

题目

摘要

Denoising diffusion implicit models (DDIM) 是基于 Denoising diffusion probabilistic models(DDPM) [1] 进行改进的更高效的迭代隐式概率模型。无需图像对抗训练就能够实现高质量的的图像生成。但是，DDPM 需要模拟马尔可夫链的多个步骤才能生成一张图像，计算开销很大。为了加速采样过程，研究者提出了一种更高效的迭代隐式概率模型，名为去噪扩散隐式模型 (DDIM)，其训练过程与 DDPM 相同。在 DDPM 中，生成过程被定义为特定马尔可夫扩散过程的逆过程。通过推广 DDPM，使用一类非马尔可夫扩散过程，这些过程会导致相同的训练目标，并且可以对应于确定性的生成过程，从而产生更快地生成高质量样本的隐式模型。实验证明，相比 DDPM，DDIM 能够更快地生成高质量样本，甚至可以快 10 倍到 50 倍。这使得我们可以在计算量和样本质量之间进行权衡，实现更加直接的语义上有意义的图像插值，并使用低误差的观测结果进行重建。

keyword: DDIM; DDPM; 马尔可夫扩散过程;

1 引言

深度生成模型在许多领域生成高质量的样本 [2] [3]。在图像生成方面 (Generative Adversarial Networks, GANS) [4] 生成的样本质量比同时期基于似然的方法更高，例如 VAE(变分自动编码器) [5]、自回归模型 (autoregressive models) van2016pixel 和标准化流 (normalizing flows) [6]。但是 GAN 可能需要很具体的优化和架构才能稳定训练，并且可能无法涵盖数据分布的模式。

最近在迭代生成模型 [7] 方面的工作，生成的样本质量与 GAN 相当，并且无需对抗训练；例如之前提到的 DDPM 和噪声条件分数网络 (Noise Conditional Score Networks, NCSN) [8]。为实现这一点，许多的去噪自编码模型都经过训练，对不同级别的高斯噪声加噪后的样本进行去噪；然后由马尔可夫链生成样本，该链从完全的噪声开始降噪变为图像，逐渐降噪生成为生成图像。这种生成性马尔科夫链过程一种基于 (Langevin dynamics) [8]，还有一种就是图像转为噪声的前向扩散过程转为逆向进行 [9]。

这种迭代生成模型的共同缺点是，需要多次迭代才能生成高质量的样本。对于 DDPM，生成过程（从噪声到图像、数据等），再类似的逆向过程（噪声图像、数据去噪），后者可能需要数千步才能产生单个样本，这比 GAN 相比要慢很多，因为 GAN 只需要输入通过一次网络进行生成，例如 DDPM 中，作者对 32×32 的 5 万张图片进行采样可能需要 20 个小时；作者在 Nivida 2080Ti GPU 上对大小为 256×256 的 50k 张图片进行采样操作，可能就需要将近 1000 个小时。

DDIM 可以缩小 DDPM 和 GAN 等模型效率上的差距，而且和 DDPM 训练目标函数和过程都相同。在复现的这篇论文的第三节，作者提出 DDPM 的马尔科夫链推广到非马尔可夫过程，再根据该过程设计恰当的反向生成马尔可夫链。非马尔可夫链，可以让模型不再需要根据马尔科夫链的特性，由前一个一步一步进行后面生成，DDIM 可以进行跳步采样，也就可以加速采样的效率。在文章五节中，研究者展示了 DDIM 相对于 DDPM 的几个经验优势。首先，当研究者使用 DDIM 的方法，将采样加速过程 $10\times$ 到 $100\times$ 时，DDIM 相比 DDPM 生成样本质量可能更好。

2 相关工作

前面摘要和前言部分提到了很多相关生成模型，这里再简单介绍一下：DDPM 及其扩散概率模型的基本概念，通过这些相关的知识可以帮助我们更好的理解复现论文 DDIM 的贡献与理论。

2.1 Denoising diffusion probabilistic model

在 2020 年，提出的 DDPM 系列模型，在当时在已有的 DPM 扩散概率模型也有关键的改进和优化，这篇论文”Denoising Diffusion Probabilistic Models (DDPMs)” [10]，解决了扩散模型生成的图像质量，不再是预测如下图 1 原始的 x_0 原图，而是预测每一个时刻添加的噪声，降低了模型的学习难度。

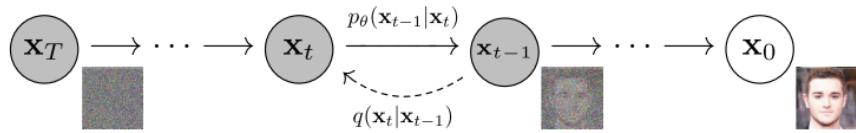


图 1. DDPM 模型流程示意图

该模型分为前向加噪和逆向去噪的两个过程，他首先需要在原始清晰的数据中不断加入噪声使其便成为高斯噪声（前向过程中）的图像，然后期望从高斯噪声的图片进行去噪还原过程称之为逆向过程，并且利用 DDPM 可以生成符合原始数据分布的新数据，以此生成图像。

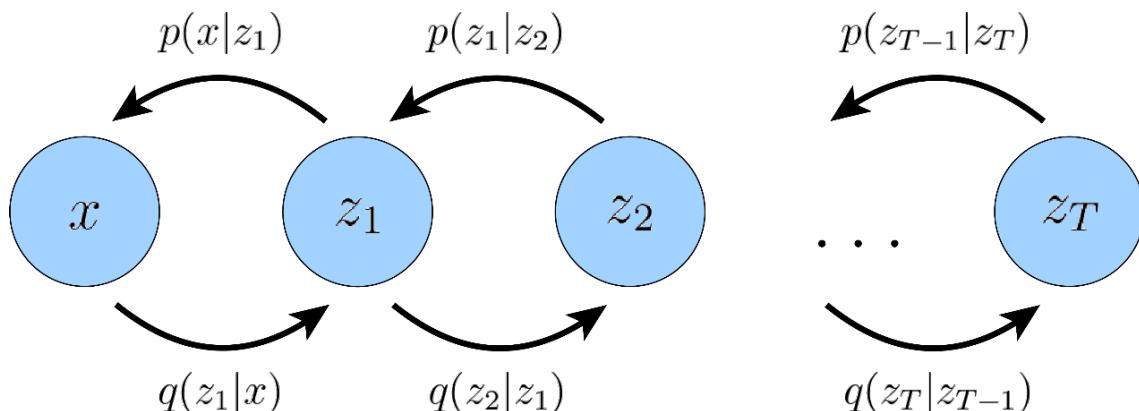


图 2. VAE 模型流程示意图

此外 VAE 模型也可以建立一种马尔可夫链式结构。整个链是双向的，但是 DDPM 不再区分 Z 和 x ，都是对 x 进行变换。

2.1.1 前向过程

给定原图片 x_0 ，前向过程是加噪的步骤，方式就是每一次在原图上每一次加上随机的噪声 $\epsilon_{t+1} \sim N(0, 1)$ （标准高斯分布的噪声）逐渐根据链式结构演变成一个标准的高斯分布 x_T 。整个网络的联合概率分布可以表示为 $p(x_{0:T})$ ，如果按照从左到右的顺序对这个联合概率进行链式拆解：

$$p(x_{0:T}) = q(x_0) \prod_{t=1}^T q(x_t | x_{t-1}) \quad (2.1)$$

但是真实图像的真实概率分布为 $P(x_0)$ 的概率密度函数是未知的，但我们能得到一批真实的图像样本，也就是有 x_0 的观测样本，此时 x_0 是观测值， $x_{1:T}$ 是未知的隐变量，这是整个马尔可夫网络的联合概率变成一个条件概率 $q(x_{1:T}|x_0)$ ，根据链式法则，可以展开为：

$$q(x_{1:T}|x_0) = q(x_1|x_0) \prod_{t=2}^T q(x_t|x_{t-1}) = \prod_{t=1}^T q(x_t|x_{t-1}) \quad (2.2)$$

根据论文假设，每一个 x_t 都是一个高斯变量，前向过程的每一个步骤的编码器 $q(x_t|x_{t-1})$ 固定为一个线性高斯变换。我们可以简单总结得到 $q(x_{1:T}|x_0)$ ，就是一个以 $\sqrt{\alpha_t} x_{t-1}$ 为均值，以 $(1-\alpha_t)I$ 为方差的高斯分布。按照线性高斯的特性，如公式 2.3 (??) 所示，它可以看作 $\sqrt{\alpha_t} x_{t-1}$ 随机噪声。这就相当于每一个步骤都在前一个步骤的基础上加上一个随机高斯噪声数据，随着 t 的增加， x_t 逐步变成一个高斯噪声数据，正如图 1 所示，这个过程好比一滴墨水滴入水中，伴随着时间的推移，墨水逐步扩散到各处，这杯水就变成了浑浊的一杯水，原来的墨水再也看不到了，这就是扩散模型。

$$q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{\alpha_t} x_{t-1}, (1 - \alpha_t)I) \quad (2.3)$$

$$\begin{aligned} x_t &= \sqrt{\alpha_t} x_{t-1} + \mathcal{N}(0, (1 - \alpha_t)I) \\ &= \sqrt{\alpha_t} x_{t-1} + \sqrt{1 - \alpha_t} \epsilon, \epsilon \sim \mathcal{N}(0, I) \end{aligned} \quad (2.4)$$

最终我们根据之前的公式 2.3 和 2.4，可以看出前向过程 $q(x_t|x_0)$ 的关键分布为：

$$q(x_t|x_0) \sim \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t)I) \quad (2.5)$$

2.1.2 后向

如果从右到左方向，正好是前向扩散过程的逆过程，它可以从一个纯粹的高斯噪声（标准的高斯噪声）随机变量 x_T 逐步去噪得到一个真实的 x_0 ，这个过程相当于生成了一个新的图片，所以称之为图像（生成）过程，而这个过程在统计概率学上，本质上是从一个联合概率分

布进行采样的过程，也可以叫做 sample 过程。对其链式分解：

$$p(x_{0:T}) = p(x_T) \prod_{t=T}^1 p(x_{t-1}|x_t) \quad (2.6)$$

其中 $p(x_T)$ 是知道的，因为之前加入的都是基于高斯分布的噪声，它是一个标准高斯分布，但是 $p_\theta(x_t|x_{t+1})$ 是难以计算的，即使贝叶斯定理得出它的表达式，但它的分母部分含有积分，难以解析，这个时候我们可以使用神经网络拟合学习条件概率分布 $p_\theta(x_t|x_{t+1})$ ，利用模拟学习好的模型取代设计的 $p_\theta(x_t|x_{t+1})$ ，这样就有了完整链式结构公式 2.6，可以通过该逆向过程，从一个随机高斯噪声 x_T 生成一张真实的图片 x_0 。我们已经知道前向过程，根据公式 2.3 我们得到 x_t 的采样过程为

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_t, \quad \bar{\alpha}_t = \prod_{i=1}^t \alpha_i, \quad \epsilon_t \sim \mathcal{N}(0, I) \quad (2.7)$$

公式 2.7 表达了 x_0 和 x_t 的关系，我们逆向过程也可以通过 x_t 得到 x_0 。

$$x_0 = \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_t}{\sqrt{\bar{\alpha}_t}}, \quad \epsilon_t \sim \mathcal{N}(0, I) \quad (2.8)$$

θ 原文定义为模型的学习参数，不带该符号的 $p(x_t|x_{t+1})$ 表示解码器的真实分布，带 θ 的 $p_\theta(x_t|x_{t+1})$ 的参数化模型学习的近似分布。

2.1.3 优化目标函数 (ELBO)

前面介绍了扩散模型的前向过程和逆向过程，主要涉及到很多随机过程，概率统计等数学理论内容。我们知道学习一个概率分布的未知参数的常用算法是极大似然估计，极大似然估计是通过极大化观测数据的对数概率（似然）实现的。上面已经介绍到了网络的概率分布为 $p(x_{0:T})$ ，其中，我们只有它的观测样本，没有 $p(x_{1:T})$ ，因此我们极大化边缘分布是观测样本的。边缘化可以得到公式 2.9

$$p(x_0) = \int p(x_{0:T}) dx_{1:T} \quad (2.9)$$

根据公式 2.9 和数学统计方法进行推导，并利用到 ELBO 进行替代。最后会得到 $\mathbb{E}_{q(x_1|x_0)} [\ln p_\theta(x_0|x_1)]$ ， $\mathbb{E}_{q(x_T|x_0)} [q(x_T|x_{T-1})p(x_T)]$ ， $\mathbb{E}_{q(x_{t-1}, x_{t+1}|x_0)} [q(x_t|x_{t-1})p_\theta(x_t|x_{t+1})]$ 三项，具体解释推导可以参考原文和网上资料，通过整合可以得到最终的目标函数，对任意 $t \in [1, T]$ ，参数化模型输入 x_t 和 t ，输出 $\hat{x}_\theta(x_t, t)$ ，模型的目标是优化预测值和真实值 x_0 的平方误差。

$$\begin{aligned} \arg \max_\theta \text{ELBO} &\Leftrightarrow \arg \max_\theta \left[\mathbb{E}_{q(x_1|x_0)} [\ln p_\theta(x_0|x_1)] - \sum_{t=2}^T \mathbb{E}_{q(x_t|x_0)} [q(x_{t-1}|x_t, x_0)p_\theta(x_{t-1}|x_t)] \right] \\ &\Leftrightarrow \arg \min_\theta \left[\mathbb{E}_{q(x_1|x_0)} [\|x_0 - \hat{x}_\theta(x_1, t=1)\|_2^2] \right] + \left[\sum_{t=2}^T \mathbb{E}_{q(x_t|x_0)} [\|(\hat{x}_\theta(x_t, t) - x_0)\|_2^2] \right] \\ &\Leftrightarrow \arg \min_\theta \sum_{t=1}^T \mathbb{E}_{q(x_t|x_0)} [\|(\hat{x}_\theta(x_t, t) - x_0)\|_2^2] \quad (2.10) \end{aligned}$$

2.2 GAN

GAN 是一种通过两个神经网络相互博弈的方式进行学习的生成模型。生成对抗网络能够在不同标注数据的情况下进行生产任务的学习。生成对抗网络主要是由一个生成器和一个判别器组成，能够在不使用不标注数据的情况下进行生成任务的学习。他会从潜在的空间进行随机取样作为输入，其输出结果通过对抗学习会模仿训练集合的真实样本。判别器的输入则为真实样本或生成器的输出，其目的就是通过输出从真实样本尽可能分辨出来，生成器和判别器不断对抗、不断模拟学习、最终目的使得判别器可以达到辨别生成器输出结果是和真的几乎相同，让他辨别不出输出结果是否真实。

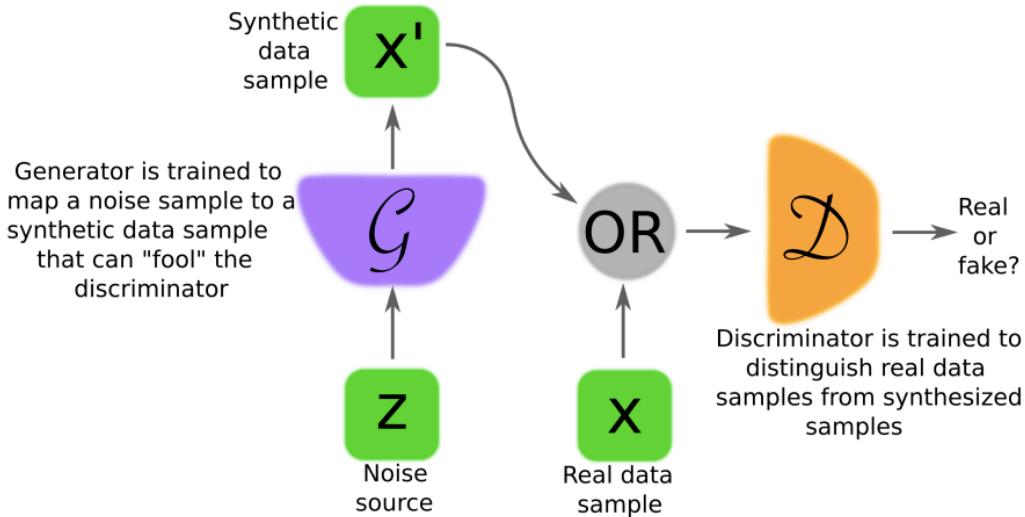


图 3. GAN 流程示意图

但是 GAN 和 DDPM 与 DDIM 这类 Diffusion models 的区别是。GAN 的训练过程是一个博弈过程，生成器和判别器相互对抗，最终生成器生成的数据越来越接近真实数据，判别器的准确率也越来越高。DDIM 是一种基于扩散过程的生成模型，它通过对数据进行扩散来生成新的数据。DDIM 的训练过程是一个最大化对数似然的过程，它通过最小化数据的负对数似然来训练模型。DDIM 的优点是可以生成高质量的图像，但是它的训练过程比较复杂，需要大量的计算资源和时间。

2.3 Score-based Generative Models

2019 年 Yang Song 提出了一种基于分数的扩散模型 [8]，实际上这种基于分数的扩散模型和基于马尔科夫的扩散模型是等价的，但是这篇论文是沿着分数（梯度） $\nabla \log p(x_t)$ 前进的，直接从分数匹配估计算法推导。在这里我们简单了解一下它的核心思想是，假设我们有一些服从某个分布的观察数据，比如大量的图像数据，但是这些观测数据的真实分布概率是未知的。但我们希望能从生成新的数据，也就是从这些观测样本中采样新的样本，比如采样生成新的图片，此时 Score-based 思想让我们虽然不知道它的形式，但是我们能得到它的分数（基于数据变量下的一阶偏导） $\nabla_x \log p_{\text{data}}(x)$ ，那么利用它的分数从观测数据的真实分布 $p_{\text{data}}(x)$ 进行采样，基于这种分数的方法有很多，这里就不一一举例，我们只需要选择一种合适的采样方法即可，从宏观来看，Score-based 就分为两步：

- 利用分数匹配方法估计出数据分布 $p_{\text{data}}(x)$ 的近似分数 $s_\theta(x) \approx \nabla_x \log p_{\text{data}}(x)$
- 使用某个基于分数 $s_\theta(x)$ 的采样算法，随机采样近似数据分布的 $p_{\text{data}}(x)$ 的样本

3 DDIM 模型方法

在 DDPM 中，生成过程被定义为马尔科夫链扩散过程的反向过程，在逆向采样过程的每一步，模型预测噪声。但是在 DDIM 中，论文提出扩散过程并不是必须遵循马尔可夫链，在基于分数的扩散模型以及基于随机微分等式的理论都有相同的结论。基于此，DDIM 的作者重新定义了扩散过程和逆过程，并提出了一种可以减少采样步骤，加速采样过程的新方法，极大的提高了图像生成的速率，代价是牺牲了一定的多样性，图像质量在和 DDPM 相比会有可接受的范围内的下降。

3.1 DDIM 方法概述

回顾在 DDPM 中，它的核心思想是构建一个马尔可夫链式结构，用神经网络来评估真实的分布得到降噪过程生成新图片，基于极大似然，基于最大下界变分法的到优化的目标，再根据高斯公式计算 KL，根据扩散过程的特性，我们可以得到重参数化简化的目标，再去参，得到最终的公式，仔细分析 DDPM 我们也发现了主要依赖于边缘分布 $q(x_t|x_0)$ 那不管中间如何，我们可以不依赖于马尔可夫链，因为马尔可夫推理速度太慢，迭代无法避免，无法进行跳跃预测。

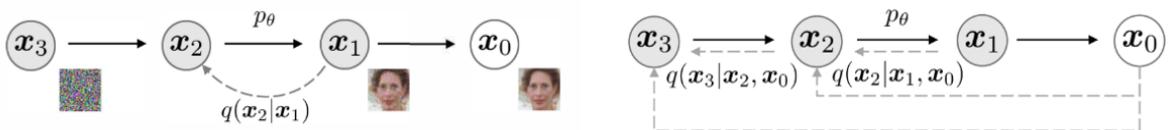


图 4. DDIM 非马尔可夫链式结构

DDIM 的公式推导大概可以归纳为：

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) \xrightarrow{\text{推导}} p(\mathbf{x}_t|\mathbf{x}_0) \xrightarrow{\text{推导}} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \xrightarrow{\text{近似}} p(\mathbf{x}_{t-1}|\mathbf{x}_t) \quad (3.1)$$

在该过程推导中，我发现了损失函数只依赖于 $p(\mathbf{x}_t|\mathbf{x}_0)$ ，采样过程只是依赖于 $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$ 。

3.2 非马尔可夫前向-逆向过程

3.2.1 回顾 DDPM 过程

在之前的第二节我已经提出了前向过程和逆向过程的概念和公式，我们关键就是得到 $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$ 的一个近似表示。根据最大似然估计理论，我们需要对数似然 $\ln p(\mathbf{x}_0)$ ，也就是之前的公式 2.9 所示，由于隐变量 $x_{1:T}$ 的存在，文章利用了 Jensen 不等式，得到了对数似然函

数的下界函数 ELBO，当满足一定条件，极大化这个下界函数和极大化对数似然是等价的。

$$\begin{aligned} [q_{x_0}] \ln p(x_0) &\geq \mathbb{E}_{q(x_{1:T}|x_0)} \left[\ln \frac{p(x_{0:T})}{q(x_{1:T}|x_0)} \right] \\ &\Rightarrow \mathbb{E}_{q(x_1|x_0)} [\ln p_\theta(x_0|x_1)] - \sum_{t=2}^T \mathbb{E}_{q(x_t|x_0)} [q(x_{t-1}|x_t, x_0) p_\theta(x_{t-1}|x_t)] \end{aligned} \quad (3.2)$$

代入各项之后，得到的最终目标函数就是一个简单的均方误差，这里记作 L_γ ，其中 γ_t 表示一些常数。

$$L_\gamma := \sum_{t=1}^T \gamma_t [q(x_t|x_0)] \|\epsilon_t - \hat{\epsilon}_\theta(x_t, t)\|_2^2, \epsilon_t \sim \mathcal{N}(0, I) \quad (3.3)$$

根据 DDPM 的最终目标函数公式 3.2, 3.3，目标函数最关键的是 KL 散度的项，是逆过程核心的转换核 $q(x_{t-1}|x_t, x_0)$ ，(逆过程：图像生成过程，图像采样过程) 和 $p_\theta(x_{t-1}|x_t)$ 模型的 KL 散度，我们最终就是要训练这个模型和转换核尽量相似，得出近似解。 $q(x_{t-1}|x_t, x_0)$ 是逆向过程的转换核， x_0 是已知的， x_t 是最直接通过前向过程 $q(x_t|x_{t-1})$ 得到的，然后可以利用线性高斯特性进行直接计算，在原始的 DDPM 模型中 $q(x_t|x_0)$ 它是联合概率的边际化得到的，如下面公式 3, 4：

$$q(x_t|x_0) = \int q(x_{1:t}|x_0) dx_{1:t-1} \quad (3.4)$$

3.2.2 非马尔可夫链式结构定义

在上一节 DDPM 中已经提到，从概率的规则来讲，不管链式结构怎么分解，最终都是需要积分消掉，但是既然是积分，可以有多种方式分解，不影响积分的结果，最终都会得到 $q(x_t|x_0)$ 。也就是说，我们可以想到一个非马尔可夫链式，作者通过重新定义 $q(x_{1:t}|x_0)$ 的分解方式，在这个给过程，如果想放弃非马尔可夫链式的假设，在不放弃 DDPM 的目标函数的情况下，只需要保证 $q(x_t|x_0)$ 和 $q(x_{t-1}|x_t, x_0)$ ，表达式和 DDPM 相同即可。现在我们重新定义分解方式，在这个过程引入一个人工定义的自由参数 σ^2 ，它代表的 $q_\sigma(x_{t-1}|x_t, x_0)$ 的方差，所以定义如下分解公式 3.5：

$$q_\sigma(x_{1:T}|x_0) = q_\sigma(x_T|x_0) \prod_{t=2}^T q_\sigma(x_{t-1}|x_t, x_0) \quad (3.5)$$

其中 $q_\sigma(x_T|x_0)$ 维持的与 DDPM 相同，类似公式 2.5

$$q_\sigma(x_T|x_0) \sim \mathcal{N}(\sqrt{\bar{\alpha}_T} x_0, (1 - \bar{\alpha}_T) I) \quad (3.6)$$

对任意的 $t > 1$ ，定义 $q_\sigma(x_{t-1}|x_t, x_0)$ 分布为：

$$q_\sigma(x_{t-1}|x_t, x_0) \sim \mathcal{N} \left(\underbrace{\sqrt{\bar{\alpha}_{t-1}} x_0 + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \frac{x_t - \sqrt{\bar{\alpha}_t} x_0}{\sqrt{1 - \bar{\alpha}_t}}}_{\text{期望}}, \underbrace{\sigma_t^2 I}_{\text{方差}} \right) \quad (3.7)$$

新的前向过程定义完成后, DDIM 还提到了, 公式 3.6 是否在 $1 \leq t \leq T$ 内任意都成立 [1], 原论文在附录 B 中也给出了证明。根据论文之前的推导, 新的公式 3.6 是没有了马尔可夫链式结构的假设, 其中 $q_\sigma(x_{t-1}|x_t, x_0)$, 仍然是逆过程的转换核, 在逆过程中, x_{t-1} 同时依赖 x_t 和 x_0 。

利用上一节公式 2.8DDPM 类似的以及 x_0, x_t, ϵ_t 三者的关系 (公式 3.8), 可以得到下面公式 3.9, 预测出 x_t

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_t, \quad \bar{\alpha}_t = \prod_{i=1}^t \alpha_i, \quad \epsilon_t \sim \mathcal{N}(0, I) \sim \mathcal{N}(\sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I) \quad (3.8)$$

$$\hat{x}_0 = f_\theta^{(t)}(x_t) = \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \hat{\epsilon}_t(x_t, t)}{\sqrt{\bar{\alpha}_t}} \quad (3.9)$$

这样我们利用上已经训练好的 DDPM 模型 $\hat{\epsilon}_t(x_t, t)$, 不需要再重新训练一个新的模型。利用公式 3.9 得到了逆向转换核 $q_\sigma(x_{t-1}|x_t, x_0)$ 近似分布:

$$\begin{aligned} p_{\theta, \sigma}(x_{t-1}|x_t) &\sim \mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}} \hat{x}_0 + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \frac{x_t - \sqrt{\bar{\alpha}_t} \hat{x}_0}{\sqrt{1 - \bar{\alpha}_t}}, \sigma_t^2 I\right) \\ &\approx q_\sigma(x_{t-1}|x_t, x_0) \end{aligned} \quad (3.10)$$

整个逆向的生成过程, 对于 x_T

$$p(x_T) = \mathcal{N}(0, I) \quad (3.11)$$

对于 $p(x_{t-1}|x_t)$

$$p(x_{t-1}|x_t) = \begin{cases} \mathcal{N}(\hat{x}_0(x_1, t=1), \sigma_1^2 I) & \text{if } t=1 \\ q_\sigma(x_{t-1}|x_t, \hat{x}_0(x_t, t)) & \text{if } 1 < t \leq T \end{cases} \quad (3.12)$$

根据上面公式 3.10, 我们能得到 x_{t-1} 具体采样公式如 3.13:

$$\begin{aligned} x_{t-1} &= \sqrt{\bar{\alpha}_{t-1}} \hat{x}_0 + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \frac{x_t - \sqrt{\bar{\alpha}_t} \hat{x}_0}{\sqrt{1 - \bar{\alpha}_t}} + \sigma_t \epsilon_t^* \\ &= \underbrace{\sqrt{\bar{\alpha}_{t-1}} \left(\frac{x_t - \sqrt{1 - \bar{\alpha}_t} \hat{\epsilon}_t(x_t, t)}{\sqrt{\bar{\alpha}_t}} \right)}_{\text{predict } x_0} + \underbrace{\sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \hat{\epsilon}_t(x_t, t)}_{\text{direction pointing to } x_t} + \underbrace{\sigma_t \epsilon_t^*}_{\text{random noise}} \end{aligned} \quad (3.13)$$

where $\epsilon_t^* \sim \mathcal{N}(0, I)$

在这个新的定义中没有了马尔可夫链式假设, 并且逆向转向核没有了仅依赖于上个状态的马尔可夫定义, 因此本文作者称之为非马尔可夫链式扩散过程。同时可以直接利用之前已经训练好的 DDPM 预测噪声模型, 不再重新训练新的模型。

3.3 加速采样

根据之前公式 3.7、3.12、3.13，多了一个方差参数 σ^2 ，它代表的是 $q_\sigma(x_{t-1}|x_t, x_0)$ 的方差。实际上，它并不是在 DDIM 新增的，这个方差在 DDPM 中也存在，只是在 DDPM 中 $q_\sigma(x_{t-1}|x_t, x_0)$ 贝叶斯推理出来的，推导的结果中这个方差有一个固定的表达式，即

$$\sigma^2 = \frac{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \quad (3.14)$$

在 DDIM 中，把 σ^2 当成一个可以人工调整的超参数，这样就可以通过调整它可以得到不一样的效果。而在 DDIM 中，可以令公式 3.14 成立，那么 DDIM 就可以退化成 DDPM，具体的推导这里就不再写出。但是将它带入到 $q_\sigma(x_{t-1}|x_t, x_0)$ 的期望中，能得到公式 3, 15

$$\mathbb{E}[q_\sigma(x_{t-1}|x_t, x_0)] = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})x_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)x_0}{(1 - \bar{\alpha}_t)} \quad (3.15)$$

可以看到，这和 DDPM 中 $q_\sigma(x_{t-1}|x_t, x_0)$ 的期望相同，也可以理解 DDIM 算是 DDPM 的扩展，DDPM 是一个 DDIM 的一个特例。可以想到 σ^2 的另一个特殊的选择是等于 0 的时候，这意味着 $q_\sigma(x_{t-1}|x_t, x_0)$ 方差为 0。最直接的公式 3.13 中的随机噪声项 $\sigma_t \epsilon_t^*$ 没了，相当于 x_{t-1} 直接等于 $q_\sigma(x_{t-1}|x_t, x_0)$ 的期望。1. 从随机采样的角度： x_{t-1} 不再是从 $q_\sigma(x_{t-1}|x_t, x_0)$ 进行随机采样，而是直接选择它的期望，又由于它是高斯分布，它的期望就是它概率密度最大的点，相当于最大概率采样。2. 从数值角度来看，没有了随机项噪声，成了确定性等式计算，不再具有随机性。

DDIM 原论文阐述方差为 0 可以加速采样过程，它从子序列的角度解释，不是很容易理解。这里我们从随机性的直观角度理解。方差为 0 时， x_t 到 x_{t-1} 的每一步，沿着期望的方向前进，自然就可以快很多，但是如果每一步都是随机采样，就会不可控，逆向到原图 x_0 步数时间就变长。

梯度和噪声的关系在 DDIM 论文中的 4.3 节可以推出梯度 $\nabla \log p(x_t)$ 和预测噪声 $\hat{\epsilon}_t(x_t, t)$ 的关系为：

$$\nabla \log p(x_t) = -\frac{1}{\sqrt{1 - \bar{\alpha}_t}} \hat{\epsilon}_t(x_t, t) \quad (3.16)$$

\hat{x}_0 可以用梯度替换：

$$\hat{x}_0 = \frac{x_t + (1 - \bar{\alpha}_t)\nabla \log p(x_t)}{\sqrt{\bar{\alpha}_t}} \quad (3.17)$$

把公式 3.17 代入到 x_{t-1} 的上面迭代公式 3.13，我们能得到 $x_{t-1} = Ax_t + B\nabla \log p(x_t) + C + \sigma_t \epsilon_t^*$ 能看出逆向过程，是沿着 x_t 的梯度在前进。意味着我们可以把各种高级的优化器算法，自适应学习率算法用在这个过程，比如原文代码用的 adam 算法。

3.4 DDIM 中的样本一致性和动态自适应调度算法

如果令 $\sigma_t \neq 0$ 意味着保留了随机项，生成的多样性更好，但是收敛速度下降，反正和 DDIM 设置 $\sigma_t = 0$ ，就会加快收敛速度，并且生成的图片将会更加一致性，如果想要兼顾速度和多样性，可能需要设计一个动态自适应的调度算法，开始一段阶段，令其为 0 加速，最后也可以让其不为 0，可以增加一点多样性，可以线性控制或者余弦控制，总之可以把自适应

调度的套路移植过来，同样系数 A 和 B 类似于学习率作用，同样可以采用自适应学习率算法。在 DDIM 论文中，虽然提出了自由方差参数 σ_t ，并做了一些实验，但论文把只有 $\sigma_t = 0$ 的情况定义为 DDIM 模型，这就是隐式（implicit）扩散模型。

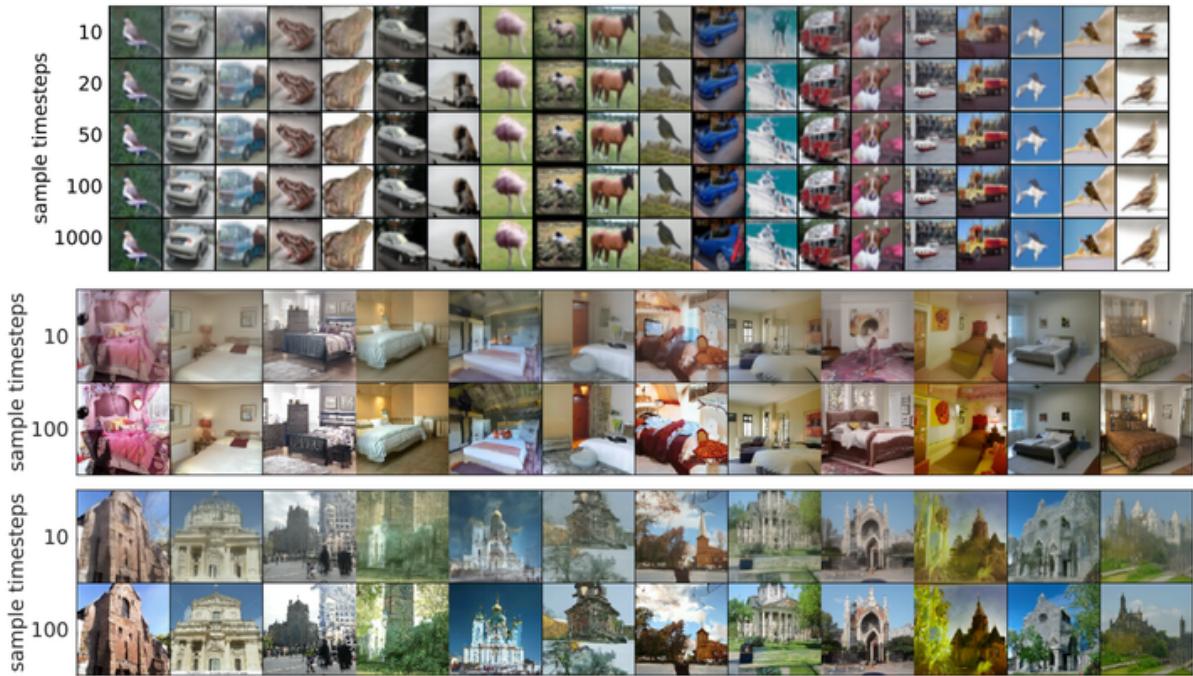


图 5. 来自不同随机 XT 和不同步数的 DDIM 的样本

对于 DDIM 生成过程是确定性的，无论生成轨迹如何，，大多数高级特征都是相似的，就像图5所示，但在 20 步生成的样本在高级特征方面已经与 1000 步生成样本相差无几，只有细节的差异。因此，看起来 x_T 本身就是图像的信息隐编码；影响样本质量的次要细节被编码在参数中，因为样本轨迹更长时，样本质量也更好，但这不会显著影响高级特征。

3.5 确定性生成过程中的插值

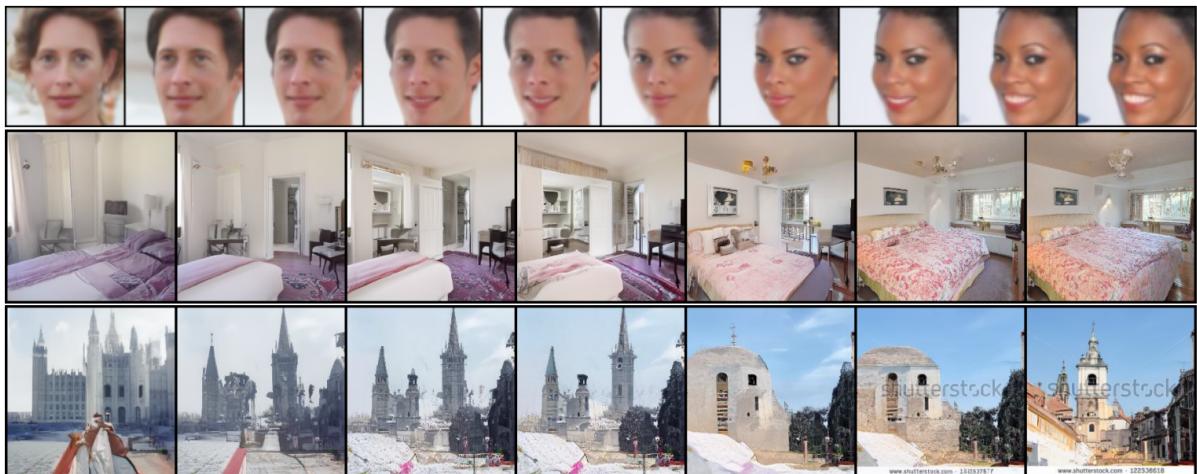


图 6. 使用步长 $\text{dim}(\cdot) = 50$ 对来自 DDIM 的样本进行插值

由于 DDIM 样本的高级特征由 x_T 编码，我们很想看看它是否会表现出类似于在其他隐式概率模型中观察到的语义插值效果，例如 GAN [4]。这不同于 [10] 中的插值过程，因为在 DDPM 中，由于随机过程，相同的 x_T 会导致不同的 x_0 （注：不注：不过如果对所有 T 噪声进行插值是可能的，如 [8] 中所做），在图 6 中，我们展示 x_T 中简单的插值可以得出两个样本之间有语义意义的插值。这允许 DDIM 直接通过隐变量在高层次上控制生成的图像，而 DDPM 则不能。

4 复现细节

4.1 与已有开源代码对比

在复现过程中引用了原文官方的代码 (<https://github.com/ermongroup/ddim>) 和 diffuser 库，进行训练和采样等实验操作。我本身的工作体现在对 DDPM 和 DDIM 公式过程的具体推导，以及用不同数据集的图像测试论文的效果。因为能力有限，并没有把能得到的创新点进行实验得出结果进行分析。

这一步引用的论文的开源代码，单独进行了生成图片和采样的操作，进行了对比，比如使用对源代码进行了拆解分析并写出小的 test，并且探索了以下五个内容：1. 验证 DDIM 论文提出的采样方式结果和 DDPM 基本相同

- 2.DDIM 的加速采样过程
- 3.DDIM 的采样确定性
4. 确定性生成过程中的插值
5. 图像重建

4.2 实验环境搭建

本文对于代码的复现使用 python 代码和 pytorch 库编写，实验环境：

CPU: AMD Ryzen 9 7940H w

Memory:32GBbytes

GPU:NVIDIA RTX 4060

大型数据训练时借用了 A40 GPU 服务器

5 实验结果分析

本文得到的实验结果，根据上一节的复现实验之后，我们展示了在考虑较少迭代次数的情况下，DDIM 在图像生成方面优于 DDPM，且与原始的 DDPM，速度得到了 10 倍乃至 100 倍的提高。另外，因为初始隐变量 X_T 固定，无论 DDIM 还可以用于对样本进行编码，这些样本从隐代码中重建，而 DDPM 则由于随机采样过程而无法做到这点 [1]。

5.1 验证 DDIM 论文提出的采样方式结果和 DDPM 基本相同

在我的 test.py 中我简单的调用 celeba 数据集中的图片。根据上文的推导，当 $t = 1$ 时我们期望 DDIM 中的前向过程能够得到和 DDPM 一样的采样结果。由于 DDIM 与 DDPM 训

练习过程几乎相同，因此我们将实验 DDPMpipeline 和 DDIMscheduler() 进行图片采样，并且选出下图进行清晰对比：

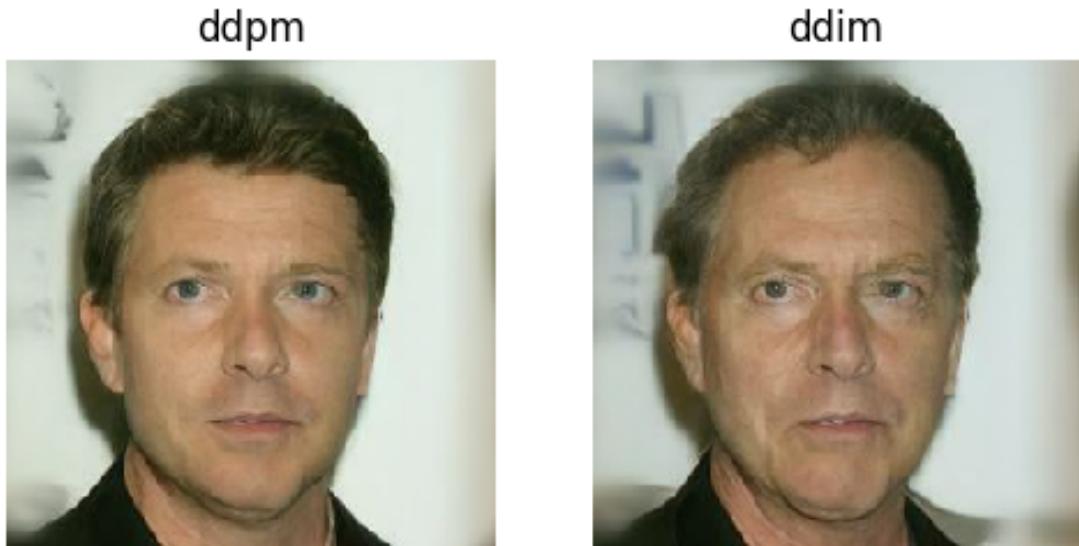


图 7. DDPM 和 DDIM 采样结果差异

通过观察，我们可以看出来两者存在显著的差别。但这不意味着两者是错误的，差异可能源于两点：1. 计算机浮点数精度问题，2.Scheduler 采样过程中存在的 clip 操作导致偏差。经过查阅资料和多次实验得到，我们只需要修改 variance 方差代码为 $variance = (1 - alpha_{prod_{tp}}rev) / (1 - alpha_{prod_t}) * (1 - self.alphas[t])$ ，并且在 scheduler 操作中进行去除 Clip 操作，将 Clip 配置设置成 False，两者采样结果就相同了。

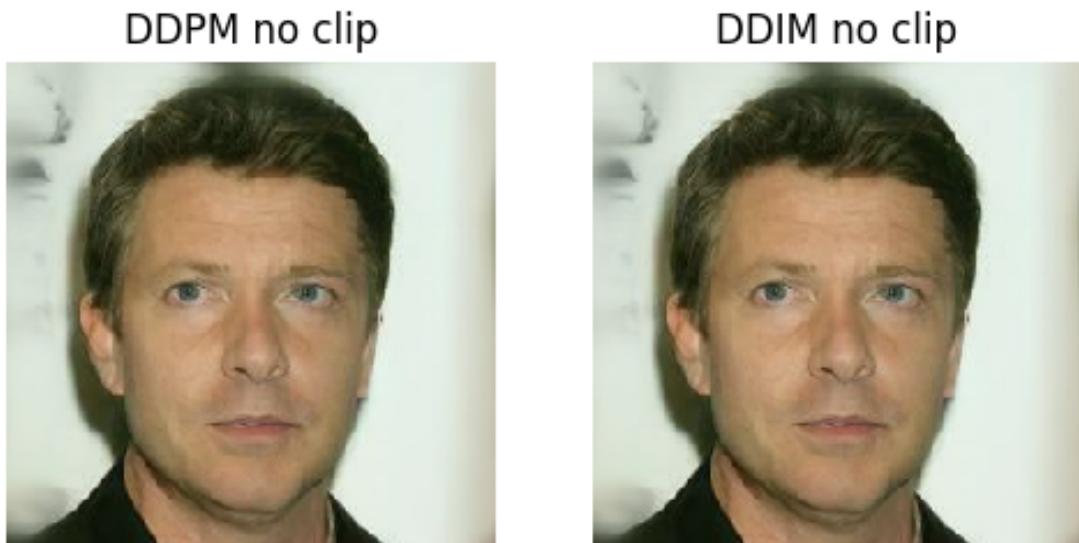


图 8. DDPM 和 DDIM 采样结果差异

5.2 样本质量和效率

原论文对不同的噪声强度和扩散步数 $\text{dim}(\cdot)$ 做了组合对比，大致上的结果是“噪声越小，加速后的生成效果越好”，如下图显示的表格所示，本人跑出来的 CelebA 和 CIFAR10 在 DDIM 模型 ($\eta = 0.0$) $\text{dim}(\tau) = 1000$ 时，FID 分数分别为 3.194 和 3.95 跟论文基本一致，其他的因为时间有限没有一一验证，但是论文的推理我已经熟悉。

Table 1: CIFAR10 and CelebA image generation measured in FID. $\eta = 1.0$ and $\hat{\sigma}$ are cases of **DDPM** (although Ho et al. (2020) only considered $T = 1000$ steps, and $S < T$ can be seen as simulating DDPMs trained with S steps), and $\eta = 0.0$ indicates **DDIM**.

S	CIFAR10 (32×32)					CelebA (64×64)				
	10	20	50	100	1000	10	20	50	100	1000
0.0	13.36	6.84	4.67	4.16	4.04	17.33	13.73	9.17	6.53	3.51
0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79	3.64
0.5	16.66	8.35	5.25	4.46	4.29	19.86	16.06	11.01	8.09	4.28
1.0	41.07	18.36	8.01	5.78	4.73	33.12	26.03	18.48	13.93	5.98
$\hat{\sigma}$	367.43	133.37	32.72	9.99	3.17	299.71	183.83	71.71	45.20	3.26

图 9. 原文的 FID 评估实验结果 (CIFAR10 和 celeba 数据集)

我用了一个比原文更简单经典的数据集 MNIST，分别用 DDPM 和 DDIM 进行采样，并且对部分中间步骤进行输出，从这个图就可以得知，DDIM 只需要采样运行五十步就能达到和 DDPM 五百步生产的样本质量相当的，如下图10所示：

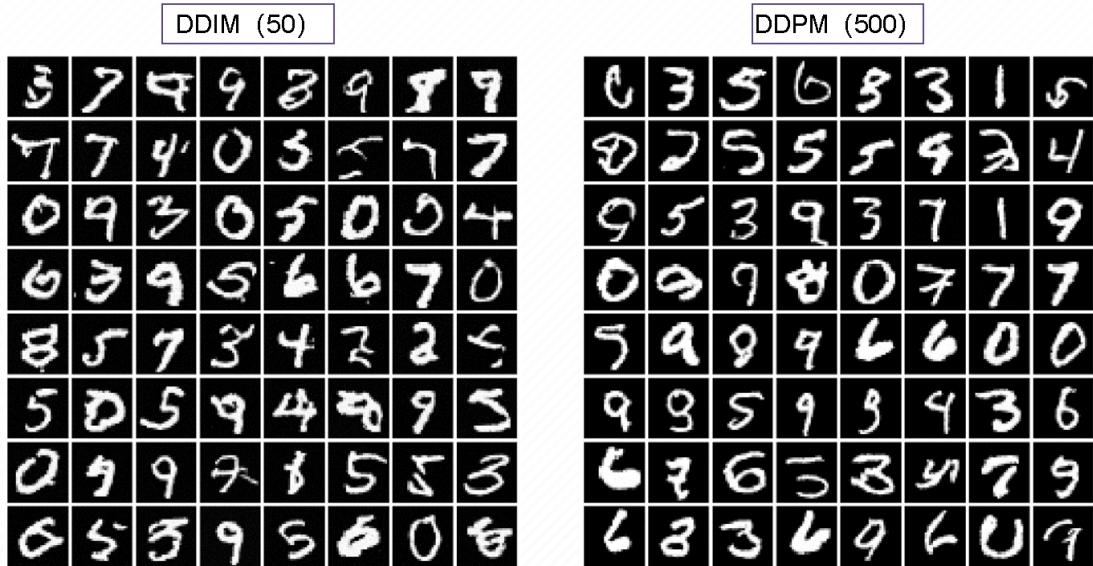


图 10. MNIST 数据集加速采样效果对照

我们又参考论文的实验步骤，选取了几个 eta 也就 η 不同的值，进行采样实验，通过下图11：



图 11. η 差异的直接对比

我们能够发现两点：1. η 越小，采样步数产生的图片质量和风格差异就越小，2. 它的减小能够很大程度的弥补采样缺陷减少带来的质量下降问题。

5.3 DDIM 的采样确定性和图像重建

对于 DDIM，生成过程是确定性的，之前已经有过介绍，论文的实验结果可以得到更一致的图像，也就是图5结果所示。简单的 test，因为在 DDIM 生成过程中， $\eta = 0$ ，因此采样过程不涉及任何随机因素，最终生成的图片仅仅由最开始输入的图片噪声 X_t 编码得到

5.4 确定性生成中的插值

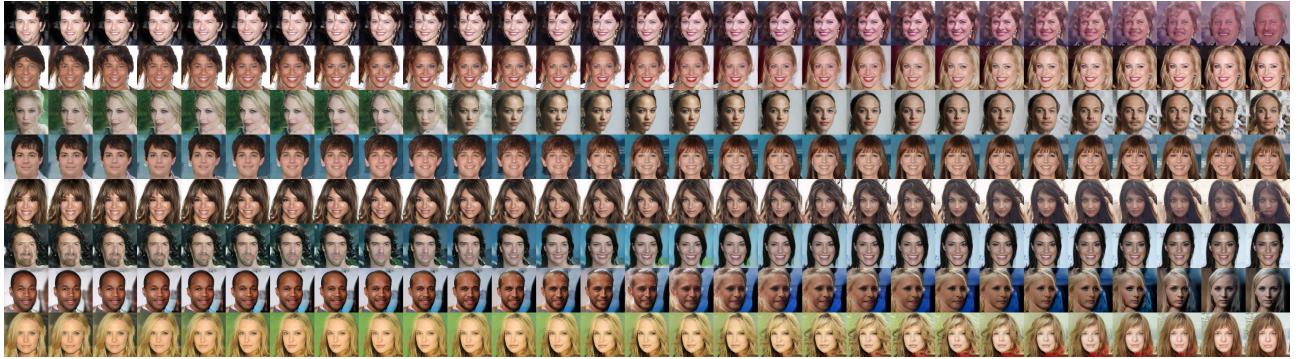


图 12. 不同 DDIM 进行插值

通过将两张不同图片的噪声，进行论文所给的 interpolation 球面线性插值。可以将两张图片进行类似风格融合的效果，如上图12所示。

也可以在 DDIM 中将原始的图片经过 T 步加噪之后变为完全噪声图像，再经过 ODE 推导的采样方式，尽可能还原图片也就是图像重建：

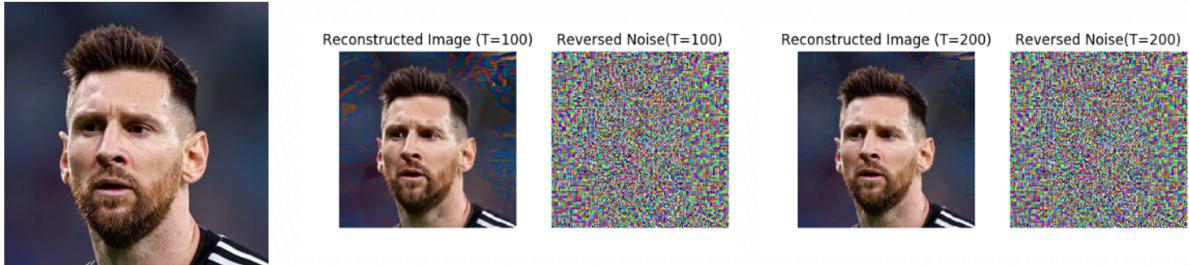


图 13. 图像重建（左图第一为原图，第二为 $T=100$ 重建，第三为 $T=100$ 重建）

论文在 CIFAR-10 测试集上使用 DDIM 重建，而我简单实验了它的重建效果，如图13，我输入一张梅西的图片，第一次我加入一百步的噪声，第二次是两百步，我们可以发现通过 ODE 的方式对图片加噪之后，变成了右边的噪声画面，重新采样之后还原了一张与原图片类似的图片，图片的还原随着时间步长的增大而增加。

6 总结与展望

该论文从纯变分的角度来介绍了 DDIM 这种使用了去噪自动编码或者得分匹配目标训练的隐式生成模型。与现有的 DDPM 和 NCSN 相比，DDIM 能够更有效果的生成高质量样本，并且能从隐空间进行有意义的插值。非马尔可夫前向过程指示了我们高斯以外的连续向前过程（但是高斯是唯一的有限方差的稳定分布，所以需要对他已有的扩散框架修改才可能推广）。

此外 DDIM 从随机性角度来说，还研究了神经 ODE 微分方程 [11] 的采样过程，去寻找减少离散化误差的角度，这个在原论文已经有了详细的思路，未来这是一个研究的方向，可

能有助于更快的步骤的同时提高样本质量。调查 DDIM 是否表现出隐式模型的其他属性也是研究重点。

DDIM 生成的样本具有一致性，无论生成轨迹是 $T=10$ 或者 100 或者 1000，大多数高级特征是相似的，如果遇到需要样本更具多样性时，模型可能效果不会那么好。这时候我们需要根据随机项 t 进行动态调度灵活设置参数，在适当的情况设置 $t = 0$ 加快收敛速度，也可以设置不为 0，增加多样性，总之通过各种方式进行动态的控制。

参考文献

- [1] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [2] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.
- [3] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.
- [4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [6] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.
- [7] Yoshua Bengio, Eric Laufer, Guillaume Alain, and Jason Yosinski. Deep generative stochastic networks trainable by backprop. In *International Conference on Machine Learning*, pages 226–234. PMLR, 2014.
- [8] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.
- [9] Jascha Sohl-Dickstein, Eric A Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv preprint arXiv:1503.03585*, March 2015.
- [10] Jonathan Ho, Aviral Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

- [11] 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.