Palette:Image-to-Image Diffusion Models

Chitwan Saharia, William Chan, Huiwen Chang, Chris A Lee, Jonathan Ho, Tim Salimans David J Fleet, Mohammad Norouzi

摘要

去噪扩散概率模型 (DDPM) 简称扩散模型，是一类深度生成模型，具有坚实的理论基础，在各种任务中都取得了令人印象深刻的结果。本文将以发表在 SIGGRAPH2022 的论文《Palette:Image-to-Image Diffusion Models》提到的 Palette 为切入点，介绍基于 DDPM 的若干图像生成任务，主要涉及到着色、图像修复、图像去裁剪、JPEG 恢复和超分辨率。我们重点介绍了 DDPM 的工作原理和方法，对其数学推导过程、算法及网络结构做了详细介绍，并且成功复现了该论文中所提到的框架，进行了实验。此外，我们对三种当下主流的深度生成模型如变分自编码器、生成对抗网络和标准化流模型的工作原理和共性局限做了概述，并对比突出了 DDPM 的优势。最后，我们在总结部分指出了所叙述论文的工作亮点、主要贡献及其局限性，并提出了个人研究心得以及 DDPM 的局限性，充分查阅资料总结了研究者们为突破 DDPM 局限性所做出的研究和未来的研究方向。

关键词：扩散模型；DDPM；图像；生成模型；网络；分辨率；采样

1 引言

扩散模型已经成为新的最先进的 (SOTA) 深度生成模型，其全称是去噪扩散概率模型 (DDPM)。扩散模型在图像合成上超越生成对抗网络 (GAN)[1][2]后，在计算机视觉、自然语言处理、时态数据建模、多模态建模、鲁棒学习、分子图建模、材料设计和逆向问题求解等众多任务中显示出巨大潜力。然而，扩散模型的原始公式存在采样过程缓慢的问题，通常需要数千个评估步骤才能绘制一个样本[2]。此外，与基于似然的模型 (如自回归模型) 相比，它难以实现有竞争力的对数似然。为了解决上述局限性，人们做出了许多努力，最近的研究或从实际的角度提高了扩散模型的性能，或从理论的角度分析了模型的可能性。

本文的复现工作是基于 Chitwan S 等人发表于 SIGGRAPH2022 上的文章《Palette: Image-to-Image Diffusion Models》。该文章提出并实现了一个图像到图像翻译任务的实现框架“Palette”，并将其运用与四种常见的图像处理任务中，分别是着色 (colorization)、图像修复 (inpainting)、图像去裁剪 (uncropping) 和 JPEG 图像恢复 (JPEG restoration)。作者用了多组不同方式的实现来对比其生成的结果，也迈出了多任务同时训练的图像生成扩散模型的第一步。作者们使用 L1 和 L2 两个不同的损失函数对模型进行训练，并对比得出了 L2 损失产生的图像有更丰富的多样性。此外，作者还对比了多任务同时训练的 Palette 和各个任务单独训练的 Palette 的实现效果，并将它们分别运用于单一数据集以及混合数据集。最后，作者还通过对比实验证实了自注意力层在网络结构中的重要性。重要的是，作者提倡基于 ImageNet 的统一评估协议，并在 Fréchet 初始距离 (FID) 、初始得分 (IS) 、分类准确性 (CA) 、感知距离 (PD) 和人类评估等多种评分标准上进行了全方位的评估。

扩散概率模型最初是作为一种潜在变量生成模型提出的，灵感来自于非平衡热力学。这类模型包括两个过程，第一个是正向过程，它通过在多个尺度上添加噪声来逐步扰动或破坏数据分布。然

后反向过程学习恢复数据结构[2]。现有工作主要从以下三个角度研究扩散模型：1) 分层潜变量模型(VAE)[3][4]，即破坏和恢复过程分别对应于深度 VAE 中的编码和解码过程；2) 基于分数的模型，用于估计受增强噪声扰动的数据分布的对数密度梯度；3) 分数模型参数化的随机微分方程 (SDEs) 的离散化，其中正向和反向过程对应于正向 SDE 和反向 SDE。

本文的其余部分组织如下：第 2 节为相关工作，将简单介绍当下主流的生成模型如变分自编码器(VAE)、GAN 和基于流的模型的实现原理以及其局限性；并介绍 Palette 实现的四个任务其他论文方法的相关工作。第 3 节介绍该论文使用的扩散模型原理及其初始论文的公式推导与算法，是论文得以实现的核心。复现工作及代码将在第 4 节介绍，说明实验所使用的环境以及操作说明。第 5 节是实验结果的展示与分析，并将该论文所使用的方法与其他图像处理方法进行比较。第 6 节是总结部分，将分别介绍这若干任务的工作亮点、主要贡献以及局限性，也包括了个人研究心得和 DDPM 的局限性及其相应的改进意见。

2 相关工作

2.1 深度生成模型

生成模型的主要目的就是要学一个给定数据集的概率密度，该数据集的数据分布设为 Pdata，其在大多数情况下是未知的比如图像、语音的一个真实分布并没有一个明确的表达式，而生成模型的目的则是产生这样一个模型 Pmodel 使其产生的数据样本与 Pdata 近似，大体可以分为两种方法：i) 显示密度估计：显示地定义 Pmodel 并用最大似然估计将其解析出来，比如参数式模型、混合模型和贝叶斯网/图模型等。ii) 隐式密度估计：学习一个模型 Pmodel 并对其进行采样而产生与真实数据样本近似或者无差、不可分辨的分布，而不是直接对数据进行建模，比如当下一些主流的深度生成模型 VAE、GAN 和基于流的模型。

2.1.1 Variational Autoencoder(VAE)

变分自编码器简称 VAE，其结构如图 1 所示，其中 $P_\theta(z)$ 为一个随机信号产生器，比如产生一个多元的高斯分布，然后经过解码器网络产生一个样本 x'，使其与真实的数据样本接近，用公式表示成如下形式：

$$p_\theta(x) = \int p_\theta(z) p_\theta(x \mid z) dz \tag{1}$$

但是其中的解码器网络 $P_\theta(x|z)$ 通常十分复杂导致 $P_\theta(z)$ 并不容易解析出 p 来，因此引入了编码器网络，这是对后验分布的一个估计，因此就可以将 $P_\theta(z)$ 写成一个下界的形式如下：

$$p_\theta(x) \geqslant E_z \left[\log p_\theta(x \mid z)\right] - D_{KL}\left(q_\varphi(z \mid x) \| p_\theta(z)\right) \tag{2}$$

因此则不需要直接计算 (1) 式这样一个边际概率，可以直接对编码器网络和解码器网络进行优化。

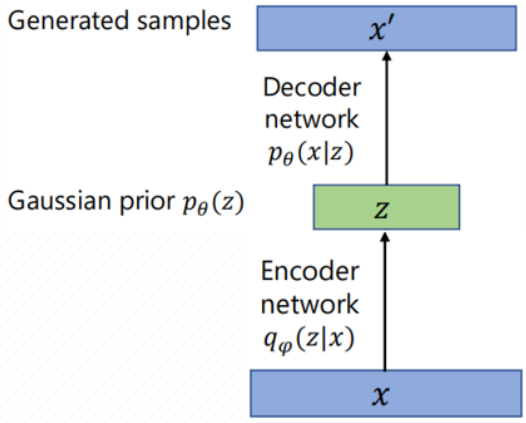

图 1: VAE 结构图

2.1.2 Generative Adversarial Networks(GAN)

生成对抗网络简称 GAN，其结构如图 2 所示，与 VAE 相似，GAN 也是通过信号产生器产生一个随机信号比如多元高斯噪声，然后进过生成器网络产生数据样本使其与真实数据分布不可区分，但其并不是从概率的角度来描述，而是从判别的角度判断产生的数据样本与真实数据是否不易区分甚至不可区分，因此 GAN 引入了一个判别器，并将生成器生成的样本和真实的数据样本同时放入判别器中，生成器的任务是让判别器无法区分两样本，而判别器的任务则是尽可能地去判别，以这样一个对抗式的任务不断优化网络。

最后可以证明当它们达到纳什均衡时，数据的分布就可以成功地学习到了网络中，因此其目标函数写成如下式 (3) 的形式：

$$\min_{\theta_g}\max_{\theta_d}\left[\mathbb{E}_{x\sim p_{\text{data}}}\log D_{\theta_d}(x)+\mathbb{E}_{z\sim p(z)}\log\left(1-D_{\theta_d}\left(G_{\theta_g}(z)\right)\right)\right] \tag{3}$$

由于这是一个对抗型的损失，因此可能会产生一些极端的训练结果比如判别器学习过快导致梯度指导无法注入到生成器中，那将很难甚至无法收敛。

2.1.3 Normalizing Flow

标准化流模型则与 VAE 和 GAN 的生成方式有所区别，其结构图如图 3 所示，将初始的简单分布如高斯，经过一系列函数之间的不断迭代耦合，最终形成一个比较复杂的分布，我们期望其与真实的数据分布一致，其中函数 f 需为可逆函数，因为向前迭代的过程要求不能出现信息消退，这也限制了该模型的表达能力；下图 4 是其可视化分布热力图；最后在数据集中学习其最大似然值：

$$\max_{\theta}\log p_X(\mathcal{D};\theta)=\sum_{\mathrm{x}\in\mathcal{D}}\log p_Z\left(\mathrm{f}_\theta^{-1}(\mathrm{x})\right)+\log\left|\det\left(\frac{\partial \mathrm{f}_\theta^{-1}(\mathrm{x})}{\partial \mathrm{x}}\right)\right| \tag{4}$$

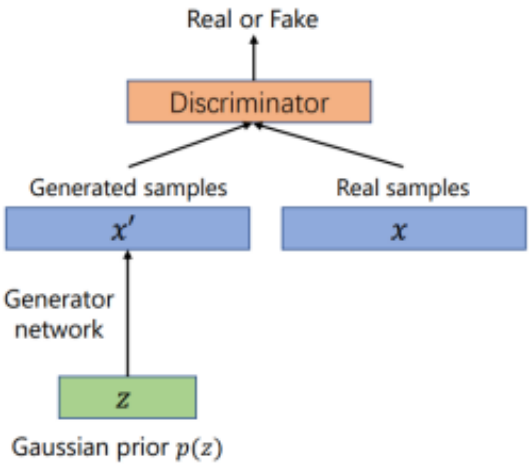

图 2: GAN 结构图

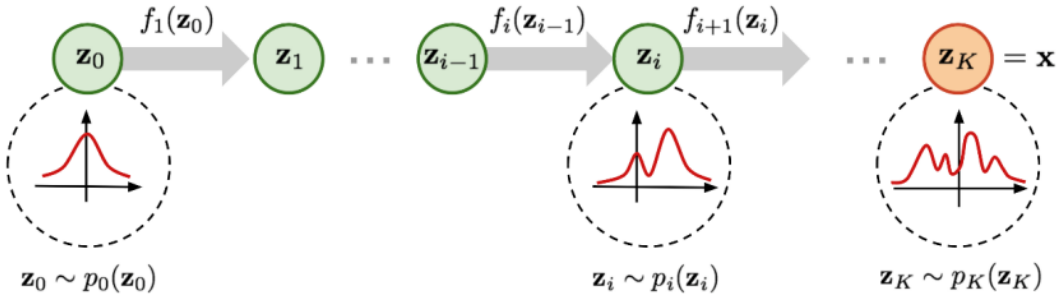

图 3: 标准化流结构图

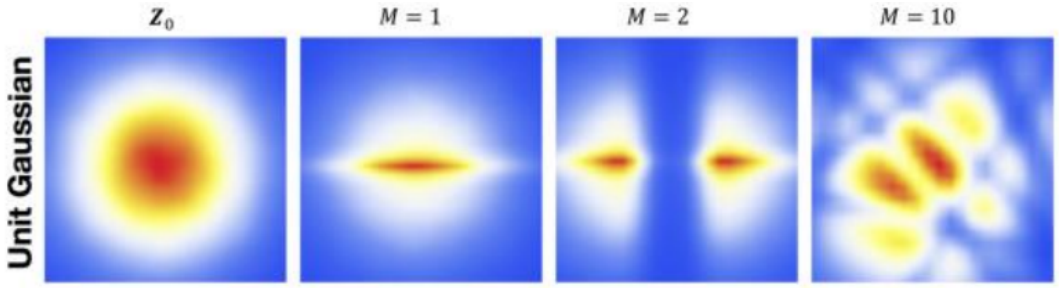

图 4: 标准化流迭代热力图

2.2 Palette 的相关工作

作者的工作受到 Isola 等人 Pix2Pix[5]的启发，它探索了大量使用 GANs 的图像到图像翻译任务。基于 GAN 的技术也被提出用于图像到图像的问题，如未配对翻译，无监督跨域生成，多域翻译，和少数镜头翻译。然而，现有的 GAN 模型在对具有一致的结构和纹理规律性的图像的整体翻译时有的并不成功。

扩散模型最近在图像生成方面取得了令人印象深刻的结果，音频合成，以及图像超分辨率[6]，以及未配对图像到图像的翻译[7]和图像编辑。我们的条件扩散模型建立在这些最新进展的基础上，显示了一套图像到图像转换任务的多功能性。大多数用于图像修复和其他线性逆问题的扩散模型都采用了无条件模型，以用于条件任务。这样做的优点是只需要训练一个模型。然而，无条件任务通常比有条件任务更难。作者将 Palette 转换为一个条件模型，如果想要一个用于多个任务的单一模型，则可以选择多任务训练。

着色 (Colorization) 是一项经过充分研究的任务，需要一定程度的场景理解。挑战包括多样化的着色，尊重语义类别，以及产生高保真的颜色。虽然之前的一些工作使用了专门的辅助分类损失，但作者发现，通用的图像到图像扩散模型在没有特定任务专门化的情况下工作得很好。

早期图像修复 (inpainting) 方法在纹理处理上取得了不错的效果，但在语义一致性方面往往做得不够好，GANs 被广泛应用，但通常需要在结构、背景、边缘、轮廓和手工设计的特征上依赖辅助目标，他们的产出缺乏多样性。

图像去裁剪 (uncropping) 被认为比图像修复更具挑战性，因为它需要在较少语义指导下生成更开放式的内容。早期的方法依赖于检索。基于 GAN 的方法现在占主导地位，但通常是特定领域的。作者表明，在大型数据集上训练的条件扩散模型可靠地解决了跨图像域的修复和去裁剪问题。

JPEG 恢复 (也叫 JPEG 伪影去除) 是去除压缩伪影的非线性逆问题。Galteri 等人将深度 CNN 架构应用于 JPEG 恢复，并成功地将 GANs 应用于伪影去除，但他们的质量因子被限制在 10 以上。而作者展示了 Palette 在去除质量因子低至 5 的压缩伪影方面的有效性。

3 本文方法

3.1 扩散模型

扩散模型是一个参数化的马尔可夫链，使用变分推理训练，在有限时间后产生与数据匹配的样本。该链的跃迁被用来反转扩散过程，这是一个逐渐在采样的相反方向向数据添加噪声的马尔可夫链，直到信号被破坏。当扩散包含少量的高斯噪声时，将采样链的跃迁设置为条件高斯就足够了，这允许一个特别简单的神经网络参数化。而作者使用的图像处理框架 Palette 正是建立在此模型基础上的。

3.1.1 正向过程

扩散模型是隐变量模型，它与其他类型的潜变量模型的区别在于近似后验 $q\left(x_{0,T} \mid x_0\right)$ 也叫正向过程或扩散过程，是固定的马尔可夫链，其根据方差表 β1,...,βT 逐步增加高斯噪声数据，数学定义如下：

$$q\left(x_{1:T} \mid x_0\right):=\prod_{t=1}^{T} q\left(x_t \mid x_{t-1}\right), \quad q\left(x_t \mid x_{t-1}\right):=\mathrm{N}\left(x_t ; \sqrt{1-\beta_t} x_{t-1}, \beta_t I\right) \tag{5}$$

再通过重参数化技巧将式 (5) 改写成下式：

$$x_t=\sqrt{1-\beta_t} x_{t-1}+\sqrt{\beta_t} \varepsilon \tag{6}$$

其中,$\beta_0=0, \beta_{\mathrm{T}}=1, \boldsymbol{\varepsilon} \sim N(0, \mathbf{I})$。

式 (6) 可以体现正向过程的马尔可夫属性，其展示了由 t-1 步数据分布推演出第 t 步数据分布的解析式。

定义：$\alpha_t=1-\beta_t$, $\bar{\alpha}_t=\prod_{i=1}^{T} \alpha_i$ 则有如下推导：

$$\mathbf{x}_t=\sqrt{\alpha_t} \mathbf{x}_{t-1}+\sqrt{1-\alpha_t} \mathbf{z}_{t-1}=\sqrt{\alpha_t \alpha_{t-1}} \mathbf{x}_{t-2}+\sqrt{1-\alpha_t \alpha_{t-1}} \bar{z}_{t-2}=\ldots=\sqrt{\bar{\alpha}_t} \mathbf{x}_0+\sqrt{1-\bar{\alpha}_t} \mathbf{z} \tag{7}$$

其中,$\mathbf{z}_{t-1}, \mathbf{z}_{t-2}, \cdots \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$，$\overline{\mathbf{Z}}_{t-2}$ 则是两个高斯噪声的叠加，由于高斯噪声的特性，其仍然是一个高斯噪声。

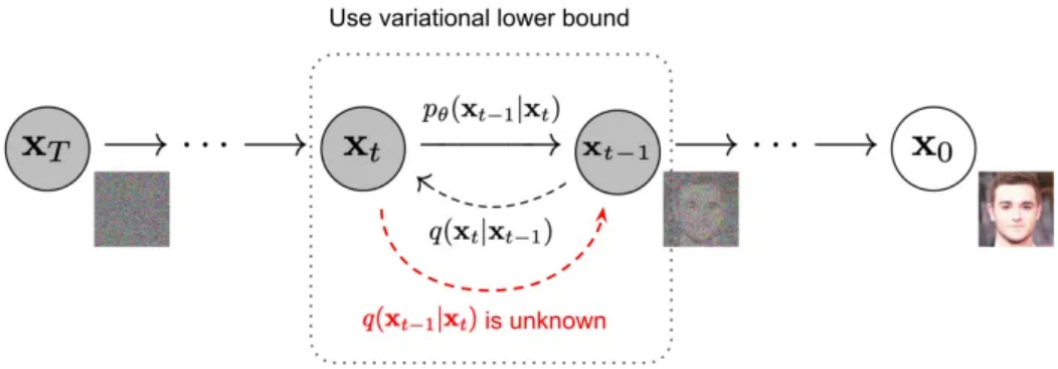

图 5: 扩散过程的有向图模型

通过以上推导，我们可以将扩散过程中第 t 步的分布用初始给定的数据分布解析给出，这为我们确定逆向过程的解析式提供了帮助。

3.1.2 逆向过程

事实上，运用数学手段可以证明当式 (5) 为高斯形式且式 (6) 中的 β_t 足够小时，我们的逆向过程 $p_\theta\left(x_{t-1} \mid x_t\right)$ 仍是一个高斯的形式，因此我们将逆向过程数学式定义如下：

$$p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right):=\mathcal{N}\left(\mathbf{x}_{t-1} ; \boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)\right) \tag{8}$$

其中，均值是我们需要训练的一个神经网络，方差可以是个固定值，也可以是个可学习的参数，在本文中均采用固定的方差值。虽然我们无法在无其他条件的情况下解析出 p_θ，但是在引入了初始数据 x_0 以及贝叶斯公式后可得下式：

$$q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)=q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0\right) \frac{q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_0\right)}{q\left(\mathbf{x}_t \mid \mathbf{x}_0\right)} \tag{9}$$

不难注意到，式 (9) 中等号右边都是我们在前向过程中可以解析给出的高斯表达式，因此将其代入式中并化简得：

$$\boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right)=\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t-\sqrt{1-\bar{\alpha}_t} \varepsilon_\theta\left(\mathbf{x}_t\right)\right)\right)=\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\varepsilon}_\theta\left(\mathbf{x}_t, t\right)\right) \tag{10}$$

将式 (10) 重参数化后，可以得到如下式子：

$$\mathbf{x}_{t-1}=\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\varepsilon}_\theta\left(\mathbf{x}_t, t\right)\right)+\sigma_t \mathbf{z} \tag{11}$$

其中，$\boldsymbol{z} \sim N(0, \mathbf{I})$。

经过上面的推导，我们在数学上得到了逆向过程的解析式，因此我们将问题重定义为如何去预测 $\varepsilon_\theta\left(\mathbf{x}_t, t\right)$，这也是我们神经网络需要学习的部分。

3.2 Palette 的算法与损失

3.2.1 算法

Palette 的算法主要分为训练部分和采样部分：

Algorithm 1 Training a denoising model f_θ

1: repeat
2: $(\boldsymbol{x}, \boldsymbol{y}_0) \sim p(\boldsymbol{x}, \boldsymbol{y})$
3: $\gamma \sim p(\gamma)$
4: $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathrm{I})$
5: Take a gradient descent step on
$\nabla_\theta \left\| f_\theta\left(\boldsymbol{x}, \sqrt{\gamma}\boldsymbol{y}_0 + \sqrt{1-\gamma}\boldsymbol{\varepsilon}, \gamma\right) - \boldsymbol{\varepsilon}\right\|_p^p$
6: until converged

Algorithm 2 Inference in T iterative refinement steps

1: $\boldsymbol{y}_T \sim \mathcal{N}(\mathbf{0}, \mathrm{I})$
2: for $t = T, \ldots, 1$ do
3: $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathrm{I})$ if $t > 1$, else $\boldsymbol{z} = \mathbf{0}$
4: $\boldsymbol{y}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta\left(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t\right)\right) + \sqrt{1-\alpha_t}\boldsymbol{z}$
5: end for
6: return $\boldsymbol{y}_0$

其中算法 1 中 x 代表 Palette 使用的是条件扩散模型，其引入了一个分类器，用于指导数据生成，使其更符合原本图像的纹理和语义；值得一提的是，将条件扩散模型用于此类图像处理工作，也是作者的主要贡献之一。第 3 步引入了一个参数序列即是 3.1.1 中提到的 α_t，第 5 步表明在前向过程的公式中利用梯度去预测所加的噪声，以便将其代入到逆向过程的式 (11) 中去。

算法 2 则是从一个高斯的开始，利用公式 (11) 将算法 1 中预测出的噪声项 f_θ 代入其中，并加上一部分的标准高斯以保证模型的多样性，最终将迭代生成的数据样本 y_0 输出。

3.2.2 损失

为了做对比试验，作者们提出了两种损失分别为 L1 和 L2 见下式 (12)，用不同的 p 值体现，并分别对它们进行训练。

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})}\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I)}\mathbb{E}_{\gamma}\left\| f_\theta(\boldsymbol{x}, \underbrace{\sqrt{\gamma}\boldsymbol{y} + \sqrt{1-\gamma}\boldsymbol{\varepsilon}}_{\widetilde{\boldsymbol{y}}}, \gamma) - \boldsymbol{\varepsilon}\right\|_p^p \tag{12}$$

通过下图 6 的 LPIPS 评估标准可以看出，L2 损失拥有更好的多样性。

	Inpainting			Colorization		
Model	**FID ↓**	**PD ↓**	**LPIPS ↑**	**FID ↓**	**PD ↓**	**LPIPS ↑**
Diffusion L_1	3.6	**41.9**	0.11	3.4	**45.8**	0.09
Diffusion L_2	3.6	43.8	**0.13**	3.4	48.0	**0.15**

图 6: L1 和 L2 损失评估参数

3.3 网络结构

Palette 使用 U-Net 结构[8][2]，并受到最近工作的影响。实际的网络架构基于 Dhariwal 和 Nichol 的 256×256 类条件 U-Net 模型[1]。作者的架构和他们的架构之间的两个主要区别是 (i) 缺乏类条件调节，以及 (ii) 继 Sahara 等之后[6]，通过拼接对源图像进行额外的条件调节。

事实上，自扩散模型诞生以来，对噪声估计的训练基本都是采用 U-Net 网络，其基础结构图如图 8 所示；大多数还会在此的基础上引入自注意力层[9]，Palette 也不例外。

自注意层已经成为最近 U-Net 扩散模型架构的重要组成部分[9][2]。虽然自注意层提供了一种直接的全局依赖形式，但它们约束了图像分辨率的泛化。在测试时推广到新的分辨率对于许多图像到图像任务来说很方便，因此以前的工作主要依赖于全卷积架构[10]。由于 Palette 作者并没有给出具体的网络架构图，所以并无法知道其更多的网络细节。

图中的每个蓝色框代表着不同的多通道特征图，箭头代表着不同的操作，其中，蓝色向右箭头是卷积的过程，红色向下的箭头为最大池化，灰色向右的箭头和虚线框代表着本结构采用了残差模块，同一层之间取一部分直接进行了复制和拼接，绿色向上的箭头表示上采样过程；与原始的 U-Net 结构 (图 7) 不同，本文采用的 U-Net 结构的输入端为 256*256 分辨率的图像，并且对输入图像分辨力的泛化能力较差，作者也对此做了一些对比试验，稍后进行介绍。

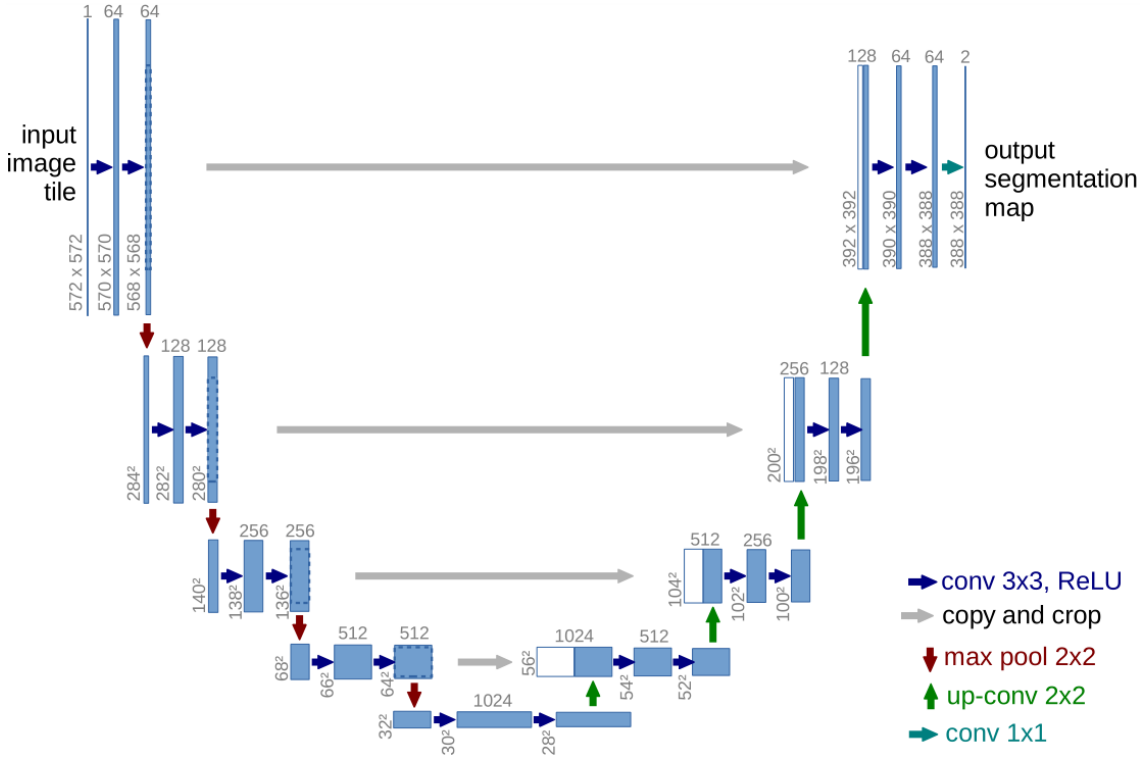

图 7: U-Net 网络结构

3.4 训练流程

有了网络结构之后，就可以介绍我们深度学习的大体训练流程了，其流程图如下图 8 所示：

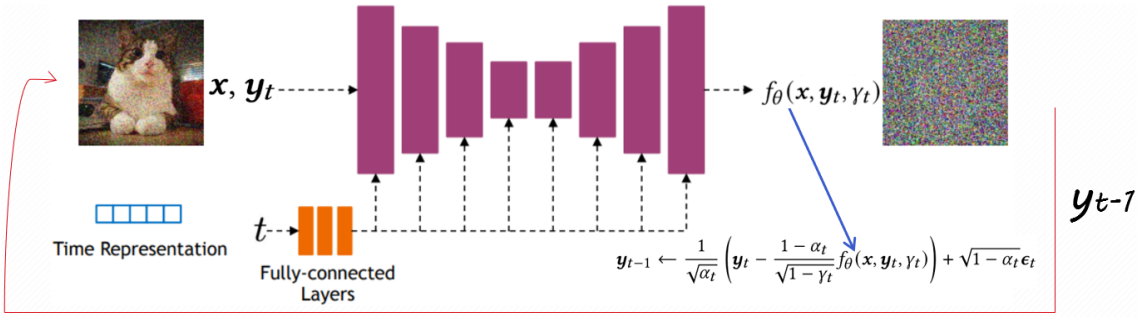

图 8: 网络训练流程图

其中左边的输入为任意时刻 t 时候的图像 (含噪声)，由于该模型使用的是条件扩散模型，所以条件 x 也被一并地输入到 U-Net 与自注意力层组成的网络中 (红色部分)，此外，此刻的步数 t 也是需要被输入的。在经过网络的预测之后，我们得到了该步所添加的噪声 f_θ，并把它代入到我们通过数学手段推导出来的公式 (11) 中，从而得出前一步的图像 y_{t-1}，之后就是一个推导迭代的过程，将 y_{t-1}、x 和 t-1 作为网络的输入端进行 y_{t-2} 的采样生成。最终经过 T 步迭代之后，生成我们的数据图像 y_0。以上就是条件扩散模型生成数据的一个大概流程。

4 复现细节

4.1 与已有开源代码对比

本文开源代码地址：https://github.com/Janspiry/Palette-Image-to-Image-Diffusion-Models. 本次复现的任务跟先前工作的对比主要有如下两点改进。

4.1.1 多任务同时训练

多任务训练是学习多个图像到图像任务的单一模型的自然方法，即盲图像增强。另一种方法是将无条件模型应用于带有归责的条件任务。例如,[11]这样做是为了图像修补；在迭代细化的每一步中，他们去噪前一步的噪声图像，然后从观察到的图像区域的像素 y 简单地替换成任何像素，然后添加噪声，并进行下一个去噪迭代。所有模型都使用相同的架构、训练数据和训练步数。重新使用的无条件模型表现不佳，部分原因是很难在像 ImageNet 这样的不同数据集上学习一个好的无条件模型，也因为在迭代细化过程中，噪声被添加到所有像素，包括观察到的像素。相比之下，Palette 直接对所有步骤进行无噪声观察。

为了更深入地探索多任务模型的潜力，通过同时训练 JPEG 恢复、图像修补和着色的 Palette 模型之间的定量比较表明，多任务 Palette 优于特定任务的 JPEG 恢复模型，但在修补和着色方面略微落后于特定任务的 Palette 模型。多任务和任务特定的 Palette 模型具有相同数量的训练步骤；希望之后可以通过更多的训练来提高多任务的表现。

4.1.2 自注意力机制的替换尝试

自注意层[9]已经成为最近 U-Net 扩散模型架构的重要组成部分。因此我们分析了这些自我注意层对修复样本质量的影响，并且得知自注意力层的引入将会限制模型输入分辨率的泛化能力。为了支持 Palette 的输入分辨率泛化，我们探索用不同的替代方案替换全局自注意层，每个替代方案都代表了大

上下文依赖性和分辨率健壮性之间的权衡。我们特别试验了以下四种配置：

(1) 全局自注意力：具有全局自注意层的基线配置，分辨率为 32×32, 16×16 和 8×8。

(2) 局部自注意力：局部自注意层在 32×32, 16×16 和 8×8 分辨率下，将特征映射划分为 4 个不重叠的查询块。

(3) 更多的 ResNet 块代替自注意力：2× 残留块在 32×32, 16×16 和 8×8 分辨率允许更深的卷积来增加接受域的大小。

(4) 无自注意力的扩张卷积：类似于 3。ResNet 区块在 32×32,16×16 和 8×8 的分辨率，随着扩张率的增加，允许接受域呈指数增长。

实验表明，全局自注意力提供了比全卷积替代方案更好的性能，重申了自注意层对此类任务的重要性。令人惊讶的是，局部自注意力的表现比全卷积的替代方法差, 采样速度比 GAN 模型慢。

4.2 实验环境搭建

- 环境搭建指令：pip install -r requirements.txt
- 服务器版本：Ubuntu 16.04.1 LTS
- GPU:Tesla P100
- PyTorch: 1.7.0
- torchvision
- numpy
- pandas
- tqdm
- tensorboardX
- scipy
- opencv-python
- clean-fid

4.3 使用说明

4.3.1 数据准备

我们大部分都是从 Kaggle 获取的，可能和官方版本略有不同，也可以从官网下载。• CelebA-HQ 调整大小 (256x256) Kaggle • Places2 官方 |Places2 Kaggle • ImageNet 官方我们使用这些数据集的默认划分进行训练和评估。我们使用的文件列表可以在 Celeba-HQ 和 Places2 中找到。当你准备好自己的数据后，你需要修改相应的配置文件以指向你的数据。

4.3.2 测试

1. 按照数据准备部分中的步骤修改配置文件以指向您的数据。
2. 按照恢复训练部分中的步骤设置模型路径。
3. 运行脚本：python run.py -p test -c config/inpainting-celebahq.json

4.3.3 评估

1. 创建两个文件夹分别保存 ground truth images 和 sample images，文件名需要一一对应。
2. 运行脚本：python eval.py -s [ground image path] -d [sample image path]

5 Palette 实验结果展示与分析

5.1 生成效果展示

Palette 的作者做了大量的实验以凸显出模型生成图像的高质量，其中包含多组与其他模型及自己不同训练方法之间的对比实验，我们只提供其中少数结果如下图 9-12 所示：

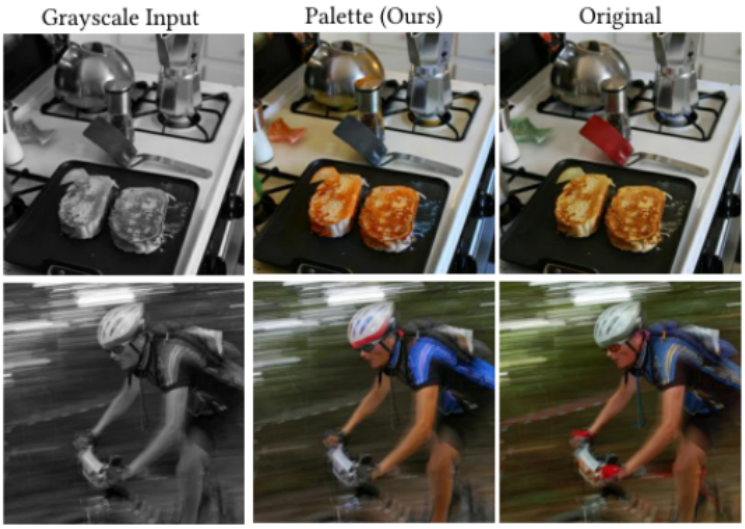

图 9: 着色 (Colorization)

图 10: 图像修复 (Inpainting)

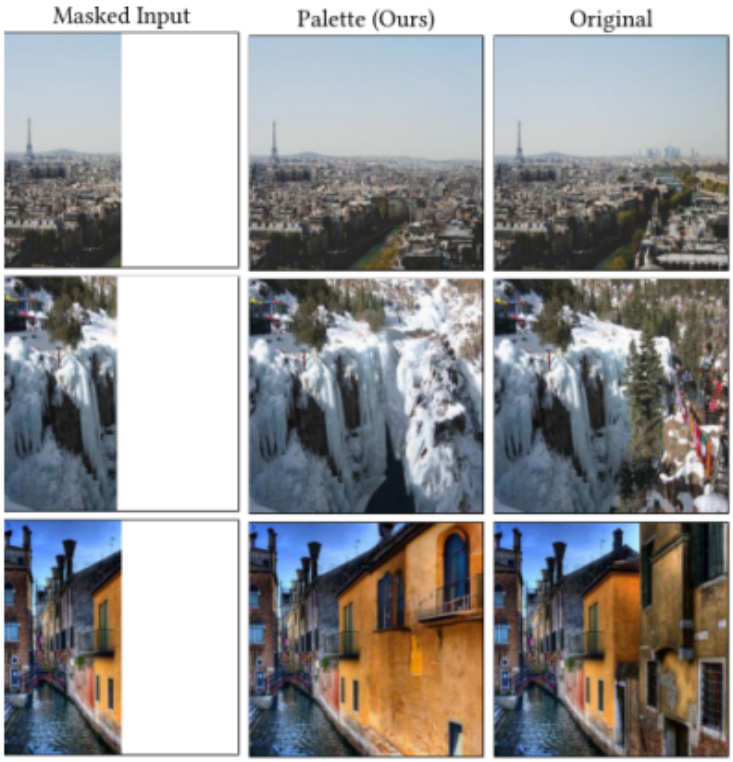

图 11: 图像去裁剪 (Uncropping)

图 12: JPEG 恢复 (JPEG Restoration)

5.2 生成多样性

图 13: Palette 生成图像的多样性展示

5.3 数据对比分析

通过 5.1 和 5.2 的图像可以直观的看出，在图像生成质量方面，Palette 模型的效果都比同任务下当前的最前沿水平还要优秀，并且在生成多样性方面，也有着十分出色的表现。接下来是生成图像的评估展示。

5.3.1 着色

Model	FID-5K ↓	IS ↑	CA ↑	PD ↓	Fool rate ↑
Prior Work					
pix2pix †	24.41	-	-	-	-
PixColor ‡	24.32	-	-	-	29.90%
Coltran ††	19.37	-	-	-	36.55%
This paper					
Regression	17.89	169.8	68.2%	60.0	39.45%
Palette	**15.78**	**200.8**	**72.5%**	**46.2**	**47.80%**
Original images	14.68	229.6	75.6%	0.0	-

图 14: 着色任务评估数据表现

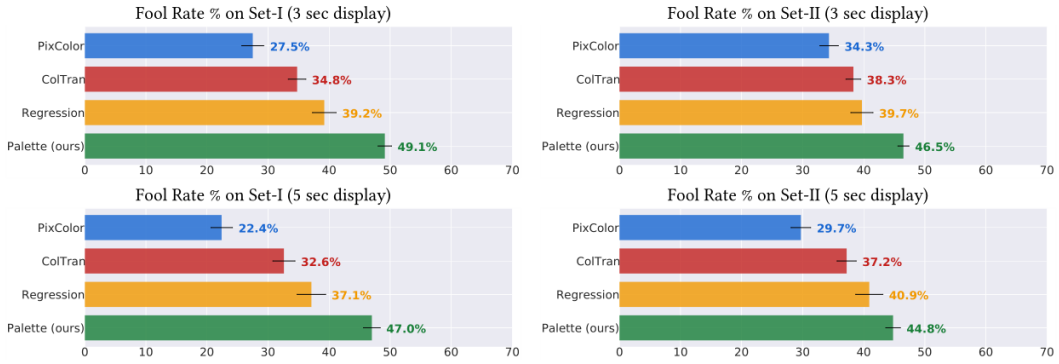

图 15: 着色任务人类愚弄率表现

5.3.2 图像修复

Mask Type	Model	ImageNet				Places2	
		FID ↓	IS ↑	CA ↑	PD ↓	FID ↓	PD ↓
10-20% Free-Form Mask	DeepFillv2 [Yu et al. 2019]	6.7	198.2	71.6%	38.6	12.2	38.1
	HiFill [Yi et al. 2020]	7.5	192.0	70.1%	46.9	13.0	55.1
	Palette (I) (Ours)	**5.1**	**221.0**	**73.8%**	15.6	11.6	22.1
	Palette (I+P) (Ours)	5.2	219.2	73.7%	**15.5**	**11.6**	**20.3**
20-30% Free-Form Mask	DeepFillv2 [Yu et al. 2019]	9.4	174.6	68.8%	64.7	13.5	63.0
	HiFill [Yi et al. 2020]	12.4	157.0	65.7%	86.2	15.7	92.8
	Co-ModGAN [Zhao et al. 2021]	-	-	-	-	12.4	51.6
	Palette (I) (Ours)	**5.2**	**208.6**	**72.6%**	**27.4**	11.8	37.7
	Palette (I+P) (Ours)	**5.2**	205.5	72.3%	27.6	**11.7**	**35.0**
30-40% Free-Form Mask	DeepFillv2 [Yu et al. 2019]	14.2	144.7	64.9%	95.5	15.8	90.1
	HiFill [Yi et al. 2020]	20.9	115.6	59.4%	131.0	20.1	132.0
	Palette (I)	**5.5**	**195.2**	**71.4%**	**39.9**	12.1	53.5
	Palette (I+P)	5.6	192.8	71.3%	40.2	**11.6**	**49.2**
128×128 Center Mask	DeepFillv2 [Yu et al. 2019]	18.0	135.3	64.3%	117.2	15.3	96.3
	HiFill [Yi et al. 2020]	20.1	126.8	62.3%	129.7	16.9	115.4
	Palette (I)	**6.4**	173.3	**69.7%**	**58.8**	12.2	62.8
	Co-ModGAN [Zhao et al. 2021]	-	-	-	-	13.7	86.2
	Palette (I+P)	6.6	**173.9**	69.3%	59.5	**11.9**	**57.3**
	Ground Truth	5.1	231.6	74.6%	0.0	11.4	0.0

图 16: 图像修复任务评估数据表现

5.3.3 图像去裁剪

Model	ImageNet				Places2	
	FID ↓	IS ↑	CA ↑	PD ↓	FID ↓	PD ↓
Boundless [Teterwak et al. 2019]	18.7	104.1	58.8%	127.9	11.8	129.3
Palette (Ours)	**5.8**	**138.1**	**63.4%**	**85.9**	**3.53**	**103.3**
Original images	2.7	250.1	76.0%	0.0	2.1	0.0

图 17: 图像去裁剪任务评估数据表现

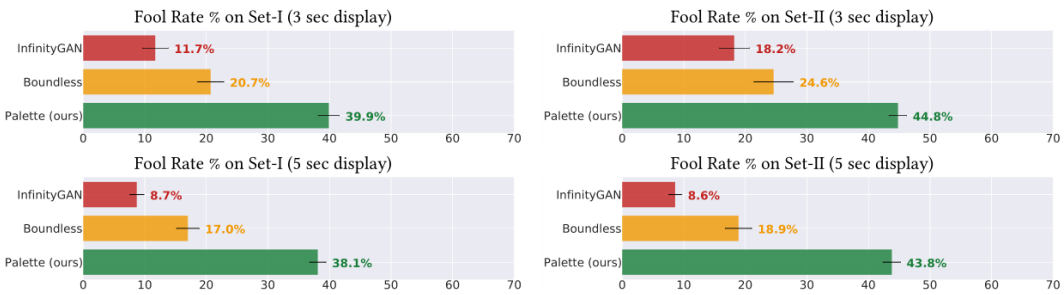

图 18: 图像去裁剪任务人类愚弄率表现

5.3.4 JPEG 恢复

QF	Model	FID-5K ↓	IS ↑	CA ↑	PD ↓
5	Regression	29.0	73.9	52.8%	155.4
	Palette (Ours)	**8.3**	**133.6**	**64.2%**	**95.5**
10	Regression	18.0	117.2	63.5%	102.2
	Palette (Ours)	**5.4**	**180.5**	**70.7%**	**58.3**
20	Regression	11.5	158.7	69.7%	65.4
	Palette (Ours)	**4.3**	**208.7**	**73.5%**	**37.1**
	Original images	2.7	250.1	76.0%	0.0

图 19: JPEG 恢复任务评估数据表现

通过上面各个表格的对比分析，我们可以看出，在当下主流的图像质量评估标准上，Palette 都有着极为出色的表现，各个指标都优于先前的相关工作；而从人类愚弄率的表格更是可以看出，其生成质量肉眼可见上的优秀。

6 总结与展望

本文通过深度生成模型为切入点，引出了基于扩散模型的 Palette 图像处理框架，对这篇论文的模型、网络结构、算法、训练以及结果做了简要的分析说明，展示了 DDPM 在图像生成、图像处理方面产生的高质量效果，在相应的领域都达到了 SOTA 的水平；因此，DDPM 为图像生成领域的推动作用是肉眼可见的。

在网络结构中不难看出，U-Net 框架看似简单，但是为 DDPM 模型的训练提供了当下来看不可替代的作用，当然，人们也在不断地对它进行优化，比如加入自注意力层，但是这会影响到输入图像对分辨率的泛化能力，因此 Palette 的作者使用了三组对比实验来尝试代替自注意力层在 U-Net 中的作用，以提高其分辨率泛化能力，遗憾的是，作者并没有找到高效的替代方法，因此为后面的研究者也提供了突破的方向。

6.1 Palette 的亮点与局限性

作者介绍了 Palette：一个用于图像到图像转换的简单通用框架。Palette 在四项图像翻译挑战 (着色、图像修复、图像去裁剪和 JPEG 恢复) 中取得了强劲的成绩，表现优于强大的 GAN 和回归基线。与许多 GAN 模型不同，Palette 产生多样化和高保真输出，并且还是在没有对特定性的任务进行训练的前提下实现的。作者还介绍了一个多任务 Palette 模型，它的性能与特定于任务的模型一样好，甚至更好。进一步探索和研究多任务扩散模型是未来工作的一个令人兴奋的途径。展示了图像到图像扩散模型的一些潜力，但我们期待看到新的应用。

6.1.1 Palette 的亮点和贡献

i) 提出了条件扩散模型用于图像处理方向的基线，为扩散模型更好地运用于图像处理提供了新的且有效的方法；

ii) 迈出了多个任务同时训练的 I2I 扩散模型运用的第一步；

iii) 为 DDPM 的网络结构 U-Net 引入的自注意力机制进行了数组对照实验，证实了自注意力层的有效性。

iv) 提供了大量的图像质量评估方法与对比实验，也进行了混合数据集和单一数据集训练的对比实验。

6.1.2 Palette 的局限性

虽然 Palette 在几个图像到图像的转换任务上取得了很好的结果，证明了新兴扩散模型的通用性和多功能性，但仍有许多重要的局限性需要解决。扩散模型在样本生成过程中通常需要大量的细化步骤 (例如，在整篇论文中对 Palette 使用了 1k 细化步骤)，导致与基于 GAN 的模型相比，推理速度明显较慢。Palette 的组规范化和自我注意层的使用阻止了它对任意输入图像分辨率的泛化，限制了它的实际可用性。与其他生成模型一样，Palette 也存在隐性偏见，在实际部署之前应该对其进行研究和缓解。

6.2 个人体会

通过上述扩散模型的若干种运用和分析，我们可以发现，虽然与先前的三种主流深度生成模型 (VAE、GAN 和 Flow-based) 相比，DDPM 的出现解决了它们的一部分共同的局限性，如优化路径不确定和随机信号不可控等因素，这是由于 DDPM 的前向学习过程是人为可控的加噪过程，每一步加噪都是根据人为设定的参数序列进行加噪的，再加上高斯噪声的特殊性，使得我们的路径变得确定和可控，而反向优化时只需要根据我们前向已知的路径进行回溯即可。

事实上，为了使我们的学习和逆向过程变得简单，我们应该把 DDPM 的迭代步骤 (即步长 T) 设置得较大，才能保证马尔可夫链上相邻的两个分布之间的差异足够小，因此，也就导致了 DDPM 当前的一个通病，就是采样迭代的推理速度太慢，这点在我们介绍的三种运用的局限性中均有所体现。此外，偏差问题的存在也是生成模型的一个局限性，该问题的解决比较困难，还需要后人的不断努力。

总而言之，新方法在不断超越旧理论的同时，也带来了一些新的挑战，这才是我们求学路上值得孜孜不倦地去追求和突破的方向。

6.3 改进意见

通过详细的考察，目前可以将 DDPM 的改进研究分为三个大类：采样效率的提高、似然最大化和数据泛化能力的改善。

i) 采样效率的提高：

近年来研究者们在提高采样速度和质量方面做出了重大努力。经过查阅整理，我们将这些采样效率增强分为两个主要部分：无学习采样和基于学习的采样，每个部分都进一步介绍了更具体的分类并给出了相应的文章。

无学习的采样：在求解扩散 SDE 时，减小离散步长可以加快采样过程。然而，这可能会导致离散化误差，并对模型性能产生负面影响。因此，已经开发了许多方法来优化离散化方案，以减少采样步骤，同时通过求解 SDE 或常微分方程 (ODE) 来保持良好的样本质量。

- SDE 求解器
- ODE 求解器

基于学习的采样：基于学习的采样也是一种有效的扩散模型采样方法，它通过自适应地使用部分步骤或在反向过程中学习抽样器来换取采样速度和样本质量。与基于求解器的方法使用前缀或确定性采样步骤不同，基于学习的采样方法通常根据学习目标选择采样步骤。

- 动态规划
- 知识蒸馏
- 早停法

ii) 似然最大化：

DDPM 最初的目的是最大化生成数据对数似然的可变长度界限 (VLB)，这相当于优化正向和反向过程之间的 KL 发散。由于对数似然与目标函数之间存在较大的变分差距，其性能可能不如其他基于似然的模型。近来有各种方法增强对数似然的最大化，并对变分下界进行了设计和分析。优化负载均衡的方法有三种：噪声调度优化、可学习反向方差、连续时间负载均衡和精确对数似然。下面将挨个列出这些工作。

- 噪声调度优化
- 确定性调度
- 可学习调度
- 可学习逆向方差
- 连续时间 VLB
- 精确对数似然

iii) 数据泛化增强：在扩散模型中，加入高斯噪声不可避免地将数据转化为连续的状态空间，这给反向去噪过程带来了困难。各种研究都专注于解决这一限制。现有的方法主要集中在将扩散模型推广到三种数据分布: 离散数据、不变结构数据和流形结构数据，当前主要工作如下。

- 离散数据
- 不变结构数据
- 流形结构数据
- 映射到流形
- 流形上扩散

参考文献

[1] DHARIWAL P, NICHOL A. Diffusion models beat gans on image synthesis[J]. Advances in Neural Information Processing Systems, 2021, 34: 8780-8794.

[2] HO J, JAIN A, ABBEEL P. Denoising diffusion probabilistic models[J]. Advances in Neural Information Processing Systems, 2020, 33: 6840-6851.

[3] LUO C. Understanding diffusion models: A unified perspective[J]. arXiv preprint arXiv:2208.11970, 2022.

[4] VAHDAT A, KAUTZ J. NVAE: A deep hierarchical variational autoencoder[J]. Advances in neural information processing systems, 2020, 33: 19667-19679.

[5] ISOLA P, ZHU J Y, ZHOU T, et al. Image-to-image translation with conditional adversarial networks [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1125-1134.

[6] SAHARIA C, HO J, CHAN W, et al. Image super-resolution via iterative refinement[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.

[7] SASAKI H, WILLCOCKS C G, BRECKON T P. Unit-ddpm: Unpaired image translation with denoising diffusion probabilistic models[J]. arXiv preprint arXiv:2104.05358, 2021.

[8] RONNEBERGER O, FISCHER P, BROX T. U-net: Convolutional networks for biomedical image segmentation[C]//Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. 2015: 234-241.

[9] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[10] GALTERI L, SEIDENARI L, BERTINI M, et al. Deep universal generative adversarial compression artifact removal[J]. IEEE Transactions on Multimedia, 2019, 21(8): 2131-2145.

[11] SONG Y, SOHL-DICKSTEIN J, KINGMA D P, et al. Score-based generative modeling through stochastic differential equations[J]. arXiv preprint arXiv:2011.13456, 2020.