_{0\to t} : \mathbb{R}^D \to \mathbb{R}^D$ 。换句话说，$\mathbf{\Psi}_{0\to t}$ 将初始状态 $\mathbf{x}_0$ 映射到 Equation (B.1.3) 在时刻 t 的解：

$$\mathbf{\Psi}_{0\to t}(\mathbf{x}_0) = \mathbf{x}_0 + \int_0^t \mathbf{f}(\mathbf{x}(\tau), \tau)\, \mathrm{d}\tau.$$

因此，整个分布也发生了移动：初始密度 p_0 被传输为新的密度 p_t ，即 $\mathbf{x}(t)$ 的规律。形式上，这被写作一个 *前推*：

$$p_t = \left(\mathbf{\Psi}_{0\to t}\right)_{\#} p_0.$$

当 $\mathbf{\Psi}_{0\to t}$ 光滑且可求逆时，这简化为熟悉的变量变换规则：

$$p_t(\mathbf{x}) = p_0\big(\mathbf{\Psi}_{t\to 0}(\mathbf{x})\big)\, \big|\det \partial_{\mathbf{x}} \mathbf{\Psi}_{t\to 0}(\mathbf{x})\big| = \int \delta(\mathbf{x} - \mathbf{\Psi}_{t\to 0}(\mathbf{x}_0))\, p_0(\mathbf{x}_0)\, \mathrm{d}\mathbf{x}_0.$$

连续性方程：密度随时间的演变。 而不是为每个时间点写出密度的分离公式，我们可以使用空间 $\mathbf{x}$ 和时间 t 的微分方程来连续描述其运动。其思想很简单：概率质量是守恒的，速度场 $\mathbf{f}$ 仅在空间中重新分布它。这给出了 *连续性方程*：

$$\frac{\partial}{\partial t} p_t(\mathbf{x}) + \nabla \cdot \big(p_t(\mathbf{x})\, \mathbf{f}(\mathbf{x}, t)\big) = 0. \tag{B.1.4}$$

此处的散度项 $\nabla \cdot (p_t \mathbf{f})$ 用于衡量流在局部对密度的扩张或压缩程度，确保总概率保持 1。

这个偏微分方程（PDE）确保了在流动过程中粒子移动时概率质量得以保持。实际上，它可以被视为变量变换公式的连续时间类比。

通过变量变换公式推导连续性方程。 从概念上讲，连续性方程也可以通过取 Equation (B.1.2) 的连续时间极限得到。然而，这里我们采用基于 Equation (B.1.1) 的更直接的推导方法。

离散化与变量变换公式。 考虑

$$\mathbf{x}_{t+\Delta t} := \mathbf{\Psi}(\mathbf{x}_t) = \mathbf{x}_t + \Delta t\, \mathbf{f}(\mathbf{x}_t, t),$$

这实际上是 Equation (B.1.3) 中常微分方程在小时间间隔 $\Delta t > 0$ 上的前向欧拉离散化。映射 $\boldsymbol{\Psi}$ 相对于 $\mathbf{x}_t$ 的雅克比矩阵展开为

$$\frac{\partial \boldsymbol{\Psi}}{\partial \mathbf{x}_t} = \mathbf{I} + \Delta t \nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}_t, t) + \mathcal{O}(\Delta t^2),$$

因此，它的行列式满足

$$\det\left(\frac{\partial \boldsymbol{\Psi}}{\partial \mathbf{x}_t}\right) = 1 + \Delta t \, \nabla \cdot \mathbf{f}(\mathbf{x}_t, t) + \mathcal{O}(\Delta t^2).$$

这使用了标准展开 $\det(\mathbf{I} + \Delta t \, \mathbf{A}) = 1 + \Delta t \, \operatorname{Tr}(\mathbf{A}) + \mathcal{O}(\Delta t^2)$ 作为 $\Delta t \to 0$ ，以及 $\nabla \cdot \mathbf{f} = \operatorname{Tr}(\nabla_{\mathbf{x}} \mathbf{f})$ 。

应用变量变换公式，对数密度的演化为

$$\log p_{t+\Delta t}(\mathbf{x}_{t+\Delta t}) = \log p_t(\mathbf{x}_t) - \Delta t \, \nabla \cdot \mathbf{f}(\mathbf{x}_t, t) + \mathcal{O}(\Delta t^2).$$

应用变量变换公式，对数密度的演化为

$$\log p_{t+\Delta t}(\mathbf{x}_{t+\Delta t}) = \log p_t(\mathbf{x}_t) - \Delta t \, \nabla \cdot \mathbf{f}(\mathbf{x}_t, t) + \mathcal{O}(\Delta t^2).$$

也就是说，

$$\log p_{t+\Delta t}(\mathbf{x}_{t+\Delta t}) - \log p_t(\mathbf{x}_t) = -\Delta t \, \nabla \cdot \mathbf{f}(\mathbf{x}_t, t) + \mathcal{O}(\Delta t^2). \tag{B.1.5}$$

使用泰勒展开。 现在，我们通过多元泰勒展开来展开左边：

$$\begin{aligned}
&\log p_{t+\Delta t}(\mathbf{x}_{t+\Delta t}) - \log p_t(\mathbf{x}_t) \\
=&\Delta t \, \partial_t \log p_t(\mathbf{x}_t) + (\mathbf{x}_{t+\Delta t} - \mathbf{x}_t)^\top \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \mathcal{O}(\Delta t^2).
\end{aligned}$$

代入 $\mathbf{x}_{t+\Delta t} - \mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\Delta t$ 得：

$$\begin{aligned}
&\log p_{t+\Delta t}(\mathbf{x}_{t+\Delta t}) - \log p_t(\mathbf{x}_t) \\
=&\Delta t \, \partial_t \log p_t(\mathbf{x}_t) + \Delta t \, \mathbf{f}(\mathbf{x}_t, t)^\top \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \mathcal{O}(\Delta t^2).
\end{aligned}$$

将各项与 Equation (B.1.5) 匹配并令 $\Delta t \to 0$ ，我们得出

$$\partial_t \log p_t(\mathbf{x}_t) = -\nabla_{\mathbf{x}_t} \cdot \mathbf{f}(\mathbf{x}_t, t) - \mathbf{f}(\mathbf{x}_t, t)^\top \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t).$$

取幂并应用乘法法则可得到连续性方程。

速度优先（拉格朗日）与密度优先（欧拉） 需要注意的是，粒子动力学与密度动力学之间存在一个关键的不对称性。从速度场出发，可以唯一确定粒子的流动，从而唯一确定密度的演化过程。相反，仅规定密度的路径并不能唯一确定速度场：许多不同的流动过程都可能导致相同的密度序列。

速度优先（欧拉：流 $\Rightarrow$ 密度）。 到目前为止，我们假设速度场 $\mathbf{f}$ 是已知的。粒子常微分方程

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}(t), t)$$

描述每个粒子的运动方式，而密度则由偏微分方程（PDE）描述。

$$\partial_t p_t + \nabla \cdot (p_t \mathbf{f}_t) = 0$$

描述了粒子分布整体演化的过程。这两种观点是相互关联的：根据常微分方程（ODE）移动粒子，会自动产生满足偏微分方程（PDE）的密度。在这种情况下，粒子流 $\boldsymbol{\Psi}_{0\to t}$ 是唯一确定的：从 $\mathbf{x}(0) \sim p_0$ 出发，每条轨迹 $\mathbf{x}(t)$ 均被固定，所得到的密度 p_t 满足连续性方程。在此情形下，粒子动力学与密度动力学完全一致。

密度优先（欧拉描述：密度 $\not\Rightarrow$ 唯一流）。 如果我们仅从密度路径 $t \mapsto p_t$ 开始（例如，在流匹配中的 Section 5.3.2），则速度场不再唯一确定。例如，若向量场 $\mathbf{w}_t$ 满足

$$\nabla_{\mathbf{x}} \cdot \big(p_t(\mathbf{x})\, \mathbf{w}_t(\mathbf{x})\big) = 0 \qquad (\text{no net flux w.r.t. } p_t),$$

因此，$\mathbf{f}_t$ 和 $\mathbf{f}_t + \mathbf{w}_t$ 都会引发相同的密度演化。因此，一条单一的密度路径可能对应多种不同的流，选择某一条特定的粒子流 $\boldsymbol{\Psi}_{0\to t}$ 相当于在这些可能性中选定一个特定的速度场。

并非每一条给定的路径 p_t 都能由在某个速度场作用下运动的粒子产生。连续性方程 (Equation (B.1.4)) 提供了密度路径能否被“流体生成”的一致性检验。我们称 p_t 为 *可实现的*（或由 $\mathbf{f}$ *生成*），如果存在一个速度场 $\mathbf{f}$，使得遵循该速度场的粒子

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}(t), t)$$

通过流映射 $\boldsymbol{\Psi}_{0\to t}$ 产生确切的密度 p_t。也就是说，当 p_t 和 $\mathbf{f}$ 一起满足 Equation (B.1.4) 时，可实现性成立。

直观上，可实现性意味着 p_t 随时间的快照可以由在某个速度场下运动的粒子来解释，而不是任意的分布序列。

当此条件成立时，密度 p_t 无非是初始密度 p_0 沿流映射 $\boldsymbol{\Psi}_{0\to t}$ 的推前像。在此情况下，熟悉的变量变换公式适用：

$$\begin{aligned} p_t &= \left(\boldsymbol{\Psi}_{0\to t}\right)_{\#} p_0 \\ &= p_0\big(\boldsymbol{\Psi}_{t\to 0}(\mathbf{x})\big)\,\big|\det \partial_{\mathbf{x}} \boldsymbol{\Psi}_{t\to 0}(\mathbf{x})\big| \\ &= \int \delta(\mathbf{x}-\boldsymbol{\Psi}_{t\to 0}(\mathbf{x}_0))\, p_0(\mathbf{x}_0)\, \mathrm{d}\mathbf{x}_0. \end{aligned}$$

（可选）调理。 如果引入一个额外的条件变量 $\mathbf{z}\sim\pi(\mathbf{z})$，那么对于每个固定的 $\mathbf{z}$，同样的推理依然适用：

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{v}_t(\mathbf{x}(t)|\mathbf{z})$$

带有前推 $p_t(\cdot|\mathbf{z}) = (\boldsymbol{\Psi}_{0\to t}(\cdot;\mathbf{z}))_{\#}p_0$，以及连续性方程

$$\partial_t p_t(\mathbf{x}|\mathbf{z}) + \nabla\cdot\big(p_t(\mathbf{x}|\mathbf{z})\,\mathbf{v}_t(\mathbf{x}|\mathbf{z})\big) = 0$$

边际密度为

$$p_t(\mathbf{x}) = \int p_t(\mathbf{x}|\mathbf{z})\pi(\mathbf{z})\,\mathrm{d}\mathbf{z}.$$

B.1.3 随机过程：福克-普朗克方程

当加入噪声时，动力学遵循如 Equation (A.2.3): 所示的随机微分方程。

$$\mathrm{d}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t),t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}(t).$$

然后，密度 $p_t(\mathbf{x})$ 满足福克-普朗克方程：

$$\begin{aligned} \frac{\partial p_t(\mathbf{x})}{\partial t} &= -\nabla\cdot\left(\mathbf{f}(\mathbf{x},t)\,p_t(\mathbf{x})\right) + \frac{1}{2}g^2(t)\,\Delta p_t(\mathbf{x}) \\ &= -\nabla\cdot\left(\left(\mathbf{f}(\mathbf{x},t) - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}}\log p_t(\mathbf{x})\right)p_t(\mathbf{x})\right). \end{aligned}$$

此处，$\Delta p_t = \nabla \cdot \nabla_{\mathbf{x}} p_t$ 为拉普拉斯算子。此处，第一项描述了由确定性漂移 $\mathbf{f}$ 引起的概率质量输运，而第二项则建模了由于方差与 $\frac{1}{2}g^2(t)$ 成比例的随机噪声导致的密度扩散（弥散）。

Fokker–Planck 方程的推导更为复杂；我们将其参考至 Section C.1.4。

B.2 连续性方程的直观理解

在本节中，我们对连续性方程给出一个物理解释，突出其作为动力系统中概率密度守恒定律的作用。

B.2.1 连续性方程的物理意义

考虑三维空间中一个微小的固定控制体（一个矩形盒子），其位于 $\mathbf{x} = (x, y, z)$ 处，边长分别为 Δx 、Δy 和 Δz 。设 $p(\mathbf{x}, t)$ 表示在位置 $\mathbf{x}$ 和时间 t 处某种守恒量（例如质量或概率）的密度。该盒子内部该量的总量为：

$$\text{Total quantity in box} = p(\mathbf{x}, t)\Delta x\Delta y\Delta z.$$

总量如何变化？ 系统的总量变化只能源于穿过盒子边界的通量。令 $\mathbf{j}(\mathbf{x}, t)$ 表示通量向量，表示单位面积上单位时间内流动的量。

***x* 方向的通量。** 左侧表面（在 x 处）的流入量约为：

$$j_x(x, y, z, t)\Delta y\Delta z,$$

且通过右侧面（在 $x + \Delta x$ 处）的流出量为：

$$j_x(x + \Delta x, y, z, t)\Delta y\Delta z.$$

因此，x 方向的净通量为：

$$\left[j_x(x, y, z, t) - j_x(x + \Delta x, y, z, t)\right]\Delta y\Delta z.$$

所有方向的净通量。 在 y 和 z 方向上也会出现类似的术语：

$$\left[j_y(x, y, z, t) - j_y(x, y + \Delta y, z, t)\right]\Delta x\Delta z,$$
$$\left[j_z(x, y, z, t) - j_z(x, y, z + \Delta z, t)\right]\Delta x\Delta y.$$

将所有贡献相加，方框的总净流出量为：

$$-\nabla \cdot \mathbf{j}(\mathbf{x}, t)\Delta x\Delta y\Delta z.$$

盒内的变化率。 盒内总量的变化率为：

$$\frac{\partial p}{\partial t}(\mathbf{x},t)\Delta x\Delta y\Delta z.$$

守恒定律 假设该量守恒（例如，总质量或概率不随时间变化），则变化率等于净外流的负值：

$$\frac{\partial p}{\partial t}(\mathbf{x},t)\Delta x\Delta y\Delta z=-\nabla\cdot\mathbf{j}(\mathbf{x},t)\Delta x\Delta y\Delta z.$$

局部形式。 消去公共体积因子（对任意小立方体均有效），得到连续性方程的局部形式：

$$\frac{\partial p}{\partial t}+\nabla\cdot\mathbf{j}=0.$$

B.2.2 由守恒定律推导连续性方程

连续性方程形式化了动力系统中某种物理量（如质量或电荷）的守恒。设 $p(\mathbf{x},t)$ 表示在位置 $\mathbf{x}\in\mathbb{R}^D$ 和时间 $t\in[0,T]$ 时守恒量的密度，令 $\mathbf{v}(\mathbf{x},t)$ 表示速度场。

第一步：控制体内的变化率。 考虑任意控制体 $V\subset\mathbb{R}^D$，其边界为 ∂V。控制体 V 内守恒量的总量为

$$\int_V p(\mathbf{x},t)\,\mathrm{d}V,$$

其时间导数给出了累积速率：

$$\frac{\partial}{\partial t}\int_V p(\mathbf{x},t)\,\mathrm{d}V.$$

第二步：通过边界净通量。 该量通过 V 从 ∂V 流出，其外法向量为 $\mathbf{n}$ 。净向外通量为

$$\int_{\partial V} p(\mathbf{x},t)\mathbf{v}(\mathbf{x},t)\cdot\mathbf{n}\,\mathrm{d}S.$$

第三步：守恒定律。 守恒律表明，V 内的累积速率为净向外通量的负值：

$$\frac{\partial}{\partial t}\int_V p\,\mathrm{d}V+\int_{\partial V} p\mathbf{v}\cdot\mathbf{n}\,\mathrm{d}S=0.$$

步骤 4：散度定理。 应用散度定理将曲面积分转化为体积分：

$$\int_{\partial V} p\mathbf{v}\cdot\mathbf{n}\,\mathrm{d}S=\int_V \nabla\cdot(p\mathbf{v})\,\mathrm{d}V.$$

因此，

$$\frac{\partial}{\partial t}\int_V p\,\mathrm{d}V+\int_V \nabla\cdot(p\mathbf{v})\,\mathrm{d}V=0.$$

第五步：局部形式。 由于控制体 V 是任意的，被积函数必须逐点消失。这得到了连续性方程。

C

扩散模型背后的原理：伊藤微积分与吉尔萨诺夫定理

基于得分的扩散模型建立在随机微分方程（SDE）之上：一个推动状态演化的漂移项，以及一个使状态产生扰动的布朗运动项。与常微分方程（ODE）路径不同，布朗运动路径处处不可导，因此普通的链式法则失效。在本节中，我们介绍两个使数学表达精确的基本工具：

- **Itô 公式** 是随机轨迹的正确链式法则。它告诉我们，当 $\mathbf{x}_t$ 遵循一个 SDE 时，函数 $\mathbf{h}(\mathbf{x}_t, t)$ 如何演化。它使得福克-普朗克方程、矩动力学、Itô 乘法法则以及得分模型训练中所用恒等式的推导成为可能。
- **Girsanov 定理** 是关于 *路径概率* 的测度变换结果。它量化了当噪声固定但漂移改变时，似然如何变化。这将分数匹配与路径空间的 KL 散度联系起来，并解释了为何在反向 SDE 中学习得分等价于最大化数据似然。

借助这些工具，标准扩散模型的推导（福克-普朗克方程、反向时间随机微分方程、训练目标和似然关系）能够清晰且严谨地展开，无需模糊处理。

C.1 伊藤公式：随机过程的链式法则

标准微积分不能直接应用于随机过程，因为维纳过程在经典意义上不可微。相反，我们使用伊藤微积分，它提供了处理随机积分的规则。

C.1.1 动机：为什么我们需要一种特殊的链式法则？

考虑一个随时间 t 平滑变化的确定性时变函数 $\mathbf{y}_t$（例如，一个常微分方程）。如果我们有一个函数 $\mathbf{h}(\mathbf{y}_t, t)$ ，通常的链式法则告诉我们：

$$\frac{\mathrm{d}\mathbf{h}}{\mathrm{d}t} = \frac{\partial \mathbf{h}}{\partial t} + \nabla_{\mathbf{y}}\mathbf{h}\frac{\mathrm{d}\mathbf{y}_t}{\mathrm{d}t}.$$

此处，$\nabla_{\mathbf{y}}\mathbf{h}$ 为 $\mathbf{h}$ 的雅克比。对于确定性路径 $\mathbf{y}_t$，此方法完全适用。

Question C.1.1

但如果 $\mathbf{x}_t$ 是一个随机过程，例如由一个随机微分方程（*SDE*）驱动的随机过程，会发生什么情况？

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t$$

如 *Equation* (A.2.3) 所示，过程 $\mathbf{h}(\mathbf{x}_t, t)$ 满足何种 *SDE*？

为什么普通的链式法则会失效？ 天真地应用经典链式法则得到

$$\mathrm{d}\mathbf{h} = \frac{\partial \mathbf{h}}{\partial t}\,\mathrm{d}t + \nabla_{\mathbf{y}}\mathbf{h} \cdot \mathrm{d}\mathbf{x}_t.$$

然而，这忽略了布朗运动增量满足 $\mathrm{d}\mathbf{w}_t = \mathcal{O}(\sqrt{\mathrm{d}t})$ 且

$$(\mathrm{d}\mathbf{w}_t)^2 = \mathrm{d}t.$$

因此，在随机微积分中，$\mathrm{d}\mathbf{w}_t$ 的二阶项不会消失，而与之不同的是，在经典微积分中，$(\mathrm{d}t)^2$ 项可以忽略不计。

Example: Simple Example–$h(x_t) = x_t^2$

To see the intuition, let us consider the simple real-valued function $h(x_t) = x_t^2 \in \mathbb{R}$ where the random variable $x_t \in \mathbb{R}$ satisfies

$$\mathrm{d}x_t = \sigma \, \mathrm{d}w_t,$$

with a constant $\sigma > 0$. If we try the classical chain rule,

$$\mathrm{d}h = 2x_t \, \mathrm{d}x_t = 2x_t \sigma \, \mathrm{d}w_t.$$

If this were true, the expectation of $h(x_t)$ would be constant in time because

$$\mathbb{E}[\mathrm{d}h] = 2\sigma \mathbb{E}[x_t \, \mathrm{d}w_t] = 2\sigma \mathbb{E}[x_t] \underbrace{\mathbb{E}[\mathrm{d}w_t]}_{=0} = 0.$$

But we know from classical Brownian motion properties (see Equation (A.2.4)) that

$$\mathbb{E}[x_t^2] = \sigma^2 t,$$

which grows linearly in time. So the ordinary chain rule misses an important term. ∎

C.1.2 从泰勒展开推导一维伊藤公式

通过泰勒展开的确定性链式法则。 为了理解为什么经典的链式法则在由随机微分方程（SDEs）定义的随机过程上失效，我们首先在确定性情景下使用泰勒展开重新回顾它。我们考虑标量情形：$y_t \in \mathbb{R}$ 和 $h(\cdot, \cdot) \in \mathbb{R}$ 。形式上处理 $\mathrm{d}y_t = y_{t+\mathrm{d}t} - y_t$ ，其中 $\mathrm{d}t \approx 0$ ，我们进行展开：

$$\begin{aligned} & h(y_{t+\mathrm{d}t}, t + \mathrm{d}t) - h(y_t, t) \\ = & \frac{\partial h}{\partial t} \mathrm{d}t + \frac{\partial h}{\partial y} \mathrm{d}y_t + \frac{1}{2} \left(\frac{\partial^2 h}{\partial y^2} (\mathrm{d}y_t)^2 + 2 \frac{\partial^2 h}{\partial t \partial y} \mathrm{d}t \, \mathrm{d}y_t + \frac{\partial^2 h}{\partial t^2} (\mathrm{d}t)^2 \right) + \mathcal{O}(\mathrm{d}t^3), \end{aligned}$$

此处，$\mathrm{d}t \, \mathrm{d}y_t = \left(\frac{\mathrm{d}y_t}{\mathrm{d}t} \right) (\mathrm{d}t)^2 = \mathcal{O}(\mathrm{d}t^2)$ ，类似地 $(\mathrm{d}t)^2 = \mathcal{O}(\mathrm{d}t^2)$ 。因此，所有灰色部分均可忽略，全微分为：

$$\mathrm{d}h = \frac{\partial h}{\partial t} \mathrm{d}t + \frac{\partial h}{\partial y} \mathrm{d}y_t + \mathcal{O}(\mathrm{d}t^2).$$

通过随机泰勒展开的伊藤公式。 现在考虑一个由 SDE 控制的随机过程 $x_t \in \mathbb{R}$：

$$\mathrm{d}x_t = f(x_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}w_t,$$

其中 w_t 为标准布朗运动。我们旨在计算标量函数 $h(x_t, t)$ 的微分。

使用随机泰勒展开 (**kloeden1992stochastic**)，它在 $\mathrm{d}x_t$ 中保留二阶项，我们有：

$$\begin{aligned}
&h(x_{t+\mathrm{d}t}, t+\mathrm{d}t) - h(x_t, t) \\
&= \frac{\partial h}{\partial t}\,\mathrm{d}t + \frac{\partial h}{\partial x}\,\mathrm{d}x_t + \frac{1}{2}\left(\frac{\partial^2 h}{\partial x^2}(\mathrm{d}x_t)^2 + 2\frac{\partial^2 h}{\partial t \partial x}\,\mathrm{d}t\,\mathrm{d}x_t + \frac{\partial^2 h}{\partial t^2}(\mathrm{d}t)^2\right) + \cdots
\end{aligned}$$

可忽略的交叉项。 根据布朗运动的缩放性质（Equation (A.2.4)），

$$\mathrm{d}w_t = \mathcal{O}(\sqrt{\mathrm{d}t}) \quad \Rightarrow \quad \mathrm{d}t \cdot \mathrm{d}w_t = \mathcal{O}((\mathrm{d}t)^{3/2}).$$

因此，

$$\mathrm{d}t \cdot \mathrm{d}x_t = \mathrm{d}t\,(f\,\mathrm{d}t + g\,\mathrm{d}w_t) = f(\mathrm{d}t)^2 + g \cdot \mathrm{d}t \cdot \mathrm{d}w_t = \mathcal{O}((\mathrm{d}t)^{3/2}).$$

因此，灰色项可以忽略：$\mathcal{O}((\mathrm{d}t)^{3/2})$ 或更小。

二阶项 $(\mathrm{d}x_t)^2$ 。 使用 SDE 展开：

$$\begin{aligned}
(\mathrm{d}x_t)^2 &= (f\,\mathrm{d}t + g\,\mathrm{d}w_t)^2 \\
&= f^2(\mathrm{d}t)^2 + 2fg\,\mathrm{d}t\,\mathrm{d}w_t + g^2(\mathrm{d}w_t)^2 \\
&= \mathcal{O}((\mathrm{d}t)^2) + \mathcal{O}((\mathrm{d}t)^{3/2}) + g^2\mathcal{O}(\mathrm{d}t) \\
&= g^2(t)\,\mathrm{d}t + \mathcal{O}((\mathrm{d}t)^{3/2}).
\end{aligned}$$

合并各项，我们得到微分：

$$\mathrm{d}h(x_t, t) = \frac{\partial h}{\partial t}\,\mathrm{d}t + \frac{\partial h}{\partial x}\,\mathrm{d}x_t + \frac{1}{2}\frac{\partial^2 h}{\partial x^2}g^2(t)\,\mathrm{d}t.$$

代入 $\mathrm{d}x_t = f(x_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}w_t$ 得：

$$\mathrm{d}h(x_t, t) = \left(\frac{\partial h}{\partial t} + f\frac{\partial h}{\partial x} + \frac{1}{2}g^2\frac{\partial^2 h}{\partial x^2}\right)\mathrm{d}t + g\frac{\partial h}{\partial x}\,\mathrm{d}w_t.$$

这是 *Itô* 公式的一维版本。

Example: Simple Example–$h(x_t) = x_t^2$

We revisit the simple example: $h(x_t) = x_t^2$, where the stochastic process $x_t \in \mathbb{R}$ satisfies

$$\mathrm{d}x_t = \sigma \, \mathrm{d}w_t,$$

with a constant $\sigma > 0$. Applying Itô' s formula correctly to $h(x_t) = x_t^2$, we obtain:

$$\mathrm{d}h(x_t) = \mathrm{d}(x_t^2) = 2x_t \, \mathrm{d}x_t + \sigma^2 \, \mathrm{d}t.$$

Substituting $\mathrm{d}x_t = \sigma \, \mathrm{d}w_t$, this becomes:

$$\mathrm{d}(x_t^2) = 2x_t\sigma \, \mathrm{d}w_t + \sigma^2 \, \mathrm{d}t.$$

C.1.3 伊藤公式：随机微分方程的链式法则

我们总结上述推导出的一维伊藤公式。通过类似的论证，该结果可自然推广到多维情景。尽管省略了详细的推导过程，但为完整性起见，我们给出一般性公式。

最后，我们通过推导 *Itô 乘法法则*来说明 Itô 公式的一个应用，该法则使得对随机过程 $\mathbf{x}_t$ 和 $\mathbf{y}_t$ 的 $\mathrm{d}(\mathbf{x}_t^\top \mathbf{y}_t)$ 计算成为可能。

1 伊藤公式。 令 $x_t \in \mathbb{R}$ 为满足 SDE 的随机过程：

$$\mathrm{d}x_t = f(x_t, t) \, \mathrm{d}t + g(t) \, \mathrm{d}w_t.$$

对于标量函数 $h\colon \mathbb{R} \times [0, T] \to \mathbb{R}$ ，过程 $h(x_t, t)$ 满足：

$$\mathrm{d}h(x_t, t) = \left(\frac{\partial h}{\partial t} + f \frac{\partial h}{\partial x} + \frac{1}{2} g^2 \frac{\partial^2 h}{\partial x^2} \right) \mathrm{d}t + g \frac{\partial h}{\partial x} \, \mathrm{d}w_t.$$

多维伊藤公式（输出为标量） 令 $\mathbf{x}_t \in \mathbb{R}^D$ 满足随机微分方程：

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t) \, \mathrm{d}t + g(t) \, \mathrm{d}\mathbf{w}_t,$$

其中 $\mathbf{f}\colon \mathbb{R}^D \times [0,T] \to \mathbb{R}^D$ 、$g\colon [0,T] \to \mathbb{R}$ 和 $\mathbf{w}_t \in \mathbb{R}^D$ 是一个 D -维布朗运动。令 $h\colon \mathbb{R}^D \times [0,T] \to \mathbb{R}$ 为一个标量函数。则 $h(\mathbf{x}_t, t)$ 满足：

$$\mathrm{d}h(\mathbf{x}_t,t) = \left(\frac{\partial h}{\partial t} + \nabla_{\mathbf{x}} h^\top \mathbf{f} + \frac{1}{2} g^2(t) \operatorname{Tr}\left(\nabla_{\mathbf{x}}^2 h\right)\right) \mathrm{d}t + g(t) \nabla_{\mathbf{x}} h^\top \,\mathrm{d}\mathbf{w}_t, \tag{C.1.1}$$

其中 $\nabla_{\mathbf{x}} h \in \mathbb{R}^D$ 为梯度，$\nabla_{\mathbf{x}}^2 h \in \mathbb{R}^{D\times D}$ 为 h 相对于 $\mathbf{x}$ 的黑塞矩阵。

Example: Itô's Product Rule

Let $\mathbf{x}_t, \mathbf{y}_t \in \mathbb{R}^D$ be vector-valued stochastic processes governed by the SDEs:

$$\begin{aligned}\mathrm{d}\mathbf{x}_t &= \mathbf{a}(\mathbf{x}_t, t)\,\mathrm{d}t + b(t)\,\mathrm{d}\mathbf{w}_t,\\ \mathrm{d}\mathbf{y}_t &= \mathbf{c}(\mathbf{y}_t, t)\,\mathrm{d}t + d(t)\,\mathrm{d}\mathbf{w}_t,\end{aligned}$$

where $\mathbf{a}, \mathbf{c} : \mathbb{R}^D \times [0,T] \to \mathbb{R}^D$ are vector fields, and $b(t), d(t) \in \mathbb{R}$ are scalar-valued functions. Here, $\mathbf{w}_t \in \mathbb{R}^D$ denotes a standard D-dimensional Brownian motion.

We aim to derive the SDE for the scalar-valued process

$$z(t) := \mathbf{x}_t^\top \mathbf{y}_t.$$

Applying the multivariate Itô formula to the bilinear function $h(\mathbf{x}, \mathbf{y}) := \mathbf{x}^\top \mathbf{y}$, we obtain:

$$\mathrm{d}(\mathbf{x}^\top \mathbf{y}) = (\mathrm{d}\mathbf{x})^\top \mathbf{y} + \mathbf{x}^\top \,\mathrm{d}\mathbf{y} + \operatorname{Tr}\left[\mathrm{d}\mathbf{x} \cdot (\mathrm{d}\mathbf{y})^\top\right].$$

The Itô correction term is computed as:

$$\begin{aligned}\mathrm{d}\mathbf{x} \cdot (\mathrm{d}\mathbf{y})^\top &= b(t)\,\mathrm{d}\mathbf{w}_t \cdot \left[d(t)\,\mathrm{d}\mathbf{w}_t\right]^\top\\ &= b(t)d(t)\,\mathrm{d}\mathbf{w}_t \cdot \mathrm{d}\mathbf{w}_t^\top\\ &= b(t)d(t)\,\mathrm{d}t \cdot \mathbf{I}_D.\end{aligned}$$

Thus,

$$\operatorname{Tr}\left[\mathrm{d}\mathbf{x} \cdot (\mathrm{d}\mathbf{y})^\top\right] = b(t)d(t)\operatorname{Tr}(\mathbf{I}_D)\,\mathrm{d}t = Db(t)d(t)\,\mathrm{d}t.$$

Putting everything together, the resulting SDE is:

$$\mathrm{d}(\mathbf{x}^\top \mathbf{y}) = (\mathrm{d}\mathbf{x})^\top \mathbf{y} + \mathbf{x}^\top \,\mathrm{d}\mathbf{y} + Db(t)d(t)\,\mathrm{d}t \tag{C.1.2}$$

C.1.4 伊藤公式的应用：福克-普朗克方程的推导

在本节中，我们应用 Equation (C.1.1) 中的 Itô 公式来推导 Fokker–Planck 方程，这是一个偏微分方程，用于刻画与由 Equation (A.2.3) 中的随机微分方程定义的 D 维扩散过程相关的概率密度 $p_t(\mathbf{x})$ 随时间的演化。

步骤 1：应用伊藤公式。 设 $\phi(\mathbf{x}, t)$ 为一个光滑检验函数 $\phi\colon \mathbb{R}^D \times [0, T] \to \mathbb{R}$ 。由 Itô 公式：

$$\mathrm{d}\phi(\mathbf{x}_t, t) = \left(\frac{\partial \phi}{\partial t} + \nabla_{\mathbf{x}} \phi^\top \mathbf{f}(\mathbf{x}_t, t) + \frac{1}{2} g^2(t) \operatorname{Tr}[\nabla_{\mathbf{x}}^2 \phi] \right) \mathrm{d}t + g(t) \nabla_{\mathbf{x}} \phi^\top \,\mathrm{d}\mathbf{w}_t.$$

第二步：取期望。 对 $p_t(\mathbf{x})$ 取期望，并注意到 $\mathbb{E}[\mathrm{d}\mathbf{w}_t] = 0$ ：

$$\mathbb{E}[\mathrm{d}\phi(\mathbf{x}_t, t)] = \mathbb{E}\left[\left(\frac{\partial \phi}{\partial t} + \nabla_{\mathbf{x}} \phi^\top \mathbf{f}(\mathbf{x}_t, t) + \frac{1}{2} g^2(t) \operatorname{Tr}[\nabla_{\mathbf{x}}^2 \phi] \right) \mathrm{d}t \right].$$

第三步：通过密度表达期望。 这种期望可表示为：

$$\mathbb{E}[\mathrm{d}\phi(\mathbf{x}_t, t)] = \int \left(\frac{\partial \phi}{\partial t} + \nabla_{\mathbf{x}} \phi^\top \mathbf{f}(\mathbf{x}, t) + \frac{1}{2} g^2(t) \operatorname{Tr}[\nabla_{\mathbf{x}}^2 \phi] \right) p_t(\mathbf{x}) \,\mathrm{d}\mathbf{x}\,\mathrm{d}t.$$

第四步：分部积分。 使用分部积分（散度定理）于 $\mathbb{R}^D$ ：

$$\int \nabla_{\mathbf{x}} \phi^\top \mathbf{f} p_t \,\mathrm{d}\mathbf{x} = -\int \phi \nabla_{\mathbf{x}} \cdot (\mathbf{f} p_t) \,\mathrm{d}\mathbf{x},$$
$$\int \operatorname{Tr}[\nabla_{\mathbf{x}}^2 \phi] p_t \,\mathrm{d}\mathbf{x} = -\int \phi \Delta p_t \,\mathrm{d}\mathbf{x}.$$

第五步：代入并整理。 代回后：

$$\mathbb{E}[\mathrm{d}\phi(\mathbf{x}_t, t)] = \int \phi(\mathbf{x}, t) \left[\frac{\partial p_t}{\partial t} + \nabla_{\mathbf{x}} \cdot (\mathbf{f} p_t) - \frac{1}{2} g^2(t) \Delta p_t \right] \mathrm{d}\mathbf{x}\,\mathrm{d}t.$$

第六步：得出福克-普朗克方程。 由于 ϕ 是任意的，被积函数必须消失：

$$\frac{\partial p_t}{\partial t} = -\nabla_{\mathbf{x}} \cdot (\mathbf{f}(\mathbf{x}, t) p_t(\mathbf{x})) + \frac{1}{2} g^2(t) \Delta p_t(\mathbf{x}),$$

这完成了福克-普朗克方程的推导。

C.1.5 伊藤公式应用：线性随机微分方程的闭式解

本小节演示如何通过使用积分因子（类似于常微分方程的情形）和伊藤公式，获得线性随机微分方程的闭式解。该方法模仿求解线性常微分方程的经典技巧，但已适应随机情景。

我们考虑如下形式的线性随机微分方程

$$\mathrm{d}\mathbf{x}_t = f(t)\mathbf{x}_t \mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t, \tag{C.1.3}$$

其中 $f(t)$ 和 $g(t)$ 是确定性函数，$\mathbf{w}_t$ 是标准维纳过程。

线性随机微分方程的闭式解。 我们使用积分因子法推导出 Equation (C.1.3) 中线性 SDE 的显式解。此类前向 SDE 在扩散模型中经常出现（参见 Section 4.1）。

步骤 1：定义一个积分因子。 设

$$\Psi(t) := \exp\left(-\int_0^t f(s)\mathrm{d}s\right), \quad \text{and define} \quad \mathbf{y}_t := \Psi(t)\mathbf{x}_t.$$

步骤 2：应用伊藤公式。 我们对函数 $\mathbf{h}(\mathbf{x}, t) := \Psi(t)\mathbf{x}$ 应用伊藤公式。这实际上是 Equation (C.1.2) 中伊藤乘法法则的一个特例。由于 $\Psi(t)$ 是确定性的，因此不存在交叉变差项，公式简化为：

$$\begin{aligned}
\mathrm{d}\mathbf{y}_t &= \mathrm{d}[\Psi(t)\mathbf{x}_t] \\
&= \Psi'(t)\mathbf{x}_t \mathrm{d}t + \Psi(t)\mathrm{d}\mathbf{x}_t \\
&= -f(t)\Psi(t)\mathbf{x}_t \mathrm{d}t + \Psi(t)\left[f(t)\mathbf{x}_t \mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t\right] \\
&= \Psi(t) g(t) \mathrm{d}\mathbf{w}_t.
\end{aligned}$$

因此，

$$\mathbf{y}_t = \mathbf{y}_0 + \int_0^t \Psi(s) g(s) \mathrm{d}\mathbf{w}(s) = \mathbf{x}_0 + \int_0^t \Psi(s) g(s) \mathrm{d}\mathbf{w}(s),$$

由于 $\Psi(0)=1$ 。

步骤 3：求解 $\mathbf{x}_t$ 。 利用 $\mathbf{x}_t=\Psi(t)^{-1}\mathbf{y}_t$ ，我们得到

$$\mathbf{x}_t=e^{\int_0^t f(s)\mathrm{d}s}\left[\mathbf{x}_0+\int_0^t e^{-\int_0^s f(r)\mathrm{d}r}g(s)\mathrm{d}\mathbf{w}(s)\right]. \tag{C.1.4}$$

这为向量值随机微分方程提供了显式解。

下面，我们演示两种计算 $p_t(\mathbf{x}_t|\mathbf{x}_0)$ 的解析形式的替代方法。

闭式解的分析。 Equation (C.1.4) 证实了 $p_t(\mathbf{x}_t|\mathbf{x}_0)$ 服从高斯分布。为此，定义

$$\phi(s):=e^{-\int_0^s f(u)\,\mathrm{d}u}g(s),$$

which 是一个确定性的矩阵值函数（假设 $f(u)$ 和 $g(s)$ 是确定性的）。伊藤积分 $\int_0^t\phi(s)\,\mathrm{d}\mathbf{w}_s$ 则是一个零均值高斯随机变量，因为它是确定性函数关于布朗运动的随机积分。因此，$\mathbf{x}_t|\mathbf{x}_0$ 是高斯随机变量的仿射变换，因此它本身也是高斯的。它的分布由其条件均值和协方差完全刻画。

我们定义条件均值和协方差（给定初始条件 $\mathbf{x}_0$ ）为

$$\mathbf{m}(t):=\mathbb{E}[\mathbf{x}_t|\mathbf{x}_0],\quad \mathbf{P}(t):=\mathbb{E}[(\mathbf{x}_t-\mathbf{m}(t))(\mathbf{x}_t-\mathbf{m}(t))^\top|\mathbf{x}_0].$$

期望. 利用期望的线性性质以及确定性函数的 Itô 积分具有零均值的事实：

$$\mathbf{m}(t)=\mathbb{E}\left[e^{\int_0^t f(s)\,\mathrm{d}s}\left(\mathbf{x}_0+\int_0^t\phi(s)\,\mathrm{d}\mathbf{w}_s\right)\middle|\mathbf{x}_0\right]=e^{\int_0^t f(s)\,\mathrm{d}s}\mathbf{x}_0.$$

协方差。 令 $\mathbf{z}_t:=\int_0^t\phi(s)\,\mathrm{d}\mathbf{w}_s$ 。则 $\mathbf{x}_t-\mathbf{m}(t)=A(t)\mathbf{z}_t$ ，所以

$$\mathbf{P}(t)=e^{2\int_0^t f(s)\,\mathrm{d}s}\mathbb{E}[\mathbf{z}_t\mathbf{z}_t^\top].$$

根据伊藤等距性[1]，

$$\mathbb{E}[\mathbf{z}_t\mathbf{z}_t^\top]=\left(\int_0^t\phi^2(s)\,\mathrm{d}s\right)\mathbf{I}_D,$$

[1] Itô 不等式将随机积分与期望下的标准积分联系起来；由于其证明需要完整的 Itô 微积分工具，我们省略证明过程。对于一个过程 $\boldsymbol{\psi}:[0,T]\to\mathbb{R}^{D\times D}$ ，Itô 不等式表述为

$$\mathbb{E}\left[\left\|\int_0^T\boldsymbol{\psi}(t)\mathrm{d}\mathbf{w}_t\right\|^2\right]=\mathbb{E}\left[\int_0^T\|\boldsymbol{\psi}(t)\|_F^2\mathrm{d}t\right],$$

因此，

$$\mathbf{P}(t) = e^{2\int_0^t f(s)\,\mathrm{d}s}\left(\int_0^t \left(e^{-\int_0^s f(u)\,\mathrm{d}u} g(s)\right)^2 \mathrm{d}s\right)\mathbf{I}_D.$$

这表明条件协方差是各向同性的。

Equation (4.3.3) 中均值和方差常微分方程的推导 另外，我们可以直接从线性 SDE Equation (C.1.3) 推导出矩演化方程。

期望演化。 对 SDE 两边取条件期望并利用线性性质：

$$\frac{\mathrm{d}\mathbf{m}(t)}{\mathrm{d}t} = \mathbb{E}[f(t)\mathbf{x}_t|\mathbf{x}_0] = f(t)\mathbb{E}[\mathbf{x}_t|\mathbf{x}_0] = f(t)\mathbf{m}(t).$$

协方差演化。 定义中心化过程 $\tilde{\mathbf{x}}_t := \mathbf{x}_t - \mathbf{m}(t)$ 。应用 Itô 的乘法法则（参见 Equation (C.1.2)）：

$$\mathrm{d}\left(\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^\top\right) = \mathrm{d}\tilde{\mathbf{x}}_t \cdot \tilde{\mathbf{x}}_t^\top + \tilde{\mathbf{x}}_t \cdot \mathrm{d}\tilde{\mathbf{x}}_t^\top + \mathrm{d}\tilde{\mathbf{x}}_t \cdot \mathrm{d}\tilde{\mathbf{x}}_t^\top.$$

从 SDE 中，我们计算：

$$\mathrm{d}\tilde{\mathbf{x}}_t = \mathrm{d}\mathbf{x}_t - \mathrm{d}\mathbf{m}(t) = f(t)\tilde{\mathbf{x}}_t\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t.$$

代入乘法法则并取期望：

$$\begin{aligned}\frac{\mathrm{d}\mathbf{P}(t)}{\mathrm{d}t} &= \mathbb{E}[f(t)\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^\top + \tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^\top f(t) + g^2(t)\mathbf{I}_D] \\ &= 2f(t)\mathbf{P}(t) + g^2(t)\mathbf{I}_D.\end{aligned}$$

因此，我们恢复了 Equation (4.3.3) 中的矩演化方程。

其中 $\|\boldsymbol{\psi}(t)\|_F^2 = \sum_{i,j=1}^D |\psi_{ij}(t)|^2$ 为 Frobenius 范数。当 $\boldsymbol{\psi}(t) \in \mathbb{R}^D$ 为（向量）时，积分结果为标量，不等式简化为

$$\mathbb{E}\left[\left(\int_0^T \boldsymbol{\psi}(t)\mathrm{d}\mathbf{w}_t\right)^2\right] = \mathbb{E}\left[\int_0^T \|\boldsymbol{\psi}(t)\|^2\mathrm{d}t\right].$$

C.2 测度的变量变换：扩散模型中的吉尔萨诺夫定理

扩散模型利用随机微分方程将简单的噪声转化为丰富的数据分布。这一转换的核心思想在于：我们可以通过仅改变随机微分方程的确定性部分（即漂移项），同时保持其潜在的随机性，来重新诠释随机性。这正是吉尔桑诺夫定理发挥作用之处。

核心思想。 考虑一个从时间 $t = 0$ 到 $t = T$ 描述数据演化的连续轨迹，记为 $\mathbf{x}_{0:T} := \{\mathbf{x}_t | t \in [0, T]\}$。Girsanov 定理解决了一个基本问题：

Question C.2.1

> 给定这条单一观测路径，如果我们假设它是由某一随机微分方程生成的，与假设它是由另一随机微分方程生成的相比，其似然分别是多少？

我们比较两种假设模型对同一轨迹的生成能力。这两种假设的随机微分方程共享相同的潜在纯随机性，由标准维纳过程（布朗运动）$\mathbf{w}_t$ 表示，但仅在确定性“推动”或“漂移”函数上有所不同。我们假设 $\mathbf{x}_0$ 在两种假设生成过程中具有相同的初始分布。

为了建立直观认识，想象 $\mathbf{x}_{0:T}$ 为纸上绘制的一条蜿蜒的曲线。一种假设是，它由一个“机器人画家”生成，该画家受漂移 $\mathbf{f}$ 的引导，并受到缩放因子为 $g(t)$ 的随机噪声扰动，从而产生似然 $p_{\mathbf{f}}(\mathbf{x}_{0:T})$。另一种假设是，存在第二个机器人，具有不同的漂移 $\tilde{\mathbf{f}}$，但使用相同的噪声过程，生成了同一条曲线，其似然为 $p_{\tilde{\mathbf{f}}}(\mathbf{x}_{0:T})$。Girsanov 定理为我们提供了一种精确的方法，用于比较这两种情况对于同一观测轨迹的似然。它量化了漂移变化如何影响生成特定轨迹的概率，同时保持随机性不变。

设置。 设 $\mathbf{x}_t \in \mathbb{R}^D$ 为我们的单一、固定、连续路径。我们考虑其在两种 SDE 模型下的似然，这两种模型仅在漂移函数 $\mathbf{f}$ 和 $\tilde{\mathbf{f}}$ 上有所不同。它们具有相同的扩散系数 $g(t) \in \mathbb{R}$ 和相同的潜在维纳过程 $\mathbf{w}_t$：

$$\begin{aligned} \mathrm{d}\mathbf{x}_t &= \mathbf{f}(\mathbf{x}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t \quad \text{(Model with drift } \mathbf{f}\text{)} \\ \mathrm{d}\mathbf{x}_t &= \tilde{\mathbf{f}}(\mathbf{x}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t \quad \text{(Model with drift } \tilde{\mathbf{f}}\text{)} \end{aligned}$$

令 $\boldsymbol{\delta}_t := \mathbf{f}(\mathbf{x}_t, t) - \tilde{\mathbf{f}}(\mathbf{x}_t, t)$ 表示给定路径 $\mathbf{x}_t$ 的漂移差异。

吉尔桑诺夫似然比。 Girsanov 定理提供了这两种解释同一观测路径方式之间的基本似然比。其内容为：

$$\frac{p_{\mathbf{f}}(\mathbf{x}_{0:T})}{p_{\tilde{\mathbf{f}}}(\mathbf{x}_{0:T})} = \exp\left(\int_0^T \boldsymbol{\delta}_t^\top g(t)^{-1}\,\mathrm{d}\mathbf{w}_t - \frac{1}{2}\int_0^T \|g(t)^{-1}\boldsymbol{\delta}_t\|^2\,\mathrm{d}t\right).$$

这个紧凑的公式是两个积分的指数形式。第一个是伊藤积分，第二个是标准的黎曼积分。该比值在扩散模型中至关重要，使我们能够连接不同的数据生成过程，并评估模型的似然。

Girsanov 定理最好被理解为测度的变量变换公式。正如微积分中的变量变换通过雅克比行列式在坐标系之间转换积分一样，Girsanov 定理提供了相应的因子（Radon–Nikodym 导数），用于在漂移改变但扩散保持不变的情况下，将两个随机过程之间的概率或期望进行转换。

C.2.1 Girsanov 定理作为似然训练与分数匹配之间的桥梁

在理解了 Girsanov 定理如何关联不同漂移假设下单一路径的似然之后，我们现在探讨其对扩散模型的影响。

回想扩散模型中的前向 SDE：

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t,$$

这诱导出一个 *路径分布* P 在完整轨迹上 $\mathbf{x}_{0:T} := \{\mathbf{x}_t\}_{t=0}^T$（即过程在整个时间区间上的联合分布）。由可学习的评分函数 $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)$ 参数化的反向时间 SDE 为

$$\mathrm{d}\mathbf{x}_t = \left[\mathbf{f}(\mathbf{x}_t, t) - g^2(t)\,\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)\right]\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}_t,$$

这进一步定义了轨迹上的另一个路径分布 $P_{\boldsymbol{\phi}}$ 。

扩散模型中的两个概念。 在扩散模型中，我们通过两种核心视角来描述随机过程 $\mathbf{x}_{0:T}$：前向过程及其逆时序对应过程。这两种视角引出了两个不同但相关的目标：

- **概念 1. 边缘分布匹配**：该目标构建了一个反向时间过程，其边缘分布 $p_t(\mathbf{x}_t)$ 与前向 SDE 的边缘分布相匹配，从时间 T 的噪声开始，最终恢复数据分布于 $t = 0$ 。如前所强调，福克-普朗克方程确保了反向时间 SDE 的边缘一致性。

- **概念 2. 联合轨迹分布匹配**：这一更强的目标旨在匹配整个轨迹上的完整联合分布 $P = p(\mathbf{x}_{0:T})$。与仅匹配单个时间步的快照不同，此条件确保了状态序列及其时间依赖关系得到忠实再现。

匹配完整路径分布 P 可确保所有边缘分布均匹配。形式上，设 $\mathbf{x}_{0:T} := \{\mathbf{x}_t | t \in [0,T]\}$ 是具有联合分布 $p(\mathbf{x}_{0:T})$ 的随机过程。假设另一个具有联合分布 $q(\mathbf{x}_{0:T})$ 的过程满足

$$p(\mathbf{x}_{0:T}) = q(\mathbf{x}_{0:T}).$$

然后对于任意 $t \in [0,T]$ ，边缘分布为

$$p_t(\mathbf{x}_t) = \int p(\mathbf{x}_{0:T})\, \mathrm{d}\mathbf{x}_{[0,T]\setminus\{t\}}, \quad q_t(\mathbf{x}_t) = \int q(\mathbf{x}_{0:T})\, \mathrm{d}\mathbf{x}_{[0,T]\setminus\{t\}},$$

这意味着

$$p_t(\mathbf{x}_t) = q_t(\mathbf{x}_t), \quad \forall t \in [0,T].$$

因此，联合路径匹配意味着边缘匹配。

然而，反过来却不成立：两个过程可能在每个时间步都具有相同的边缘分布，但在时间相关性上却存在显著差异。边缘匹配无法捕捉这些时间间的依赖关系，而这些关系仅存在于联合分布中。

吉尔桑诺夫连接了两个目标。 虽然反向时间 SDE 主要针对边际匹配（概念 1），但吉尔萨诺夫定理揭示了一个更深层次的联系：时间上的得分匹配同样能促进联合路径匹配（概念 2）。

更精确地说，Girsanov 定理关联了前向路径分布 P 与学成的反向路径分布 $P_{\boldsymbol{\phi}}$ 。基于得分的扩散模型的目标函数是这两种路径测度之间的 KL 散度：

$$\mathcal{D}_{\mathrm{KL}}(P \| P_{\boldsymbol{\phi}}) = \frac{1}{2}\mathbb{E}_P \left[\int_0^T g^2(t) \left\| \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t) \right\|^2 \mathrm{d}t \right] + \text{Const.}, \tag{C.2.1}$$

此处，常数项不依赖于 $\boldsymbol{\phi}$ ，我们利用了在 P 下 Itô 积分的期望为零这一事实。该表达式表明，最小化联合路径之间的 KL 散度等价于学习一个近似真实得分 $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ 的得分函数 $\mathbf{s}_{\boldsymbol{\phi}}$ 。因此，尽管分数匹配被表述为边缘目标，但其实际上有效促进了整个联合路径分布的一致性。

隐式似然训练 除了匹配路径分布外，分数匹配还隐式地使扩散模型能够实现生成式建模的一个基本目标：近似数据的似然 (**song2021maximum**)。

通过一个强大的概念——测度变换公式，这种联系变得清晰起来。该公式在吉尔桑诺夫定理的启示下，使我们能够表达在学成模型下数据在 $t = 0$ ($p_{\boldsymbol{\phi}}(\mathbf{x}_0)$) 处的边缘似然函数的对数。

$$\log p_{\boldsymbol{\phi}}(\mathbf{x}_0) = \log \int p_T(\mathbf{x}_T) \cdot \frac{p_{\boldsymbol{\phi}}(\mathbf{x}_{0:T})}{p(\mathbf{x}_{0:T})} p(\mathbf{x}_{0:T}) \, \mathrm{d}\mathbf{x}_{0:T}. \tag{C.2.2}$$

此处，$p_T(\mathbf{x}_T)$ 是由前向 SDE 给出的时间 T 下的噪声已知分布。项 $\frac{p_{\boldsymbol{\phi}}(\mathbf{x}_{0:T})}{p(\mathbf{x}_{0:T})}$ 是给定路径 $\mathbf{x}_{0:T}$ 下，学成的反向过程与前向过程之间的密度比——该对象由吉尔萨诺夫定理精确量化。本质上，此公式通过根据我们学成的反向动力学对观测轨迹的解释程度，对已知噪声似然进行重赋权，从而计算生成数据的似然。

我们进一步将联系回 Equation (C.2.1) 中的 KL 最小化，该问题关注的是完整前向路径分布与学成反向路径分布之间的差异。这两个概念，Equation (C.2.1) 与 Equation (C.2.2)，紧密交织：优化这一分数匹配目标（训练损失）直接转化为学习 Girsanov 密度比，从而隐式地最大化数据似然 ($p_{\boldsymbol{\phi}}(\mathbf{x}_0)$)。这一优美的关联巧妙地将 Girsanov 定理、基于分数的学习以及生成式建模的最终目标——为真实数据分配高概率——融为一体。

D

补充材料与证明

D.1 变分视角

D.1.1 定理 2.2.1：边缘与条件 KL 最小化之间的等价性

证明。 Equation (2.2.3). 的推导

我们首先展开右侧的期望：

$$\begin{aligned}&\mathbb{E}_{p(\mathbf{x}_0,\mathbf{x}_i)}\left[\mathcal{D}_{\mathrm{KL}}\big(p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x}_0)\|p_\phi(\mathbf{x}_{i-1}|\mathbf{x}_i)\big)\right]\\=&\int\int p(\mathbf{x}_0,\mathbf{x}_i)\mathcal{D}_{\mathrm{KL}}\big(p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x}_0)\|p_\phi(\mathbf{x}_{i-1}|\mathbf{x}_i)\big)\,\mathrm{d}\mathbf{x}_0\,\mathrm{d}\mathbf{x}_i.\end{aligned}$$

根据 KL 散度的定义，

$$\mathcal{D}_{\mathrm{KL}}\big(p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x}_0)\|p_\phi(\mathbf{x}_{i-1}|\mathbf{x}_i)\big)=\int p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x}_0)\log\frac{p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x}_0)}{p_\phi(\mathbf{x}_{i-1}|\mathbf{x}_i)}\,\mathrm{d}\mathbf{x}_{i-1}.$$

将其代入期望，我们得到

$$\int\int\int p(\mathbf{x}_0,\mathbf{x}_i)p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x}_0)\log\frac{p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x}_0)}{p_\phi(\mathbf{x}_{i-1}|\mathbf{x}_i)}\,\mathrm{d}\mathbf{x}_{i-1}\,\mathrm{d}\mathbf{x}_0\,\mathrm{d}\mathbf{x}_i.$$

使用概率的链式法则，

$$p(\mathbf{x}_0, \mathbf{x}_i) = p(\mathbf{x}_i)p(\mathbf{x}_0|\mathbf{x}_i),$$

我们将积分重写为

$$\int p(\mathbf{x}_i) \int p(\mathbf{x}_0|\mathbf{x}_i) \int p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}_0) \log \frac{p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}_0)}{p_\phi(\mathbf{x}_{i-1}|\mathbf{x}_i)} \, \mathrm{d}\mathbf{x}_{i-1} \, \mathrm{d}\mathbf{x}_0 \, \mathrm{d}\mathbf{x}_i.$$

这使我们能够以嵌套形式表达期望：

$$\mathbb{E}_{p(\mathbf{x}_i)} \left[\mathbb{E}_{p(\mathbf{x}_0|\mathbf{x}_i)} \left[\mathbb{E}_{p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}_0)} \left[\log \frac{p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}_0)}{p_\phi(\mathbf{x}_{i-1}|\mathbf{x}_i)} \right] \right] \right].$$

接下来，我们应用对数的分解：

$$\log \frac{p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}_0)}{p_\phi(\mathbf{x}_{i-1}|\mathbf{x}_i)} = \log \frac{p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}_0)}{p(\mathbf{x}_{i-1}|\mathbf{x}_i)} + \log \frac{p(\mathbf{x}_{i-1}|\mathbf{x}_i)}{p_\phi(\mathbf{x}_{i-1}|\mathbf{x}_i)}.$$

将其代回期望式中，得到两项：

$$\begin{aligned} &\mathbb{E}_{p(\mathbf{x}_i)} \left[\mathbb{E}_{p(\mathbf{x}_0|\mathbf{x}_i)} \left[\mathbb{E}_{p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}_0)} \left[\log \frac{p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}_0)}{p(\mathbf{x}_{i-1}|\mathbf{x}_i)} \right] \right] \right] \\ &+ \mathbb{E}_{p(\mathbf{x}_i)} \left[\mathbb{E}_{p(\mathbf{x}_0|\mathbf{x}_i)} \left[\mathbb{E}_{p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}_0)} \left[\log \frac{p(\mathbf{x}_{i-1}|\mathbf{x}_i)}{p_\phi(\mathbf{x}_{i-1}|\mathbf{x}_i)} \right] \right] \right]. \end{aligned}$$

由于第二项对数项不依赖于 $\mathbf{x}_0$ ，根据全概率定律

$$\mathbb{E}_{p(\mathbf{x}_0|\mathbf{x}_i)} \left[\mathbb{E}_{p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}_0)} \left[\log \frac{p(\mathbf{x}_{i-1}|\mathbf{x}_i)}{p_\phi(\mathbf{x}_{i-1}|\mathbf{x}_i)} \right] \right] = \mathbb{E}_{p(\mathbf{x}_{i-1}|\mathbf{x}_i)} \left[\log \frac{p(\mathbf{x}_{i-1}|\mathbf{x}_i)}{p_\phi(\mathbf{x}_{i-1}|\mathbf{x}_i)} \right].$$

同样，第一项是 KL 散度

$$\mathbb{E}_{p(\mathbf{x}_0|\mathbf{x}_i)} \left[\mathcal{D}_{\mathrm{KL}}\big(p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}_0) \| p(\mathbf{x}_{i-1}|\mathbf{x}_i)\big) \right].$$

综上所述，我们得到如下分解：

$$\begin{aligned} &\mathbb{E}_{p(\mathbf{x}_0, \mathbf{x}_i)} \left[\mathcal{D}_{\mathrm{KL}}\big(p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}_0) \| p_\phi(\mathbf{x}_{i-1}|\mathbf{x}_i)\big) \right] \\ =&\mathbb{E}_{p(\mathbf{x}_i)} \left[\mathbb{E}_{p(\mathbf{x}_0|\mathbf{x}_i)} \left[\mathcal{D}_{\mathrm{KL}}\big(p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}_0) \| p(\mathbf{x}_{i-1}|\mathbf{x}_i)\big) \right] \right] \\ &+ \mathbb{E}_{p(\mathbf{x}_i)} \left[\mathcal{D}_{\mathrm{KL}}\big(p(\mathbf{x}_{i-1}|\mathbf{x}_i) \| p_\phi(\mathbf{x}_{i-1}|\mathbf{x}_i)\big) \right]. \end{aligned}$$

最优性证明。证明：

$$p^*(\mathbf{x}_{i-1}|\mathbf{x}_i) = p(\mathbf{x}_{i-1}|\mathbf{x}_i) = \mathbb{E}_{p(\mathbf{x}|\mathbf{x}_i)}\left[p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x})\right], \quad \mathbf{x}_i \sim p_i.$$

第一个恒等式源于以下事实：当参数化足够表达时，KL 散度 $\mathcal{D}_{\mathrm{KL}}(p\|p_{\boldsymbol{\phi}})$ 在 $p^* = p$ 时取得最小值。第二个恒等式直接来自全概率定律。 ■

D.1.2 定理 2.2.3：扩散模型的 ELBO

证明。 为书写简便，我们记 $\mathbf{x}_{0:L} := (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_L)$ 。

步骤 1：应用詹森不等式。边缘对数似然由下式给出：

$$\log p_{\boldsymbol{\phi}}(\mathbf{x}) = \log \int p_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{x}_{0:L})\, \mathrm{d}\mathbf{x}_0 \cdots \mathrm{d}\mathbf{x}_L,$$

联合分布为：

$$p_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{x}_{0:L}) = p_{\mathrm{prior}}(\mathbf{x}_L) \prod_{i=1}^{L} p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i) \cdot p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{x}_0).$$

我们引入变分分布 $p(\mathbf{x}_{0:L}|\mathbf{x})$ 并重写为：

$$\log p_{\boldsymbol{\phi}}(\mathbf{x}) = \log \int p(\mathbf{x}_{0:L}|\mathbf{x}) \frac{p_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{x}_{0:L})}{p(\mathbf{x}_{0:L}|\mathbf{x})}\, \mathrm{d}\mathbf{x}_0 \cdots \mathrm{d}\mathbf{x}_L.$$

应用 Jensen 不等式 ($\log \mathbb{E}[Z] \geq \mathbb{E}[\log Z]$)，我们得到 ELBO：

$$\log p_{\boldsymbol{\phi}}(\mathbf{x}) \geq \mathbb{E}_{p(\mathbf{x}_{0:L}|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{x}_{0:L})}{p(\mathbf{x}_{0:L}|\mathbf{x})}\right] =: \mathcal{L}_{\mathrm{ELBO}},$$

因此，

$$-\log p_{\boldsymbol{\phi}}(\mathbf{x}) \leq -\mathcal{L}_{\mathrm{ELBO}}.$$

步骤 2：展开 ELBO。假设变分分布可分解为：

$$p(\mathbf{x}_{0:L}|\mathbf{x}) = p(\mathbf{x}_L|\mathbf{x}) \prod_{i=1}^{L} p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}).$$

将联合分布和变分分布代入 ELBO：

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{p(\mathbf{x}_{0:L}|\mathbf{x})}\Big[\log p_{\text{prior}}(\mathbf{x}_L) + \sum_{i=1}^{L} \log p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i) + \log p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{x}_0) - \log p(\mathbf{x}_L|\mathbf{x}) - \sum_{i=1}^{L} \log p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x})\Big].$$

我们现在通过根据各项的依赖关系进行分组并应用边缘化来计算负的 ELBO：

$$-\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{p(\mathbf{x}_0|\mathbf{x})}\left[-\log p_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{x}_0)\right] + \mathbb{E}_{p(\mathbf{x}_L|\mathbf{x})}\left[\log \frac{p(\mathbf{x}_L|\mathbf{x})}{p_{\text{prior}}(\mathbf{x}_L)}\right] + \sum_{i=1}^{L} \mathbb{E}_{p(\mathbf{x}_i|\mathbf{x})}\left[\mathbb{E}_{p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x})}\left[\log \frac{p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x})}{p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i)}\right]\right].$$

为了证明最后一项，我们使用因子分解：

$$p(\mathbf{x}_i, \mathbf{x}_{i-1}|\mathbf{x}) = p(\mathbf{x}_i|\mathbf{x}) \cdot p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}),$$

这导致：

$$\begin{aligned}
&\mathbb{E}_{p(\mathbf{x}_{0:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x})}{p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i)}\right] \\
=&\int p(\mathbf{x}_i|\mathbf{x}) \left[\int p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}) \log \frac{p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x})}{p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i)} \, \mathrm{d}\mathbf{x}_{i-1}\right] \mathrm{d}\mathbf{x}_i \\
=&\mathbb{E}_{p(\mathbf{x}_i|\mathbf{x})}\left[\mathbb{E}_{p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x})}\left[\log \frac{p(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x})}{p_{\boldsymbol{\phi}}(\mathbf{x}_{i-1}|\mathbf{x}_i)}\right]\right].
\end{aligned}$$

我们可以将 $-\mathcal{L}_{\text{ELBO}}$ 中的三个术语称为：

$$\mathcal{L}_{\text{recon.}}, \quad \mathcal{L}_{\text{prior}}, \quad \mathcal{L}_{\text{diffusion}},$$

分别对应重构损失、先验 KL 散度和逐步扩散 KL 散度。推导完成。 ■

D.2 基于得分的视角

D.2.1 命题 3.2.1：通过分部积分实现易处理的分数匹配

证明。 **展开** $\mathcal{L}_{\mathsf{SM}}(\boldsymbol{\phi})$ 。我们展开期望内的平方差：

$$\begin{aligned}\mathcal{L}_{\mathrm{SM}}(\boldsymbol{\phi}) &= \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}(\mathbf{x})}\left[\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})\|_2^2 - 2\langle \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{s}(\mathbf{x})\rangle + \|\mathbf{s}(\mathbf{x})\|_2^2\right] \\ &= \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}(\mathbf{x})}\left[\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})\|_2^2\right] - \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}(\mathbf{x})}\left[\langle \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{s}(\mathbf{x})\rangle\right] \\ &\quad + \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}(\mathbf{x})}\left[\|\mathbf{s}(\mathbf{x})\|_2^2\right].\end{aligned}$$

我们现在关注交叉项：

$$\mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}(\mathbf{x})}\left[\langle \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{s}(\mathbf{x})\rangle\right].$$

利用这一事实

$$\nabla_{\mathbf{x}} \log p_{\mathrm{data}}(\mathbf{x}) = \frac{\nabla_{\mathbf{x}} p_{\mathrm{data}}(\mathbf{x})}{p_{\mathrm{data}}(\mathbf{x})},$$

并且假设 $p_{\mathrm{data}}(\mathbf{x})$ 不为零（例如，在其支撑集上），叉积项变为：

$$\begin{aligned}\mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}(\mathbf{x})}\left[\langle \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{s}(\mathbf{x})\rangle\right] &= \int \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})^\top \nabla_{\mathbf{x}} \log p_{\mathrm{data}}(\mathbf{x}) p_{\mathrm{data}}(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &= \int \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})^\top \nabla_{\mathbf{x}} p_{\mathrm{data}}(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &= \sum_{i=1}^{D} \int s_{\boldsymbol{\phi}}^{(i)}(\mathbf{x}) \partial_{x_i} p_{\mathrm{data}}(\mathbf{x})\,\mathrm{d}\mathbf{x},\end{aligned}$$

其中 $s_{\boldsymbol{\phi}}^{(i)}(\mathbf{x})$ 是评分函数的 i -th 分量

$$\mathbf{s}_{\boldsymbol{\phi}} = \left(s_{\boldsymbol{\phi}}^{(1)}, s_{\boldsymbol{\phi}}^{(2)}, \ldots, s_{\boldsymbol{\phi}}^{(D)}\right).$$

Integration by Parts. 我们使用以下分部积分公式 (**evans10**)，该公式由微积分基本定理推导得出：

Lemma. 设 u, v 为半径为 $R > 0$ 的球 $\mathbb{B}(\mathbf{0}, R) \subset \mathbb{R}^D$ 上的可微实值函数。则对于 $i = 1, \ldots, D$ ，有如下公式成立：

$$\int_{\mathbb{B}(\mathbf{0},R)} u \partial_{x_i} v\,\mathrm{d}\mathbf{x} = -\int_{\mathbb{B}(\mathbf{0},R)} v \partial_{x_i} u\,\mathrm{d}\mathbf{x} + \int_{\partial\mathbb{B}(\mathbf{0},R)} uv\nu_i\,\mathrm{d}S,$$

其中 $\nu = (\nu_1, \ldots, \nu_D)$ 是边界 $\partial\mathbb{B}(\mathbf{0}, R)$ 的外单位法向量——半径为 $R > 0$ 的球面，$\mathrm{d}S$ 是 $\partial\mathbb{B}(\mathbf{0}, R)$ 上的面积测度。

我们将此公式应用于所有 $i = 1, \ldots, D$ 的 $u(\mathbf{x}) := s_{\boldsymbol{\phi}}^{(i)}(\mathbf{x})$ 和 $v(\mathbf{x}) = p_{\text{data}}(\mathbf{x})$，假设

$$|u(\mathbf{x})v(\mathbf{x})| \to 0 \quad \text{as } R \to \infty.$$

对所有 $i = 1, \ldots, D$ 求和，我们得到：

$$\begin{aligned}\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\langle \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{s}(\mathbf{x})\rangle\right] &= -\sum_{i=1}^{D}\int \partial_{x_i} s_{\boldsymbol{\phi}}^{(i)}(\mathbf{x}) p_{\text{data}}(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &= -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\mathrm{Tr}\left(\nabla_{\mathbf{x}}\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})\right)\right].\end{aligned}$$

综合所有结果，我们得到：

$$\begin{aligned}\mathcal{L}_{\text{SM}}(\boldsymbol{\phi}) = &\underbrace{\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\mathrm{Tr}\left(\nabla_{\mathbf{x}}\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})\right) + \frac{1}{2}\left\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x})\right\|_2^2\right]}_{\widetilde{\mathcal{L}}_{\text{SM}}(\boldsymbol{\phi})} \\ &+ \underbrace{\frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\left\|\mathbf{s}(\mathbf{x})\right\|_2^2\right]}_{=:C},\end{aligned}$$

其中 C 仅依赖于分布 p_{data}，证明完毕。 ■

D.2.2 定理 3.3.1：SM 与 DSM 最小化之间的等价性

证明。 将 $\mathcal{L}_{\text{SM}}(\boldsymbol{\phi}; \sigma)$ 和 $\mathcal{L}_{\text{DSM}}(\boldsymbol{\phi}; \sigma)$ 展开后，我们得到：

$$\begin{aligned}\mathcal{L}_{\text{SM}}(\boldsymbol{\phi}; \sigma) = \frac{1}{2}\mathbb{E}_{\tilde{\mathbf{x}}\sim p_\sigma(\tilde{\mathbf{x}})}\Big[&\left\|\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}}; \sigma)\right\|_2^2 - 2\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}}; \sigma)^\top \nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}}) \\ &+ \left\|\nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}})\right\|_2^2\Big], \\ \mathcal{L}_{\text{DSM}}(\boldsymbol{\phi}; \sigma) = \frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}\Big[&\left\|\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}}; \sigma)\right\|_2^2 - 2\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}}; \sigma)^\top \nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) \\ &+ \left\|\nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})\right\|_2^2\Big].\end{aligned}$$

将两个方程相减，得到：

$$
\begin{aligned}
&\mathcal{L}_{\mathrm{SM}}(\boldsymbol{\phi};\sigma)-\mathcal{L}_{\mathrm{DSM}}(\boldsymbol{\phi};\sigma)\\
=&\frac{1}{2}\left(\mathbb{E}_{\tilde{\mathbf{x}}\sim p_\sigma(\tilde{\mathbf{x}})}\left\|\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma)\right\|_2^2-\mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}\left\|\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma)\right\|_2^2\right)\\
&-\left(\mathbb{E}_{\tilde{\mathbf{x}}\sim p_\sigma(\tilde{\mathbf{x}})}\left[\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma)^\top\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}})\right]\right.\\
&\qquad\left.-\mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}\left[\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\mathbf{x})^\top\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})\right]\right)\\
&+\frac{1}{2}\left(\mathbb{E}_{\tilde{\mathbf{x}}\sim p_\sigma(\tilde{\mathbf{x}})}\left\|\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}})\right\|_2^2-\mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}\left\|\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})\right\|_2^2\right).
\end{aligned}
$$

接下来，我们逐项处理。对于第一项，由于 $p_\sigma(\tilde{\mathbf{x}})=\int p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})p_{\mathrm{data}}(\mathbf{x})\,\mathrm{d}\mathbf{x}$ ，我们可以将其重写为：

$$
\begin{aligned}
\mathbb{E}_{\tilde{\mathbf{x}}\sim p_\sigma(\tilde{\mathbf{x}})}\left\|\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma)\right\|_2^2&=\int\left(\int p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})p_{\mathrm{data}}(\mathbf{x})\,\mathrm{d}\mathbf{x}\right)\left\|\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma)\right\|_2^2\,\mathrm{d}\tilde{\mathbf{x}}\\
&=\int p_{\mathrm{data}}(\mathbf{x})\int p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})\left\|\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma)\right\|_2^2\,\mathrm{d}\tilde{\mathbf{x}}\,\mathrm{d}\mathbf{x}\\
&=\mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}\left\|\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma)\right\|_2^2.
\end{aligned}
$$

因此，第一项为零。对于第二项：

$$
\begin{aligned}
&\mathbb{E}_{\tilde{\mathbf{x}}\sim p_\sigma(\tilde{\mathbf{x}})}\left[\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma)^\top\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}})\right]\\
=&\int p_\sigma(\tilde{\mathbf{x}})\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma)^\top\frac{\nabla_{\tilde{\mathbf{x}}}p_\sigma(\tilde{\mathbf{x}})}{p_\sigma(\tilde{\mathbf{x}})}\,\mathrm{d}\tilde{\mathbf{x}}\\
=&\int\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma)^\top\nabla_{\tilde{\mathbf{x}}}\int p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})p_{\mathrm{data}}(\mathbf{x})\,\mathrm{d}\mathbf{x}\,\mathrm{d}\tilde{\mathbf{x}}\\
=&\int\int\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma)^\top\nabla_{\tilde{\mathbf{x}}}p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})p_{\mathrm{data}}(\mathbf{x})\,\mathrm{d}\tilde{\mathbf{x}}\,\mathrm{d}\mathbf{x}\\
=&\mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}\left[\mathbf{s}_{\boldsymbol{\phi}}(\tilde{\mathbf{x}};\sigma)^\top\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})\right].
\end{aligned}
\tag{D.2.1}
$$

因此，它也是零。对于第三项，注意到：

$$
C:=\frac{1}{2}\left(\mathbb{E}_{\tilde{\mathbf{x}}\sim p_\sigma(\tilde{\mathbf{x}})}\left\|\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}})\right\|_2^2-\mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}\left\|\nabla_{\tilde{\mathbf{x}}}\log p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})\right\|_2^2\right)
$$

仅依赖于 $p_{\text{data}}(\mathbf{x})$ 和 $p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})$ ，因此相对于 $\boldsymbol{\phi}$ 为常数。 ■

D.2.3 引理 3.3.2：Tweedie 公式

我们首先陈述一个更一般化的 Tweedie 公式形式，该形式考虑了随时间变化的高斯扰动，并在下面提供其证明。

具有时变参数的 Tweedie 恒等式。 设 $\mathbf{x}_t \sim \mathcal{N}(\cdot; \alpha_t \mathbf{x}_0, \sigma_t^2 \mathbf{I})$ 为高斯随机向量。则 Tweedie 公式表明

$$\alpha_t \mathbb{E}_{\mathbf{x}_0 \sim p(\mathbf{x}_0|\mathbf{x}_t)}[\mathbf{x}_0|\mathbf{x}_t] = \mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t),$$

其中期望是关于在观测到 $\mathbf{x}_t$ 后 $\mathbf{x}_0$ 的后验分布 $p(\mathbf{x}_0|\mathbf{x}_t)$ 取的，且 $p_t(\mathbf{x}_t)$ 是 $\mathbf{x}_t$ 的边缘密度。

证明。

边缘密度及其得分。 我们回顾一下，$\mathbf{x}_t$ 的边缘密度由下式给出

$$p_t(\mathbf{x}_t) = \int p_t(\mathbf{x}_t|\mathbf{x}_0) p_0(\mathbf{x}_0) \, \mathrm{d}\mathbf{x}_0.$$

我们现在计算评分函数：

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = \frac{\nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t)}{p_t(\mathbf{x}_t)} = \frac{1}{p_t(\mathbf{x}_t)} \int \nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t|\mathbf{x}_0) p_0(\mathbf{x}_0) \, \mathrm{d}\mathbf{x}_0.$$

因此，我们需要计算条件密度的梯度。

条件与重排的梯度。 条件高斯密度的梯度为：

$$\nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t|\mathbf{x}_0) = -p_t(\mathbf{x}_t|\mathbf{x}_0) \cdot \sigma_t^{-2}(\mathbf{x}_t - \alpha_t \mathbf{x}_0).$$

将其代入前式，得：

$$\begin{aligned}\nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t) &= \int \nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t|\mathbf{x}_0) p_0(\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_0 \\ &= -\sigma_t^{-2}\int (\mathbf{x}_t - \alpha_t\mathbf{x}_0) p_t(\mathbf{x}_t|\mathbf{x}_0) p_0(\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_0 \\ &= -\sigma_t^{-2}\int (\mathbf{x}_t - \alpha_t\mathbf{x}_0) p(\mathbf{x}_0|\mathbf{x}_t) p_t(\mathbf{x}_t)\,\mathrm{d}\mathbf{x}_0 \\ &= -p_t(\mathbf{x}_t)\sigma_t^{-2}\left(\mathbf{x}_t - \alpha_t \mathbb{E}_{p(\mathbf{x}_0|\mathbf{x}_t)}[\mathbf{x}_0|\mathbf{x}_t]\right).\end{aligned}$$

两边同时除以 $p_t(\mathbf{x}_t)$ ，得到：

$$\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t) = -\sigma_t^{-2}\left(\mathbf{x}_t - \alpha_t \mathbb{E}_{p(\mathbf{x}_0|\mathbf{x}_t)}[\mathbf{x}_0|\mathbf{x}_t]\right).$$

整理得：

$$\mathbf{x}_t + \sigma_t^2\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t) = \alpha_t \mathbb{E}_{p(\mathbf{x}_0|\mathbf{x}_t)}[\mathbf{x}_0|\mathbf{x}_t].$$

推导至此完成。

D.2.4 Stein 恒等式与代理 SURE 目标

斯坦因恒等式。 Stein 恒等式是将未知密度下的期望转化为可观测函数及其导数的期望的分部积分技巧，这消去了配分函数，从而在无需计算未知密度或配分函数的情况下，实现无偏、易处理的目标函数和检验。我们首先从最简单的单变量情形开始，然后将其推广到证明 SURE 替代损失所需的形态。

1D，标准正态情形。 若 $z \sim \mathcal{N}(0,1)$ 与 f 具有适当的衰减性，则 Stein 恒等式表述为：

$$\mathbb{E}[f'(z)] = \mathbb{E}[Zf(z)].$$

记 $\phi(z) := \frac{1}{\sqrt{2\pi}}e^{-z^2/2}$ 为一维标准正态密度。证明通过分部积分完成，利用 $\phi'(z) = -z\phi(z)$ 及边界项消失的性质。为了精确说明这一点，我们进行计算

$$\mathbb{E}[f'(Z)] = \int_{-\infty}^{\infty} f'(z)\phi(z)\,\mathrm{d}z.$$

通过分部积分法，结合 $u = f(z)$ 与 $\mathrm{d}v = \phi'(z)\,\mathrm{d}z$ ，我们得到

$$\int f'(z)\phi(z)\,\mathrm{d}z = \Big[f(z)\phi(z)\Big]_{-\infty}^{\infty} - \int f(z)\phi'(z)\,\mathrm{d}z.$$

由于 $\phi'(z) = -z\phi(z)$ 和 $f(z)\phi(z) \to 0$ 满足 $|z| \to \infty$ (衰减条件)，边界项消失，我们得到

$$\mathbb{E}[f'(z)] = \int f(z)z\phi(z)\,\mathrm{d}z = \mathbb{E}[zf(z)].$$

这完成了推导，并证明了斯坦因恒等式的二维情况。

多元标准正态情形。 若 $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$ 且 $g : \mathbb{R}^D \to \mathbb{R}$ ，则 Stein's 恒等式为

$$\mathbb{E}[\nabla g(\mathbf{z})] = \mathbb{E}[\mathbf{z}g(\mathbf{z})].$$

等价地，对于 $\mathbf{u} : \mathbb{R}^D \to \mathbb{R}^D$ ，

$$\mathbb{E}[\nabla_{\tilde{\mathbf{x}}} \cdot \mathbf{u}(\mathbf{z})] = \mathbb{E}[\mathbf{z}^\top \mathbf{u}(\mathbf{z})]. \tag{D.2.2}$$

SURE 所需的恒等式。 其中 $\tilde{\mathbf{x}} = \mathbf{x} + \sigma\mathbf{z}$ ， $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$ ，以及任意适当正则性的向量函数 $\mathbf{a}$ ，

$$\mathbb{E}\big[(\tilde{\mathbf{x}} - \mathbf{x})^\top \mathbf{a}(\tilde{\mathbf{x}})\big|\mathbf{x}\big] = \sigma^2 \mathbb{E}\big[\nabla_{\tilde{\mathbf{x}}} \cdot \mathbf{a}(\tilde{\mathbf{x}})\big|\mathbf{x}\big]. \tag{D.2.3}$$

这是通过应用 Equation (D.2.2) 并使用链式法则得到的。

从条件均方误差推导 SURE。 设 $\mathbf{D} : \mathbb{R}^D \to \mathbb{R}^D$ 为去噪器，并定义

$$R(\mathbf{D}; \mathbf{x}) := \mathbb{E}\left[\|\mathbf{D}(\tilde{\mathbf{x}}) - \mathbf{x}\|_2^2 \big|\mathbf{x}\right].$$

在 $\tilde{\mathbf{x}}$ 附近展开：

$$\begin{aligned}
&R(\mathbf{D};\mathbf{x}) \\
&= \mathbb{E}\left[\|\mathbf{D}(\tilde{\mathbf{x}})-\tilde{\mathbf{x}}\|^2|\mathbf{x}\right] + 2\mathbb{E}\left[(\mathbf{D}(\tilde{\mathbf{x}})-\tilde{\mathbf{x}})^\top(\tilde{\mathbf{x}}-\mathbf{x})|\mathbf{x}\right] + \mathbb{E}\left[\|\tilde{\mathbf{x}}-\mathbf{x}\|^2|\mathbf{x}\right] \\
&= \mathbb{E}\left[\|\mathbf{D}(\tilde{\mathbf{x}})-\tilde{\mathbf{x}}\|^2|\mathbf{x}\right] + 2\Big(\underbrace{\mathbb{E}[(\tilde{\mathbf{x}}-\mathbf{x})^\top\mathbf{D}(\tilde{\mathbf{x}})|\mathbf{x}]}_{\substack{\sigma^2\mathbb{E}[\nabla_{\tilde{\mathbf{x}}}\cdot\mathbf{D}(\tilde{\mathbf{x}})|\mathbf{x}] \\ \text{by Equation (D.2.3)}}} - \underbrace{\mathbb{E}[(\tilde{\mathbf{x}}-\mathbf{x})^\top\tilde{\mathbf{x}}|\mathbf{x}]}_{\sigma^2 D}\Big) \\
&\qquad + \underbrace{\mathbb{E}[\|\tilde{\mathbf{x}}-\mathbf{x}\|^2|\mathbf{x}]}_{\sigma^2 D} \\
&= \mathbb{E}\left[\|\mathbf{D}(\tilde{\mathbf{x}})-\tilde{\mathbf{x}}\|^2 + 2\sigma^2\nabla_{\tilde{\mathbf{x}}}\cdot\mathbf{D}(\tilde{\mathbf{x}}) - D\sigma^2|\mathbf{x}\right].
\end{aligned}$$

因此，可观测的代理

$$\mathrm{SURE}(\mathbf{D};\tilde{\mathbf{x}}) := \|\mathbf{D}(\tilde{\mathbf{x}})-\tilde{\mathbf{x}}\|_2^2 + 2\sigma^2\nabla_{\tilde{\mathbf{x}}}\cdot\mathbf{D}(\tilde{\mathbf{x}}) - D\sigma^2$$

满足 $\mathbb{E}\left[\mathrm{SURE}(\mathbf{D};\tilde{\mathbf{x}})|\mathbf{x}\right] = R(\mathbf{D};\mathbf{x})$。因此，最小化 SURE（在期望意义上或经验上）等价于仅使用噪声观测值来最小化真实的条件均方误差。

D.2.5 定理 4.1.1：通过福克-普朗克方程实现边缘对齐

证明。

第一部分：前向随机微分方程的福克-普朗克方程。 考虑前向随机微分方程：

$$\mathrm{d}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t),t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}(t).$$

扩散矩阵为 $\sigma(t) = g(t)I_D$ ，因此 $\sigma(t)\sigma(t)^T = g^2(t)I_D$ 。变量 $\mathbf{x}(t)$ 的边缘密度 $p_t(\mathbf{x})$ 的福克-普朗克方程为：

$$\partial_t p_t(\mathbf{x}) = -\nabla_{\mathbf{x}}\cdot\left[\mathbf{f}(\mathbf{x},t)p_t(\mathbf{x})\right] + \frac{1}{2}\sum_{i,j=1}^{D}\frac{\partial^2}{\partial x_i\partial x_j}\left[(g^2(t)\delta_{ij})p_t(\mathbf{x})\right].$$

计算扩散项：

$$\sum_{i,j=1}^{D}\frac{\partial^2}{\partial x_i\partial x_j}\left[g^2(t)\delta_{ij}p_t(\mathbf{x})\right] = \sum_{i=1}^{D}\frac{\partial^2}{\partial x_i^2}\left[g^2(t)p_t(\mathbf{x})\right] = g^2(t)\Delta_{\mathbf{x}}p_t(\mathbf{x}).$$

因此：

$$\partial_t p_t(\mathbf{x}) = -\nabla_{\mathbf{x}} \cdot \big[\mathbf{f}(\mathbf{x}, t) p_t(\mathbf{x})\big] + \frac{1}{2} g^2(t) \Delta_{\mathbf{x}} p_t(\mathbf{x}).$$

现在，使用以下内容重写：

$$\tilde{\mathbf{f}}(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t) - \frac{1}{2} g^2(t) \nabla_{\mathbf{x}} \log p_t(\mathbf{x}).$$

由于 $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) = \frac{\nabla_{\mathbf{x}} p_t(\mathbf{x})}{p_t(\mathbf{x})}$ ，计算：

$$\nabla_{\mathbf{x}} \cdot \big[\tilde{\mathbf{f}}(\mathbf{x}, t) p_t(\mathbf{x})\big] = \nabla_{\mathbf{x}} \cdot \left[\mathbf{f}(\mathbf{x}, t) p_t(\mathbf{x}) - \frac{1}{2} g^2(t) \frac{\nabla_{\mathbf{x}} p_t(\mathbf{x})}{p_t(\mathbf{x})} p_t(\mathbf{x})\right].$$

第二项是：

$$\nabla_{\mathbf{x}} \cdot \left[-\frac{1}{2} g^2(t) \nabla_{\mathbf{x}} p_t(\mathbf{x})\right] = -\frac{1}{2} g^2(t) \Delta_{\mathbf{x}} p_t(\mathbf{x}).$$

因此：

$$\nabla_{\mathbf{x}} \cdot \big[\tilde{\mathbf{f}}(\mathbf{x}, t) p_t(\mathbf{x})\big] = \nabla_{\mathbf{x}} \cdot \big[\mathbf{f}(\mathbf{x}, t) p_t(\mathbf{x})\big] - \frac{1}{2} g^2(t) \Delta_{\mathbf{x}} p_t(\mathbf{x}).$$

因此：

$$\partial_t p_t(\mathbf{x}) = -\nabla_{\mathbf{x}} \cdot \big[\tilde{\mathbf{f}}(\mathbf{x}, t) p_t(\mathbf{x})\big],$$

验证福克-普朗克方程的两种形式。

第二部分：PF-ODE 边际密度。 考虑 PF-ODE：

$$\frac{\mathrm{d}\tilde{\mathbf{x}}(t)}{\mathrm{d}t} = \tilde{\mathbf{f}}(\tilde{\mathbf{x}}(t), t), \quad \tilde{\mathbf{f}}(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t) - \frac{1}{2} g^2(t) \nabla_{\mathbf{x}} \log p_t(\mathbf{x}).$$

正向方向： $\tilde{\mathbf{x}}(0) \sim p_0$。令 $\tilde{\mathbf{x}}(t)$ 沿着 PF-ODE 进行演化，其中 $\tilde{\mathbf{x}}(0) \sim p_0$。流映射 $\boldsymbol{\Psi}_t : \mathbb{R}^D \to \mathbb{R}^D$ 定义为：

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\Psi}_t(\mathbf{x}_0) = \tilde{\mathbf{f}}(\boldsymbol{\Psi}_t(\mathbf{x}_0), t), \quad \boldsymbol{\Psi}_0(\mathbf{x}_0) = \mathbf{x}_0.$$

自从 $\tilde{\mathbf{x}}(t) = \boldsymbol{\Psi}_t(\tilde{\mathbf{x}}(0))$ 以来，密度 $\tilde{p}_t(\mathbf{x})$ 的 $\tilde{\mathbf{x}}(t)$ 满足连续性方程：

$$\partial_t \tilde{p}_t(\mathbf{x}) = -\nabla_{\mathbf{x}} \cdot \big[\tilde{\mathbf{f}}(\mathbf{x}, t) \tilde{p}_t(\mathbf{x})\big].$$

由于 $\tilde{\mathbf{x}}(0) \sim p_0$ ，我们有 $\tilde{p}_0(\mathbf{x}) = p_0(\mathbf{x})$ 。由第 1 部分，$p_t(\mathbf{x})$ 满足：

$$\partial_t p_t(\mathbf{x}) = -\nabla_{\mathbf{x}} \cdot \left[\tilde{\mathbf{f}}(\mathbf{x}, t) p_t(\mathbf{x})\right].$$

$\tilde{p}_t$ 与 p_t 满足相同的连续性方程，且具有相同的初值条件 p_0 。假设足够光滑（例如 $\tilde{\mathbf{f}} \in C^1$ ），则解在某个适当的函数空间中是唯一的，因此 $\tilde{p}_t = p_t$ 。于是，对所有 $t \in [0, T]$ ，有 $\tilde{\mathbf{x}}(t) \sim p_t$ 。

反向路径：$\tilde{\mathbf{x}}(T) \sim p_T$。现在，令 $\tilde{\mathbf{x}}(t)$ 从 $t = T$ 逆向沿 PF-ODE 走到 $t = 0$ ，其中 $\tilde{\mathbf{x}}(T) \sim p_T$ 。该常微分方程为：

$$\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{x}}(t) = \tilde{\mathbf{f}}(\tilde{\mathbf{x}}(t), t).$$

令 $s = T - t$ ，则反向微分方程变为：

$$\frac{\mathrm{d}}{\mathrm{d}s}\tilde{\mathbf{x}}(T - s) = -\tilde{\mathbf{f}}(\tilde{\mathbf{x}}(T - s), T - s).$$

密度 $\tilde{p}_{T-s}(\mathbf{x})$ 的 $\tilde{\mathbf{x}}(T - s)$ 满足：

$$\partial_s \tilde{p}_{T-s}(\mathbf{x}) = \nabla_{\mathbf{x}} \cdot \left[\tilde{\mathbf{f}}(\mathbf{x}, T - s)\tilde{p}_{T-s}(\mathbf{x})\right].$$

由于 $\tilde{\mathbf{x}}(T) \sim p_T$ ，我们有 $\tilde{p}_T = p_T$ 。在 $t = T - s$ 时，p_t 的福克-普朗克方程为：

$$\partial_t p_{T-s}(\mathbf{x}) = -\nabla_{\mathbf{x}} \cdot \left[\tilde{\mathbf{f}}(\mathbf{x}, T - s) p_{T-s}(\mathbf{x})\right].$$

由于 $\partial_t = -\partial_s$ ，我们得到：

$$\partial_s p_{T-s}(\mathbf{x}) = \nabla_{\mathbf{x}} \cdot \left[\tilde{\mathbf{f}}(\mathbf{x}, T - s) p_{T-s}(\mathbf{x})\right].$$

$\tilde{p}_{T-s}$ 与 p_{T-s} 在 $s = 0$ 处满足相同的初值条件（$\tilde{p}_T = p_T$ ），且满足相同的偏微分方程。由唯一性可知 $\tilde{p}_{T-s} = p_{T-s}$ ，因此 $\tilde{\mathbf{x}}(t) = \tilde{\mathbf{x}}(T - s) \sim p_{T-s} = p_t$ ，对所有 $t \in [0, T]$ 成立。

第三部分：**反向时间 SDE 的边缘密度**。 考虑反向时间的随机微分方程：

$$\mathrm{d}\bar{\mathbf{x}}(t) = \left[\mathbf{f}(\bar{\mathbf{x}}(t), t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\bar{\mathbf{x}}(t))\right] \mathrm{d}t + g(t)\, \mathrm{d}\bar{\mathbf{w}}(t),$$

其中 $\bar{\mathbf{x}}(0) \sim p_T$ ， $\bar{\mathbf{w}}(t)$ 为反向时间下的标准维纳过程，定义为 $\bar{\mathbf{w}}(t) = \mathbf{w}(T - t) - \mathbf{w}(T)$ 。我们需要证明 $\bar{\mathbf{x}}(t) \sim p_{T-t}$ 。

重写漂移：

$$\mathbf{f}(\mathbf{x}, t) = \tilde{\mathbf{f}}(\mathbf{x}, t) + \frac{1}{2} g^2(t) \nabla_{\mathbf{x}} \log p_t(\mathbf{x}),$$

所以：

$$\mathbf{f}(\mathbf{x}, t) - g^2(t) \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) = \tilde{\mathbf{f}}(\mathbf{x}, t) - \frac{1}{2} g^2(t) \nabla_{\mathbf{x}} \log p_t(\mathbf{x}).$$

反向时间 SDE 为：

$$\mathrm{d}\bar{\mathbf{x}}(t) = \left[\tilde{\mathbf{f}}(\bar{\mathbf{x}}(t), t) - \frac{1}{2} g^2(t) \nabla_{\mathbf{x}} \log p_t(\bar{\mathbf{x}}(t)) \right] \mathrm{d}t + g(t) \, \mathrm{d}\bar{\mathbf{w}}(t).$$

令 $s = T - t$ ， 则 $\bar{\mathbf{x}}(t) = \bar{\mathbf{x}}(T - s)$ ， 且 $\mathrm{d}t = -\,\mathrm{d}s$ 。随机微分方程变为：

$$\begin{aligned} \mathrm{d}\bar{\mathbf{x}}(T - s) = & \left[-\tilde{\mathbf{f}}(\bar{\mathbf{x}}(T - s), T - s) + \frac{1}{2} g^2(T - s) \nabla_{\mathbf{x}} \log p_{T-s}(\bar{\mathbf{x}}(T - s)) \right] \mathrm{d}s \\ & + g(T - s) \, \mathrm{d}\bar{\mathbf{w}}(T - s). \end{aligned}$$

由于 $\bar{\mathbf{w}}(t) = \mathbf{w}(T - t) - \mathbf{w}(T)$ ， 我们有 $\mathrm{d}\bar{\mathbf{w}}(T - s) = -\,\mathrm{d}\mathbf{w}(s)$ ， 其中 $\mathbf{w}(s)$ 为标准维纳过程。因此，令 $\bar{\mathbf{w}}'(s) = -\mathbf{w}(s)$ ， 为标准维纳过程，于是：

$$\mathrm{d}\bar{\mathbf{x}}(s) = \left[\tilde{\mathbf{f}}(\bar{\mathbf{x}}(s), T - s) - \frac{1}{2} g^2(T - s) \nabla_{\mathbf{x}} \log p_{T-s}(\bar{\mathbf{x}}(s)) \right] \mathrm{d}s + g(T - s) \, \mathrm{d}\bar{\mathbf{w}}'(s).$$

Fokker-Planck 方程为密度 $\bar{p}_s(\mathbf{x})$ 的 $\bar{\mathbf{x}}(s)$ 是：

$$\begin{aligned} \partial_s \bar{p}_s(\mathbf{x}) = & -\nabla_{\mathbf{x}} \cdot \left[\left(\tilde{\mathbf{f}}(\mathbf{x}, T - s) - \frac{1}{2} g^2(T - s) \nabla_{\mathbf{x}} \log p_{T-s}(\mathbf{x}) \right) \bar{p}_s(\mathbf{x}) \right] \\ & + \frac{1}{2} g^2(T - s) \Delta_{\mathbf{x}} \bar{p}_s(\mathbf{x}). \end{aligned}$$

假设 $\bar{p}_s = p_{T-s}$。关于 p_{T-s} 的福克-普朗克方程为：

$$\partial_t p_{T-s}(\mathbf{x}) = -\nabla_{\mathbf{x}} \cdot \left[\tilde{\mathbf{f}}(\mathbf{x}, T - s) p_{T-s}(\mathbf{x}) \right].$$

由于 $\partial_t = -\partial_s$ ：

$$\partial_s p_{T-s}(\mathbf{x}) = \nabla_{\mathbf{x}} \cdot \left[\tilde{\mathbf{f}}(\mathbf{x}, T - s) p_{T-s}(\mathbf{x}) \right].$$

将 $\bar{p}_s = p_{T-s}$ 代入：

$$\partial_s p_{T-s} = -\nabla_{\mathbf{x}} \cdot \left[\tilde{\mathbf{f}}(\mathbf{x}, T-s) p_{T-s}(\mathbf{x}) - \frac{1}{2} g^2(T-s) \frac{\nabla_{\mathbf{x}} p_{T-s}(\mathbf{x})}{p_{T-s}(\mathbf{x})} p_{T-s}(\mathbf{x}) \right] + \frac{1}{2} g^2(T-s) \Delta_{\mathbf{x}} p_{T-s}(\mathbf{x}).$$

额外的项是：

$$\begin{aligned} & -\nabla_{\mathbf{x}} \cdot \left[-\frac{1}{2} g^2(T-s) \nabla_{\mathbf{x}} p_{T-s}(\mathbf{x}) \right] + \frac{1}{2} g^2(T-s) \Delta_{\mathbf{x}} p_{T-s}(\mathbf{x}) \\ =& \frac{1}{2} g^2(T-s) \Delta_{\mathbf{x}} p_{T-s}(\mathbf{x}) - \frac{1}{2} g^2(T-s) \Delta_{\mathbf{x}} p_{T-s}(\mathbf{x}) = 0. \end{aligned}$$

因此，$\bar{p}_s = p_{T-s}$ 满足福克-普朗克方程。由于 $\bar{\mathbf{x}}(0) \sim p_T$ ，我们有 $\bar{p}_0 = p_T$ ，符合初始条件。在足够光滑性条件下，唯一性保证 $\bar{p}_s = p_{T-s}$ ，所以 $\bar{\mathbf{x}}(t) = \bar{\mathbf{x}}(T-s) \sim p_{T-t}$ 。 ■

D.2.6 命题 4.2.1：SM 和 DSM 的最小化器

证明。 为了找到最小化器 $\mathbf{s}^*$ ，我们首先考虑一个固定的时间 t ，并分析目标函数中的内层期望：

$$\mathcal{J}(t, \boldsymbol{\phi}) := \mathbb{E}_{\mathbf{x}_0 \sim p_{\text{data}}} \mathbb{E}_{\mathbf{x}_t \sim p_t(\cdot|\mathbf{x}_0)} \left[\| \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0) \|_2^2 \right].$$

为了使该期望最小化，我们需要找到 $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)$ 使得对每个 $\mathbf{x}_t$ 的期望平方误差最小。我们可以利用 X_0 和 X_t 的联合分布重写此期望：

$$\mathcal{J}(t, \boldsymbol{\phi}) = \iint p_{\text{data}}(\mathbf{x}_0) p_t(\mathbf{x}_t|\mathbf{x}_0) \| \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{x}_0) \|_2^2 \, \mathrm{d}\mathbf{x}_0 \, \mathrm{d}\mathbf{x}_t.$$

对于每个固定的 $\mathbf{x}_t$ ，我们需要进行最小化：

$$\int p(\mathbf{x}_0|X_t = \mathbf{x}_t) p_t(\mathbf{x}_t) \| \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{x}_0) \|_2^2 \, \mathrm{d}\mathbf{x}_0.$$

由于 $p_t(\mathbf{x}_t)$ 相对于 $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)$ 为常数，这等价于最小化：

$$\int p(\mathbf{x}_0|X_t = \mathbf{x}_t) \| \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{x}_0) \|_2^2 \, \mathrm{d}\mathbf{x}_0$$

当 $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)$ 等于条件期望时，该式达到最小值：

$$\mathbf{s}^*(\mathbf{x}_t, t) = \mathbb{E}_{X_0 \sim p(X_0|X_t=\mathbf{x}_t)} \left[\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|X_0)\right].$$

现在我们需要证明其等于 $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$。根据贝叶斯规则和边缘概率的定义：

$$p_t(\mathbf{x}_t) = \int p_t(\mathbf{x}_t|\mathbf{x}_0) p_{\text{data}}(\mathbf{x}_0)\, \mathrm{d}\mathbf{x}_0.$$

对数化后再对 $\mathbf{x}_t$ 求梯度：

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = \frac{\nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t)}{p_t(\mathbf{x}_t)} = \frac{\nabla_{\mathbf{x}_t} \int p_t(\mathbf{x}_t|\mathbf{x}_0) p_{\text{data}}(\mathbf{x}_0)\, \mathrm{d}\mathbf{x}_0}{\int p_t(\mathbf{x}_t|\mathbf{x}_0) p_{\text{data}}(\mathbf{x}_0)\, \mathrm{d}\mathbf{x}_0}.$$

在适当的正则性条件下，我们可以交换梯度与积分运算：

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = \frac{\int \nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t|\mathbf{x}_0) p_{\text{data}}(\mathbf{x}_0)\, \mathrm{d}\mathbf{x}_0}{\int p_t(\mathbf{x}_t|\mathbf{x}_0) p_{\text{data}}(\mathbf{x}_0)\, \mathrm{d}\mathbf{x}_0}.$$

■

D.2.7 高斯分布的闭式评分函数

为便于今后参考，我们将一般多元正态分布的得分公式总结如下引理：

Lemma D.2.1: Score of Gaussian

Let $\mathbf{x} \in \mathbb{R}^D$ and consider the multivariate normal distribution

$$p(\tilde{\mathbf{x}}|\mathbf{x}) := \mathcal{N}(\tilde{\mathbf{x}}; \boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

where $\boldsymbol{\mu} \in \mathbb{R}^D$ is the mean and $\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}$ is an invertible covariance matrix. Its score function is

$$\nabla_{\tilde{\mathbf{x}}} \log p(\tilde{\mathbf{x}}|\mathbf{x}) = -\boldsymbol{\Sigma}^{-1}(\tilde{\mathbf{x}} - \boldsymbol{\mu}). \tag{D.2.4}$$

Proof for Lemma.

The density function of $p(\tilde{\mathbf{x}}|\mathbf{x})$ is given by:

$$p(\tilde{\mathbf{x}}|\mathbf{x}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\tilde{\mathbf{x}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\tilde{\mathbf{x}} - \boldsymbol{\mu})\right).$$

To compute the score function, we first take the log of $p(\tilde{\mathbf{x}}|\mathbf{x})$:

$$\log p(\tilde{\mathbf{x}}|\mathbf{x}) = -\frac{D}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}(\tilde{\mathbf{x}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\tilde{\mathbf{x}} - \boldsymbol{\mu}).$$

Now, we compute the gradient of $\log p(\tilde{\mathbf{x}}|\mathbf{x})$ with respect to $\tilde{\mathbf{x}}$:

$$\nabla_{\tilde{\mathbf{x}}} \log p(\tilde{\mathbf{x}}|\mathbf{x}) = -\frac{1}{2}\nabla_{\tilde{\mathbf{x}}}\left((\tilde{\mathbf{x}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\tilde{\mathbf{x}} - \boldsymbol{\mu})\right).$$

Using the chain rule, we get:

$$\nabla_{\tilde{\mathbf{x}}}\left((\tilde{\mathbf{x}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\tilde{\mathbf{x}} - \boldsymbol{\mu})\right) = 2\boldsymbol{\Sigma}^{-1}(\tilde{\mathbf{x}} - \boldsymbol{\mu}).$$

Thus, the score function is:

$$\nabla_{\tilde{\mathbf{x}}} \log p(\tilde{\mathbf{x}}|\mathbf{x}) = -\boldsymbol{\Sigma}^{-1}(\tilde{\mathbf{x}} - \boldsymbol{\mu}). \tag{D.2.5}$$

D.3 基于流的视角

D.3.1 引理 5.1.1：瞬时变量变换

证明。 **方法 1：变量变换公式。**我们将 $p(\mathbf{x}(t),t)$ 记为 $p_t(\mathbf{x}_t)$ 。从常微分方程的离散化开始

$$\mathbf{z}_{t+\Delta t} = \mathbf{z}_t + \Delta t \mathbf{F}(\mathbf{z}_t, t),$$

变换变量公式用于归一化流 (Equation (5.1.1)) 给出

$$\begin{aligned}
\log p_{t+\Delta t}(\mathbf{z}_{t+\Delta t}) &= \log p_t(\mathbf{z}_t) - \log \left| \det \left(\mathbf{I} + \Delta t \nabla_{\mathbf{z}} \mathbf{F}(\mathbf{z}_t, t) \right) \right| \\
&= \log p_t(\mathbf{z}_t) - \operatorname{Tr} \Big(\log(\mathbf{I} + \Delta t \nabla_{\mathbf{z}} \mathbf{F}(\mathbf{z}_t, t)) \Big) \\
&= \log p_t(\mathbf{z}_t) - \Delta t \operatorname{Tr} \left(\nabla_{\mathbf{z}} \mathbf{F}(\mathbf{z}_t, t) \right) + \mathcal{O}(\Delta t^2),
\end{aligned}$$

其中我们使用了 $\log\det = \operatorname{Tr}\log$ 以及对小量 Δt 的展开。取极限 $\Delta t \to 0$ 可得对数密度的连续时间微分方程。事实上，Equation (5.1.6) 中也应用了同样的技巧。

方法 2：连续性方程。我们也可以利用连续性方程，它本质上是变量变换公式：

$$\partial_t p(\mathbf{z}, t) = -\nabla_{\mathbf{z}} \cdot \big(\mathbf{F}(\mathbf{z}, t) p(\mathbf{z}, t) \big).$$

展开散度，

$$\partial_t p = -\big((\nabla_{\mathbf{z}} \cdot \mathbf{F}) p + \mathbf{F} \cdot \nabla_{\mathbf{z}} p \big).$$

沿满足 $\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} = \mathbf{F}(\mathbf{z}(t), t)$ 的轨迹 $\mathbf{z}(t)$ ，总导数为

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} p(\mathbf{z}(t), t) &= \nabla_{\mathbf{z}} p \cdot \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} + \partial_t p \\
&= \nabla_{\mathbf{z}} p \cdot \mathbf{F} - \big((\nabla_{\mathbf{z}} \cdot \mathbf{F}) p + \mathbf{F} \cdot \nabla_{\mathbf{z}} p \big) \\
&= -(\nabla_{\mathbf{z}} \cdot \mathbf{F}) p.
\end{aligned}$$

除以 $p(\mathbf{z}(t), t)$ ，我们得出

$$\frac{\mathrm{d}}{\mathrm{d}t} \log p(\mathbf{z}(t), t) = -\nabla_{\mathbf{z}} \cdot \mathbf{F}(\mathbf{z}(t), t).$$

■

D.3.2 定理 5.2.2：质量守恒准则

一些先决条件：流场图与流动诱导密度。 对于任意初值位置 $\mathbf{x}_0 \in \mathbb{R}^D$，流映射 $\boldsymbol{\Psi}_t : \mathbb{R}^D \to \mathbb{R}^D$ 是该常微分方程的唯一解

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\Psi}_t(\mathbf{x}_0) = \mathbf{v}_t\big(\boldsymbol{\Psi}_t(\mathbf{x}_0)\big), \quad \boldsymbol{\Psi}_0(\mathbf{x}_0) = \mathbf{x}_0.$$

在我们的正则性假设下，$\boldsymbol{\Psi}_t$ 关于 t 和 $\mathbf{x}_0$ 均是连续可微的。

流动诱导的密度 p_t^{fwd} 是初始密度 p_0 通过 $\boldsymbol{\Psi}_t$ 的前推：

$$p_t^{\mathrm{fwd}}(\mathbf{x}) = p_0\big(\boldsymbol{\Psi}_t^{-1}(\mathbf{x})\big)\Big|\det\big(\nabla\boldsymbol{\Psi}_t^{-1}(\mathbf{x})\big)\Big|.$$

这给出了在时间 t 从 $p_0 = p_{\mathrm{data}}$ 出发并受速度场 $\mathbf{v}_t$ 演化而来的粒子在 $\mathbf{x}$ 处的密度。

非正式证明： **充分条件**：$p_t^{\mathrm{fwd}} = p_t \Rightarrow$ **连续性方程**。在 Section B.1.2 中，我们通过取离散变量变换公式的连续时间极限，得到了连续性方程的 *强解*。在该方法中，假设密度 p_t 具有足够的光滑性，使得所有导数在经典意义下存在且偏微分方程逐点成立。在此，我们提供一种互补的推导方式，即在 *弱意义*下进行：连续性方程仅在对任意光滑检验函数积分后被施加，这降低了对 p_t 以及速度场 $\mathbf{v}_t$ 的正则性要求。这种弱形式不仅更具严谨性（因为它能容纳正则性较低的解），而且是偏微分方程理论和数值分析中的标准框架。

对于任意具有紧支集的光滑测试函数 $\varphi(\mathbf{x})$，前推性质给出：

$$\begin{aligned}\int p_t^{\mathrm{fwd}}(\mathbf{x})\varphi(\mathbf{x})\,\mathrm{d}\mathbf{x} &= \int p_0(\boldsymbol{\Psi}_t^{-1}(\mathbf{x}))\left|\det\left(\nabla\boldsymbol{\Psi}_t^{-1}(\mathbf{x})\right)\right|\varphi(\mathbf{x})\,\mathrm{d}\mathbf{x}\\ &= \int p_0(\mathbf{y})\varphi(\boldsymbol{\Psi}_t(\mathbf{y}))\,\mathrm{d}\mathbf{y},\end{aligned}$$

其中第二个等式由变量替换 $\mathbf{x} = \boldsymbol{\Psi}_t(\mathbf{y})$ 得到，其中 $\mathrm{d}\mathbf{y} = \left|\det\left(\nabla\boldsymbol{\Psi}_t^{-1}(\mathbf{x})\right)\right|\mathrm{d}\mathbf{x}$ 。

对两边关于 t 求导：

$$\frac{\mathrm{d}}{\mathrm{d}t}\int p_t^{\mathrm{fwd}}(\mathbf{x})\varphi(\mathbf{x})\,\mathrm{d}\mathbf{x} = \frac{\mathrm{d}}{\mathrm{d}t}\int p_0(\mathbf{y})\varphi(\boldsymbol{\Psi}_t(\mathbf{y}))\,\mathrm{d}\mathbf{y}.$$

左边是：

$$\int \frac{\partial p_t^{\mathrm{fwd}}}{\partial t}(\mathbf{x})\varphi(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

在右侧：

$$\int p_0(\mathbf{y})\nabla\varphi(\mathbf{\Psi}_t(\mathbf{y}))\cdot\mathbf{v}_t(\mathbf{\Psi}_t(\mathbf{y}))\,\mathrm{d}\mathbf{y},$$

由于 $\frac{\partial\mathbf{\Psi}_t}{\partial t}(\mathbf{y})=\mathbf{v}_t(\mathbf{\Psi}_t(\mathbf{y}))$ 。将变量变为 $\mathbf{x}=\mathbf{\Psi}_t(\mathbf{y})$ ，因此

$$\mathrm{d}\mathbf{y}=\left|\det\left(\nabla\mathbf{\Psi}_t^{-1}(\mathbf{x})\right)\right|\mathrm{d}\mathbf{x},$$

并且

$$p_0(\mathbf{y})=p_t^{\mathrm{fwd}}(\mathbf{x})\left|\det\left(\nabla\mathbf{\Psi}_t(\mathbf{y})\right)\right|=\frac{p_t^{\mathrm{fwd}}(\mathbf{x})}{\left|\det\left(\nabla\mathbf{\Psi}_t^{-1}(\mathbf{x})\right)\right|}.$$

因此，右边变为：

$$\int p_t^{\mathrm{fwd}}(\mathbf{x})\nabla\varphi(\mathbf{x})\cdot\mathbf{v}_t(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

使用分部积分法，利用 φ 的紧支集：

$$\int p_t^{\mathrm{fwd}}(\mathbf{x})\nabla\varphi(\mathbf{x})\cdot\mathbf{v}_t(\mathbf{x})\,\mathrm{d}\mathbf{x}=-\int\varphi(\mathbf{x})\nabla\cdot(p_t^{\mathrm{fwd}}(\mathbf{x})\mathbf{v}_t(\mathbf{x}))\,\mathrm{d}\mathbf{x}.$$

两边相等：

$$\int\left[\frac{\partial p_t^{\mathrm{fwd}}}{\partial t}(\mathbf{x})+\nabla\cdot(p_t^{\mathrm{fwd}}(\mathbf{x})\mathbf{v}_t(\mathbf{x}))\right]\varphi(\mathbf{x})\,\mathrm{d}\mathbf{x}=0.$$

由于 φ 是任意的，我们得出：

$$\frac{\partial p_t^{\mathrm{fwd}}}{\partial t}+\nabla\cdot(p_t^{\mathrm{fwd}}\mathbf{v}_t)=0.$$

给定 $p_t^{\mathrm{fwd}}=p_t$ ，可得：

$$\frac{\partial p_t}{\partial t}+\nabla\cdot(p_t\mathbf{v}_t)=0.$$

必要条件：**连续性方程** $\Rightarrow p_t^{\mathrm{fwd}}=p_t$ 。假设 p_t 满足连续性方程：

$$\frac{\partial p_t}{\partial t}+\nabla\cdot(p_t\mathbf{v}_t)=0,$$

初始条件为 $p_0(\mathbf{x})=p_{\mathrm{data}}(\mathbf{x})$ 。我们可知 p_t^{fwd} 满足相同的连续性方程，如上所示，且：

$$p_0^{\mathrm{fwd}}(\mathbf{x})=p_0(\mathbf{\Psi}_0^{-1}(\mathbf{x}))\left|\det\left(\nabla\mathbf{\Psi}_0^{-1}(\mathbf{x})\right)\right|=p_0(\mathbf{x}),$$

由于 $\mathbf{\Psi}_0(\mathbf{x})=\mathbf{x}$。因此，两个密度具有相同的初始条件 $p_0=p_{\mathrm{data}}$。

连续性方程可改写为：

$$\frac{\partial p}{\partial t} + \mathbf{v}_t \cdot \nabla p + p \nabla \cdot \mathbf{v}_t = 0.$$

这是一个一阶线性偏微分方程。假设 $\mathbf{v}_t$ 连续可微且全局利普希茨连续，p_t 足够光滑，则特征线方法保证在光滑函数空间中存在唯一解。由于 p_t 和 p_t^{fwd} 满足相同的偏微分方程和初始条件，我们得出结论：

$$p_t(\mathbf{x}) = p_t^{\text{fwd}}(\mathbf{x})$$

对所有 $t \in [0, 1]$ 和 $\mathbf{x} \in \mathbb{R}^D$。

这就完成了等价性的证明。 ■

D.4 理论补充：对扩散模型的统一与系统性视角

D.4.1 命题 6.3.1：参数化的等价性

证明： 与定理 4.2.1 中关于 DSM 损失的情况类似，匹配损失的全局最优解

$$\mathbb{E}_t \left[\omega(t) \, \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[\| \cdot \|_2^2 \right] \right]$$

当内部期望达到时

$$\mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[\| \cdot \|_2^2 \right]$$

对于每个固定的 t，该式被最小化。由于这是一个标准的均方误差问题，最小值是唯一的。从降噪分数匹配 (**vincent2011connection**)，定理 4.2.1 表明最优评分函数满足

$$\mathbf{s}^*(\mathbf{x}_t, t) = \mathbb{E}_{p(\mathbf{x}_0|\mathbf{x}_t)} \left[\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t|\mathbf{x}_0) \right] = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t).$$

利用恒等式 $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t|\mathbf{x}_0) = -\frac{1}{\sigma_t}\boldsymbol{\epsilon}$ 对 $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ，我们得到

$$\mathbf{s}^*(\mathbf{x}_t, t) = -\frac{1}{\sigma_t} \boldsymbol{\epsilon}^*(\mathbf{x}_t, t),$$

其中 $\boldsymbol{\epsilon}^*(\mathbf{x}_t, t) = \mathbb{E}_{p(\mathbf{x}_0|\mathbf{x}_t)}[\boldsymbol{\epsilon}|\mathbf{x}_t]$ 是针对 $\mathcal{L}_{\text{noise}}(\boldsymbol{\phi})$ 的最优 $\boldsymbol{\epsilon}$ 预测。对于 $\mathbf{x}$ 预测损失 $\mathcal{L}_{\text{clean}}$ ，最优估计量是

$$\mathbf{x}^*(\mathbf{x}_t, t) = \mathbb{E}_{p(\mathbf{x}_0|\mathbf{x}_t)}[\mathbf{x}_0|\mathbf{x}_t],$$

根据特威迪公式，这与得分相关

$$\alpha_t \, \mathbf{x}^*(\mathbf{x}_t, t) = \mathbf{x}_t + \sigma_t^2 \, \mathbf{s}^*(\mathbf{x}_t, t).$$

对于 $\mathbf{v}$ -预测损失 $\mathcal{L}_{\text{velocity}}$ ，最优估计量是

$$\begin{aligned} \mathbf{v}^*(\mathbf{x}_t, t) &= \mathbb{E}_{p(\mathbf{x}_0|\mathbf{x}_t)}[\alpha_t' \mathbf{x}_0 + \sigma_t' \boldsymbol{\epsilon}|\mathbf{x}_t] \\ &= \alpha_t' \mathbf{x}^* + \sigma_t' \boldsymbol{\epsilon}^*. \end{aligned}$$

将 $\mathbf{x}^*$ 和 $\boldsymbol{\epsilon}^*$ 用 $\mathbf{s}^*$ 表示的表达式代入，得到

$$\begin{aligned}\mathbf{v}^*(\mathbf{x}_t, t) &= \frac{\alpha_t'}{\alpha_t}\mathbf{x}_t + \left(\frac{\alpha_t'}{\alpha_t}\sigma_t^2 - \sigma_t\sigma_t'\right)\mathbf{s}^*(\mathbf{x}_t, t) \\ &= f(t)\mathbf{x}_t - \frac{1}{2}g^2(t)\mathbf{s}^*(\mathbf{x}_t, t),\end{aligned}$$

这就完成了推导。

■

D.5 理论补充：快速扩散基础生成器的学习

D.5.1 知识蒸馏损失作为通用框架的一个实例 Equation (10.1.4)

我们从 Oracle KD 损失开始

$$\mathcal{L}_{\mathrm{KD}}^{\mathrm{oracle}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}_T \sim p_T} \left\| \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_T, T, 0) - \boldsymbol{\Psi}_{T\to 0}(\mathbf{x}_T) \right\|_2^2,$$

其中包含 $p_T = p_{\mathrm{prior}}$ 。对于确定性常微分方程流映射 $\boldsymbol{\Psi}$ （沿轨迹构成半群且双射），边际分布满足推送前恒等式

$$p_t = \boldsymbol{\Psi}_{0\to t} \sharp\, p_{\mathrm{data}} = \boldsymbol{\Psi}_{T\to t} \sharp\, p_{\mathrm{prior}};$$

因此，

$$\boldsymbol{\Psi}_{s\to T} \sharp\, p_s = p_T \quad \text{and} \quad \boldsymbol{\Psi}_{T\to 0} \circ \boldsymbol{\Psi}_{s\to T} = \boldsymbol{\Psi}_{s\to 0}.$$

变量替换 $\mathbf{x}_T = \boldsymbol{\Psi}_{s\to T}(\mathbf{x}_s)$ 为 $\mathbf{x}_s \sim p_s$ 得

$$\mathcal{L}_{\mathrm{KD}}^{\mathrm{oracle}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}_s \sim p_s} \Big\| \mathbf{G}_{\boldsymbol{\theta}}\big(\boldsymbol{\Psi}_{s\to T}(\mathbf{x}_s), T, 0\big) - \boldsymbol{\Psi}_{s\to 0}(\mathbf{x}_s) \Big\|_2^2.$$

定义拉回的学生 $\widetilde{\mathbf{G}}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, 0) := \mathbf{G}_{\boldsymbol{\theta}}(\boldsymbol{\Psi}_{s\to T}(\mathbf{x}_s), T, 0)$ 以在统一的流映射形式下表达相同的损失（在 $t = 0$ 处）：

$$\mathcal{L}_{\mathrm{KD}}^{\mathrm{oracle}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}_s \sim p_s} \big\| \widetilde{\mathbf{G}}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, 0) - \boldsymbol{\Psi}_{s\to 0}(\mathbf{x}_s) \big\|_2^2.$$

该推导依赖于通过预言机流进行变量变换以及半群性质。

■

D.5.2 参数–流解释对于命题 10.2.1

由命题 10.2.1 的推导可知，VSD 的梯度可被解释为参数诱导的传输流，其中调整模型参数会移动数据空间中的粒子，使其运动方向与学生分布和教师分布之间的得分差异相匹配。

令 $t \sim p(t)$ ， $\mathbf{z} \sim p(\mathbf{z})$ ， $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 和

$$\hat{\mathbf{x}}_t = \alpha_t\, \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z}) + \sigma_t\, \boldsymbol{\epsilon}.$$

定义 样本（粒子）速度

$$\hat{\mathbf{v}}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t) := \partial_{\boldsymbol{\theta}}\hat{\mathbf{x}}_t = \alpha_t\, \partial_{\boldsymbol{\theta}}\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z}),$$

以及在 $\mathbf{x}$–空间中的速度场作为条件平均

$$\mathbf{v}_{\boldsymbol{\theta}}(\mathbf{x}) := \mathbb{E}\big[\hat{\mathbf{v}}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t)\big|\hat{\mathbf{x}}_t = \mathbf{x}\big].$$

根据此定义，对于每个固定的 t，密度服从参数-流连续性方程

$$\partial_{\boldsymbol{\theta}} p_t^{\boldsymbol{\theta}}(\mathbf{x}) + \nabla_{\mathbf{x}} \cdot \big(p_t^{\boldsymbol{\theta}}(\mathbf{x})\mathbf{v}_{\boldsymbol{\theta}}(\mathbf{x})\big) = 0.$$

此处 $\mathbf{v}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t) = \partial_{\boldsymbol{\theta}}\hat{\mathbf{x}}_t$ 是数据空间中的 参数诱导粒子速度（固定 t 时）。等价地，在每个固定的 t 下，密度满足 $\boldsymbol{\theta}$ 中的连续性方程：

$$\partial_{\boldsymbol{\theta}} p_t^{\boldsymbol{\theta}}(\mathbf{x}) + \nabla_{\mathbf{x}} \cdot \big(p_t^{\boldsymbol{\theta}}(\mathbf{x})\mathbf{v}_{\boldsymbol{\theta}}(\mathbf{x})\big) = 0.$$

因此，$\mathcal{L}_{\mathrm{VSD}}$ 的梯度具有输运形式，

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{VSD}} = \mathbb{E}\big[\omega(t)\big\langle \underbrace{\nabla_{\mathbf{x}}\log p_t^{\boldsymbol{\theta}} - \nabla_{\mathbf{x}}\log p_t}_{\text{score mismatch at fixed } t},\ \underbrace{\mathbf{v}_{\boldsymbol{\theta}}}_{\text{parameter–flow velocity}} \big\rangle\big],$$

即调整参数-流，使粒子运动与局部得分差异对齐，从而减小轨迹上的散度。

D.5.3 定理 11.2.1：CM 在 $o(\Delta s)$ 意义下等于 CT

证明：步骤 1：带有 Oracle 得分的 DDIM 更新即为条件均值。

$$\begin{aligned}
\hat{\mathbf{x}}_{s'} &:= \frac{\alpha_{s'}}{\alpha_s}\mathbf{x}_s + \sigma_s^2\left(\frac{\alpha_{s'}}{\alpha_s} - \frac{\sigma_{s'}}{\sigma_s}\right)\nabla_{\mathbf{x}_s}\log p_s(\mathbf{x}_s) \\
&= \frac{\alpha_{s'}}{\alpha_s}\big(\mathbf{x}_s + \sigma_s^2\nabla_{\mathbf{x}_s}\log p_s(\mathbf{x}_s)\big) - \sigma_{s'}\sigma_s\nabla_{\mathbf{x}_s}\log p_s(\mathbf{x}_s) \\
&= \alpha_{s'}\mathbb{E}[\mathbf{x}_0|\mathbf{x}_s] + \sigma_{s'}\Big(\frac{\mathbf{x}_s - \alpha_s\mathbb{E}[\mathbf{x}_0|\mathbf{x}_s]}{\sigma_s}\Big) \\
&= \alpha_{s'}\mathbb{E}[\mathbf{x}_0|\mathbf{x}_s] + \sigma_{s'}\mathbb{E}[\boldsymbol{\epsilon}|\mathbf{x}_s] \\
&= \mathbb{E}[\alpha_{s'}\mathbf{x}_0 + \sigma_{s'}\boldsymbol{\epsilon}|\mathbf{x}_s] \\
&= \mathbb{E}[\mathbf{x}_{s'}|\mathbf{x}_s].
\end{aligned}$$

此处,我们在第三和第四等式中使用了 Tweedie 公式 $\mathbb{E}[\mathbf{x}_0|\mathbf{x}_s] = \frac{\mathbf{x}_s+\sigma_s^2\nabla_{\mathbf{x}_s}\log p_s(\mathbf{x}_s)}{\alpha_s}$ 和 $\mathbf{x}_s = \alpha_s\mathbf{x}_0 + \sigma_s\boldsymbol{\epsilon}$。

第 2 步：在条件均值 $\hat{\mathbf{x}}_{s'}$ 附近展开 CT

$$
\begin{aligned}
\mathcal{L}_{\mathrm{CT}} &= \mathbb{E}_{s,\mathbf{x}_s}\mathbb{E}_{\mathbf{x}_{s'}|\mathbf{x}_s}\Big[w(s)d\big(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s,s),\mathbf{f}_{\boldsymbol{\theta}^-}(\mathbf{x}_{s'},s')\big)\Big] \\
&\overset{(1)}{=} \mathbb{E}_{s,\mathbf{x}_s}\mathbb{E}_{\mathbf{x}_{s'}|\mathbf{x}_s}\Big[w(s)d\big(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s,s),\mathbf{f}_{\boldsymbol{\theta}^-}(\hat{\mathbf{x}}_{s'},s')\big) \\
&\qquad\qquad + w(s)\partial_2 d\big(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s,s),\mathbf{f}_{\boldsymbol{\theta}^-}(\hat{\mathbf{x}}_{s'},s')\big)\big[\partial_1\mathbf{f}_{\boldsymbol{\theta}^-}(\hat{\mathbf{x}}_{s'},s')(\mathbf{x}_{s'}-\hat{\mathbf{x}}_{s'})\big] \\
&\qquad\qquad + w(s)\mathcal{O}\big(\|\mathbf{x}_{s'}-\hat{\mathbf{x}}_{s'}\|^2\big)\Big] \\
&\overset{(2)}{=} \mathbb{E}_{s,\mathbf{x}_s}\Big[w(s)d\big(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s,s),\mathbf{f}_{\boldsymbol{\theta}^-}(\hat{\mathbf{x}}_{s'},s')\big)\Big] \\
&\quad + \mathbb{E}_{s,\mathbf{x}_s}\Big[w(s)\partial_2 d\big(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s,s),\mathbf{f}_{\boldsymbol{\theta}^-}(\hat{\mathbf{x}}_{s'},s')\big)\big[\partial_1\mathbf{f}_{\boldsymbol{\theta}^-}(\hat{\mathbf{x}}_{s'},s')\mathbb{E}_{\mathbf{x}_{s'}|\mathbf{x}_s}(\mathbf{x}_{s'}-\hat{\mathbf{x}}_{s'})\big]\Big] \\
&\quad + \mathbb{E}_{s,\mathbf{x}_s}\mathbb{E}_{\mathbf{x}_{s'}|\mathbf{x}_s}\Big[w(s)\mathcal{O}\big(\|\mathbf{x}_{s'}-\hat{\mathbf{x}}_{s'}\|^2\big)\Big] \\
&\overset{(3)}{=} \mathbb{E}_{s,\mathbf{x}_s}\Big[w(s)d\big(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s,s),\mathbf{f}_{\boldsymbol{\theta}^-}(\hat{\mathbf{x}}_{s'},s')\big)\Big] + \mathbb{E}_{s,\mathbf{x}_s}\mathbb{E}_{\mathbf{x}_{s'}|\mathbf{x}_s}\Big[w(s)\mathcal{O}\big(\|\mathbf{x}_{s'}-\hat{\mathbf{x}}_{s'}\|^2\big)\Big] \\
&= \mathbb{E}_{s,\mathbf{x}_s}\Big[w(s)d\big(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s,s),\mathbf{f}_{\boldsymbol{\theta}^-}(\hat{\mathbf{x}}_{s'},s')\big)\Big] + \mathcal{O}\Big(\mathbb{E}_{s,\mathbf{x}_s}\mathbb{E}_{\mathbf{x}_{s'}|\mathbf{x}_s}\|\mathbf{x}_{s'}-\hat{\mathbf{x}}_{s'}\|^2\Big) \\
&= \mathcal{L}_{\mathrm{CM}} + \mathcal{O}\big(\mathbb{E}_{s,\mathbf{x}_s}\mathbb{E}_{\mathbf{x}_{s'}|\mathbf{x}_s}\|\mathbf{x}_{s'}-\hat{\mathbf{x}}_{s'}\|^2\big) \\
&\overset{(4)}{=} \mathcal{L}_{\mathrm{CM}} + o(\Delta s)
\end{aligned}
$$

此处，(1) 对

$$h(\mathbf{x}') := d(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s,s),\mathbf{f}_{\boldsymbol{\theta}^-}(\mathbf{x}',s'))$$

在其第二个参数中围绕 $\hat{\mathbf{x}}_{s'}$。(2) 应用塔性质

$$\mathbb{E}_{s,\mathbf{x}_s}\mathbb{E}_{\mathbf{x}_{s'}|\mathbf{x}_s}[\cdot] = \mathbb{E}_{s,\mathbf{x}_s}[\cdot]$$

并且注意到，在 $\mathbb{E}_{\mathbf{x}_{s'}|\mathbf{x}_s}$ 内部，唯一的 $\mathbf{x}_{s'}$ 依赖性是通过 $(\mathbf{x}_{s'}-\hat{\mathbf{x}}_{s'})$，因此所有其他因子被视为常数，内部期望简化为 $\mathbb{E}_{\mathbf{x}_{s'}|\mathbf{x}_s}(\mathbf{x}_{s'}-\hat{\mathbf{x}}_{s'})$ 和 $\mathbb{E}_{\mathbf{x}_{s'}|\mathbf{x}_s}\|\mathbf{x}_{s'}-\hat{\mathbf{x}}_{s'}\|^2$。(3) 使用了 $\mathbb{E}[\mathbf{x}_{s'}-\hat{\mathbf{x}}_{s'}|\mathbf{x}_s] = \mathbf{0}$，因为 $\hat{\mathbf{x}}_{s'} = \mathbb{E}[\mathbf{x}_{s'}|\mathbf{x}_s]$。(4) 使用了线性的–高斯调度器，其给出

$$\mathbb{E}[\|\mathbf{x}_{s'}-\hat{\mathbf{x}}_{s'}\|^2|\mathbf{x}_s] = \mathcal{O}(\Delta s^2),$$

从而导致总余数为 $o(\Delta s)$ 。

■

D.5.4 命题 11.3.1：CT 梯度的连续时间极限

证明： 我们通过证明来简化记号：Equation (11.3.2):

$$\mathcal{L}_{\mathrm{CM}}^{\Delta s}(\boldsymbol{\theta},\boldsymbol{\theta}^{-}) := \mathbb{E}\left[\omega(s)\left\|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s,s)-\mathbf{f}_{\boldsymbol{\theta}^{-}}\left(\boldsymbol{\Psi}_{s\to s-\Delta s}(\mathbf{x}_s),s-\Delta s\right)\right\|_2^2\right].$$

为书写简便，我们记 $\tilde{\mathbf{x}}_{s-\Delta s} := \boldsymbol{\Psi}_{s\to s-\Delta s}(\mathbf{x}_s)$

扩展均方误差损失后，我们将得到

$$\begin{aligned}
&\|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s,s)-\mathbf{f}_{\boldsymbol{\theta}^{-}}(\tilde{\mathbf{x}}_{s-\Delta s},s-\Delta s)\|_2^2\\
=&\|\underbrace{\left(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s,s)-\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)\right)}_{=:\delta\mathbf{f}}+\left(\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)-\mathbf{f}_{\boldsymbol{\theta}^{-}}(\tilde{\mathbf{x}}_{s-\Delta s},s-\Delta s)\right)\|_2^2\\
=&\|\delta\mathbf{f}\|_2^2+2\delta\mathbf{f}^{\top}\cdot\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)\Delta s+\left\|\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)\right\|_2^2|\Delta s|^2+\mathcal{O}\left(|\Delta s|^3\right).
\end{aligned}$$

在此，我们在 $(\mathbf{x}_s,s)$ 处应用泰勒展开：

$$\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)-\mathbf{f}_{\boldsymbol{\theta}^{-}}\left(\tilde{\mathbf{x}}_{s-\Delta s},s-\Delta s\right)=\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)\,\Delta s+\mathcal{O}(|\Delta s|^2),$$

结合预言流映射的一阶展开，

$$\mathbf{x}_s-\boldsymbol{\Psi}_{s\to s-\Delta s}(\mathbf{x}_s)=\mathbf{v}^{*}(\mathbf{x}_s,s)\,\Delta s+\mathcal{O}(|\Delta s|^2),$$

以及全微分恒等式，

$$\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)=\partial_s\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)+\left(\partial_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)\right)\mathbf{v}^{*}(\mathbf{x}_s,s),$$

化简表达式。

由于 $\boldsymbol{\theta}^{-}$ 被视为常数，且与 $\boldsymbol{\theta}$ 完全相同（即对其无梯度传播），因此 $\mathcal{L}_{\mathrm{CM}}^{\Delta s}(\boldsymbol{\theta},\boldsymbol{\theta}^{-})$ 相对于 $\boldsymbol{\theta}$ 的梯度为：

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{CM}}^{\Delta s}(\boldsymbol{\theta},\boldsymbol{\theta}^{-})=2\mathbb{E}\left[\omega(s)\,\nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_s,s)^{\top}\cdot\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)\right]\Delta s+\mathcal{O}\left(|\Delta s|^2\right).$$

除以 Δs 并取极限得：

$$\lim_{\Delta s\to 0}\frac{1}{\Delta s}\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{CM}}^{\Delta s}(\boldsymbol{\theta},\boldsymbol{\theta}^{-})=\nabla_{\boldsymbol{\theta}}\mathbb{E}\left[2\omega(t)\mathbf{f}_{\boldsymbol{\theta}}^{\top}(\mathbf{x}_s,s)\cdot\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_s,s)\right].$$

这证明了恒等式。

■

D.6 （可选）扩散模型阐明（EDM）

我们引入了针对神经网络参数化设计的具体准则，用于 $\mathbf{x}$ -预测模型，如《阐明扩散模型》（EDM）（**karras2022elucidating**）所提出的。EDM 提供了一种简单的配方，使训练过程更易于优化且更加可靠。$\mathbf{x}$ -预测模型表示为时间相关的跳跃连接与缩放残差（Equation (D.6.1)）的结合。核心思想是在所有时刻对输入和回归目标进行单位方差归一化，并调整跳跃路径，以确保残差误差不会随时间演化而被放大。该配方已成为扩散模型实现中的广泛采用的默认方案，并可自然扩展至流映射学习，特别是保真度模型（Consistency Models）家族。

D.6.1 神经网络设计的标准 $\mathbf{x}_\phi$

EDM 将 $\mathbf{x}$ -预测模型的参数化表示为，用略微滥用的符号记作 $\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}, t)$ [1]，以以下形式：

$$\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}, t) := c_{\text{skip}}(t)\mathbf{x} + c_{\text{out}}(t)\mathbf{F}_{\boldsymbol{\phi}}\left(c_{\text{in}}(t)\mathbf{x}, c_{\text{noise}}(t)\right). \tag{D.6.1}$$

此处，$c_{\text{skip}}(t)$ ，$c_{\text{out}}(t)$ ，$c_{\text{in}}(t)$ ，和 $c_{\text{noise}}(t)$ 为时间相关函数。它们的选择基于实际考量，旨在提升训练过程中的稳定性与性能，相关内容将很快介绍。

将其代入 Equation (6.2.5)，经过直接的代数运算后，目标函数变为[2]：

$$\mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t}\left[\omega(t)c_{\text{out}}^2(t)\left\|\mathbf{F}_{\boldsymbol{\phi}}\left(c_{\text{in}}(t)\mathbf{x}_t, c_{\text{noise}}(t)\right) - \mathbf{x}_{\text{tgt}}(t)\right\|_2^2\right]. \tag{D.6.2}$$

---

[1] 如前所述，所有四种预测类型都是等价的，可以简化为 $\mathbf{x}$ -预测情形。EDM 采用这种表述方式，它不仅得到了充分研究，而且与生成流图模型干净样本的目标天然契合。

[2] 我们从 $\mathbf{x}$-预测扩散损失出发，通过代入 Equation (D.6.1) 中给出的参数化形式：

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t}\left[\omega(t)\left\|\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \mathbf{x}_0\right\|_2^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t}\left[\omega(t)\left\|c_{\text{skip}}(t)\underbrace{\left(\alpha_t\mathbf{x}_0 + \sigma_t\boldsymbol{\epsilon}\right)}_{\mathbf{x}_t} + c_{\text{out}}(t)\mathbf{F}_{\boldsymbol{\phi}}\left(c_{\text{in}}(t)\mathbf{x}_t, c_{\text{noise}}(t)\right) - \mathbf{x}_0\right\|_2^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t}\left[\omega(t)c_{\text{out}}^2(t)\left\|\mathbf{F}_{\boldsymbol{\phi}}\left(c_{\text{in}}(t)\mathbf{x}, c_{\text{noise}}(t)\right) - \underbrace{\left(\frac{\left(1 - c_{\text{skip}}(t)\alpha_t\right)\mathbf{x}_0 - c_{\text{skip}}(t)\sigma_t\boldsymbol{\epsilon}}{c_{\text{out}}(t)}\right)}_{\mathbf{x}_{\text{tgt}}(t)}\right\|_2^2\right] \\
&= \text{Equation (D.6.1)}.
\end{aligned}$$

此处，回归目标 $\mathbf{x}_{\text{tgt}}(t)$ 的获取方式为：

$$\mathbf{x}_{\text{tgt}}(t) = \frac{(1 - c_{\text{skip}}(t)\alpha_t)\,\mathbf{x}_0 - c_{\text{skip}}(t)\sigma_t\boldsymbol{\epsilon}}{c_{\text{out}}(t)}.$$

通过引入 p_{data} 的标准差，记为 σ_{d} ，EDM 提出了网络参数化的一个设计准则，可描述为 *单位方差准则*。

输入的单位方差

$$\begin{aligned}
&\text{Var}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[c_{\text{in}}(t)\mathbf{x}_t\right] = 1 \\
\iff &c_{\text{in}}^2(t)\text{Var}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\alpha_t\mathbf{x}_0 + \sigma_t\boldsymbol{\epsilon}\right] = 1 \\
\iff &c_{\text{in}}(t) = \frac{1}{\sqrt{\sigma_{\text{d}}^2\alpha_t^2 + \sigma_t^2}},
\end{aligned}$$

取正根。

训练目标的单位方差

$$\begin{aligned}
&\text{Var}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\mathbf{x}_{\text{tgt}}(t)\right] = 1 \\
\iff &c_{\text{out}}^2(t) = \left(1 - c_{\text{skip}}(t)\alpha_t\right)^2\sigma_{\text{d}}^2 + c_{\text{skip}}^2(t)\sigma_t^2,
\end{aligned} \tag{D.6.3}$$

使用中心化数据 ($\mathbb{E}[\mathbf{x}_0] = \mathbf{0}$)。

从 $\mathbf{F}_\phi$ 到 $\mathbf{x}_\phi$ 最小化误差放大。 EDM 的目标是减轻网络学习误差从 $\mathbf{F}_\phi$ 到 $\mathbf{x}_\phi$ 的放大。这是通过选择 c_{skip} 来最小化 c_{out} 实现的。

$$c_{\text{skip}}^* \in \arg\min_{c_{\text{skip}}} c_{\text{out}}^2.$$

使用求解 $\frac{\partial c_{\text{out}}}{\partial c_{\text{skip}}} = 0$ 的标准方法得到临界点 c_{skip}^* ，我们获得

$$c_{\text{skip}}^*(t) = \frac{\alpha_t\sigma_{\text{d}}^2}{\alpha_t^2\sigma_{\text{d}}^2 + \sigma_t^2}.$$

将其代入 Equation (D.6.3)，最优值为

$$c_{\text{out}}^*(t) = \pm\frac{\sigma_t\sigma_{\text{d}}}{\sqrt{\alpha_t^2\sigma_{\text{d}}^2 + \sigma_t^2}}.$$

按照惯例，我们对输出尺度采用非负分支：

$$c_{\text{out}}^{*}(t) = \frac{\sigma_t \sigma_{\text{d}}}{\sqrt{\alpha_t^2 \sigma_{\text{d}}^2 + \sigma_t^2}} \quad (\geq 0),$$

这确保了当 σ_t 足够大时，$c_{\text{out}}(0) = 0$ 和 $c_{\text{out}}(t) \to \sigma_{\text{d}}$ 成立，从而得到直观的极限 $\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_t, 0) \approx \mathbf{x}_t$ 和 $\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) \approx \sigma_{\text{d}}\, \mathbf{F}_{\boldsymbol{\phi}}(\cdot)$。

我们将这些系数总结如下：

记 $R_t := \alpha_t^2 \sigma_{\text{d}}^2 + \sigma_t^2$ ，我们有如下选择：

$$c_{\text{in}}(t) = \frac{1}{\sqrt{R_t}}, \quad c_{\text{skip}}(t) = \frac{\alpha_t \sigma_{\text{d}}^2}{R_t}, \quad c_{\text{out}}(t) = \frac{\sigma_t \sigma_{\text{d}}}{\sqrt{R_t}}. \tag{D.6.4}$$

使用 Equation (D.6.4)，回归目标 $\mathbf{x}_{\text{tgt}}(t)$ 简化为

$$\mathbf{x}_{\text{tgt}}(t) = \frac{1}{\sigma_{\text{d}}} \frac{\sigma_t \mathbf{x}_0 - \alpha_t \sigma_{\text{d}}^2 \boldsymbol{\epsilon}}{\sqrt{R_t}}.$$

此外，将这些表达式代入 Equation (D.6.1) 和 Equation (D.6.2) 可以简化参数化和损失函数，得到：

$$\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}, t) = \frac{\alpha_t \sigma_{\text{d}}^2}{R_t} \mathbf{x} + \frac{\sigma_t \sigma_{\text{d}}}{\sqrt{R_t}} \mathbf{F}_{\boldsymbol{\phi}} \left(\frac{1}{\sqrt{R_t}} \mathbf{x}, c_{\text{noise}}(t) \right),$$

并且

$$\mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t} \left[\omega(t) \frac{\sigma_t^2}{R_t} \left\| \sigma_{\text{d}} \mathbf{F}_{\boldsymbol{\phi}} \left(\frac{1}{\sqrt{R_t}} \mathbf{x}_t, c_{\text{noise}}(t) \right) - \left(\frac{\sigma_t \mathbf{x}_0 - \alpha_t \sigma_{\text{d}}^2 \boldsymbol{\epsilon}}{\sqrt{R_t}} \right) \right\|_2^2 \right].$$

在此参数化及条件 $\alpha_0 \approx 1$ 和 $\sigma_0 \approx 0$ 下，我们观察到

$$c_{\text{skip}}(0) \approx 1 \quad \text{and} \quad c_{\text{out}}(0) \approx 0.$$

$c_{\text{noise}}(t)$ 的选择。 它为 $\mathbf{F}_{\boldsymbol{\phi}}$ 提供了一个噪声等级嵌入；任何将噪声等级 σ_t（例如 $c_{\text{noise}}(t) = \log \sigma_t$ ）进行一一映射的方法都是合适的。

D.6.2 一种常见的 EDM 特例：$\alpha_t = 1$ ，$\sigma_t = t$

我们考虑在 EDM 中使用的简化噪声调度，其中 $\alpha_t = 1$ 和 $\sigma_t = t$ ，这也在 Section 11.4 中关于 CTM 的讨论中出现。在此情景下，前向过程变为

$$\mathbf{x}_t = \mathbf{x}_0 + t\boldsymbol{\epsilon}, \quad \text{with } \mathbf{x}_0 \sim p_{\text{data}},\ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

对应于扰动核

$$p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_0, t^2\mathbf{I}).$$

因此，终端时间的先验分布设置为

$$p_{\text{prior}}(\mathbf{x}_T) := \mathcal{N}(\mathbf{x}_T; \mathbf{0}, T^2\mathbf{I}).$$

扰动核所引起的边缘密度由卷积给出：

$$p_t(\mathbf{x}) = \int \mathcal{N}(\cdot; \mathbf{0}, t^2\mathbf{I}) p_{\text{data}}(\mathbf{x}_0)\, \mathrm{d}\mathbf{x}_0.$$

在此设置下，基于 $\mathbf{x}$ -预测 $\mathbf{x}_{\boldsymbol{\phi}^\times}$ 的 PF-ODE（见 Equation (6.3.2)）简化为

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \frac{\mathbf{x}(t) - \mathbf{x}_{\boldsymbol{\phi}^\times}(\mathbf{x}(t), t)}{t}.$$

将此表述代入 Equation (D.6.4)，神经网络参数化系数变为

$$c_{\text{in}}(t) = \frac{1}{\sqrt{\sigma_{\text{d}}^2 + t^2}}, \quad c_{\text{skip}}(t) = \frac{\sigma_{\text{d}}^2}{\sigma_{\text{d}}^2 + t^2}, \quad c_{\text{out}}(t) = \pm\frac{t\sigma_{\text{d}}}{\sqrt{\sigma_{\text{d}}^2 + t^2}}. \tag{D.6.5}$$

从这些表达式中，我们观察到：

- 当 $t \approx 0$ 时，噪声水平可忽略不计，因此 $c_{\text{skip}} \approx 1$ 且 $c_{\text{out}} \approx 0$ 。在此极限下，跳跃路径占主导地位，网络本质上直接传递其输入，

$$\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}, t) \approx \mathbf{x}.$$

- 当 $t \gg 0$ 时，输入受到噪声的严重干扰，因此 $c_{\text{skip}} \approx 0$ 且 $c_{\text{out}} \approx \sigma_{\text{d}}$。在此区域中，跳跃路径消失，模型输出完全由学成的残差决定，

$$\mathbf{x}_{\boldsymbol{\phi}}(\mathbf{x}, t) \approx \sigma_{\text{d}} \mathbf{F}_{\boldsymbol{\phi}}\left(c_{\text{in}}(t)\mathbf{x}, c_{\text{noise}}(t)\right),$$

这意味着网络 $\mathbf{F}_\phi$ 从规范化的噪声输入中预测一个缩放后的干净信号代理；在高噪声水平下，模型输出完全由学成的降噪函数决定。

简而言之，参数化在小 t 时平滑地从恒等映射过渡到大 t 时在标准化输入上的缩放残差预测器。