

# 理解扩散模型：一种统一的视角

Calvin Luo

Google Research, Brain Team

[calvinluo@google.com](mailto:calvinluo@google.com)

2025 年 5 月 7 日

## 目录

<b>Introduction: Generative Models</b> . . . . .	1
<b>Background: ELBO, VAE, and Hierarchical VAE</b> . . . . .	1
<b>Evidence Lower Bound</b> . . . . .	2
<b>Variational Autoencoders</b> . . . . .	3
<b>Hierarchical Variational Autoencoders</b> . . . . .	5
<b>Variational Diffusion Models</b> . . . . .	6
<b>Learning Diffusion Noise Parameters</b> . . . . .	14
<b>Three Equivalent Interpretations</b> . . . . .	15
<b>Score-based Generative Models</b> . . . . .	18
<b>Guidance</b> . . . . .	21
<b>Classifier Guidance</b> . . . . .	22
<b>Classifier-Free Guidance</b> . . . . .	22
<b>Closing</b> . . . . .	23

## 简介：生成式模型

给定从感兴趣的分布中观察到的样本  $x$ , **生成式模型**的目标是学习建模其真实数据分布  $p(x)$ 。一旦学成, 我们可以随意从我们的近似模型中生成新的样本。此外, 在某些表述下, 我们还可以使用学成的模型来评估观察到的数据或采样数据的似然。

当前文献中有一些著名的方向，我们将在高层次上简要介绍。生成对抗网络（GANs）对复杂分布的采样过程进行建模，这种分布是通过对抗方式学习的。另一类生成模型被称为“基于似然的”，其目标是学习一个能为观测到的数据样本分配高似然的模型。这包括自回归模型、归一化流和变分自编码器（VAEs）。另一种类似的方法是基于能量的建模，在这种方法中，分布被学习为一个任意灵活的能量函数，然后进行规范化。基于得分的生成模型与此高度相关；它们不直接学习能量函数本身，而是将基于能量的模型的得分作为神经网络进行学习。在本工作中，我们探讨并回顾了扩散模型，正如我们将展示的那样，扩散模型既有基于似然的解释，也有基于得分的解释。我们详细展示了这些模型背后的数学原理，旨在让任何人能够跟上思路并理解扩散模型是什么以及它们是如何工作的。

## 背景：ELBO、VAE 和分层 VAE

对于许多模态，我们可以将我们观察到的数据视为由相关的不可见的潜在变量表示或生成的，我们可以用随机变量  $\mathbf{z}$  来表示。表达这个想法的最佳直觉是通过柏拉图的 *Allegory of the Cave*。在寓言中，一群人一生都被锁在一个洞穴里，只能看到火光前经过的不可见三维物体投射在他们面前墙上的二维阴影。对这些人来说，他们所观察到的一切实际上是由他们永远无法看到的高维抽象概念决定的。

同样，我们在现实世界中遇到的物体也可能是某些更高级表示的函数；例如，这些表示可能包含诸如颜色、大小、形状等抽象属性。然后，我们所观察到的可以被解释为这些抽象概念的三维投影或实例，就像洞穴中的人观察到的是三维物体的二维投影一样。尽管洞穴中的人永远无法看到（甚至完全理解）隐藏的物体，但他们仍然可以对它们进行推理并得出推论；同样，我们可以近似描述我们所观察数据的潜在表示。

尽管柏拉图的寓言说明了潜变量作为可能无法观测的表示，这些表示决定了观察结果，但这个类比有一个需要注意的地方：在生成式建模中，我们通常寻求学习低维的潜表示，而不是高维的。这是因为如果没有强有力的知识先验，试图学习比观察结果更高维度的表示是徒劳的。另一方面，学习低维潜变量也可以看作是一种压缩形式，并且可能会揭示描述观察结果的语义上有意义的结构。

### 证据下界

数学上，我们可以将潜在变量和我们观察到的数据建模为联合分布  $p(\mathbf{x}, \mathbf{z})$ 。回忆一种生成式建模的方法，称为“基于似然”，就是学习一个模型来最大化所有观察到的  $\mathbf{x}$  的似然  $p(\mathbf{x})$ 。我们有两种方法可以操作这个联合分布以恢复我们观察数据的似然  $p(\mathbf{x})$ ；我们可以显式地 *marginalize* 掉潜在变量  $\mathbf{z}$ ：

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z} \quad (1)$$

或者，我们也可以引用 *chain rule of probability*:

$$p(\mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{z}|\mathbf{x})} \quad (2)$$

直接计算并最大化似然  $p(\mathbf{x})$  是困难的，因为它要么涉及在方程 1 中对所有潜变量  $\mathbf{z}$  进行积分，这对复杂模型来说是不可处理的，要么涉及在方程 2 中拥有一个真实值潜变量编码器  $p(\mathbf{z}|\mathbf{x})$ 。然而，使用这两个方程，我们可以推导出一个称为证据下界（ELBO）的项，正如其名称所示，它是证据的一个 *lower bound*。在这种情况下，证据被量化为观测数据的对数似然函数。然后，最大化 ELBO 成为优化潜变量模型的代理目标；在最好的情况下，当 ELBO 被强大地参数化并且被完美优化时，它就与证据完全等价。形式上，ELBO 的方程是：

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] \quad (3)$$

为了使关系与证据明确，我们可以用数学方式表示为：

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] \quad (4)$$

在这里， $q_\phi(\mathbf{z}|\mathbf{x})$  是一个具有参数  $\phi$  的灵活近似变分分布，我们希望对其进行优化。直观地说，它可以被看作是一个可参数化的模型，该模型通过给定的观测值  $\mathbf{x}$  来估计潜在变量的真实分布；换句话说，它旨在近似真实后验  $p(\mathbf{z}|\mathbf{x})$ 。当我们探索变分自编码器时，我们会看到，通过调整参数  $\phi$  以最大化 ELBO 来提高下界，我们就可以获得可用于建模真实数据分布并从中采样的组件，从而学习一个生成式模型。现在，让我们尝试更深入地了解为什么 ELBO 是我们想要最大化的目标。

让我们从使用方程 1 开始推导 ELBO，

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z} \quad (\text{Apply Equation 1}) \quad (5)$$

$$= \log \int \frac{p(\mathbf{x}, \mathbf{z}) q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} d\mathbf{z} \quad (\text{Multiply by } 1 = \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}) \quad (6)$$

$$= \log \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[ \frac{p(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] \quad (\text{Definition of Expectation}) \quad (7)$$

$$\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] \quad (\text{Apply Jensen's Inequality}) \quad (8)$$

在这个推导中，我们直接通过应用 Jensen 不等式得到了我们的下界。然而，这并没有给我们提供太多关于内部实际发生情况的有用信息；关键的是，这个证明并没有给出确切的理由说明为什么 ELBO 实际上是证据的下界，因为 Jensen 不等式把它轻轻带过了。此外，仅仅知道 ELBO 确实是数据的下界，并不能真正告诉我们为什么我们要将其作为目标进行最大化。为了更好地理解证据和 ELBO 之间的关系，让我们进行另一次推导，这次使用方程 2：

$$\log p(\mathbf{x}) = \log p(\mathbf{x}) \int q_\phi(\mathbf{z}|\mathbf{x}) d\mathbf{z} \quad (\text{Multiply by } 1 = \int q_\phi(\mathbf{z}|\mathbf{x}) d\mathbf{z}) \quad (9)$$

$$= \int q_\phi(\mathbf{z}|\mathbf{x}) (\log p(\mathbf{x})) d\mathbf{z} \quad (\text{Bring evidence into integral}) \quad (10)$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p(\mathbf{x})] \quad (\text{Definition of Expectation}) \quad (11)$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{z}|\mathbf{x})} \right] \quad (\text{Apply Equation 2}) \quad (12)$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z}) q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x}) q_\phi(\mathbf{z}|\mathbf{x})} \right] \quad (\text{Multiply by } 1 = \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}) \quad (13)$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[ \log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x})} \right] \quad (\text{Split the Expectation}) \quad (14)$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] + D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}|\mathbf{x})) \quad (\text{Definition of KL Divergence}) \quad (15)$$

$$\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] \quad (\text{KL Divergence always } \geq 0) \quad (16)$$

从这个推导中，我们可以明显地从公式 15 看出，证据等于 ELBO 加上近似后验  $q_\phi(\mathbf{z}|\mathbf{x})$  与真实后验  $p(\mathbf{z}|\mathbf{x})$  之间的 KL 散度。事实上，正是这个 KL 散度项在第一次推导中的公式 8 中通过 Jensen 不等式被神奇地消去了。理解这个项是理解 ELBO 与证据之间的关系，以及为什么优化 ELBO 本身是一个合适的目标的关键。

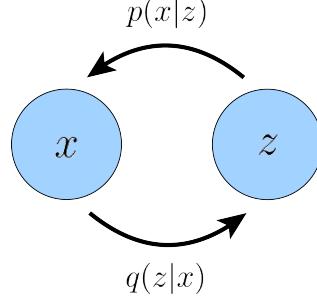


图 1: 变分自编码器的图形表示。此处, 编码器  $q(z|x)$  定义了观测值  $x$  上的潜在变量  $z$  的分布, 而  $p(x|z)$  将潜在变量解码为观测值。

首先, 我们现在知道为什么 ELBO 确实是一个下界: 证据和 ELBO 之间的差异是一个严格非负的 KL 项, 因此 ELBO 的值永远不可能超过证据。

其次, 我们探讨为何要最大化 ELBO。在引入了我们希望建模的潜变量  $z$  之后, 我们的目标是学习描述我们观测数据的潜在结构。换句话说, 我们希望优化变分后验  $q_\phi(z|x)$  的参数, 使其精确匹配真实后验分布  $p(z|x)$ , 这是通过最小化它们的 KL 散度 (理想情况下为零) 来实现的。不幸的是, 直接最小化这个 KL 散度项是难以处理的, 因为我们无法获得真实值  $p(z|x)$  分布。然而, 注意到方程 15 左边的数据似然 (因此也是我们的证据项  $\log p(x)$ ) 相对于  $\phi$  始终是一个常数, 因为它通过对联合分布  $p(x,z)$  中的所有潜变量  $z$  进行边缘化计算得出, 并且与  $\phi$  完全无关。由于 ELBO 和 KL 散度项的和是一个常数, 因此相对于  $\phi$  对 ELBO 项进行最大化必然导致 KL 散度项的同等最小化。因此, 可以将最大化 ELBO 作为学习如何完美建模真实潜后验分布的代理; 我们对 ELBO 的优化越多, 近似后验就越接近真实后验。此外, 一旦训练完成, ELBO 也可以用来估计观测数据或生成数据的似然, 因为它被训练为近似模型证据  $\log p(x)$ 。

## 变分自编码器

在变分自编码器 (VAE) 的默认公式 [1] 中, 我们直接最大化 ELBO。这种方法是变分的, 因为我们从由  $\phi$  参数化的潜在后验分布族中优化最佳  $q_\phi(z|x)$ 。它被称为自编码器, 因为它类似于传统的自编码器模型, 其中输入数据在经过中间瓶颈表示步骤后被训练以预测自身。为了明确这种联系, 让我们进一步分析 ELBO 项:

$$\mathbb{E}_{q_\phi(z|x)} \left[ \log \frac{p(x,z)}{q_\phi(z|x)} \right] = \mathbb{E}_{q_\phi(z|x)} \left[ \log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)} \right] \quad (\text{Chain Rule of Probability}) \quad (17)$$

$$= \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)] + \mathbb{E}_{q_\phi(z|x)} \left[ \log \frac{p(z)}{q_\phi(z|x)} \right] \quad (\text{Split the Expectation}) \quad (18)$$

$$= \underbrace{\mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)]}_{\text{reconstruction term}} - \underbrace{D_{\text{KL}}(q_\phi(z|x) \parallel p(z))}_{\text{prior matching term}} \quad (\text{Definition of KL Divergence}) \quad (19)$$

在这种情况下, 我们学习一个中间瓶颈分布  $q_\phi(z|x)$ , 它可以被视为一个 编码器; 它将输入转换为潜在变量的分布。同时, 我们学习一个确定性函数  $p_\theta(x|z)$  将给定的潜在向量  $z$  转换为一个观测值  $x$ , 这可以被解释为一个 解码器。

方程 19 中的两个项都有直观的描述: 第一项衡量了从我们的变分分布中解码器的重构似然; 这确保了学成的分布正在建模有效的潜在变量, 这些潜在变量可以从原始数据中重新生成。第二项衡量了学成的变分分布与对潜在变量的先验信念之间的相似性。最小化该项会鼓励编码器实际学习一个分布, 而不是坍缩成一个

Dirac Delta 函数。因此，最大化 ELBO 等同于最大化其第一项并最小化其第二项。

VAE 的一个定义特征是如何联合优化参数  $\phi$  和  $\theta$  的 ELBO。VAE 的编码器通常被选择为建模一个对角协方差的多元高斯分布，而先验通常被选择为标准的多元高斯分布：

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), \sigma_\phi^2(\mathbf{x})\mathbf{I}) \quad (20)$$

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, \mathbf{I}) \quad (21)$$

然后，ELBO 的 KL 散度项可以解析计算，而重构项可以使用蒙特卡罗估计进行近似。我们的目标可以重写为：

$$\arg \max_{\phi, \theta} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})) \approx \arg \max_{\phi, \theta} \sum_{l=1}^L \log p_\theta(\mathbf{x}|\mathbf{z}^{(l)}) - D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})) \quad (22)$$

其中，潜在变量  $\{\mathbf{z}^{(l)}\}_{l=1}^L$  是从  $q_\phi(\mathbf{z}|\mathbf{x})$  中采样得到的，对于数据集中的每个观测值  $\mathbf{x}$ 。然而，在这种默认设置中会出现一个问题：我们计算损失的每个  $\mathbf{z}^{(l)}$  都是通过随机采样过程生成的，这通常是不可微的。幸运的是，当  $q_\phi(\mathbf{z}|\mathbf{x})$  被设计用来建模某些分布时，包括多元高斯分布，可以通过重参数化技巧来解决这个问题。

重参数化技巧将随机变量重写为噪声变量的确定性函数；这使得可以通过梯度下降优化非随机项。例如，来自任意均值  $\mu$  与方差  $\sigma^2$  的正态分布  $x \sim \mathcal{N}(x; \mu, \sigma^2)$  的样本可以重写为：

$$x = \mu + \sigma \epsilon \quad \text{with } \epsilon \sim \mathcal{N}(\epsilon; 0, \mathbf{I})$$

换句话说，任意高斯分布可以解释为标准高斯分布（其中  $\epsilon$  是一个样本）通过加法将其均值从零转移到目标均值  $\mu$ ，并通过目标方差  $\sigma^2$  将其方差拉伸。因此，通过重参数化技巧，可以从任意高斯分布中进行采样，方法是先从标准高斯分布中进行采样，然后将结果按目标标准差进行缩放，并将其按目标均值进行平移。

在变分自编码器中，每个  $\mathbf{z}$  从而作为输入  $\mathbf{x}$  和辅助噪声变量  $\epsilon$  的确定性函数进行计算：

$$\mathbf{z} = \mu_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x}) \odot \epsilon \quad \text{with } \epsilon \sim \mathcal{N}(\epsilon; 0, \mathbf{I})$$

其中  $\odot$  表示逐元素积。在  $\mathbf{z}$  的重参数化版本下，可以按照需要相对于  $\phi$  计算梯度，以优化  $\mu_\phi$  和  $\sigma_\phi$ 。因此，VAE 使用重参数化技巧和蒙特卡罗估计来联合优化  $\phi$  和  $\theta$  的 ELBO。

在训练完变分自编码器后，可以通过直接从潜在空间  $p(\mathbf{z})$  采样，然后将其通过解码器进行新数据生成。当  $\mathbf{z}$  的维度小于输入  $\mathbf{x}$  的维度时，变分自编码器尤其有趣，因为我们可能会学到紧凑且有用的表示。此外，当学习到语义上有意义的潜在空间时，可以在将潜在向量传递给解码器之前对其进行编辑，以更精确地控制生成的数据。

## 分层变分自编码器

一种分层变分自编码器 (HVAE) [2, 3] 是一种扩展到潜在变量多个层次的 VAE 的推广。在这种表述下，潜在变量本身被解释为从其他更高级、更抽象的潜在变量生成的。直观地说，就像我们将三维观测对象视为由更高层次的抽象潜在变量生成一样，柏拉图洞穴中的居民将三维物体视为生成其二维观测的潜在变量。因此，从柏拉图洞穴居民的角度来看，他们的观测可以被视为由深度为二（或更多）的潜在层次建模的。

而在一般的具有  $T$  层级的 HVAE 中，每个潜在变量都可以依赖于所有之前的潜在变量，在本文中我们关注一种特殊的案例，我们称之为马尔可夫 HVAE (MHVAE)。在 MHVAE 中，生成过程是一个马尔可夫链；也

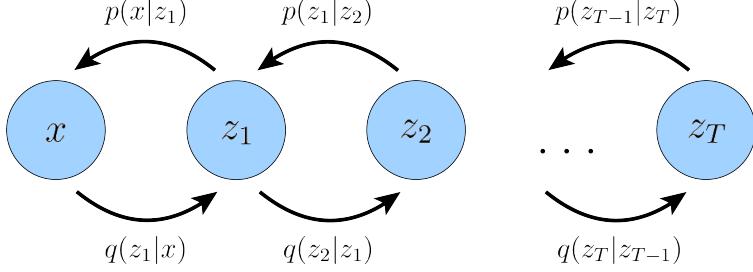


图 2: 一个具有  $T$  个层次潜在变量的马尔可夫链分层变分自编码器。生成过程被建模为一个马尔可夫链，其中每个潜在  $z_t$  仅从之前的潜在  $z_{t+1}$  生成。

就是说，每一层向下的转移都是马尔可夫性的，其中解码每个潜在变量  $z_t$  仅依赖于前一个潜在变量  $z_{t+1}$ 。直观上和视觉上，这可以看作是将 VAEs 逐层堆叠在一起，如图2 所示；另一个恰当的描述该模型的术语是递归 VAE。数学上，我们将马尔可夫 HVAE 的联合分布和后验表示为：

$$p(\mathbf{x}, \mathbf{z}_{1:T}) = p(\mathbf{z}_T)p_\theta(\mathbf{x}|\mathbf{z}_1) \prod_{t=2}^T p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t) \quad (23)$$

$$q_\phi(\mathbf{z}_{1:T}|\mathbf{x}) = q_\phi(\mathbf{z}_1|\mathbf{x}) \prod_{t=2}^T q_\phi(\mathbf{z}_t|\mathbf{z}_{t-1}) \quad (24)$$

然后，我们可以很容易地将 ELBO 扩展为：

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}, \mathbf{z}_{1:T}) d\mathbf{z}_{1:T} \quad (\text{Apply Equation 1}) \quad (25)$$

$$= \log \int \frac{p(\mathbf{x}, \mathbf{z}_{1:T}) q_\phi(\mathbf{z}_{1:T}|\mathbf{x})}{q_\phi(\mathbf{z}_{1:T}|\mathbf{x})} d\mathbf{z}_{1:T} \quad (\text{Multiply by 1} = \frac{q_\phi(\mathbf{z}_{1:T}|\mathbf{x})}{q_\phi(\mathbf{z}_{1:T}|\mathbf{x})}) \quad (26)$$

$$= \log \mathbb{E}_{q_\phi(\mathbf{z}_{1:T}|\mathbf{x})} \left[ \frac{p(\mathbf{x}, \mathbf{z}_{1:T})}{q_\phi(\mathbf{z}_{1:T}|\mathbf{x})} \right] \quad (\text{Definition of Expectation}) \quad (27)$$

$$\geq \mathbb{E}_{q_\phi(\mathbf{z}_{1:T}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z}_{1:T})}{q_\phi(\mathbf{z}_{1:T}|\mathbf{x})} \right] \quad (\text{Apply Jensen's Inequality}) \quad (28)$$

We can then plug our joint distribution (Equation 23) and posterior (Equation 24) into Equation 28 to produce an alternate form:

$$\mathbb{E}_{q_\phi(\mathbf{z}_{1:T}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z}_{1:T})}{q_\phi(\mathbf{z}_{1:T}|\mathbf{x})} \right] = \mathbb{E}_{q_\phi(\mathbf{z}_{1:T}|\mathbf{x})} \left[ \log \frac{p(\mathbf{z}_T)p_\theta(\mathbf{x}|\mathbf{z}_1) \prod_{t=2}^T p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t)}{q_\phi(\mathbf{z}_1|\mathbf{x}) \prod_{t=2}^T q_\phi(\mathbf{z}_t|\mathbf{z}_{t-1})} \right] \quad (29)$$

正如我们将在下面展示的，当我们研究变分扩散模型时，这个目标可以进一步分解为可解释的组成部分。

## 变分扩散模型

考虑变分扩散模型 (VDM) [4, 5, 6] 的最简单方式是将其简单地视为一个具有三个关键限制的马尔可夫层次变分自编码器：

- 潜在维度恰好等于数据维度
- 每个时间步的潜在编码器结构不是学成的；它是作为线性高斯模型预先定义的。换句话说，它是一个以前一时间步输出为中心的高斯分布。

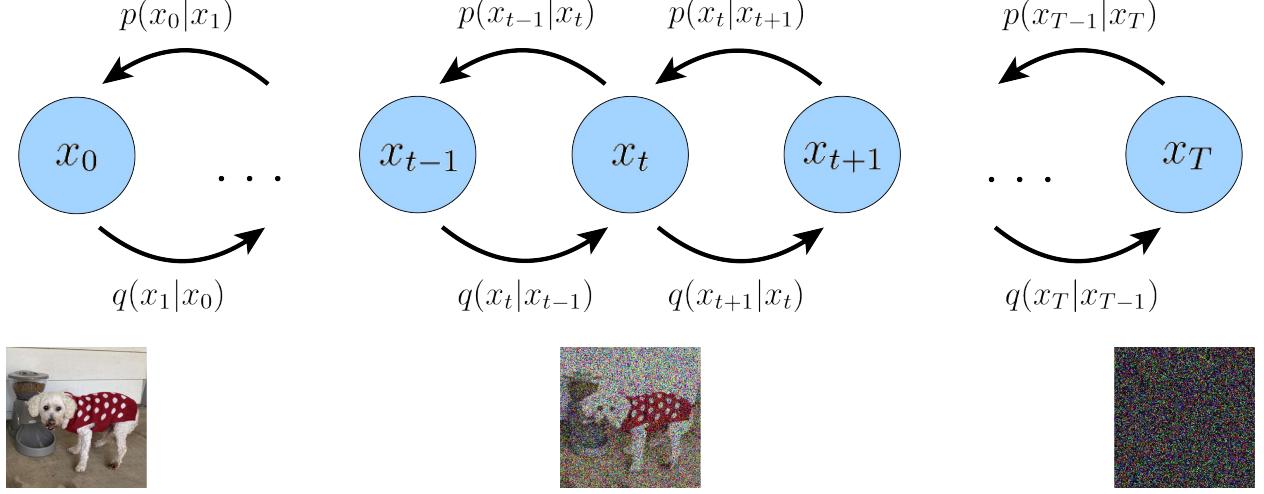


图 3: 一种变分扩散模型的视觉表示;  $\mathbf{x}_0$  表示真实数据观测值, 如自然图像,  $\mathbf{x}_T$  表示纯高斯噪声,  $\mathbf{x}_t$  是  $\mathbf{x}_0$  的中间噪声版本。每个  $q(\mathbf{x}_t|\mathbf{x}_{t-1})$  被建模为一个高斯分布, 其均值使用前一状态的输出。

- 潜在编码器的高斯参数随时间变化, 使得最终时间步  $T$  的潜在分布为标准正态分布

此外, 我们显式地在标准马尔可夫分层变分自编码器的分层转移中保持马尔可夫性质。

让我们进一步探讨这些假设的含义。从第一个限制条件出发, 稍微滥用一下符号, 我们现在可以将真实数据样本和潜变量都表示为  $\mathbf{x}_t$ , 其中  $t = 0$  表示真实数据样本,  $t \in [1, T]$  表示一个对应的具有由  $t$  索引层次结构的潜变量。VDM 后验与 MHVAE 后验 (公式 24) 相同, 但现在可以重写为:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}) \quad (30)$$

从第二个假设中, 我们知道编码器中每个潜变量的分布是围绕其前一层次潜变量的高斯分布。与马尔可夫 HVAE 不同, 每个时间步  $t$  的编码器结构不是学成的; 它被固定为一个线性高斯模型, 其中均值和标准差可以事先作为超参数 [5] 设置, 或者作为参数 [6] 进行学成。我们用均值  $\mu_t(\mathbf{x}_t) = \sqrt{\alpha_t} \mathbf{x}_{t-1}$  和方差  $\Sigma_t(\mathbf{x}_t) = (1 - \alpha_t) \mathbf{I}$  对高斯编码器进行参数化, 其中系数的形式被选择以使潜变量的方差保持在相似的尺度; 换句话说, 编码过程是方差保持的。请注意, 允许使用其他高斯参数化方式, 并导致类似的推导。主要的观点是  $\alpha_t$  是一个 (可能可学成) 的系数, 可以根据层次深度  $t$  变化, 以提高灵活性。数学上, 编码器转移表示为:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t} \mathbf{x}_{t-1}, (1 - \alpha_t) \mathbf{I}) \quad (31)$$

从第三个假设中, 我们知道  $\alpha_t$  随时间演化遵循一个固定或可学习的调度, 其结构使得最终潜在变量  $p(\mathbf{x}_T)$  的分布是一个标准正态分布。然后我们可以更新马尔可夫 HVAE (方程 23) 的联合分布, 以写出 VDM 的联合分布:

$$p(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \quad (32)$$

where,

$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; 0, \mathbf{I}) \quad (33)$$

总体而言，这一组假设所描述的是图像输入随时间的稳定噪声化；我们逐步通过添加高斯噪声来破坏图像，直到最终它完全与纯高斯噪声相同。从视觉上看，这个过程如图 3 所示。

注意，我们的编码器分布  $q(\mathbf{x}_t | \mathbf{x}_{t-1})$  已不再由  $\phi$  参数化，因为它们在每个时间步都被完全建模为具有定义均值和方差参数的高斯分布。因此，在 VDM 中，我们只关心学习条件  $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$ ，以便可以模拟新数据。在优化 VDM 后，采样过程就像从  $p(\mathbf{x}_T)$  采样高斯噪声并迭代运行降噪转移  $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$  进行  $T$  步以生成一个新颖的  $\mathbf{x}_0$ 。

与任何 HVAE 一样，VDM 可以通过最大化 ELBO 来优化，其推导如下：

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} \quad (34)$$

$$= \log \int \frac{p(\mathbf{x}_{0:T}) q(\mathbf{x}_{1:T} | \mathbf{x}_0)}{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} d\mathbf{x}_{1:T} \quad (35)$$

$$= \log \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \left[ \frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \right] \quad (36)$$

$$\geq \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \right] \quad (37)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)}{\prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1})} \right] \quad (38)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T) p_\theta(\mathbf{x}_0 | \mathbf{x}_1) \prod_{t=2}^T p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)}{q(\mathbf{x}_T | \mathbf{x}_{T-1}) \prod_{t=1}^{T-1} q(\mathbf{x}_t | \mathbf{x}_{t-1})} \right] \quad (39)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T) p_\theta(\mathbf{x}_0 | \mathbf{x}_1) \prod_{t=1}^{T-1} p_\theta(\mathbf{x}_t | \mathbf{x}_{t+1})}{q(\mathbf{x}_T | \mathbf{x}_{T-1}) \prod_{t=1}^{T-1} q(\mathbf{x}_t | \mathbf{x}_{t-1})} \right] \quad (40)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T) p_\theta(\mathbf{x}_0 | \mathbf{x}_1)}{q(\mathbf{x}_T | \mathbf{x}_{T-1})} \right] + \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \left[ \log \prod_{t=1}^{T-1} \frac{p_\theta(\mathbf{x}_t | \mathbf{x}_{t+1})}{q(\mathbf{x}_t | \mathbf{x}_{t-1})} \right] \quad (41)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0 | \mathbf{x}_1)] + \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T | \mathbf{x}_{T-1})} \right] + \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \left[ \sum_{t=1}^{T-1} \log \frac{p_\theta(\mathbf{x}_t | \mathbf{x}_{t+1})}{q(\mathbf{x}_t | \mathbf{x}_{t-1})} \right] \quad (42)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0 | \mathbf{x}_1)] + \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T | \mathbf{x}_{T-1})} \right] + \sum_{t=1}^{T-1} \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \left[ \log \frac{p_\theta(\mathbf{x}_t | \mathbf{x}_{t+1})}{q(\mathbf{x}_t | \mathbf{x}_{t-1})} \right] \quad (43)$$

$$= \mathbb{E}_{q(\mathbf{x}_1 | \mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0 | \mathbf{x}_1)] + \mathbb{E}_{q(\mathbf{x}_{T-1}, \mathbf{x}_T | \mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T | \mathbf{x}_{T-1})} \right] + \sum_{t=1}^{T-1} \mathbb{E}_{q(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1} | \mathbf{x}_0)} \left[ \log \frac{p_\theta(\mathbf{x}_t | \mathbf{x}_{t+1})}{q(\mathbf{x}_t | \mathbf{x}_{t-1})} \right] \quad (44)$$

$$\begin{aligned} &= \underbrace{\mathbb{E}_{q(\mathbf{x}_1 | \mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0 | \mathbf{x}_1)]}_{\text{reconstruction term}} - \underbrace{\mathbb{E}_{q(\mathbf{x}_{T-1} | \mathbf{x}_0)} [D_{\text{KL}}(q(\mathbf{x}_T | \mathbf{x}_{T-1}) \| p(\mathbf{x}_T))]}_{\text{prior matching term}} \\ &\quad - \sum_{t=1}^{T-1} \underbrace{\mathbb{E}_{q(\mathbf{x}_{t-1}, \mathbf{x}_{t+1} | \mathbf{x}_0)} [D_{\text{KL}}(q(\mathbf{x}_t | \mathbf{x}_{t-1}) \| p_\theta(\mathbf{x}_t | \mathbf{x}_{t+1}))]}_{\text{consistency term}} \end{aligned} \quad (45)$$

ELBO 的导出形式可以以其各个组成部分来解释：

1.  $\mathbb{E}_{q(\mathbf{x}_1 | \mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0 | \mathbf{x}_1)]$  可以被解释为一个 重构项，预测给定第一步潜在变量的原始数据样本的对数概率。该项也出现在标准的 VAE 中，可以以类似的方式进行训练。
2.  $\mathbb{E}_{q(\mathbf{x}_{T-1} | \mathbf{x}_0)} [D_{\text{KL}}(q(\mathbf{x}_T | \mathbf{x}_{T-1}) \| p(\mathbf{x}_T))]$  是一个 先验匹配项；当最终潜在分布与高斯先验匹配时，该项被最小化。此术语不需要最优化，因为它没有可训练参数；此外，由于我们假设了足够大的  $T$  使得最终分布为高斯分布，因此该术语实际上变为零。
3.  $\mathbb{E}_{q(\mathbf{x}_{t-1}, \mathbf{x}_{t+1} | \mathbf{x}_0)} [D_{\text{KL}}(q(\mathbf{x}_t | \mathbf{x}_{t-1}) \| p_\theta(\mathbf{x}_t | \mathbf{x}_{t+1}))]$  是一个 一致性项；它努力使  $\mathbf{x}_t$  处的分布一致，从正

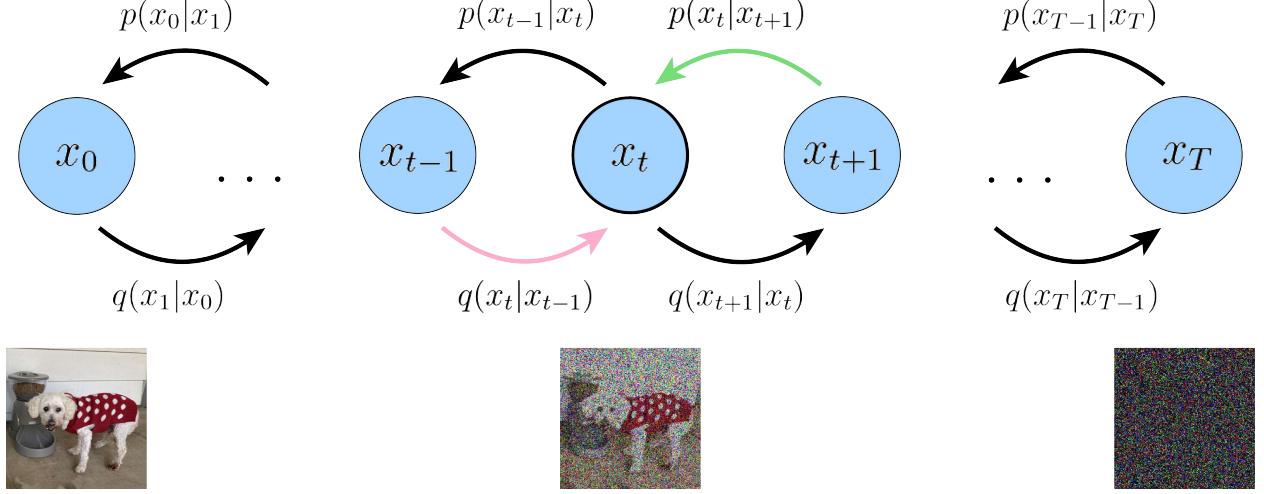


图 4: 在我们的第一个推导中, 可以通过确保对于每个中间  $\mathbf{x}_t$ , 其上方的潜在变量的后验  $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})$  与它之前的潜在变量的高斯扰动  $q(\mathbf{x}_t|\mathbf{x}_{t-1})$  匹配, 从而对 VDM 进行优化。在这张图中, 对于每个中间  $\mathbf{x}_t$ , 我们最小化由粉红色和绿色箭头表示的分布之间的差异。

向和反向过程中来看。也就是说, 从更嘈杂的图像进行降噪步骤应该与从更清晰的图像进行的相应加噪步骤相匹配, 对于每个中间时间步; 这在数学上由 KL 散度体现。当训练  $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})$  以匹配高斯分布  $q(\mathbf{x}_t|\mathbf{x}_{t-1})$  时, 该术语被最小化, 该分布定义在方程 31 中。

从视觉上看, ELBO 的这种解释在图 4 中有所描绘。优化 VDM 的成本主要由第三项主导, 因为我们必须对所有时间步  $t$  进行优化。

在这种推导下, ELBO 的所有项都被计算为期望, 因此可以使用蒙特卡罗估计进行近似。然而, 实际上使用我们刚刚推导出的项来优化 ELBO 可能是次优的; 因为一致性项是针对每个时间步对两个随机变量  $\{\mathbf{x}_{t-1}, \mathbf{x}_{t+1}\}$  取期望计算的, 其蒙特卡罗估计的方差可能比每个时间步仅使用一个随机变量进行估计的项的方差更高。由于它是通过求和  $T - 1$  个一致性项得到的, 因此当  $T$  值较大时, ELBO 的最终估计值可能具有较高的方差。

让我们尝试推导出一种 ELBO 的形式, 其中每个项都是仅对一个随机变量取期望来计算的。关键的见解是, 我们可以将编码器转移重写为  $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)$ , 其中额外的条件项由于马尔可夫性质而多余。然后, 根据贝叶斯规则, 我们可以将每个转移重写为:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) = \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} \quad (46)$$

借助这个新方程, 我们可以重新尝试推导, 从方程 37 中的 ELBO 开始:

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] \quad (47)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \quad (48)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)\prod_{t=2}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_1|\mathbf{x}_0)\prod_{t=2}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \quad (49)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)\prod_{t=2}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_1|\mathbf{x}_0)\prod_{t=2}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)} \right] \quad (50)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p_\theta(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \prod_{t=2}^T \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)} \right] \quad (51)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \prod_{t=2}^T \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}} \right] \quad (52)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \prod_{t=2}^T \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}} \right] \quad (53)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{q(\mathbf{x}_T|\mathbf{x}_0)} + \log \prod_{t=2}^T \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right] \quad (54)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_T|\mathbf{x}_0)} + \sum_{t=2}^T \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right] \quad (55)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)] + \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} \right] + \sum_{t=2}^T \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right] \quad (56)$$

$$= \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)] + \mathbb{E}_{q(\mathbf{x}_T|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} \right] + \sum_{t=2}^T \mathbb{E}_{q(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{x}_0)} \left[ \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right] \quad (57)$$

$$= \underbrace{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)]}_{\text{reconstruction term}} - \underbrace{D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^T \underbrace{\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} [D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))]}_{\text{denoising matching term}} \quad (58)$$

因此，我们成功地为 ELBO 提供了一个可以以较低方差进行估计的解释，因为每个项都是逐次计算最多一个随机变量的期望。这种表述也具有优美的解释，当检查每个单独的项时就会显现出来：

1.  $\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)]$  可以解释为一个重构项；类似于普通 VAE 中的对应项，该术语可以使用蒙特卡罗估计进行近似和优化。
2.  $D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))$  表示最终加噪输入的分布与标准高斯先验的接近程度。它没有可训练参数，并且在我们的假设下也等于零。
3.  $\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} [D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))]$  是一个降噪匹配项。我们学习期望的降噪转移步骤  $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$  作为对易处理的、真实降噪转移步骤  $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$  的近似。 $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$  转移步骤可以作为真实信号，因为它定义了如何在知道最终完全降噪图像  $\mathbf{x}_0$  应该是什么的情况下对噪声图像  $\mathbf{x}_t$  进行降噪。因此，当两个降噪步骤尽可能接近时，该术语会被最小化，这通过它们的 KL 散度来衡量。

作为旁注，可以观察到在两种 ELBO 推导（方程45和方程58）过程中，仅使用了马尔可夫假设；因此这些公式对于任何任意的马尔可夫 HVAE 都成立。此外，当我们设置  $T = 1$  时，VDM 的两种 ELBO 解释将确切地重现普通 VAE 的 ELBO 方程，如方程19所示。

在ELBO的推导中，最优化成本的大部分再次位于求和项中，它主导了重构项。而每个KL散度项  $D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))$  对于任意后验分布在任意复杂的马尔可夫 HVAE 中都难以最小化，这是由于同时学习编码器带来的复杂性。而在 VDM 中，我们可以利用高斯转移假设使最优化变得易处理。根据贝叶斯规则，我们有：

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}$$

As we already know that  $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) = q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I})$  from our assumption regarding encoder transitions (Equation 31), what remains is deriving for the forms of  $q(\mathbf{x}_t|\mathbf{x}_0)$  and  $q(\mathbf{x}_{t-1}|\mathbf{x}_0)$ . Fortunately, these are also made tractable by utilizing the fact that the encoder transitions of a VDM are

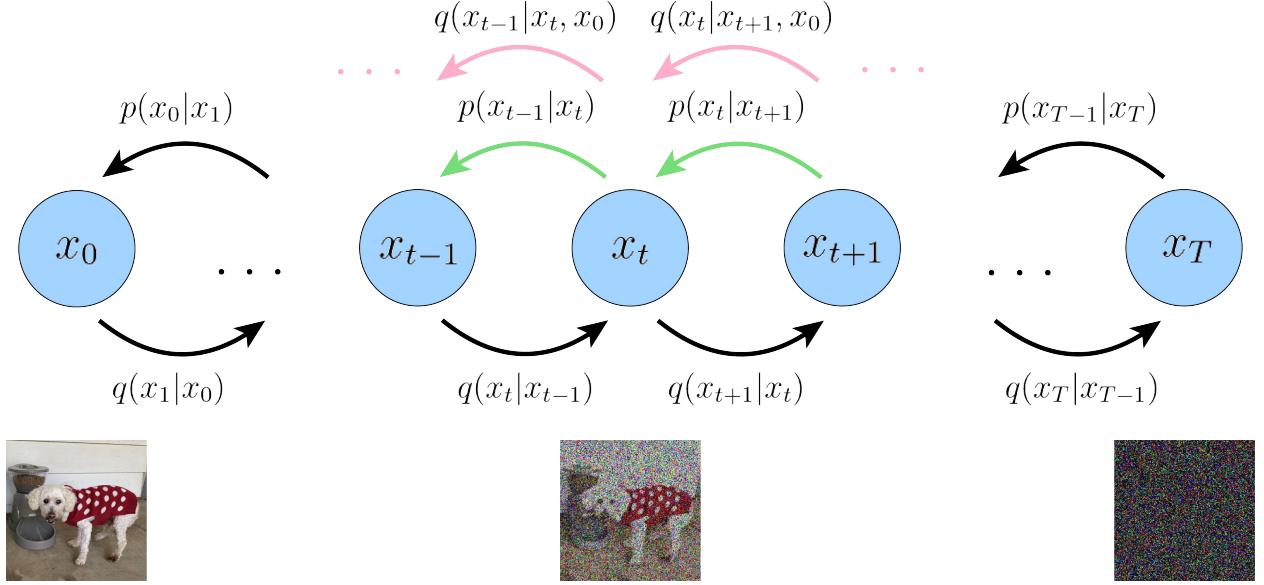


图 5: 图中展示了一种方差更低的优化 VDM 的方法; 我们使用贝叶斯规则计算真实降噪步骤  $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$  的形式，并最小化其与我们近似降噪步骤  $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$  的 KL 散度。这再次通过将绿色箭头所表示的分布与粉色箭头的分布匹配来直观表示。这里运用了艺术自由；在完整图像中，每个粉色箭头也必须来自  $\mathbf{x}_0$ ，因为它也是一个条件项。

linear Gaussian models. Recall that under the reparameterization trick, samples  $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_{t-1})$  can be rewritten as:

$$\mathbf{x}_t = \sqrt{\alpha_t} \mathbf{x}_{t-1} + \sqrt{1 - \alpha_t} \epsilon \quad \text{with } \epsilon \sim \mathcal{N}(\epsilon; 0, \mathbf{I}) \quad (59)$$

并且类似地，样本  $\mathbf{x}_{t-1} \sim q(\mathbf{x}_{t-1}|\mathbf{x}_{t-2})$  可以重写为：

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{1 - \alpha_{t-1}} \epsilon \quad \text{with } \epsilon \sim \mathcal{N}(\epsilon; 0, \mathbf{I}) \quad (60)$$

然后，通过反复应用重参数化技巧，可以递归地推导出  $q(\mathbf{x}_t|\mathbf{x}_0)$  的形式。假设我们有  $2T$  个随机噪声变量

$\{\epsilon_t^*, \epsilon_t\}_{t=0}^T \stackrel{\text{iid}}{\sim} \mathcal{N}(\epsilon; 0, \mathbf{I})$ 。那么，对于任意样本  $\mathbf{x}_t \sim q(\mathbf{x}_t | \mathbf{x}_0)$ ，我们可以将其重写为：

$$\mathbf{x}_t = \sqrt{\alpha_t} \mathbf{x}_{t-1} + \sqrt{1 - \alpha_t} \epsilon_{t-1}^* \quad (61)$$

$$= \sqrt{\alpha_t} \left( \sqrt{\alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{1 - \alpha_{t-1}} \epsilon_{t-2}^* \right) + \sqrt{1 - \alpha_t} \epsilon_{t-1}^* \quad (62)$$

$$= \sqrt{\alpha_t \alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{\alpha_t - \alpha_t \alpha_{t-1}} \epsilon_{t-2}^* + \sqrt{1 - \alpha_t} \epsilon_{t-1}^* \quad (63)$$

$$= \sqrt{\alpha_t \alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{\sqrt{\alpha_t - \alpha_t \alpha_{t-1}}^2 + \sqrt{1 - \alpha_t}^2} \epsilon_{t-2} \quad (64)$$

$$= \sqrt{\alpha_t \alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{\alpha_t - \alpha_t \alpha_{t-1} + 1 - \alpha_t} \epsilon_{t-2} \quad (65)$$

$$= \sqrt{\alpha_t \alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{1 - \alpha_t \alpha_{t-1}} \epsilon_{t-2} \quad (66)$$

$$= \dots \quad (67)$$

$$= \sqrt{\prod_{i=1}^t \alpha_i} \mathbf{x}_0 + \sqrt{1 - \prod_{i=1}^t \alpha_i} \epsilon_0 \quad (68)$$

$$= \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_0 \quad (69)$$

$$\sim \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}) \quad (70)$$

其中在公式 64 中，我们利用了 sum of two independent Gaussian random variables 仍然是一个高斯分布，其均值为两个均值之和，方差为两个方差之和。将  $\sqrt{1 - \alpha_t} \epsilon_{t-1}^*$  解释为从高斯分布  $\mathcal{N}(0, (1 - \alpha_t) \mathbf{I})$  中采样的样本，将  $\sqrt{\alpha_t - \alpha_t \alpha_{t-1}} \epsilon_{t-2}^*$  解释为从高斯分布  $\mathcal{N}(0, (\alpha_t - \alpha_t \alpha_{t-1}) \mathbf{I})$  中采样的样本，那么它们的和就可以视为从高斯分布  $\mathcal{N}(0, (1 - \alpha_t + \alpha_t - \alpha_t \alpha_{t-1}) \mathbf{I}) = \mathcal{N}(0, (1 - \alpha_t \alpha_{t-1}) \mathbf{I})$  中采样的随机变量。该分布的一个样本可以使用重参数化技巧表示为  $\sqrt{1 - \alpha_t \alpha_{t-1}} \epsilon_{t-2}$ ，如公式 66 所示。

我们因此推导出了  $q(\mathbf{x}_t | \mathbf{x}_0)$  的高斯形式。这个推导可以修改以同样得出描述  $q(\mathbf{x}_{t-1} | \mathbf{x}_0)$  的高斯参数化形式。现在，既然知道了  $q(\mathbf{x}_t | \mathbf{x}_0)$  和  $q(\mathbf{x}_{t-1} | \mathbf{x}_0)$  的形式，我们可以通过代入贝叶斯规则展开来计算  $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$  的形式：

$$q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_0)q(\mathbf{x}_{t-1} | \mathbf{x}_0)}{q(\mathbf{x}_t | \mathbf{x}_0)} \quad (71)$$

$$= \frac{\mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t} \mathbf{x}_{t-1}, (1 - \alpha_t) \mathbf{I}) \mathcal{N}(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0, (1 - \bar{\alpha}_{t-1}) \mathbf{I})}{\mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I})} \quad (72)$$

$$\propto \exp \left\{ - \left[ \frac{(\mathbf{x}_t - \sqrt{\alpha_t} \mathbf{x}_{t-1})^2}{2(1 - \alpha_t)} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0)^2}{2(1 - \bar{\alpha}_{t-1})} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \mathbf{x}_0)^2}{2(1 - \bar{\alpha}_t)} \right] \right\} \quad (73)$$

$$= \exp \left\{ - \frac{1}{2} \left[ \frac{(\mathbf{x}_t - \sqrt{\alpha_t} \mathbf{x}_{t-1})^2}{1 - \alpha_t} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0)^2}{1 - \bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \mathbf{x}_0)^2}{1 - \bar{\alpha}_t} \right] \right\} \quad (74)$$

$$= \exp \left\{ - \frac{1}{2} \left[ \frac{(-2\sqrt{\alpha_t} \mathbf{x}_t \mathbf{x}_{t-1} + \alpha_t \mathbf{x}_{t-1}^2)}{1 - \alpha_t} + \frac{(\mathbf{x}_{t-1}^2 - 2\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_{t-1} \mathbf{x}_0)}{1 - \bar{\alpha}_{t-1}} + C(\mathbf{x}_t, \mathbf{x}_0) \right] \right\} \quad (75)$$

$$\propto \exp \left\{ - \frac{1}{2} \left[ - \frac{2\sqrt{\alpha_t} \mathbf{x}_t \mathbf{x}_{t-1}}{1 - \alpha_t} + \frac{\alpha_t \mathbf{x}_{t-1}^2}{1 - \alpha_t} + \frac{\mathbf{x}_{t-1}^2}{1 - \bar{\alpha}_{t-1}} - \frac{2\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_{t-1} \mathbf{x}_0}{1 - \bar{\alpha}_{t-1}} \right] \right\} \quad (76)$$

$$= \exp \left\{ - \frac{1}{2} \left[ \left( \frac{\alpha_t}{1 - \alpha_t} + \frac{1}{1 - \bar{\alpha}_{t-1}} \right) \mathbf{x}_{t-1}^2 - 2 \left( \frac{\sqrt{\alpha_t} \mathbf{x}_t}{1 - \alpha_t} + \frac{\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0}{1 - \bar{\alpha}_{t-1}} \right) \mathbf{x}_{t-1} \right] \right\} \quad (77)$$

$$= \exp \left\{ - \frac{1}{2} \left[ \frac{\alpha_t(1 - \bar{\alpha}_{t-1}) + 1 - \alpha_t}{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \mathbf{x}_{t-1}^2 - 2 \left( \frac{\sqrt{\alpha_t} \mathbf{x}_t}{1 - \alpha_t} + \frac{\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0}{1 - \bar{\alpha}_{t-1}} \right) \mathbf{x}_{t-1} \right] \right\} \quad (78)$$

$$= \exp \left\{ - \frac{1}{2} \left[ \frac{\alpha_t - \bar{\alpha}_t + 1 - \alpha_t}{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \mathbf{x}_{t-1}^2 - 2 \left( \frac{\sqrt{\alpha_t} \mathbf{x}_t}{1 - \alpha_t} + \frac{\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0}{1 - \bar{\alpha}_{t-1}} \right) \mathbf{x}_{t-1} \right] \right\} \quad (79)$$

$$= \exp \left\{ - \frac{1}{2} \left[ \frac{1 - \bar{\alpha}_t}{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \mathbf{x}_{t-1}^2 - 2 \left( \frac{\sqrt{\alpha_t} \mathbf{x}_t}{1 - \alpha_t} + \frac{\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0}{1 - \bar{\alpha}_{t-1}} \right) \mathbf{x}_{t-1} \right] \right\} \quad (80)$$

$$= \exp \left\{ - \frac{1}{2} \left( \frac{1 - \bar{\alpha}_t}{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \right) \left[ \mathbf{x}_{t-1}^2 - 2 \frac{\left( \frac{\sqrt{\alpha_t} \mathbf{x}_t}{1 - \alpha_t} + \frac{\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0}{1 - \bar{\alpha}_{t-1}} \right)}{\frac{1 - \bar{\alpha}_t}{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}} \mathbf{x}_{t-1} \right] \right\} \quad (81)$$

$$= \exp \left\{ - \frac{1}{2} \left( \frac{1 - \bar{\alpha}_t}{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \right) \left[ \mathbf{x}_{t-1}^2 - 2 \frac{\left( \frac{\sqrt{\alpha_t} \mathbf{x}_t}{1 - \alpha_t} + \frac{\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0}{1 - \bar{\alpha}_{t-1}} \right) (1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{x}_{t-1} \right] \right\} \quad (82)$$

$$= \exp \left\{ - \frac{1}{2} \left( \frac{1}{\frac{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}} \right) \left[ \mathbf{x}_{t-1}^2 - 2 \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1}) \mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t) \mathbf{x}_0}{1 - \bar{\alpha}_t} \mathbf{x}_{t-1} \right] \right\} \quad (83)$$

$$\propto \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1}) \mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t) \mathbf{x}_0}{1 - \bar{\alpha}_t}}_{\mu_q(\mathbf{x}_t, \mathbf{x}_0)}, \underbrace{\frac{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{I}}_{\Sigma_q(t)}) \quad (84)$$

其中在公式 75 中， $C(\mathbf{x}_t, \mathbf{x}_0)$  是关于  $\mathbf{x}_{t-1}$  的常数项，其计算仅涉及  $\mathbf{x}_t$ 、 $\mathbf{x}_0$  和  $\alpha$  的值；该术语在公式 84 中隐式返回，以完成方阵。

因此，我们已经证明，在每一步中， $\mathbf{x}_{t-1} \sim q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$  服从正态分布，均值  $\mu_q(\mathbf{x}_t, \mathbf{x}_0)$  是  $\mathbf{x}_t$  和  $\mathbf{x}_0$  的函数，方差  $\Sigma_q(t)$  是  $\alpha$  系数的函数。这些  $\alpha$  系数在每个时间步都是已知且固定的；它们要么在作为超参数建模时被永久设置，要么被视为试图对它们进行建模的网络的当前推理输出。根据公式 84，我们可以将方差方程重写为  $\Sigma_q(t) = \sigma_q^2(t) \mathbf{I}$ ，其中：

$$\sigma_q^2(t) = \frac{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \quad (85)$$

为了尽可能接近匹配近似降噪转移步骤  $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$  与真实降噪转移步骤  $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$ ，我们也可以将其建模为高斯分布。此外，由于所有  $\alpha$  项在每个时间步都被已知冻结，我们可以立即构建近似降噪转移步骤的

方差也为此  $\Sigma_q(t) = \sigma_q^2(t)\mathbf{I}$ 。然而，我们必须将它的均值  $\mu_\theta(\mathbf{x}_t, t)$  参数化为  $\mathbf{x}_t$  的函数，因为  $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$  并不依赖于  $\mathbf{x}_0$ 。

回想一下，KL Divergence between two Gaussian distributions 是：

$$D_{\text{KL}}(\mathcal{N}(\mathbf{x}; \mu_x, \Sigma_x) \| \mathcal{N}(\mathbf{y}; \mu_y, \Sigma_y)) = \frac{1}{2} \left[ \log \frac{|\Sigma_y|}{|\Sigma_x|} - d + \text{tr}(\Sigma_y^{-1} \Sigma_x) + (\mu_y - \mu_x)^T \Sigma_y^{-1} (\mu_y - \mu_x) \right] \quad (86)$$

在我们的情况下，可以将两个高斯分布的方差设置为完全匹配，优化 KL 散度项就转化为最小化两个分布均值之间的差异：

$$\begin{aligned} & \arg \min_{\theta} D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) \\ &= \arg \min_{\theta} D_{\text{KL}}(\mathcal{N}(\mathbf{x}_{t-1}; \mu_q, \Sigma_q(t)) \| \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta, \Sigma_q(t))) \end{aligned} \quad (87)$$

$$= \arg \min_{\theta} \frac{1}{2} \left[ \log \frac{|\Sigma_q(t)|}{|\Sigma_q(t)|} - d + \text{tr}(\Sigma_q(t)^{-1} \Sigma_q(t)) + (\mu_\theta - \mu_q)^T \Sigma_q(t)^{-1} (\mu_\theta - \mu_q) \right] \quad (88)$$

$$= \arg \min_{\theta} \frac{1}{2} [\log 1 - d + d + (\mu_\theta - \mu_q)^T \Sigma_q(t)^{-1} (\mu_\theta - \mu_q)] \quad (89)$$

$$= \arg \min_{\theta} \frac{1}{2} [(\mu_\theta - \mu_q)^T \Sigma_q(t)^{-1} (\mu_\theta - \mu_q)] \quad (90)$$

$$= \arg \min_{\theta} \frac{1}{2} [(\mu_\theta - \mu_q)^T (\sigma_q^2(t)\mathbf{I})^{-1} (\mu_\theta - \mu_q)] \quad (91)$$

$$= \arg \min_{\theta} \frac{1}{2\sigma_q^2(t)} [\|\mu_\theta - \mu_q\|_2^2] \quad (92)$$

其中我们用  $\mu_q$  作为  $\mu_q(\mathbf{x}_t, \mathbf{x}_0)$  的简写，用  $\mu_\theta$  作为  $\mu_\theta(\mathbf{x}_t, t)$  的简写以简化表达。换句话说，我们想要优化一个与  $\mu_q(\mathbf{x}_t, \mathbf{x}_0)$  匹配的  $\mu_\theta(\mathbf{x}_t, t)$ ，根据我们推导出的方程 84，其形式为：

$$\mu_q(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\mathbf{x}_0}{1 - \bar{\alpha}_t} \quad (93)$$

由于  $\mu_\theta(\mathbf{x}_t, t)$  也对  $\mathbf{x}_t$  进行条件判断，我们可以将  $\mu_q(\mathbf{x}_t, \mathbf{x}_0)$  设置为以下形式以紧密匹配：

$$\mu_\theta(\mathbf{x}_t, t) = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t)}{1 - \bar{\alpha}_t} \quad (94)$$

其中  $\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t)$  由一个神经网络参数化，该神经网络旨在从噪声图像  $\mathbf{x}_t$  和时间索引  $t$  预测  $\mathbf{x}_0$ 。然后，最优化问题简化为：

$$\begin{aligned} & \arg \min_{\theta} D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) \\ &= \arg \min_{\theta} D_{\text{KL}}(\mathcal{N}(\mathbf{x}_{t-1}; \mu_q, \Sigma_q(t)) \| \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta, \Sigma_q(t))) \end{aligned} \quad (95)$$

$$= \arg \min_{\theta} \frac{1}{2\sigma_q^2(t)} \left[ \left\| \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t)}{1 - \bar{\alpha}_t} - \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\mathbf{x}_0}{1 - \bar{\alpha}_t} \right\|_2^2 \right] \quad (96)$$

$$= \arg \min_{\theta} \frac{1}{2\sigma_q^2(t)} \left[ \left\| \frac{\sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t)}{1 - \bar{\alpha}_t} - \frac{\sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\mathbf{x}_0}{1 - \bar{\alpha}_t} \right\|_2^2 \right] \quad (97)$$

$$= \arg \min_{\theta} \frac{1}{2\sigma_q^2(t)} \left[ \left\| \frac{\sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)}{1 - \bar{\alpha}_t} (\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0) \right\|_2^2 \right] \quad (98)$$

$$= \arg \min_{\theta} \frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1 - \alpha_t)^2}{(1 - \bar{\alpha}_t)^2} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (99)$$

因此，优化一个 VDM 归结为学习一个神经网络，从任意噪声化的版本中预测原始真实值图像 [5]。此外，通过在所有噪声级别上最小化我们推导出的 ELBO 目标的求和项（方程 58）可以近似为在所有时间步上最小化期望：

$$\arg \min_{\theta} \mathbb{E}_{t \sim U\{2, T\}} [\mathbb{E}_{q(\mathbf{x}_t | \mathbf{x}_0)} [D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t))]] \quad (100)$$

然后可以使用随机样本在时间步上进行优化。

### 学习扩散噪声参数

让我们研究一下如何联合学习 VDM 的噪声参数。一种可能的方法是使用神经网络  $\alpha_t$  来建模  $\hat{\alpha}_\eta(t)$ ，其参数为  $\eta$ 。然而，这种方法效率不高，因为在每个时间步  $t$  必须进行多次推理以计算  $\bar{\alpha}_t$ 。虽然缓存可以减轻这种计算成本，但我们也推导出一种学习扩散噪声参数的替代方法。通过将方程 85 中的方差公式代入方程 99 中推导出的每一步目标，我们可以减少：

$$\frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] = \frac{1}{2 \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (101)$$

$$= \frac{1}{2} \frac{1-\bar{\alpha}_t}{(1-\alpha_t)(1-\bar{\alpha}_{t-1})} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (102)$$

$$= \frac{1}{2} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)}{(1-\bar{\alpha}_{t-1})(1-\bar{\alpha}_t)} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (103)$$

$$= \frac{1}{2} \frac{\bar{\alpha}_{t-1}-\bar{\alpha}_t}{(1-\bar{\alpha}_{t-1})(1-\bar{\alpha}_t)} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (104)$$

$$= \frac{1}{2} \frac{\bar{\alpha}_{t-1}-\bar{\alpha}_{t-1}\bar{\alpha}_t+\bar{\alpha}_{t-1}\bar{\alpha}_t-\bar{\alpha}_t}{(1-\bar{\alpha}_{t-1})(1-\bar{\alpha}_t)} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (105)$$

$$= \frac{1}{2} \frac{\bar{\alpha}_{t-1}(1-\bar{\alpha}_t)-\bar{\alpha}_t(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_{t-1})(1-\bar{\alpha}_t)} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (106)$$

$$= \frac{1}{2} \left( \frac{\bar{\alpha}_{t-1}(1-\bar{\alpha}_t)}{(1-\bar{\alpha}_{t-1})(1-\bar{\alpha}_t)} - \frac{\bar{\alpha}_t(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_{t-1})(1-\bar{\alpha}_t)} \right) [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (107)$$

$$= \frac{1}{2} \left( \frac{\bar{\alpha}_{t-1}}{1-\bar{\alpha}_{t-1}} - \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t} \right) [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (108)$$

从公式 70 可以看出  $q(\mathbf{x}_t | \mathbf{x}_0)$  是形式为  $\mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1-\bar{\alpha}_t) \mathbf{I})$  的高斯分布。然后，根据 signal-to-noise ratio (SNR) 的定义为  $\text{SNR} = \frac{\mu^2}{\sigma^2}$ ，我们可以将每个时间步的 SNR  $t$  表示为：

$$\text{SNR}(t) = \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t} \quad (109)$$

然后，我们推导出的方程 108（以及方程 99）可以简化为：

$$\frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] = \frac{1}{2} (\text{SNR}(t-1) - \text{SNR}(t)) [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (110)$$

正如名称所示，信噪比 (SNR) 表示原始信号与存在的噪声量之间的比率；较高的 SNR 表示信号更多，较低的 SNR 表示噪声更多。在扩散模型中，我们要求 SNR 随着步数  $t$  的增加而单调递减；这形式化了被扰动的输入  $\mathbf{x}_t$  随时间推移变得越来越嘈杂的概念，直到它与标准高斯分布在  $t=T$  处完全相同。

按照公式 110 中目标的简化，我们可以直接使用神经网络在每个时间步对信噪比进行参数化，并与扩散模型一起学习。由于信噪比必须随时间单调减少，我们可以将其表示为：

$$\text{SNR}(t) = \exp(-\omega_\eta(t)) \quad (111)$$

其中  $\omega_\eta(t)$  被建模为一个具有参数  $\eta$  的单调递增神经网络。将  $\omega_\eta(t)$  取反得到一个单调递减函数，而指数运算则使结果项为正数。注意，方程 100 中的目标现在还需要对  $\eta$  进行优化。通过将方程 111 中的信噪比参数化方法与方程 109 中信噪比的定义相结合，我们还可以显式推导出  $\bar{\alpha}_t$  和  $1 - \bar{\alpha}_t$  值的优美形式：

$$\frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} = \exp(-\omega_\eta(t)) \quad (112)$$

$$\therefore \bar{\alpha}_t = \text{sigmoid}(-\omega_\eta(t)) \quad (113)$$

$$\therefore 1 - \bar{\alpha}_t = \text{sigmoid}(\omega_\eta(t)) \quad (114)$$

这些术语对于各种计算是必要的；例如，在最优化过程中，它们用于使用重参数化技巧从输入  $\mathbf{x}_0$  生成任意噪声的  $\mathbf{x}_t$ ，如公式 69 中推导得出。

### 三种等效解释

正如我们之前所证明的，一个变分扩散模型可以通过简单地学习一个神经网络来训练，该网络用于从任意加噪版本  $\mathbf{x}_t$  及其时间索引  $t$  预测原始自然图像  $\mathbf{x}_0$ 。然而， $\mathbf{x}_0$  还有另外两种等价的参数化形式，这为变分扩散模型带来了另外两种解释。

首先，我们可以利用重参数化技巧。在推导  $q(\mathbf{x}_t | \mathbf{x}_0)$  的形式时，我们可以重新排列方程 69 来证明：

$$\mathbf{x}_0 = \frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_0}{\sqrt{\bar{\alpha}_t}} \quad (115)$$

将此代入我们之前推导出的真实降噪转移均值  $\mu_q(\mathbf{x}_t, \mathbf{x}_0)$ ，我们可以重新推导为：

$$\mu_q(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\mathbf{x}_0}{1 - \bar{\alpha}_t} \quad (116)$$

$$= \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t) \frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_0}{\sqrt{\bar{\alpha}_t}}}{1 - \bar{\alpha}_t} \quad (117)$$

$$= \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + (1 - \alpha_t) \frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_0}{\sqrt{\bar{\alpha}_t}}}{1 - \bar{\alpha}_t} \quad (118)$$

$$= \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t}{1 - \bar{\alpha}_t} + \frac{(1 - \alpha_t)\mathbf{x}_t}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} - \frac{(1 - \alpha_t)\sqrt{1 - \bar{\alpha}_t} \epsilon_0}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} \quad (119)$$

$$= \left( \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} + \frac{1 - \alpha_t}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} \right) \mathbf{x}_t - \frac{(1 - \alpha_t)\sqrt{1 - \bar{\alpha}_t}}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} \epsilon_0 \quad (120)$$

$$= \left( \frac{\alpha_t(1 - \bar{\alpha}_{t-1})}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} + \frac{1 - \alpha_t}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} \right) \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}\sqrt{\alpha_t}} \epsilon_0 \quad (121)$$

$$= \frac{\alpha_t - \bar{\alpha}_t + 1 - \alpha_t}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}\sqrt{\alpha_t}} \epsilon_0 \quad (122)$$

$$= \frac{1 - \bar{\alpha}_t}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}\sqrt{\alpha_t}} \epsilon_0 \quad (123)$$

$$= \frac{1}{\sqrt{\alpha_t}} \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}\sqrt{\alpha_t}} \epsilon_0 \quad (124)$$

因此，我们可以将我们的近似降噪转移均值  $\mu_\theta(\mathbf{x}_t, t)$  设置为：

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}\sqrt{\alpha_t}} \hat{\epsilon}_\theta(\mathbf{x}_t, t) \quad (125)$$

并且相应的最优化问题变为：

$$\begin{aligned} & \arg \min_{\theta} D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t)) \\ &= \arg \min_{\theta} D_{\text{KL}}(\mathcal{N}(\mathbf{x}_{t-1}; \mu_q, \Sigma_q(t)) \| \mathcal{N}(\mathbf{x}_{t-1}; \mu_{\theta}, \Sigma_q(t))) \end{aligned} \quad (126)$$

$$= \arg \min_{\theta} \frac{1}{2\sigma_q^2(t)} \left[ \left\| \frac{1}{\sqrt{\alpha_t}} \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} \hat{\epsilon}_{\theta}(\mathbf{x}_t, t) - \frac{1}{\sqrt{\alpha_t}} \mathbf{x}_t + \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} \epsilon_0 \right\|_2^2 \right] \quad (127)$$

$$= \arg \min_{\theta} \frac{1}{2\sigma_q^2(t)} \left[ \left\| \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} \epsilon_0 - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} \hat{\epsilon}_{\theta}(\mathbf{x}_t, t) \right\|_2^2 \right] \quad (128)$$

$$= \arg \min_{\theta} \frac{1}{2\sigma_q^2(t)} \left[ \left\| \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} (\epsilon_0 - \hat{\epsilon}_{\theta}(\mathbf{x}_t, t)) \right\|_2^2 \right] \quad (129)$$

$$= \arg \min_{\theta} \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t} [\|\epsilon_0 - \hat{\epsilon}_{\theta}(\mathbf{x}_t, t)\|_2^2] \quad (130)$$

在这里， $\hat{\epsilon}_{\theta}(\mathbf{x}_t, t)$  是一个神经网络，它学习预测决定  $\mathbf{x}_t$  的源噪声  $\epsilon_0 \sim \mathcal{N}(\epsilon; 0, \mathbf{I})$ ，从  $\mathbf{x}_0$ 。因此，我们已经证明，通过预测原始图像  $\mathbf{x}_0$  来学习 VDM 等同于学习预测噪声；然而，实证研究表明，预测噪声取得了更好的性能 [5, 7]。

为了推导变分扩散模型的第三种常见解释，我们引用 Tweedie 公式 [8]。在英文中，Tweedie 公式指出，给定从指数族分布中抽取的样本，该分布的真实均值可以通过样本的极大似然估计（即经验均值）加上一个涉及估计得分的校正项来估计。在仅有一个观测样本的情况下，经验均值就是该样本本身。它常用于减轻样本偏差；如果观测样本都位于潜在分布的一端，那么负得分会变得很大，并将样本的朴素极大似然估计向真实均值进行修正。

数学上，对于高斯变量  $\mathbf{z} \sim \mathcal{N}(\mathbf{z}; \mu_z, \Sigma_z)$ ，Tweedie 公式表述为：

$$\mathbb{E}[\mu_z | \mathbf{z}] = \mathbf{z} + \Sigma_z \nabla_{\mathbf{z}} \log p(\mathbf{z})$$

在这种情况下，我们将其用于预测给定其样本的  $\mathbf{x}_t$  的真实后验均值。根据公式 70，我们知道：

$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1-\bar{\alpha}_t) \mathbf{I})$$

然后，根据 Tweedie 公式，我们有：

$$\mathbb{E}[\mu_{x_t} | \mathbf{x}_t] = \mathbf{x}_t + (1-\bar{\alpha}_t) \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) \quad (131)$$

其中我们写作  $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$  作为  $\nabla \log p(\mathbf{x}_t)$  以简化符号。根据 Tweedie 公式，对于由  $\mathbf{x}_t$  生成的真实均值  $\mu_{x_t} = \sqrt{\bar{\alpha}_t} \mathbf{x}_0$  的最佳估计定义为：

$$\sqrt{\bar{\alpha}_t} \mathbf{x}_0 = \mathbf{x}_t + (1-\bar{\alpha}_t) \nabla \log p(\mathbf{x}_t) \quad (132)$$

$$\therefore \mathbf{x}_0 = \frac{\mathbf{x}_t + (1-\bar{\alpha}_t) \nabla \log p(\mathbf{x}_t)}{\sqrt{\bar{\alpha}_t}} \quad (133)$$

然后，我们可以将方程 133 再次代入我们的真实降噪转移均值  $\mu_q(\mathbf{x}_t, \mathbf{x}_0)$ ，推导出一种新的形式：

$$\mu_q(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\mathbf{x}_0}{1 - \bar{\alpha}_t} \quad (134)$$

$$= \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\frac{\mathbf{x}_t + (1 - \bar{\alpha}_t)\nabla \log p(\mathbf{x}_t)}{\sqrt{\bar{\alpha}_t}}}{1 - \bar{\alpha}_t} \quad (135)$$

$$= \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + (1 - \alpha_t)\frac{\mathbf{x}_t + (1 - \bar{\alpha}_t)\nabla \log p(\mathbf{x}_t)}{\sqrt{\alpha_t}}}{1 - \bar{\alpha}_t} \quad (136)$$

$$= \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t}{1 - \bar{\alpha}_t} + \frac{(1 - \alpha_t)\mathbf{x}_t}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} + \frac{(1 - \alpha_t)(1 - \bar{\alpha}_t)\nabla \log p(\mathbf{x}_t)}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} \quad (137)$$

$$= \left( \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} + \frac{1 - \alpha_t}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} \right) \mathbf{x}_t + \frac{1 - \alpha_t}{\sqrt{\alpha_t}} \nabla \log p(\mathbf{x}_t) \quad (138)$$

$$= \left( \frac{\alpha_t(1 - \bar{\alpha}_{t-1})}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} + \frac{1 - \alpha_t}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} \right) \mathbf{x}_t + \frac{1 - \alpha_t}{\sqrt{\alpha_t}} \nabla \log p(\mathbf{x}_t) \quad (139)$$

$$= \frac{\alpha_t - \bar{\alpha}_t + 1 - \alpha_t}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} \mathbf{x}_t + \frac{1 - \alpha_t}{\sqrt{\alpha_t}} \nabla \log p(\mathbf{x}_t) \quad (140)$$

$$= \frac{1 - \bar{\alpha}_t}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} \mathbf{x}_t + \frac{1 - \alpha_t}{\sqrt{\alpha_t}} \nabla \log p(\mathbf{x}_t) \quad (141)$$

$$= \frac{1}{\sqrt{\alpha_t}} \mathbf{x}_t + \frac{1 - \alpha_t}{\sqrt{\alpha_t}} \nabla \log p(\mathbf{x}_t) \quad (142)$$

因此，我们也可以将我们的近似降噪转移均值  $\mu_\theta(\mathbf{x}_t, t)$  设置为：

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \mathbf{x}_t + \frac{1 - \alpha_t}{\sqrt{\alpha_t}} \mathbf{s}_\theta(\mathbf{x}_t, t) \quad (143)$$

并且相应的最优化问题变为：

$$\begin{aligned} & \arg \min_{\theta} D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)) \\ &= \arg \min_{\theta} D_{\text{KL}}(\mathcal{N}(\mathbf{x}_{t-1}; \mu_q, \Sigma_q(t)) \| \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta, \Sigma_q(t))) \end{aligned} \quad (144)$$

$$= \arg \min_{\theta} \frac{1}{2\sigma_q^2(t)} \left[ \left\| \frac{1}{\sqrt{\alpha_t}} \mathbf{x}_t + \frac{1 - \alpha_t}{\sqrt{\alpha_t}} \mathbf{s}_\theta(\mathbf{x}_t, t) - \frac{1}{\sqrt{\alpha_t}} \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{\alpha_t}} \nabla \log p(\mathbf{x}_t) \right\|_2^2 \right] \quad (145)$$

$$= \arg \min_{\theta} \frac{1}{2\sigma_q^2(t)} \left[ \left\| \frac{1 - \alpha_t}{\sqrt{\alpha_t}} \mathbf{s}_\theta(\mathbf{x}_t, t) - \frac{1 - \alpha_t}{\sqrt{\alpha_t}} \nabla \log p(\mathbf{x}_t) \right\|_2^2 \right] \quad (146)$$

$$= \arg \min_{\theta} \frac{1}{2\sigma_q^2(t)} \left[ \left\| \frac{1 - \alpha_t}{\sqrt{\alpha_t}} (\mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla \log p(\mathbf{x}_t)) \right\|_2^2 \right] \quad (147)$$

$$= \arg \min_{\theta} \frac{1}{2\sigma_q^2(t)} \frac{(1 - \alpha_t)^2}{\alpha_t} \left[ \|\mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla \log p(\mathbf{x}_t)\|_2^2 \right] \quad (148)$$

在这里， $\mathbf{s}_\theta(\mathbf{x}_t, t)$  是一个神经网络，它学习预测评分函数  $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ ，这是数据空间中  $\mathbf{x}_t$  的梯度，适用于任何任意噪声水平  $t$ 。

细心的读者会注意到评分函数  $\nabla \log p(\mathbf{x}_t)$  在形式上与源噪声  $\epsilon_0$  非常相似。这可以通过将 Tweedie 公式（方

程 133) 与重参数化技巧 (方程 115) 结合来明确展示:

$$\mathbf{x}_0 = \frac{\mathbf{x}_t + (1 - \bar{\alpha}_t) \nabla \log p(\mathbf{x}_t)}{\sqrt{\bar{\alpha}_t}} = \frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_0}{\sqrt{\bar{\alpha}_t}} \quad (149)$$

$$\therefore (1 - \bar{\alpha}_t) \nabla \log p(\mathbf{x}_t) = -\sqrt{1 - \bar{\alpha}_t} \epsilon_0 \quad (150)$$

$$\nabla \log p(\mathbf{x}_t) = -\frac{1}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_0 \quad (151)$$

结果表明, 这两个项之间的差异是一个随时间缩放的常数因子! 评分函数衡量了如何在数据空间中移动以最大化对数概率; 直观上, 由于源噪声被添加到自然图像中以使其失真, 因此向相反方向移动可以“去噪”图像, 并且是提高后续对数概率的最佳更新。我们的数学证明验证了这种直觉; 我们明确证明了学习建模评分函数等同于建模源噪声的负数 (忽略一个缩放因子)。

因此, 我们推导出了三个等效的目标来优化 VDM: 学习一个神经网络以预测原始图像  $\mathbf{x}_0$ , 源噪声  $\epsilon_0$ , 或在任意噪声水平下的图像得分  $\nabla \log p(\mathbf{x}_t)$ 。VDM 可以通过随机采样时间步  $t$  并最小化预测值与真实值目标之间的范数来可扩展地进行训练。

## 基于得分的生成式模型

我们已经证明, 通过优化神经网络  $s_\theta(\mathbf{x}_t, t)$  来预测评分函数  $\nabla \log p(\mathbf{x}_t)$ , 可以简单地学习到一个变分扩散模型。然而, 在我们的推导中, 评分项是通过应用 Tweedie 公式得到的; 这并不一定为我们提供关于评分函数到底是什么以及为什么值得建模的深刻直觉或洞察。幸运的是, 我们可以参考另一类生成式模型——基于评分的生成模型 [9, 10, 11], 以获得这种直觉。事实上, 我们可以证明之前推导出的 VDM 公式与基于评分的生成建模公式等价, 这使我们能够灵活地在两种解释之间切换。

为了开始理解为什么优化一个评分函数是有意义的, 我们绕道重新回顾基于能量的模型 [12, 13]。任意灵活的概率分布都可以写成以下形式:

$$p_\theta(\mathbf{x}) = \frac{1}{Z_\theta} e^{-f_\theta(\mathbf{x})} \quad (152)$$

其中  $f_\theta(\mathbf{x})$  是一个任意灵活的、可参数化的函数, 称为能量函数, 通常由神经网络建模, 而  $Z_\theta$  是一个归一化常数, 以确保  $\int p_\theta(\mathbf{x}) d\mathbf{x} = 1$ 。学习这种分布的一种方法是极大似然; 然而, 这需要可计算的归一化常数  $Z_\theta = \int e^{-f_\theta(\mathbf{x})} d\mathbf{x}$ , 对于复杂的  $f_\theta(\mathbf{x})$  函数来说, 这可能无法实现。

避免计算或建模归一化常数的一种方法是使用神经网络  $s_\theta(\mathbf{x})$  来学习分布  $p(\mathbf{x})$  的评分函数  $\nabla \log p(\mathbf{x})$ 。这是受到以下观察结果的启发: 对等式 152 两边取对数的导数得到:

$$\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) = \nabla_{\mathbf{x}} \log \left( \frac{1}{Z_\theta} e^{-f_\theta(\mathbf{x})} \right) \quad (153)$$

$$= \nabla_{\mathbf{x}} \log \frac{1}{Z_\theta} + \nabla_{\mathbf{x}} \log e^{-f_\theta(\mathbf{x})} \quad (154)$$

$$= -\nabla_{\mathbf{x}} f_\theta(\mathbf{x}) \quad (155)$$

$$\approx s_\theta(\mathbf{x}) \quad (156)$$

这可以自由表示为一个不涉及任何规范化常数的神经网络。得分模型可以通过最小化与真实得分函数的 Fisher 散度来优化:

$$\mathbb{E}_{p(\mathbf{x})} \left[ \|s_\theta(\mathbf{x}) - \nabla \log p(\mathbf{x})\|_2^2 \right] \quad (157)$$

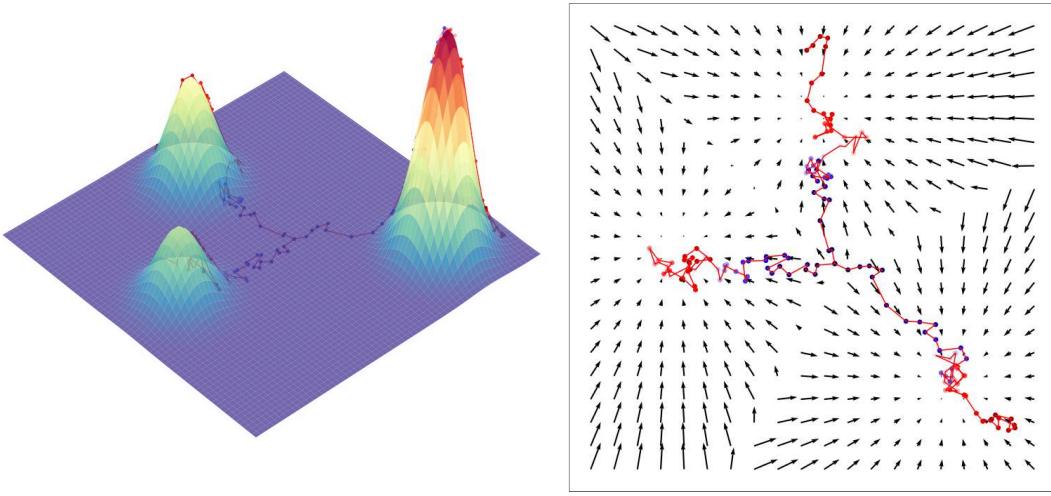


图 6: 使用 Langevin 动力学生成的三个随机采样轨迹的可视化，所有轨迹都从同一初始点开始，用于高斯混合模型。左图在三维等高线图上绘制了这些采样轨迹，而右图则将采样轨迹与真实得分函数进行了对比。从相同的初始点出发，由于 Langevin 动力学采样过程中存在随机噪声项，我们能够从不同的模式中生成样本；如果没有这个噪声项，从固定点进行采样每次试验都会确定性地沿着得分函数到达同一个模式。

得分函数代表什么？对于每个  $\mathbf{x}$ ，对其对数似然关于  $\mathbf{x}$  求梯度，本质上描述了在数据空间中为了进一步增加其似然性应该移动的方向。直观地说，得分函数定义了数据  $\mathbf{x}$  所处整个空间上的一个向量场，指向各个模式。从视觉上看，这在图6的右图中有所描绘。然后，通过学习真实数据分布的得分函数，我们可以在同一空间中的任意点开始，并迭代地沿着得分方向移动，直到达到一个模式。这种采样过程称为 Langevin 动力学，其数学描述如下：

$$\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + c \nabla \log p(\mathbf{x}_i) + \sqrt{2c\epsilon}, \quad i = 0, 1, \dots, K \quad (158)$$

其中  $\mathbf{x}_0$  是从先验分布（如均匀分布）中随机采样的，而  $\epsilon \sim \mathcal{N}(\epsilon; 0, \mathbf{I})$  是一个额外的噪声项，以确保生成的样本不会总是坍缩到一个模式上，而是在其周围保持多样性。此外，由于学成的得分函数是确定性的，涉及噪声项的采样会为生成过程增加随机性，使我们能够避免确定性的轨迹。当采样从位于多个模式之间的位置开始时，这尤其有用。Langevin 动力学采样以及噪声项的优势如图 6 所示。

注意，方程 157 中的目标函数依赖于访问真实得分函数，而对我们来说，对于复杂分布（如建模自然图像的分布）而言，这是不可用的。幸运的是，已推导出一些称为分数匹配 [14, 15, 16, 17] 的替代技术，可以在不知道真实得分函数的情况下最小化这种费舍尔散度，并且可以使用随机梯度下降进行优化。

共同地，将分布表示为评分函数并使用它通过马尔可夫链蒙特卡罗技术（如 Langevin 动力学）生成样本的学习过程被称为基于评分的生成建模 [9, 10, 11]。

vanilla 分数匹配有三个主要问题，如 Song and Ermon [9] 所详细描述。首先，当  $\mathbf{x}$  位于高维空间中的低维流形上时，分数函数是未定义的。这可以从数学上看出：所有不在低维流形上的点的概率都为零，其对数是未定义的。当尝试在自然图像上学习生成式模型时，这尤其不方便，因为已知自然图像位于整个环境空间的低维流形上。

其次，通过原始分数匹配训练的估计分数函数在低密度区域将不准确。这从我们在公式 157 中最小化的目标中可以明显看出。因为它是对  $p(\mathbf{x})$  的期望，并且明确地在来自它的样本上进行训练，因此模型将不会接收到对很少见或未见过的例子的准确学习信号。这是有问题的，因为我们的采样策略涉及从高维空间中的随机位置开始，这很可能是随机噪声，并根据学成的分数函数进行移动。由于我们遵循的是噪声较大或不准确的分数估计，最终生成的样本可能也是次优的，或者需要更多的迭代才能得到准确的输出。

最后，即使使用真实值得分进行采样，Langevin 动力学采样也可能无法充分混合。假设真实数据分布是两个不相交分布的混合体：

$$p(\mathbf{x}) = c_1 p_1(\mathbf{x}) + c_2 p_2(\mathbf{x}) \quad (159)$$

然后，当计算得分时，这些混合系数会丢失，因为对数运算将系数从分布中分离出来，并且梯度运算将其置零。为了可视化这一点，请注意右侧图 6 中显示的真实得分函数忽略了三个分布之间的不同权重；从图中所示的初始点进行 Langevin 动力学采样，到达每个模式的可能性大致相同，尽管右下角的模式在实际的高斯混合中具有更高的权重。

结果发现，通过向数据中添加多级高斯噪声可以同时解决这三个缺点。首先，由于高斯噪声分布的支撑集是整个空间，扰动后的数据样本将不再局限于低维流形。其次，添加较大的高斯噪声会增加每个模式在数据分布中覆盖的区域，在低密度区域添加更多的训练信号。最后，添加具有递增方差的多级高斯噪声将产生尊重真实值混合系数的中间分布。

形式上，我们可以选择一个正的噪声水平序列  $\{\sigma_t\}_{t=1}^T$  并定义一系列逐步扰动的数据分布：

$$p_{\sigma_t}(\mathbf{x}_t) = \int p(\mathbf{x}) \mathcal{N}(\mathbf{x}_t; \mathbf{x}, \sigma_t^2 \mathbf{I}) d\mathbf{x} \quad (160)$$

然后，使用分数匹配学习神经网络  $s_\theta(\mathbf{x}, t)$ ，以同时学习所有噪声水平的评分函数：

$$\arg \min_{\theta} \sum_{t=1}^T \lambda(t) \mathbb{E}_{p_{\sigma_t}(\mathbf{x}_t)} \left[ \|s_\theta(\mathbf{x}, t) - \nabla \log p_{\sigma_t}(\mathbf{x}_t)\|_2^2 \right] \quad (161)$$

其中  $\lambda(t)$  是一个正的权重函数，它根据噪声水平  $t$  进行条件约束。请注意，这个目标几乎与方程 148 中推导出的目标完全匹配，用于训练变分扩散模型。此外，作者提出使用退火朗之万动力学采样作为一种生成过程，其中样本是通过依次对每个  $t = T, T-1, \dots, 2, 1$  运行朗之万动力学产生的。初始化是从某个固定的先验（如均匀分布）中选择的，每个后续采样步骤都从前一次仿真最终的样本开始。由于噪声水平在时间步长  $t$  中逐渐降低，并且我们随时间减少步长，样本最终会收敛到一个真实模式。这直接类似于在变分扩散模型的马尔可夫 HVAE 解释中进行的采样过程，其中一个随机初始化的数据向量在逐渐降低的噪声水平上被迭代优化。

因此，我们在训练目标和采样过程方面建立了变分扩散模型与基于得分的生成模型之间的显式联系。

一个问题是如何自然地将扩散模型推广到无限数量的步骤。在马尔可夫 HVAE 观点下，这可以解释为将层次结构的数量扩展到无限  $T \rightarrow \infty$ 。从等效的基于评分的生成式模型角度来看，这更清晰；在无限数量的噪声尺度下，图像在连续时间上的扰动可以表示为随机过程，因此可以用随机微分方程 (SDE) 来描述。然后通过反转 SDE 进行采样，这自然需要在每个连续值的噪声水平上估计评分函数 [10]。SDE 的不同参数化本质上描述了不同时间的扰动方案，从而实现了对噪声过程的灵活建模 [6]。

## 指导

到目前为止，我们一直专注于建模数据分布  $p(\mathbf{x})$ 。然而，我们通常也对学习条件分布  $p(\mathbf{x}|y)$  感兴趣，这将使我们能够通过条件信息  $y$  显式控制生成的数据。这是级联扩散模型 [18] 以及最先进的图像-文本模型如 DALL-E 2 [19] 和 Imagen [7] 的基础。

将条件信息添加到每个迭代的步骤信息旁边是一种自然的方式。回想我们从公式 32 中的联合分布：

$$p(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$$

然后，为了将其转换为条件扩散模型，我们可以在每个转移步骤中简单地添加任意的条件信息  $y$ ，如下所示：

$$p(\mathbf{x}_{0:T} | y) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t, y) \quad (162)$$

例如， $y$  可以是图像-文本生成中的文本编码，或者是在进行超分辨率的低分辨率图像。因此，我们能够像以前一样通过预测每个期望的解释和实现的  $\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t, y) \approx \mathbf{x}_0$ 、 $\hat{\epsilon}_\theta(\mathbf{x}_t, t, y) \approx \epsilon_0$  或  $s_\theta(\mathbf{x}_t, t, y) \approx \nabla \log p(\mathbf{x}_t | y)$  来学习 VDM 的核心神经网络。

这种原始公式的缺点是，以此方式训练的条件扩散模型可能会学习忽略或弱化任何给定的条件信息。因此提出了引导机制，作为一种更明确地控制模型对条件信息赋予权重的方法，但代价是样本多样性会降低。引导的两种最流行形式分别为分类器引导 [10, 20] 和无分类器引导 [21]。

## 分类器指导

让我们从扩散模型的基于得分的公式开始，我们的目标是在任意噪声水平  $t$  处学习  $\nabla \log p(\mathbf{x}_t|y)$ ，即条件模型的得分。请记住，为了简洁起见， $\nabla$  是  $\nabla_{\mathbf{x}_t}$  的缩写。根据贝叶斯规则，我们可以推导出以下等效形式：

$$\nabla \log p(\mathbf{x}_t|y) = \nabla \log \left( \frac{p(\mathbf{x}_t)p(y|\mathbf{x}_t)}{p(y)} \right) \quad (163)$$

$$= \nabla \log p(\mathbf{x}_t) + \nabla \log p(y|\mathbf{x}_t) - \nabla \log p(y) \quad (164)$$

$$= \underbrace{\nabla \log p(\mathbf{x}_t)}_{\text{unconditional score}} + \underbrace{\nabla \log p(y|\mathbf{x}_t)}_{\text{adversarial gradient}} \quad (165)$$

其中我们利用了  $\log p(y)$  关于  $\mathbf{x}_t$  的梯度为零的事实。

我们最终推导的结果可以解释为学习一个无条件的评分函数，结合分类器  $p(y|\mathbf{x}_t)$  的对抗梯度。因此，在分类器引导 [10, 20] 中，无条件扩散模型的评分函数如前所述进行学习，同时还还有一个分类器，该分类器接收任意噪声的  $\mathbf{x}_t$  并尝试预测条件信息  $y$ 。然后，在采样过程中，用于退火朗之万动力学的整体条件评分函数被计算为无条件评分函数和噪声分类器的对抗梯度之和。

为了引入细粒度控制以鼓励或抑制模型考虑条件信息，分类器引导通过一个  $\gamma$  超参数项对噪声分类器的对抗梯度进行缩放。在分类器引导下学习到的得分函数可以总结为：

$$\nabla \log p(\mathbf{x}_t|y) = \nabla \log p(\mathbf{x}_t) + \gamma \nabla \log p(y|\mathbf{x}_t) \quad (166)$$

直观上，当  $\gamma = 0$  时，条件扩散模型会完全忽略条件信息，而当  $\gamma$  时较大时，条件扩散模型会学习生成大量依赖于条件信息的样本。这将付出样本多样性减少的代价，因为它只会生成那些即使在噪声水平下也容易从提供的条件信息中重新生成的数据。

分类器指导的一个显著缺点是其对一个单独学习的分类器的依赖。由于分类器必须处理任意噪声输入，而大多数现有的预训练分类模型并未优化为此目的，因此必须在扩散模型的同时临时学习它。

## 无分类器指导

在无分类器引导 [21] 中，作者放弃了训练一个分离的分类器模型，而是选择使用一个无条件扩散模型和一个条件扩散模型。为了推导无分类器引导下的评分函数，我们可以先重新排列方程 165 以显示：

$$\nabla \log p(y|\mathbf{x}_t) = \nabla \log p(\mathbf{x}_t|y) - \nabla \log p(\mathbf{x}_t) \quad (167)$$

然后，将其代入方程 166，我们得到：

$$\nabla \log p(\mathbf{x}_t|y) = \nabla \log p(\mathbf{x}_t) + \gamma (\nabla \log p(\mathbf{x}_t|y) - \nabla \log p(\mathbf{x}_t)) \quad (168)$$

$$= \nabla \log p(\mathbf{x}_t) + \gamma \nabla \log p(\mathbf{x}_t|y) - \gamma \nabla \log p(\mathbf{x}_t) \quad (169)$$

$$= \underbrace{\gamma \nabla \log p(\mathbf{x}_t|y)}_{\text{conditional score}} + \underbrace{(1 - \gamma) \nabla \log p(\mathbf{x}_t)}_{\text{unconditional score}} \quad (170)$$

再一次， $\gamma$  是一个控制我们的学成条件模型对条件信息关注程度的术语。当  $\gamma = 0$  时，学成条件模型完全忽略条件器并学习一个无条件扩散模型。当  $\gamma = 1$  时，模型显式地学习普通的条件分布而没有指导。当  $\gamma > 1$  时，扩散模型不仅优先考虑条件得分函数，还朝着远离无条件得分函数的方向移动。换句话说，它减少了不使用条件信息的样本生成概率，以利于明确使用条件信息的样本。这也导致了样本多样性的减少，但生成的样本更准确地匹配条件信息。

因为学习两个分离的扩散模型成本很高，我们可以将条件扩散模型和无条件扩散模型一起学习为一个单一的条件模型；通过用固定常数值（如零）替换条件信息，可以查询无条件扩散模型。这本质上是在对条件信息进行随机暂退法处理。无分类器指导之所以优雅，是因为它在仅需要训练单一扩散模型的情况下，使我们能够更好地控制条件生成过程。

## 关闭

让我们回顾一下我们探索过程中的发现。首先，我们将变分扩散模型作为马尔可夫分层变分自编码器的一个特例推导出来，在这种情况下，三个关键假设使得 ELBO 的易处理计算和可扩展最优化成为可能。然后，我们证明了优化一个 VDM 最终归结为学习一个神经网络来预测三个潜在目标之一：从任何任意噪声化的图像中预测原始源图像，从任何任意噪声化的图像中预测原始源噪声，或在任何任意噪声水平下预测噪声图像的评分函数。接着，我们深入探讨了学习评分函数的含义，并将其明确地与基于评分的生成建模视角联系起来。最后，我们介绍了如何使用扩散模型来学习条件分布。

总之，扩散模型作为生成式模型展现出了惊人的能力；事实上，它们推动了当前最先进的文本条件图像生成模型，如 Imagen 和 DALL-E 2。此外，使这些模型成为可能的数学原理非常优雅。然而，仍然存在一些需要考虑的缺点：

- 我们人类自然建模和生成数据的方式不太可能是这样；我们不会生成作为随机噪声的样本，并逐步去除噪声。
- VDM 不会产生可解释的潜在变量。与 VAE 希望通过其编码器的最优化学习到结构化的潜在空间不同，在 VDM 中，每个时间步的编码器已经作为线性高斯模型给出，无法灵活地进行最优化。因此，中间的潜在变量仅被限制为原始输入的噪声版本。
- 潜在变量被限制为与原始输入相同的维度，进一步阻碍了学习有意义的压缩潜在结构的努力。
- 采样是一个昂贵的过程，因为必须在两种公式下运行多次降噪步骤。回想一下，其中一个限制是选择了足够多的时间步  $T$  以确保最终的潜在变量是完全高斯噪声；在采样过程中，我们必须遍历所有这些时间步来生成一个样本。

最后需要指出的是，扩散模型的成功突显了分层变分自编码器作为生成式模型的威力。我们已经证明，当我们将模型推广到无限潜在层次时，即使编码器是简单的，并且潜在维度是固定的，并且假设存在马尔可夫转移，我们仍然能够学习到强大的数据模型。这表明，在一般的深度分层变分自编码器情况下，可以通过学习复杂的编码器和语义上有意义的潜在空间来实现进一步的性能提升。

**致谢：**我要感谢 Josh Dillon、Yang Song、Durk Kingma、Ben Poole、Jonathan Ho、Yiding Jiang、Ting Chen、Jeremy Cohen 和 Chen Sun 对本作品初稿的审阅以及提供许多有帮助的修改和意见。非常感谢！

## 参考文献

- [1] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. [arXiv preprint arXiv:1312.6114](#), 2013.
- [2] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. [Advances in neural information processing systems](#), 29, 2016.
- [3] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. [Advances in neural information processing systems](#), 29, 2016.
- [4] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In [International Conference on Machine Learning](#), pages 2256–2265. PMLR, 2015.
- [5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. [Advances in Neural Information Processing Systems](#), 33:6840–6851, 2020.
- [6] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. [Advances in neural information processing systems](#), 34:21696–21707, 2021.
- [7] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. [arXiv preprint arXiv:2205.11487](#), 2022.
- [8] Bradley Efron. Tweedie’s formula and selection bias. [Journal of the American Statistical Association](#), 106(496):1602–1614, 2011.
- [9] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. [Advances in Neural Information Processing Systems](#), 32, 2019.
- [10] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. [arXiv preprint arXiv:2011.13456](#), 2020.
- [11] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. [Advances in neural information processing systems](#), 33:12438–12448, 2020.
- [12] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. [Predicting structured data](#), 1(0), 2006.
- [13] Yang Song and Diederik P Kingma. How to train your energy-based models. [arXiv preprint arXiv:2101.03288](#), 2021.
- [14] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. [Journal of Machine Learning Research](#), 6(4), 2005.
- [15] Saeed Saremi, Arash Mehrjou, Bernhard Schölkopf, and Aapo Hyvärinen. Deep energy estimator networks. [arXiv preprint arXiv:1805.08306](#), 2018.
- [16] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In [Uncertainty in Artificial Intelligence](#), pages 574–584. PMLR, 2020.
- [17] Pascal Vincent. A connection between score matching and denoising autoencoders. [Neural computation](#), 23(7):1661–1674, 2011.

- [18] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23:47–1, 2022.
- [19] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- [20] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.
- [21] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.