

Bilateral Denoising Diffusion Models

Max W. Y. Lam, Jun Wang, Rongjie Huang[□], Dan Su, Dong Yu

摘要

去噪扩散概率模型 (ddpm) 是一种具有竞争力的生成模型,但对有效采样提出了挑战。在本文中,提出了一种新的双边去噪扩散模型 (BDDMs),其生成高质量样本的步骤显著减少。从双边建模目标出发, bddm 分别用分数网络和调度网络参数化正向和反向过程。我们表明,可以推导出一个比标准证据下界更紧的新下界,作为训练两个网络的替代目标。特别是, bddm 是高效的,易于训练的,并且能够通过优化推断噪声计划进一步改进任何预训练的 DDPM。我们的实验证明, bddm 只需要 3 个采样步骤就可以生成高保真的样本,与 ddpm 使用 1000 个采样步骤 (仅 16 个采样步骤) 生成的样本质量相当,甚至更高 (加速 62 倍)。

关键词: ddpm; 加速采样

1 引言

深度生成模型在图像生成、语音合成、自然语言生成和用于下游任务的无监督表示学习方面取得了巨大的进步。目前成功的方法主要分为两个分支:基于生成对抗网络 (GAN) 的模型和基于似然的模型。前者使用对抗性训练程序,但训练可能不稳定,其训练目标不适合与其他 GAN 模型进行比较;后者使用对数似然或替代目标进行训练,但它们也有内在的局限性。例如,自回归模型在高维数据上的采样速度非常慢,缩放特性也很差,因为它的采样本质上是一个顺序过程。同样,基于流的模型依赖于专门的架构来构建规范化的概率模型,因此模型训练的参数效率较低,通常比其他基于可能性的生成模型更困难。另一方面,对于那些使用替代损失的方法,例如变分自编码器中的证据下限和基于能量的模型中的对比散度,它们提高了推理速度,但生成的样本质量往往低于自回归模型和 GAN 模型,并且通常只适用于低维数据。

扩散概率模型 (dpms) 是一类很有前途的基于可能性的模型,它阐明了生成与当前最先进的 (SOTA) 自回归模型和 GAN 模型相当甚至更好的高质量样本。Sohl-Dickstein 等引入了使用迭代正向过程来破坏给定分布的结构,同时学习反向过程来恢复数据中的结构的思想。在不同的建模目标下,基于分数的生成模型使用与匹配的分数的神经网络通过朗之万动力学生成样本。作为 dpm 的一项开创性工作,Ho 等提出了去噪扩散概率模型 (DDPMs),该模型利用了上述两种思想之间的联系,即扩散框架可以被视为为得分匹配提供支持的具体方法。ddpm 首次呈现出与 GAN 模型相当的高质量图像合成结果,特别是在几个高分辨率图像基准上。随后,分数匹配和去噪扩散模型被应用于语音合成,如 Wavegrad 和并行贡献 Diffwave,并证明了在匹配 SOTA 自回归方法时,生成高保真音频样本的能力优于非自回归对抗模型。

尽管结果令人信服,但扩散模型比其他生成模型 (如 GANs 和 V AEs) 慢了两到三个数量级。生成扩散模型的主要局限性是在训练过程中需要多达数千个扩散步骤来学习目标分布;因此,在采样时往往需要大量的去噪步骤。最近进行了广泛的调查,以加速抽样过程,有效地产生高质量的产出。在 WaveGrad 中,使用网格搜索算法生成高保真音频样本,共分六步。从另一个方面来看,Song 等人使

用了神经概率流 ODE 来实现快速确定的多步采样过程。一种称为去噪扩散隐式模型 (ddim) 的并行贡献考虑了非马尔可夫扩散过程，并使用噪声表的子序列来加速去噪过程。最近的一项工作探索了一种学生-教师方法，将 ddim 抽样过程提取为单步模型。

先前的扩散模型只考虑了可选的反向过程形式或使用额外的知识来进行更好的条件建模。值得注意的是，我们注意到采样步骤的减少基本上取决于噪声计划的选择，这在传统上被认为是预先确定的正向过程。因此，我们有动力直接学习网络训练框架内的噪声调度。有了这样的激励，我们提出了双边去噪扩散模型 (BDDM)，它分别用一个分数网络和一个调度网络参数化正向和反向过程。bddm 有几个很好的特性。首先，在 bddm 中，我们证明了可以推导出比传统 ELBO 更紧的对数边际似然的新下界。其次，我们还可以推导出学习评分网络和调度网络的新目标。有趣的是，在合理的条件下，我们为评分网络推导的损失类似于 ddpm 中使用的目标。第三，bddm 允许有效的训练，只增加原始 DDPM 评分网络训练时间的一小部分。特别地，BDDM 是高效的，训练简单，也可以应用于任何预训练的 DDPM 优化噪声调度。我们的实验证明，bddm 只需要 3 个采样步骤就可以生成高保真的样本，并且生成的样本质量与 SOTA ddpm 相当，甚至更高，使用 1000 个步骤，只需要 16 个采样步骤 (加速 62 倍)。

2 相关工作

2.1 概率扩散模型 (DPM)

给定来自未知数据分布 $p_{\text{data}}(\mathbf{x}_0)$ ，扩散概率模型定义了正向过程 $q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})$ ，在 T 步扩散后将任何复杂的数据分布转换为简单的、可处理的分布。为了恢复正向扩散过程，使用定义为有限时间马尔可夫链 $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ 的逆向生成过程，通过边际似然 $p_\theta(\mathbf{x}_0) = \int \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) d\mathbf{x}_{1:T}$ 对数据分布建模，其中变分参数 θ 通过最大化标准证据下限 (ELBO) 来学习^{bddm}:

$$\mathcal{F}_{\text{elbo}} := \mathbb{E}_q \left[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1) - \sum_{t=2}^T \mathbb{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) || p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) - \mathbb{KL}(q(\mathbf{x}_T|\mathbf{x}_0) || p(\mathbf{x}_T)) \right]. \quad (1)$$

2.2 概率去噪扩散模型 (DDPM)

去噪扩散概率模型 (DDPMs) 作为 dpm 的扩展，应用分数匹配技术对生成过程进行参数化。特别是，ddpm 考虑了一个用噪声表参数化高斯扩散过程 $\beta \in^T$ 且 $0 < \beta_1, \dots, \beta_T < 1$:

$$q_\beta(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^T q_{\beta_t}(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad \text{其中} \quad q_{\beta_t}(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \quad (2)$$

我们可以利用各向同性高斯分布的一个很好的性质，直接用封闭形式表示 \mathbf{x}_t :

$$q_\beta(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \alpha_t\mathbf{x}_0, (1-\alpha_t^2)\mathbf{I}), \quad \text{其中} \quad \alpha_t = \prod_{i=1}^t \sqrt{1-\beta_i} \quad (3)$$

然后，利用评分网络 $\epsilon_\theta(\mathbf{x}_t, \alpha_t)$ 定义一个更复杂的反向过程:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}\left(\frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t^2}}\epsilon_\theta(\mathbf{x}_t, \alpha_t)\right), \frac{1-\alpha_{t-1}^2}{1-\alpha_t^2}\beta_t\mathbf{I}\right), \quad (4)$$

在这种情况下，学习 θ 需要用反向传播训练分数网络。请注意，使用式 (1) 中的完整 ELBO 来训练分数网络需要分数网络的 T 次向前传递和通过 T 个扩散步骤的反向传播，这使得训练在计算上是禁止的。为了有效地训练分数网络，DDPMs 提出了一种训练启发式的样本 $t \sim \text{Uniform}(\{1, \dots, T\})$ ，并计

算以下简化训练损失:

$$\mathcal{L}_{\text{ddpm}}^{(t)}(\theta) := \left\| \epsilon_t - \epsilon_\theta \left(\alpha_t \mathbf{x}_0 + \sqrt{1 - \alpha_t^2} \epsilon_t, \alpha_t \right) \right\|_2^2, \quad (5)$$

可以看作是 $\mathbb{KL}(q_\beta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) || p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))$ 的重加权项。

3 双边去噪扩散模型 (BDDM)

3.1 本文方法概述

双边去噪扩散模型 (BDDMs) 的目的是:(1) 为在扩散步骤 t 执行的训练提供理论基础, 这与原始 DDPM 中的训练启发式一致;(2) 允许基于理论直接学习采样的噪声计划。BDDMs 的基本思想是使用正向过程直接采样 $\mathbf{x}_t \sim q_\beta(\mathbf{x}_t|\mathbf{x}_0)$, 然后从采样的 \mathbf{x}_t 开始考虑反向过程 $p_\theta(\mathbf{x}_{0:t-1}|\mathbf{x}_t)$ 。在一个积分中同时考虑正向过程和反向过程, 称之为 bddm 中的双边建模。

3.2 网络架构

正向过程主要运用了 Unet 架构, 如图 1 所示:

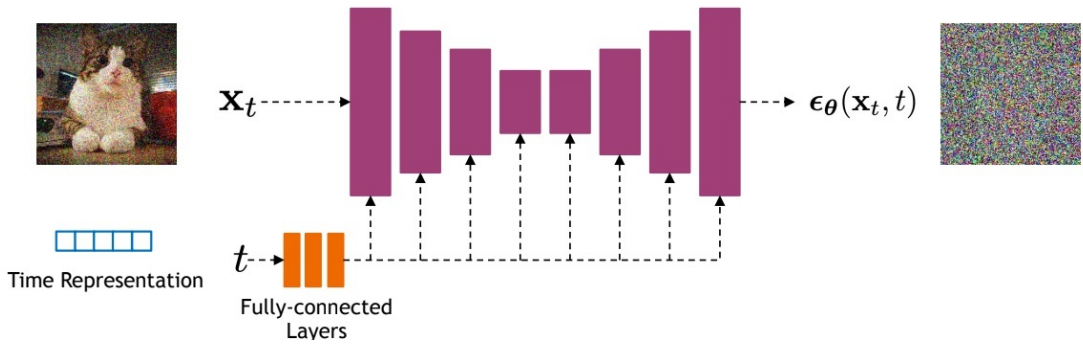


图 1: 方法示意图

反向过程主要运用了一个卷积神经网络来获得每一步的噪声大小。

3.3 损失函数定义

正向过程损失:

$$\mathcal{L}_{\text{ddpm}}^{(t)}(\theta) := \left\| \epsilon_t - \epsilon_\theta \left(\alpha_t \mathbf{x}_0 + \sqrt{1 - \alpha_t^2} \epsilon_t, \alpha_t \right) \right\|_2^2, \quad (6)$$

反向过程损失:

$$\mathcal{L}_{\text{step}}^{(t)}(\phi; \theta^*) = \frac{1}{2(1 - \hat{\beta}_t(\phi) - \alpha_t^2)} \left\| \sqrt{1 - \alpha_t^2} \epsilon_t - \frac{\hat{\beta}_t(\phi)}{\sqrt{1 - \alpha_t^2}} \epsilon_{\theta^*}(\mathbf{x}_t, \alpha_t) \right\|_2^2 + C_t(\phi), \quad (7)$$

4 复现细节

复现主要参照了原论文提供的伪代码, 实现了分数网络和噪声调度网络, 伪代码如图 2 所示:

Algorithm 1 Training θ

```

1: Given  $T, \{\beta_t\}_{t=1}^T$ 
2: Compute  $\alpha_t = \prod_{i=1}^t \sqrt{1 - \beta_i}$  from  $t = 1$  to  $T$ 
3: repeat
4:  $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0)$ 
5:  $t \sim \text{Uniform}(\{1, \dots, T\})$ 
6:  $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
7:  $\mathbf{x}_t = \hat{\alpha}_t \mathbf{x}_0 + \sqrt{1 - \hat{\alpha}_t^2} \epsilon_t$ 
8:  $\mathcal{L}_{\text{ddpm}}^{(t)} = \|\epsilon_t - \epsilon_{\theta}(\mathbf{x}_t, \alpha_t)\|_2^2$ 
9: Take a gradient descent step on  $\nabla_{\theta} \mathcal{L}_{\text{ddpm}}^{(t)}$ 
10: until converged

```

Algorithm 3 Noise Scheduling

```

1: Given  $\hat{\alpha}_N, \hat{\beta}_N$ 
2:  $\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
3: for  $n = N, \dots, 2$  do
4:  $\mathbf{x}_{n-1} \sim p_{\theta}(\mathbf{x}_{n-1} | \mathbf{x}_n; \hat{\alpha}_n, \hat{\beta}_n)$ 
5:  $\hat{\alpha}_{n-1} = \frac{\hat{\alpha}_n}{\sqrt{1 - \hat{\beta}_n}}$ 
6:  $\hat{\beta}_{n-1} = \min\{1 - \hat{\alpha}_{n-1}^2, \hat{\beta}_n\} \sigma_{\phi}(\mathbf{x}_{n-1})$ 
7: if  $\hat{\beta}_{n-1} < \beta_1$  then
8:   return  $\hat{\beta}_n, \dots, \hat{\beta}_N$ 
9: end if
10: end for
11: return  $\hat{\beta}_1, \dots, \hat{\beta}_N$ 

```

Algorithm 2 Training ϕ

```

1: Given  $\theta^*, \tau, T, \{\alpha_t, \beta_t\}_{t=1}^T$ 
2: repeat
3:  $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0)$ 
4:  $t \sim \text{Uniform}(\{2, \dots, T - \tau\})$ 
5:  $\hat{\alpha}_n = \alpha_t, \hat{\beta}_{n+1} = 1 - (\alpha_{t+\tau}/\alpha_t)^2$ 
6:  $\delta_n = \sqrt{1 - \hat{\alpha}_n^2}$ 
7:  $\epsilon_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
8:  $\mathbf{x}_n = \hat{\alpha}_n \mathbf{x}_0 + \delta_n \epsilon_n$ 
9:  $\epsilon_{\theta^*}^{(n)} = \epsilon_{\theta^*}(\mathbf{x}_n, \hat{\alpha}_n)$ 
10:  $\hat{\beta}_n = \min\{\delta_n^2, \hat{\beta}_{n+1}\} \sigma_{\phi}(\mathbf{x}_n)$ 
11:  $C_n = 4^{-1} \log(\delta_n^2/\hat{\beta}_n) + 2^{-1} D(\hat{\beta}_n/\delta_n^2 - 1)$ 
12:  $\mathcal{L}_{\text{step}}^{(n)} = \frac{1}{2(\delta_n^2 - \hat{\beta}_n)} \left\| \delta_n \epsilon_n - \frac{\hat{\beta}_n}{\delta_n} \epsilon_{\theta^*}^{(n)} \right\|_2^2 + C_n$ 
13: Take a gradient descent step on  $\nabla_{\phi} \mathcal{L}_{\text{step}}^{(n)}$ 
14: until converged

```

Algorithm 4 Sampling

```

1: Given  $\{\hat{\beta}_n\}_{n=1}^N, \mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: Compute  $\alpha_n = \prod_{i=1}^n \sqrt{1 - \hat{\beta}_i}$  from  $n = 1$  to  $N$ 
3: for  $n = N, \dots, 1$  do
4:  $\mathbf{x}_{n-1} \sim p_{\theta}(\mathbf{x}_{n-1} | \mathbf{x}_n; \alpha_n, \hat{\beta}_n)$ 
5: end for
6: return  $\mathbf{x}_0$ 

```

图 2: 伪代码

4.1 与已有开源代码对比

代码主要实现了论文中伪代码所展示的正向过程的 Unet 的网络、高斯去噪过程和反向调度噪声网络。Unet 网络架构参考了 <https://github.com/lucidrains/denoising-diffusion-pytorch> 中提供的代码。噪声调度网络由自己设计的卷积神经网络来实现，通过输入一张去噪过程中的噪声图片，输出一个处于 0-1 之间的比例来获得下一步去噪的噪声尺度。

5 实验结果分析

本次实验所用数据集为手写数据集，实验结果如图 3 所示，其中包含对高斯噪声进行去噪处理，获得生成的数字图片的过程：

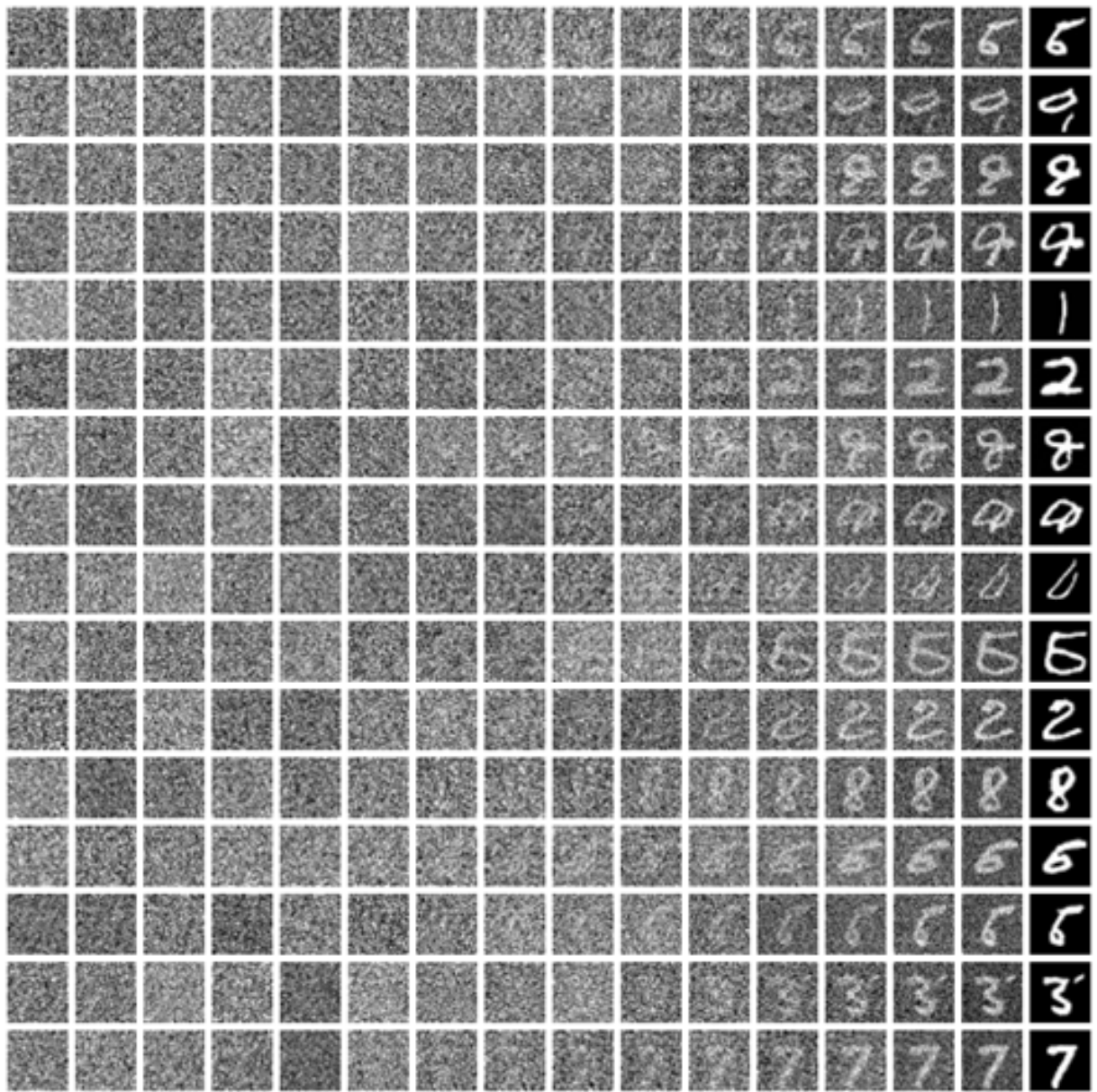


图 3: 实验结果

原始的去噪过程需要经过 500 个时间步，使用训练后的噪声调度网络进行去噪只需要经过 435 个时间步，且生成结果没有明显的退化。

6 总结与展望

本次实验通过 Unet 实现正向的分数网络实现图片生成，再通过卷积神经网络实现反向的噪声调度过程实现去噪过程的加速，最终实现了双边去噪扩散模型，并且有效地对去噪过程进行了加速。但是其中的噪声调度网络有所不足，加速效果不够显著。噪声调度网络可以加以改进，例如运用更加有效的 self——attention 架构或者加上一些提升性能的 trick，更大程度的提升扩散模型的生成图片速度。