《人工智能导论》探究报告
图像生成

姓名：张铭徐　　学号:2113615　　Diffusion
姓名：张惠程　　学号:2112241　　VAE
姓名：张润哲　　学号:2112362　　GAN

目录

作业正文

1 问题描述

生成模型是目前爆火的一个研究方向，据 Microsoft 对于 ChatGPT-4 的研究称 “ChatGPT-4 可以看成是通用型人工智能 (AGI) 的早期版本”；其独特的推理能力和理解语义能力迅速在全球掀起了大模型研究的一股热潮。不仅仅是 NLP 领域，CV 领域也有相应的工作，例如 meta 研究发布的《Segment Anything》这篇论文，也是图像分割领域的里程碑，可以零次迁移到新的图像分布和任务。研究者们发现，其零次迁移的性能令人印象深刻，甚至超过以前的全监督结果。本文从四篇经典的图像生成相关的文献入手，对论文的动机，研究方法，实验结果进行相应的总结分析，并试图总结出图像生成领域目前的局限性以及未来的研究方向。

2 核心内容

2.1 **变分自编码器：**Auto-Encoding Variational Bayes

变分自编码器 (VAE) 是自编码器的一种变体，为更加深刻的理解 VAE，下面我们先对 Auto-encoder 的相关内容做一些详细的说明介绍。

2.1.1 **背景知识** (**自编码器** Auto-encoder)

自编码器是一种无监督学习的神经网络模型，用于学习输入数据的低维表示。它由两部分组成：编码器（Encoder）和解码器（Decoder）。编码器将输入数据映射到潜在空间中的低维表示，而解码器则将该低维表示映射回原始数据的重构。下面是自编码器的基本原理和组成部分，原理图如图2.1所示：

- 编码器（Encoder）：编码器接收输入数据并将其映射到潜在空间中的低维表示。这个映射过程可以由多个隐藏层组成的神经网络完成。每个隐藏层将输入数据通过一系列非线性变换，最终输出一个低维的编码表示。编码器的目标是将输入数据压缩成更紧凑的表示，捕捉输入数据的重要特征。
- 解码器（Decoder）：解码器接收编码器的输出，也就是低维表示，并将其映射回原始数据的重构。解码器通常与编码器对称，并由一系列隐藏层和反向的非线性变换组成。最后一个隐藏层的输出经过一个适当的激活函数，如 Sigmoid 或 ReLU，以生成重构数据。解码器的目标是尽可能准确地还原原

始数据。

- 损失函数（Loss Function）：自编码器的训练依赖于定义良好的损失函数。常见的损失函数是重构损失，它衡量解码器的输出与原始数据之间的差异。重构损失可以使用均方误差（MSE）或二进制交叉熵（BCE）等来计算。训练过程中，自编码器通过最小化损失函数来优化编码器和解码器的参数。
- 自编码器的训练过程可以通过反向传播和梯度下降等优化算法来实现。一旦自编码器训练完成，它可以用于多种任务，如数据压缩、特征提取和图像去噪等。通过学习到数据的紧凑表示，自编码器可以发现数据的潜在结构和特征，并用于生成、重构或分类等任务。

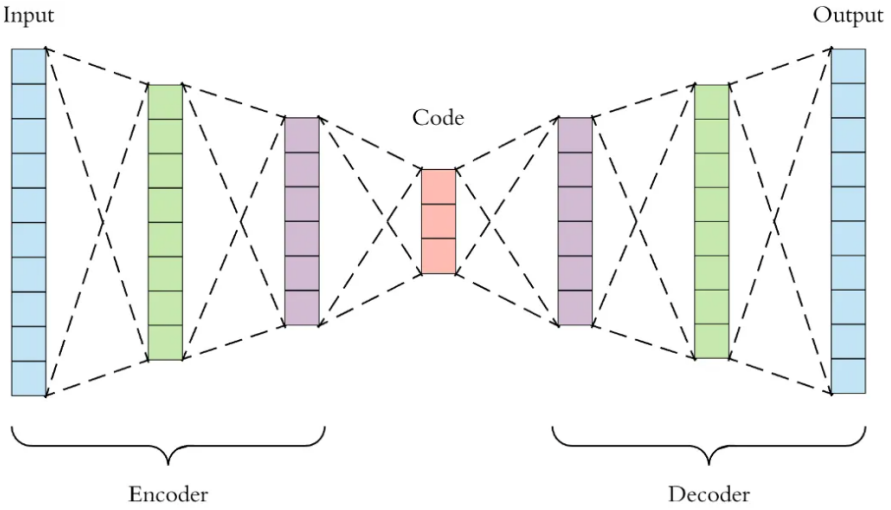


图 2.1: 自编码器示意图

除了基本的自编码器，还有一些变体，如稀疏自编码器（Sparse Autoencoder）、去噪自编码器（Denoising Autoencoder）和变分自编码器（VAE）。这些变体在结构和目标函数上有所不同，以应对不同的学习任务和数据特点。

2.1.2 论文动机

在上面介绍自编码器的原理中，我们不难发现自编码器有一些问题：由于自编码器是将原输入数据通过一系列非线性变换映射到潜在空间中，每一个输入的向量对应潜在空间的一个向量点，换言之，潜在空间中的点是通过编码器对每个输入数据进行独立映射得到的。这些点的分布取决于输入数据的特性和编码器的学习能力。如果训练数据集中存在多个簇或模式，潜在空间中的点可能会反映这些簇或模式的分布。

我们通过 decoder，可以将任意一个潜在空间上的点重新升维成原数据规模的向量，也就是说，我们可以通过对样本空间进行采样，将采样后的点通过 decoder

生成出与原本数据分布类似的样本。但是这样存在一些问题：auto-encoder 并不是学习整个潜在空间的样本分布，而是训练出了一个映射模型，所以它无法通过直接从空间中随机采样来生成新的样本。

在自编码器中，生成新样本的常见方法是从编码器输出的潜在向量周围进行采样。也就是说，在潜在空间中，通常选择一些固定的采样点或在编码器输出潜在向量的附近进行采样。通过将这些采样点传递给解码器，可以生成与这些点附近的样本相似的重构样本。这种方法的缺点是生成的样本通常局限于编码器训练过程中见过的样本附近的分布。它难以生成超出训练数据分布范围的全新样本。

变分自编码器 (VAE)[1] 的动机便是通过引入概率推断的思想，允许对潜在空间中的点进行随机采样，并通过解码器生成与这些采样点对应的新样本。这种方法使得生成的样本具有更大的多样性，并能够在更广泛的数据分布中进行采样，这使得 VAE 能够更好地描述数据的分布，并在生成任务中展现出更强的表现。

2.1.3 方法

我们考虑学习潜在空间的分布即 $p(z)$ 的分布，我们考虑生成数据的过程，可以认为是：

- 1. 对先验分布 $p(z)$ 进行采样得到一个 z_i。
- 2. 根据上面得到的 z_i，从条件分布 $P(X|z_i)$ 中采样得到一个数据点 x_i。值得注意的是，在这里，我们可以认为该条件分布是一个 decoder 的过程。

如果我们能够对这个过程加以建模，学习到这个过程，那么我们就能够解决上面自编码器存在的问题即只能对部分数据点进行采样映射，生成有用的数据。

我们不妨设 z_i $N(0,1)$ 即我们将潜在编码视为服从若干维度均独立的标准正态分布，我们输入的 z_i 经过 decoder 处理后，同样会得到服从于一个若干维度的且均独立的正态分布的数据向量。其中 decoder 中的参数，便是决定了输出数据向量的各项的均值和方差。

接下来我们考察模型本身的意义：由于模型的本质是生成模型，我们倾向于让神经网络学习尽可能的拟合样本数据点本身的分布 $p(x)$，然后我们便可以从中采样，经过映射处理后得到一些可能的数据点。

在给出对应公式前，让我们定义一些符号：

- 输入数据：x
- 潜在编码（潜在向量）：z

- 编码器（推断模型）：$q_\phi(z|x)$，这里的 ϕ 表示编码器的参数
- 解码器（生成模型）：$p_\theta(x|z)$，这里的 θ 表示解码器的参数
- 先验分布：$p(z)$，通常假设为标准正态分布 $p(z) = \mathcal{N}(0, I)$

VAE 的目标是最大化对数似然函数的下界，称为变分下界，即：

$$\log p(x) \geq \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \mathrm{KL}(q_\phi(z|x)||p(z))$$

其中，第一项 $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$ 表示重构项，衡量解码器生成 x 的能力；第二项 $\mathrm{KL}(q_\phi(z|x)||p(z))$ 表示正则化项，衡量编码器输出与先验分布之间的差异。

为了训练 VAE，我们使用重参数化技巧，使得可以对潜在变量进行采样并进行反向传播。重参数化技巧通过将潜在变量表示为编码器的输出和一个随机噪声项的乘积来实现。具体地，对于标准正态分布的先验 $p(z)$，我们可以通过以下公式进行采样：

$$z = \mu + \sigma \odot \epsilon$$

其中，μ 和 σ 是编码器输出的均值和标准差，ϵ 是从标准正态分布 $\mathcal{N}(0, I)$ 中采样得到的噪声，$\odot$ 表示逐元素乘法。

接下来，我们可以通过编码器将输入数据 x 映射到潜在变量的分布 $q_\phi(z|x)$。编码器通常采用神经网络结构，输出均值向量 μ 和标准差向量 σ。

最后，我们使用解码器将潜在变量 z 映射回重构数据 $\hat{x}$。解码器通常也采用神经网络结构，输出重构数据的分布 $p_\theta(x|z)$。通过最大化变分下界，我们可以同时学习编码器和解码器的参数 ϕ 和 θ，以实现对数据分布的建模和生成。

我们考虑更加细致的分析 VAE：

- 编码器（推断模型）：
  编码器的作用是将输入数据映射到潜在空间中的潜在变量。通常，编码器采用神经网络结构，将输入数据 x 转换为潜在变量的分布 $q_\phi(z|x)$。
  具体而言，编码器网络接收输入数据 x，通过一系列神经网络层将其转换为潜在变量的分布的参数。这些参数包括均值向量 μ 和方差向量 σ，这些向量可以表示为：

  $$\mu = \mathrm{EncoderMean}(x)$$
  $$\sigma = \mathrm{EncoderStd}(x)$$

  然后，我们可以使用这些参数来对潜在变量 z 进行采样，通常使用重参数化技巧。具体地，我们使用均值向量和标准差向量来生成一个噪声项 ϵ，并通

过如下公式计算潜在变量 z：

$$z = \mu + \sigma \odot \epsilon$$
$$\epsilon \sim \mathcal{N}(0, I)$$

这样，编码器将输入数据 x 映射到了潜在空间中的潜在变量 z。

- 解码器（生成模型）：解码器的作用是从潜在变量 z 中生成数据样本。解码器通常也采用神经网络结构，将潜在变量 z 转换为数据样本的分布 $p_\theta(x|z)$。解码器网络接收潜在变量 z，通过一系列神经网络层将其转换为数据样本的分布的参数。具体而言，解码器网络将潜在变量映射为中间表示 h'，然后通过一层全连接层生成数据样本的均值向量 μ'：

$$\mathrm{h}' = \mathrm{Decoder}(z)$$
$$\mu' = \mathrm{W}'_\mu \mathrm{h}' + \mathrm{b}'_\mu$$

其中，W'_μ 和 b'_μ 是解码器网络的参数。
最终，我们可以从生成的均值向量 μ' 中采样得到生成的数据样本 $\hat{x}$。

- 损失函数：
VAE 的损失函数包括两个部分：重构损失和正则化损失。重构损失（Reconstruction Loss）：用于衡量解码器生成的数据样本与原始数据之间的差异。通常使用平均平方误差（Mean Squared Error，MSE）作为重构损失函数。对于给定的数据样本 x 和解码器生成的数据样本 $\hat{x}$，重构损失可以表示为：

$$\mathcal{L}_{\mathrm{rec}} = ||x - \hat{x}||^2$$

正则化损失（Regularization Loss）：用于衡量编码器输出的潜在变量分布与先验分布之间的差异。通常使用 KL 散度（Kullback-Leibler Divergence，KL Divergence）作为正则化损失函数。对于给定的编码器输出的潜在变量分布 $q_\phi(z|x)$ 和先验分布 $p(z)$，正则化损失可以表示为：

$$\mathcal{L}_{\mathrm{KL}} = \mathrm{KL}(q_\phi(z|x)||p(z))$$

最终，VAE 的损失函数是重构损失和正则化损失的组合，通常是将它们加权求和：

$$\mathcal{L} = \lambda_{\mathrm{rec}} \cdot \mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{KL}} \cdot \mathcal{L}_{\mathrm{KL}}$$

其中，λ_{rec} 和 λ_{KL} 是控制两个损失项权重的超参数。

VAE 的训练过程是通过最小化损失函数，使用反向传播和优化算法（如梯度下降）来更新编码器和解码器的参数。这样，我们可以训练 VAE 以学习数据分布并生成新样本。针对整体网络的过程，我们总结出了其算法流程图，如算法1所示：

Algorithm 1 VAE 算法流程图

Input: 设置步数 k（作为超参数），用于更新编码器和解码器

for 每个训练迭代 do
  for k 步 do
    从输入数据分布 $p_{\text{data}}(x)$ 中采样大小为 m 的样本 $\{x^{(1)}, \ldots, x^{(m)}\}$
    使用编码器将样本映射到潜在空间的均值和方差参数：

$$\mu^{(i)}, \sigma^{(i)} = \text{Encoder}(x^{(i)})$$

    从参数化的潜在空间分布 $q(z|x)$ 中采样大小为 m 的潜在变量 $\{z^{(1)}, \ldots, z^{(m)}\}$
    使用解码器将潜在变量映射回重构的数据空间：

$$\hat{x}^{(i)} = \text{Decoder}(z^{(i)})$$

    计算重构损失：$\mathcal{L}_{\text{rec}}^{(i)} = \text{ReconstructionLoss}(x^{(i)}, \hat{x}^{(i)})$
    计算 KL 散度损失：$\mathcal{L}_{\text{KL}}^{(i)} = \text{KLDivergenceLoss}(\mu^{(i)}, \sigma^{(i)})$
    计算总体损失函数：$\mathcal{L}^{(i)} = \mathcal{L}_{\text{rec}}^{(i)} + \mathcal{L}_{\text{KL}}^{(i)}$
    使用反向传播和优化算法更新编码器和解码器的参数：

$$\nabla_{\theta_e, \theta_d} \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}^{(i)}$$

  end for
end for

2.1.4 **实验结果**

作者使用了 MNIST 数据集和 Frey Face 数据集对自己的方法进行了测试，结果我们总结如下：

- 基准比较：作者使用了传统的自编码器（AE）和受限玻尔兹曼机（RBM）作为基准模型进行比较。实验结果表明，变分自编码器（VAE）在生成样本和潜在空间插值等方面表现出更好的性能。
- 潜在空间的连续性：通过对潜在空间进行插值实验，作者展示了 VAE 的潜在空间具有良好的连续性。在潜在空间中进行线性插值操作可以产生平滑的过渡样本。
- 数据生成：作者使用 VAE 生成了 MNIST 手写数字数据集和 Frey Faces 数据集的新样本。结果表明，VAE 能够生成与原始数据集相似但不完全相同的新样本。

- 可解释性和潜在空间分布：通过对潜在空间中的点进行解码并观察生成的图像，作者展示了 VAE 对于潜在空间中的不同区域具有语义上的解释性。在潜在空间中，不同区域对应于不同的数字或面部特征。
- 对比学习：作者在训练过程中引入了对比学习的方法，以进一步提高 VAE 的性能。实验结果表明，对比学习可以改善 VAE 生成样本的质量和多样性。

总之，实验结果证明了 VAE 在样本生成、潜在空间连续性和可解释性等方面的优越性，并且还展示了对比学习对于改善 VAE 的性能的重要性。这些实验结果为 VAE 的应用和研究提供了强有力的支持。

2.2 **生成对抗网络：**Generative Adversarial Nets

2.2.1 **论文动机**

《Generative Adversarial Nets》[2] 是由 Ian Goodfellow 等人于 2014 年提出的重要论文，提出了一种新颖的生成模型，称为生成对抗网络（GAN）。

在这篇论文中，作者提出了一种通过对抗过程估计生成模型的框架，其中包含两个模型：一个生成模型 G 和一个判别模型 D。生成模型 G 用来生成伪造数据，而判别模型 D 用来评估一个数据样本是真实数据还是伪造数据。这两个模型通过对抗过程相互训练，最终得到一个能够生成类似于真实数据的生成模型。

GAN 是一种非常有效的生成模型，已经在图像生成、语音合成、自然语言处理等领域取得了广泛应用。这篇论文对于深度学习和生成模型的发展具有重要意义，并且一直受到广泛的研究和引用。

在此前的生成模型领域，都存在一些问题，例如生成效果不好或过度平滑等；GAN 的诞生开创了生成模型的一个全新的世界。其主要利用了博弈论的原理：训练两个神经网络分别是 D 和 G，分别是判别网络 D 和生成模型 G，生成模型是学习给定样本的数据分布，并尽可能的生成出符合给定样本数据分布的全新数据；判别器的作用是判断给定的样本是生成器生成出来的还是原始数据。我们期望于判别模型尽可能的能够分清给定数据到底是生成器生成的，还是原始数据分布；并且期望于生成器能够尽可能的逼近原始数据，做到以假乱真的效果。

最终我们可以达到纳什平衡：判别器对于给定的数据有 50% 的概率认为是原始数据，有 50% 的概率认为是生成器生成的数据。这样就达到了我们 GAN 的收敛效果。事实上，可以用验钞机和罪犯的例子来理解：印假钞的罪犯期望于能够做出以假乱真的假币，在外人眼里与真钞毫无区别；而我们的验钞机则想要区分出一张钞票的真伪。

具体而言，生成对抗网络 (GAN) 的研究动机可以从以下两方面进行考虑：

- 模型生成质量：在 GAN 之前，确实已经有了部分的生成模型，但是这些模型或多或少都存在相应的问题，例如生成质量较低，生成的结果过度平滑即过度相似于训练样本等问题；GAN 便视图用对抗的手段来解决这些问题。
- 训练的困难性：由于 GAN 是一种无监督性的学习模型，对于训练数据的标注需求比较弱，我们可以使用未经标注的数据训练该网络；同时在先前的模型中，例如深度信念网络 (DBNs) 需要马尔科夫蒙特卡洛采样 (MCMC) 等较为复杂的技术手段对训练过程加以干预，让训练过程变得较为困难。

2.2.2 方法

GAN 由两个主要组件组成：生成器（Generator）和判别器（Discriminator），它们通过对抗训练的方式相互竞争，从而达到生成逼真样本的目的。

生成器的目标是学习生成与真实样本相似的数据，而判别器的目标是区分生成器生成的样本和真实样本。两者通过博弈过程相互学习，直到达到一个平衡点，生成器能够生成逼真的样本，判别器无法区分真实样本和生成样本。接下来我们将对生成器和判别器做更加细致的分析讨论：

- 生成器（Generator）：生成器将一个随机噪声向量 z 作为输入，通过一系列神经网络层生成一个与真实样本相似的数据样本 x。生成器可以表示为函数 $G(z;\theta_g)$，其中 θ_g 是生成器的参数。
- 判别器（Discriminator）：判别器接收一个数据样本 x，并输出一个介于 0 和 1 之间的概率，表示 x 是真实样本的概率。判别器可以表示为函数 $D(x;\theta_d)$，其中 θ_d 是判别器的参数。

而对于对抗过程：在对抗训练中，生成器和判别器通过博弈的方式相互学习。生成器试图最小化判别器将生成样本判别为假的概率，而判别器试图最大化判别正确的能力。这可以用以下公式表示：

$$\min_{\theta_g}\max_{\theta_d} V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

其中，第一项表示判别器将真实样本判别为真的期望，第二项表示判别器将生成样本判别为假的期望。

对于我们生成样本的训练目标：生成器的目标是最大化生成样本被判别器判别为真实样本的概率，可以表示为：

$$\min_{\theta_g} V(G,D) = \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

对于判别器的训练目标：判别器的目标是最小化生成样本被判别器判别为真的概率，同时最大化真实数据被判别为真的概率，则对应了：

$$\max_{\theta_d} V(G,D) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)$$

对于训练过程：事实上，我们的生成网络和判别网络共享一个损失函数即上面所提到的这个，我们在训练时，首先更新判别网络 D 的参数：我们考虑，最终的目的是想要让模型对于给定样本的判别能力尽可能的强，同时对于生成器生成的数据，我们期望犯错的概率尽可能的小，所以在更新 D 的参数时，我们首先需要最大化 $D(x)$，同时最最大化 $1-D(G(z))$，值得注意的是，对于后面的 $1-D(G(z))$，代表的是判别器成功将生成数据判定为生成数据的概率。同理，在更新生成器时，我们需要最小化 $1-D(G(z))$，也就是说最大化 $D(G((z))$，这与我们上面所讨论的原理是相一致的。至于训练的一些细节，例如更新多少次判别网络 D 的参数后更新生成网络参数是可以人为调整的，不同的次数会对结果的收敛时间和效果产生影响。

由于 GAN 是基于博弈的学习过程，我们可以给出 GAN 达到纳什平衡时的状态：对于判别器而言，判别器有 50% 的概率认为给定的真实数据是伪造的数据，有 50% 的概率认为生成的数据是伪造的数据，换言之，我们的生成模型已经达到了以假乱真的效果；同时判别器也达到了最佳的判别状态。下图2.2展示了 GAN 的训练过程，算法2则展示了 GAN 的整体的算法流程图：

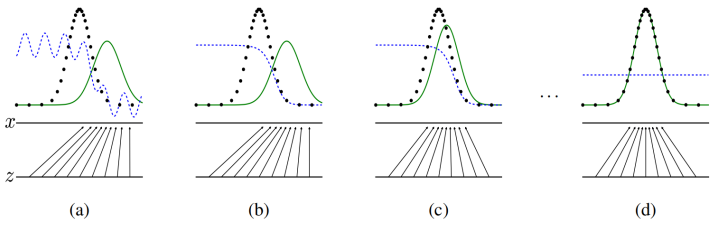


图 2.2: GAN_training

在图2.2中，(a),(b),(c),(d) 代表训练的不同阶段；蓝色的虚线代表判别网络 D，黑色的点线代表真实数据的分布 p_x，绿色的实线代表生成的数据分布 p_g，我们可以看到，我们先更新判别网络，使其判别能力尽可能的强，然后通过判别网路来更新生成网络，使生成网络生成的数据分布尽可能的拟合原始数据分布；最后经过若干轮的迭代，达到收敛点即纳什平衡点。

Algorithm 2 小批量更新 GAN 算法

Input: 设置步数 k（作为超参数），用于更新判别器

1: for 每个训练迭代 do

2: for k 步 do

3: 从噪声先验分布 $p_g(z)$ 中采样大小为 m 的噪声样本 $\{z^{(1)}, \ldots, z^{(m)}\}$

4: 从数据生成分布 $p_{\text{data}}(x)$ 中采样大小为 m 的样本 $\{x^{(1)}, \ldots, x^{(m)}\}$

5: 使用判别器的随机梯度下降更新参数：

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \right]$$

6: end for

7: 从噪声先验分布 $pg(z)$ 中采样大小为 m 的噪声样本 $\{z^{(1)}, \ldots, z^{(m)}\}$

8: 使用生成器的随机梯度下降更新参数：

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)$$

9: end for

2.2.3 **实验结果**

在论文中，作者使用了 MNIST，Toronto Face Database (TFD) 两个数据集对结果进行验证，具体的实验结果请见表1。我们考虑对结果进行分析：作者使用了基于 Parzen 窗口的对数似然估计方法来评估模型在 MNIST 数据集上的性能。对抗网络在 MNIST 数据集上的对数似然估计为 225 ± 2，这比 DBN (138 ± 2)，Stacked CAE (121 ± 1.6) 和 Deep GSN (214 ± 1.1) 的结果都要好。对于 TFD 数据集，作者同样使用了基于 Parzen 窗口的对数似然估计方法。对抗网络在 TFD 数据集上的对数似然估计为 2057 ± 26，这比 DBN (1909 ± 66) 和 Deep GSN (1890 ± 29) 的结果要好，但比 Stacked CAE (2110 ± 50) 的结果稍差。

Model	MNIST	TFD
DBN	138 ± 2	1909 ± 66
Stacked CAE	121 ± 1.6	2110 ± 50
Deep GSN	214 ± 1.1	1890 ± 29
Adversarial nets	225 ± 2	2057 ± 26

表 1: 基于 Parzen 窗口的对数似然估计

此外，作者还使用了 CIFAR-10 进行实验，虽然作者在论文中并未给出对抗网络在 CIFAR-10 数据集上的具体实验结果，但他们展示了一些从训练后的生成器

网络中抽取的样本。他们认为这些样本至少可以与文献中的更好的生成模型竞争，并突显了对抗框架的潜力。

作者在文中详细对比了提出的生成对抗方法和目前已有的生成模型的优劣性，对于结果，我们汇总成了表格2

	深度有向图模型	**深度无向图模型**	**生成自编码器**	**对抗模型**
训练	训练过程需要推理	训练过程需要推理	需要 MCMC 来近似分区函数梯度	需要同步判别器和生成器
推理	学习近似推理	变分推理	基于 MCMC 的推理	学习近似推理
采样	无困难	需要马尔可夫链	需要马尔可夫链	无困难
评估 $p(x)$	不可行，可以用 AIS 近似	不可行，可以用 AIS 近似	不明确表示，可以用 Parzen 密度估计近似	不明确表示，可以用 Parzen 密度估计近似
模型设计	几乎所有模型都遇到极大困难	需要精心设计以确保多个属性	任何可微函数在理论上都是允许的	任何可微函数在理论上都是允许的

表 2: 不同生成模型对比

2.3 **扩散模型：**Denoising Diffusion Probabilistic Models

2.3.1 **论文动机**

在前面我们讨论了基于 GAN 和 VAE 的生成模型，我们发现，这些生成模型的本质都是通过一个隐变量 z 来生成样本数据分布的 x，其主要思想是通过 decoder 或者是对抗生成网络等模型来学习样本的分布，通过不断最小化生成样本与输入样本的误差来训练出一个映射，我们期望于生成样本与输入样本的概率分布尽可能的相似，将给定的数据映射到原来的数据分布中。

我们考虑这种方法的问题，虽然看起来这种方法十分奏效和新颖，但是我们仔细考虑“相似”，其实并没有一个绝对公允或者可解释的评判标准来评测输出数据和给定数据分布之间的差异，我们 KL 散度虽然可以用来判断概率分布的相似度，但是 KL 散度是度量两个概率分布已知的分布之间的差异，而我们的输入数据的分布和生成数据的分布都是未知的，这就导致了我们的可解释性较差。

事实上，我们 VAE 做的很好，好的地方就在于，其具有噪声的鲁棒性，但是

VAE 有一个做了好几年的核心问题。上面我们所讨论的 VAE 中，涉及到了变分后验 $P(X|Z)$，这个后验概率是通过训练下降 Loss 学出来的，我们目前已有的方法对于变分后验的表达代价和计算代价都是难以估计的同时也是难以兼得的。接下来我们就引入了 diffusion(扩散模型)[3]。

事实上，扩散模型的本质是一个马尔科夫链，马尔科夫链有一个比较好的性质是平稳性，一个概率分布如果随时间变化，那么在马尔可夫链的作用下，它一定会趋于某种平稳分布（例如高斯分布）。只要终止时间足够长，概率分布就会趋近于这个平稳分布。我们考虑扩散模型，扩散模型的本质是不断在当前的状态下对图像加噪声，也就是说，在这里的马尔科夫的转移过程，每一次转移都是在对图像加噪。扩散模型最后的稳定状态是一个各向同性的噪声图片。

考虑完马尔科夫的前向过程后，我们考虑马尔科夫的逆向过程。事实上，如果我们可以知道如何从当前状态 x_t 转移到前一个状态 x_{t-1}，那么根据马尔科夫链的传递性，我们就可以从各向同性的高斯分布的噪声中逐渐去噪，得到服从原本概率分布的生成数据。

接下来我们考虑这种方法的好处：

- 逐步演化：扩散模型通过马尔科夫链来逐步演化噪声样本，使其逐渐接近目标分布。在每个扩散步骤中，样本通过与当前状态相关的转移函数进行转换，从而实现了样本的渐进转变。这种逐步演化的过程可以提高生成样本的质量，使其更接近目标分布，从而避免了传统生成模型中可能出现的模式崩溃或低质量样本的问题。
- 去噪效果：扩散模型的马尔科夫链通过逆向扩散过程，可以实现对噪声样本的去噪。在每个扩散步骤中，样本逐渐恢复到干净的样本，从而实现了去噪的目标。这对于处理具有噪声的数据是非常有用的，可以提高数据的质量和准确性。
- 灵活性和控制性：马尔科夫链可以根据具体的需求进行设计和控制，从而提供了扩散模型的灵活性。通过调整马尔科夫链的转移函数、步骤数以及扩散过程的强度，可以实现对生成过程的精细控制。这使得扩散模型能够生成具有不同特性和多样性的样本，满足不同应用场景的需求。
- 训练效率：扩散模型的马尔科夫链在训练过程中可以使用有效的数值方法进行迭代更新。相比于传统生成模型中的优化算法，如梯度下降，马尔科夫链的更新方法可以更高效地训练模型，并且具有更好的收敛性。

综上所述，扩散模型的动机是为了解决以下几个方面：

- 采样质量：传统的生成模型在采样时可能会产生低质量的样本或者陷入某些

模式中。扩散模型通过引入扩散过程，可以有效地生成高质量的样本，因为扩散过程允许样本从简单的分布逐渐演化到目标分布，从而避免了模式崩溃的问题。

- 训练效率：传统的生成模型在训练过程中可能会面临训练不稳定的挑战，尤其是当模型复杂度增加时。扩散模型通过迭代的扩散过程来优化模型，相比于传统的优化方法，可以更高效地训练模型并达到更好的收敛性。
- 扩散模型提供了一种灵活的生成框架，可以通过控制扩散过程的参数和步骤来实现不同的生成效果。这使得扩散模型在生成图像、视频、语言和其他复杂数据类型时具有更大的应用潜力。

2.3.2 方法

我们在这个小节尝试讨论一下扩散模型所应用的数学方法：

扩散模型基于一个连续的时间变量 t 来描述样本的逐步演化过程。给定一个初始噪声样本 x_0，通过一系列的扩散步骤，将其转化为目标分布 $p(x)$ 的样本。每个扩散步骤可以表示为一个概率密度函数的变换。假设在时间 t 处的样本为 x_t，扩散步骤可以定义为以下概率密度函数的变换：

$$x_{t+1} = x_t + \sqrt{\beta_t} \cdot \epsilon_t$$

其中，$\epsilon_t \sim \mathcal{N}(0, I)$ 是从标准正态分布中采样的噪声，β_t 是控制扩散强度的可学习参数。这个变换可以被视为在时间步长 t 处，当前样本与噪声之间的线性组合。

对于整个扩散过程，我们可以将其视为一个马尔科夫链，由一系列的扩散步骤组成。通过在连续时间 t 上积分，可以将扩散过程转化为概率密度函数的连续变换。这个连续变换可以表示为如下的随机微分方程：

$$dx_t = \sqrt{\beta(t)} \cdot dW_t$$

其中，dW_t 是 Wiener 过程（布朗运动）的微分，$\beta(t)$ 是时间的可学习参数。

在扩散模型中，训练的目标是通过最小化生成样本和真实样本之间的差异来优化模型参数。为此，我们可以引入一个损失函数，被称为 Denoising Score Matching (DSM) 损失函数。

DSM 损失函数可以通过比较生成样本 x_t 和真实样本 x 的局部对数密度来定

义，如下所示：

$$\mathcal{L}(x_t, x) = \frac{1}{2}\|\nabla_x \log p(x_t) - \nabla_x \log p(x)\|^2$$

其中，∇_x 表示对 x 求梯度的操作。

为了优化扩散模型，我们使用随机梯度下降等优化算法来最小化 DSM 损失函数。我们将生成样本 x_t 初始化为初始噪声样本 x_0，然后通过多个扩散步骤逐渐演化为目标分布 $p(x)$ 的样本。在每个扩散步骤中，我们通过对损失函数的梯度进行反向传播，来更新模型的参数，以使生成样本逐渐接近真实样本的分布。

通过迭代训练过程，扩散模型的参数可以被优化，使得生成样本能够更好地逼近目标分布，并产生高质量的样本。

此外，作者使用了反向自回归模型（IAF）来参数化概率密度函数。假设潜在变量 z 服从一个已知的先验分布 $p(z)$，通过 IAF 模型的变换可以得到生成样本 x 的条件分布：

$$x = g_\theta(z) = \mu + \sigma \odot z$$

其中 $g_\theta(\cdot)$ 是 IAF 模型的变换函数，μ 和 σ 是可学习的参数，z 是通过逆变换从已知先验分布 $p(z)$ 中采样得到的。

我们考虑训练的算法：算法首先对模型参数进行初始化。然后，在每个训练迭代中，对于数据集中的每个样本，算法进行以下操作：

- 从噪声先验分布 $p(z)$ 中采样初始噪声样本 x_0，作为扩散过程的起始点。
- 使用扩散步数 K 进行循环迭代。在每个迭代步骤中，通过引入逐渐减小的扩散参数 $\beta(t)$，从标准正态分布 $\epsilon_t \sim \mathcal{N}(0, I)$ 中采样噪声，并将其与当前样本进行相加来生成下一个时间步的样本 x_{t+1}。通过递增 t 的值，逐步逼近真实数据分布。
- 在扩散过程中，使用 Denoising Score Matching (DSM) 损失函数计算梯度。该损失函数通过比较样本 x_t 的对数概率密度函数（通过模型计算）和真实样本 $x^{(i)}$ 的对数概率密度函数之间的差异来进行训练。通过对所有扩散步骤的梯度求和并除以 T，得到模型参数的梯度。
- 使用梯度下降算法来更新模型参数 θ，使其逐步逼近真实数据分布。
- 重复上述步骤，直到达到预定的训练步数。

该算法通过逐步扩散和逆扩散过程，利用梯度下降算法训练扩散模型，使其

能够生成与真实数据分布相似的样本。通过控制扩散步数和调整扩散参数，可以灵活地控制生成样本的多样性和质量。对于上述的讨论过程，我们整理出了如下所示的算法流程图3：

Algorithm 3 Denoising Diffusion Probabilistic Models

Input: 数据样本集合 $\{x^{(1)}, \ldots, x^{(n)}\}$，训练步数 T，扩散步数 K

初始化参数 θ
for 每个训练迭代 do
    for 每个样本 $x^{(i)}$ do
        从噪声先验分布 $p(z)$ 中采样初始噪声样本 x_0
        for $t = 0$ to $T - 1$ do
            计算扩散参数 $\beta(t)$
            从标准正态分布 $\epsilon_t \sim \mathcal{N}(0, I)$ 中采样噪声
            更新 $x_{t+1} = x_t + \sqrt{\beta(t)} \cdot \epsilon_t$
        end for
        使用 Denoising Score Matching (DSM) 损失函数计算梯度：
        $\nabla_\theta = \frac{1}{T} \sum_{t=0}^{T-1} \left[\nabla_x \log p(x_t) - \nabla_x \log p(x^{(i)})\right]$
        更新模型参数 θ 使用梯度下降算法
    end for
end for

2.3.3 实验结果

我们在这一节中，总结了作者的实验结果，并将其汇总成了表格，如表3和表4所示，具体的介绍如下：

表3显示了目前主流的算法在 CIFAR10 上的起始分数、FID 分数和负对数似然（无损编码长度）。Diffusion 模型的 FID 得分为 3.17，即无条件模型达成了比文献中的大多数模型（包括类条件模型）更好的样本质量。值得注意的是，DDPM 的 FID 分数是根据训练集计算的，而作者提到，当他们相对于测试集计算它时，得分为 5.24，这仍然优于文献中的许多训练集 FID 得分。

在表4中，该模型的结果展示了反向过程参数化和训练目标对样本质量的影响。作者发现预测的基线选项只有在训练真正的变分界、而非未加权均方误差时才能很好地工作，如作者所提议的，那样使用固定方差的变分界进行训练时，预测 ϵ 的性能大致与预测均值一样好，但其在使用作者提出的简化目标进行训练时要好得多。

表 3: CIFAR10 results. NLL measured in bits/dim.

Model	IS	FID	NLL (Test/Train)
Conditional			
EBM	8.30	37.9	
JEM	8.76	38.4	
BigGAN	9.22	14.73	
StyleGAN2 + ADA (v1)	10.06	2.67	
Unconditional			
Diffusion (original)			≤5.40
Gated PixelCNN	4.60	65.93	3.03 (2.90)
Sparse Transformer			2.80
PixelIQN	5.29	49.46	
EBM	6.78	38.2	
NCSNv2			31.75
NCSN	8.87±0.12	25.32	
SNGAN	8.22±0.05	21.7	
SNGAN-DDLS	9.09±0.10	15.42	
StyleGAN2 + ADA (v1)	9.74±0.05	3.26	
Ours (L, fixed isotropic Σ)	7.67±0.13	13.51	≤3.70 (3.69)
Ours (Lsimple)	9.46±0.11	3.17	≤3.75 (3.72)

表 4: Objective, IS, and FID scores.

Objective	IS	FID
$\overline{\boldsymbol{\mu}}$ prediction (baseline)		
L, learned diagonal	7.28 ± 0.10	23.69
L, fixed isotropic	8.06 ± 0.09	13.22
$\|\overline{\boldsymbol{\mu}} - \overline{\boldsymbol{\mu}}_\theta\|^2$	-	-
$\boldsymbol{\epsilon}$ prediction		
L, learned diagonal		-
L, fixed isotropic	7.67 ± 0.13	13.51
$\|\overline{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon}_\theta\|^2$ (L_{simple})	9.46 ± 0.11	3.17

2.4 **稳定扩散模型** LDMs

在去年 “AI 作画” 火出了圈，用户惊叹于人工智能的创造力，只需要输入自然语言的 prompt，人工智能就可以输出一幅满足自然语言意境指导的画作。AI 作画近期取得如此巨大进展的原因有很大的功劳归属于 Stable Diffusion 的开源。Stable

diffusion 是一个基于 Latent Diffusion Models（潜在扩散模型，LDMs）的文图生成（text-to-image）模型。

stable diffusion 旨在通过不停去除噪音来获得期望结果的一个生成式模型。在 AI 绘画早期，扩散是发生在像素空间 pixel space 的，不仅效果不好而且单张图大约需要 10-15 分钟，后来英国初创公司 Stability AI 对模型进行了改进，把核心计算从像素空间（pixel space）改到了潜空间 (latent space) 中，使得稳定性与像素质量都得到了极大提升，并且速度提高了近 100 倍，故名 stable diffusion。

Latent Diffusion Models 通过在一个潜在表示空间中迭代“去噪”数据来生成图像，然后将表示结果解码为完整的图像，让文图生成能够在消费级 GPU 上，在 10 秒级别时间生成图片，大大降低了落地门槛，也带来了文图生成领域的大火。

我们本文中来讨论一下 stable diffusion[4] 后面的论文原理。

2.4.1 论文动机

扩散模型通过将图像形成过程分解为一个马尔科夫过程，即将该过程分解为一系列去噪自编码器的顺序应用，其在图像生成领域及其他领域取得了最先进的合成结果。此外，它们的构建允许使用导向机制来控制图像生成过程而无需重新训练。然而，扩散模型通常直接在像素空间中对图像进行操作，性能较强的扩散模型通常需要消耗极大的计算资源，推断过程也由于顺序评估而变得极为昂贵。

作者考虑到：为了在有限的计算资源上构建并训练扩散模型，同时保持其质量和灵活性，作者考虑将它们应用于强大的预训练自编码器的潜在空间中。与先前的工作相比，通过在这种空间上训练扩散模型，可以在复杂性和细节之间达到近乎最优的平衡状态，极大地提升了视觉保真度。

通过在模型架构中引入交叉注意力层，作者将扩散模型转化为强大而灵活的生成器，可以用于一般的条件输入，如文本或边界框，从而实现了高分辨率合成的卷积方式。作者将该模型命名为潜在扩散模型（LDMs）。潜在扩散模型在图像修复和类别条件图像合成方面取得了最新的最佳分数，并在无条件图像生成、文本到图像合成和超分辨率等各种任务上表现出极具竞争力的性能，同时大大降低了与基于像素的 DM 相比的计算要求。

总的来看，本文的动机可以总结为以下几点：

- 解决传统扩散模型的资源消耗问题。
- 解决传统模型无法合成高分辨率的问题。
- 考虑通过 VAE 和 DM 的结合，将训练复杂度和模型生成质量达到一个最优

的平衡。

2.4.2 方法

为了降低训练高分辨率图像合成的扩散模型的计算需求，作者观察到虽然扩散模型可以通过对相关损失项进行欠采样来忽略感知上不相关的细节，但它们仍然需要在像素空间中进行昂贵的函数评估，这导致计算时间和能源资源的巨大需求。

为了克服这个缺点，作者在本文中提出了一个明确的压缩学习阶段和生成学习阶段的分离方法，如图2.3。为了实现这一点，作者利用了一个自编码模型，该模型学习到与图像空间在感知上等价但计算复杂度显著降低的空间。

这种方法具有以下几个优点：

- 通过回避高维图像空间中对数据的操作，我们获得了计算效率更高的扩散模型，因为采样是在低维空间中进行的。
- 利用扩散模型从其 UNet 架构继承的归纳偏差，使其对具有空间结构的数据特别有效，因此减轻了先前方法所需的激进的降低质量的压缩级别的需求。
- 最后，LDMs 便可以成为通用的压缩模型，其潜在空间可以用于训练多个生成模型，并且还可以用于其他下游应用，如单图像的 CLIP 引导合成。

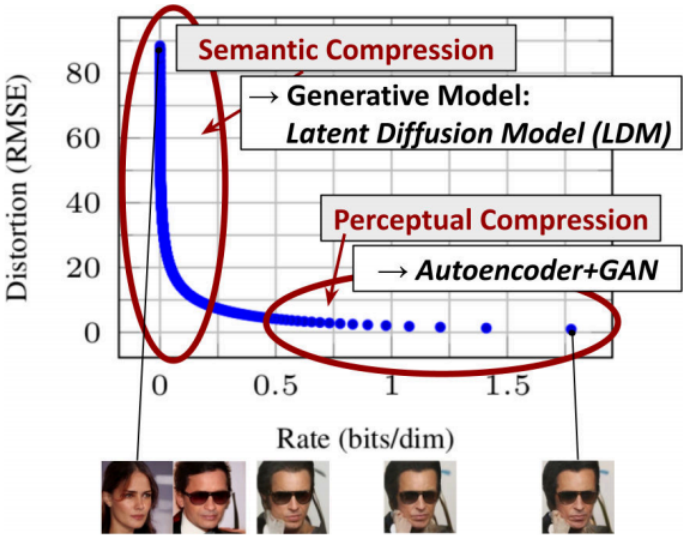


图 2.3: 阶段学习分离方法

对于图2.3，我们考虑：数字图像中的大部分像素值对应于无法感知的细节。虽然扩散模型通过最小化相应的损失项来抑制这种语义上无意义的信息，但在训练

过程中，训练和推断仍然需要对所有像素进行评估，导致了多余的计算和不必要的昂贵的优化和推断过程。为此，作者提出了潜在扩散模型（LDMs）作为一种有效的生成模型和一个单独的轻度压缩阶段，仅消除无法感知的细节。

作者设计了一个 LDMs 的框架，如下图2.4所示：

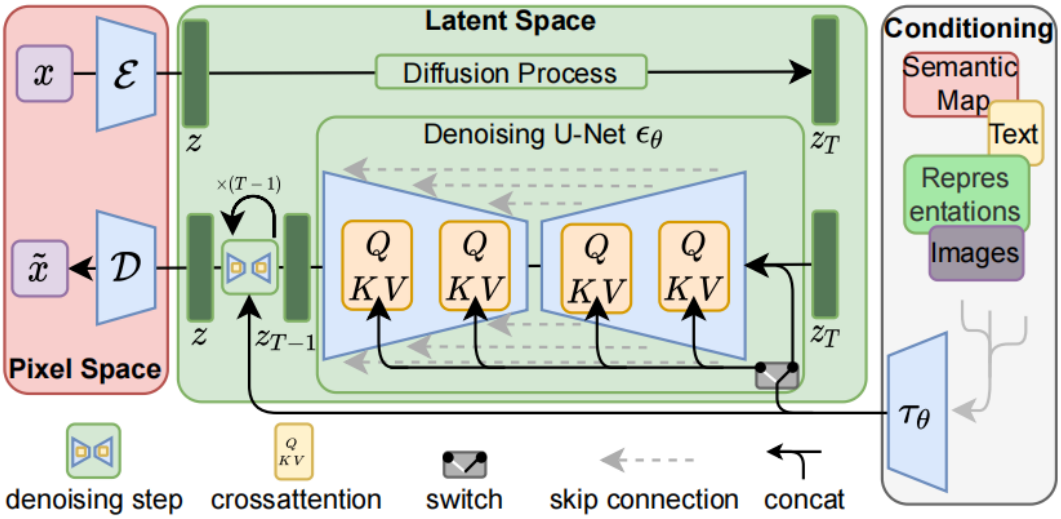


图 2.4: LDMs 框架图

我们考虑论文中的组成部分：

- 自动编码器：预训练的自动编码器包含编码器函数（表示为 E）和解码器函数（表示为 D）。对于输入图像 $x \in \mathbb{R}^{H\times W\times 3}$（RGB 空间），编码器将其编码为潜在表示 $z = E(x)$，解码器从潜在表示重构图像，得到 $\tilde{x} = D(z) = D(E(x))$，其中 $z \in \mathbb{R}^{h\times w\times c}$。编码器通过因子 $f = H/h = W/w$ 对图像进行下采样，并尝试不同的下采样因子。
- 压缩正则化：LDMs 使用正则化技术确保潜在空间具有较低的方差，实现压缩效果。常用的两个变体包括：
  a. KL 正则化（KL-reg.）：该变体对学习到的潜在空间施加轻微的 KL 惩罚，使其向标准正态分布靠近，类似于 VAE（变分自编码器）的思想。
  b. VQ 正则化（VQ-reg.）：该变体在解码器中使用向量量化层（vector quantization layer），将潜在空间离散化为离散的向量，以实现压缩效果。这种模型可以被解释为 VQGAN，但是将量化层吸收到解码器中。由于后续的潜在扩散模型（DM）设计用于处理学习到的潜在空间 $z = E(x)$ 的二维结构，所以可以使用相对较轻的压缩率并实现很好的重构效果。这与先前的方法不同，先前的方法依赖于学习空间 z 的任意一维顺序，以自回归方式对其分布进行

建模，从而忽略了 z 的许多固有结构。因此，LDMs 可以更好地保留图像 x 的细节。

- 交叉注意力机制通常由以下几个步骤组成：
  1. 特征提取：首先，对条件输入和图像特征进行特征提取。条件输入可以通过相应的编码器网络转换为特征向量，而图像特征则可以通过预训练的卷积神经网络（如 VGG）提取。
  2. 相似度计算：然后，通过计算条件输入特征和图像特征之间的相似度，得到一个相似度矩阵。这可以通过计算它们之间的内积或其他相似性度量来完成。
  3. 注意力权重计算：基于相似度矩阵，使用合适的注意力机制（如 softmax 函数）计算条件输入特征对于每个图像位置的注意力权重。这些权重表示了生成图像时应该关注条件输入的程度。
  4. 特征融合：将注意力权重与图像特征进行加权融合，得到最终的融合特征表示。这样，生成图像时可以同时考虑图像自身的特征和条件输入的相关信息。

2.4.3 **实验结果**

在我们给出具体的实验结果之前，我们考虑一些评估图像生成的指标：

- FID（Fréchet Inception Distance）：FID 是一种广泛使用的评估指标，用于衡量生成图像与真实图像之间的差异。它通过计算生成图像和真实图像在预训练的图像分类网络（如 InceptionNet）中特征表示的统计距离来衡量图像质量。较低的 FID 值表示生成图像与真实图像之间的差异较小，质量较高。
- IS（Inception Score）：IS 是另一种常用的评估指标，用于衡量生成图像的多样性和真实度。它基于预训练的图像分类网络，计算生成图像的类别分布的熵和条件分布的 KL 散度。较高的 IS 值表示生成图像具有更好的多样性和真实度。
- SSIM（Structural Similarity Index）：SSIM 是一种衡量图像相似性的指标，它考虑了图像的亮度、对比度和结构信息。在无条件图像合成中，SSIM 可以用来评估生成图像与真实图像之间的结构相似性。
- PSNR（Peak Signal-to-Noise Ratio）：PSNR 是一种衡量图像质量的传统指标，它通过计算生成图像和真实图像之间的均方误差来衡量图像的失真程度。较高的 PSNR 值表示生成图像与真实图像之间的失真较小。

作者首先对比了目前主流的方法在无条件图像合成上的数据指标，如2.5所示：

CelebA-HQ 256 × 256				FFHQ 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑	Method	FID ↓	Prec. ↑	Recall ↑
DC-VAE [61]	15.8	-	-	ImageBART [21]	9.57	-	-
VQGAN+T. [23] (k = 400)	10.2	-	-	U-Net GAN (+aug) [75]	10.9(7.6)	-	-
PGGAN [38]	8.0	-	-	UDM [42]	5.54	-	-
LSGM [90]	7.22	-	-	StyleGAN [40]	4.16	0.71	0.46
UDM [42]	7.16	-	-	ProjectedGAN [74]	3.08	0.65	0.46
$LDM-4$ (ours, $500-s^{\dagger}$)	5.11	0.72	0.49	$LDM-4$ (ours, 200-s)	4.98	0.73	0.50
LSUN-Churches 256 × 256				LSUN-Bedrooms 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑	Method	FID ↓	Prec. ↑	Recall ↑
DDPM [29]	7.89	-	-	ImageBART [21]	5.51	-	-
ImageBART [21]	7.32	-	-	DDPM [29]	4.9	-	-
PGGAN [38]	6.42	-	-	UDM [42]	4.57	-	-
StyleGAN [40]	4.21	-	-	StyleGAN [40]	2.35	0.59	0.48
StyleGAN2 [41]	3.86	-	-	ADM[15]	1.90	0.66	0.51
ProjectedGAN [74]	1.59	0.61	0.44	ProjectedGAN [74]	1.52	0.61	0.34
$LDM-8^{*}$ (ours, 200 − s)	4.02	0.64	0.52	$LDM-4$ (ours, 200-s)	2.95	0.66	0.48

图 2.5: 无条件图像合成的评估指标。

接下来作者比较了自己的方法和主流的几种方法在文图合成中的性能指标：如下所示：

表 5: 文本条件图像合成评估

Method	DALL-E	CogView	Lafite	LDM-KL-8	LDM-KL-8-G∗
FID ↓	27.50	27.10	26.94	23.35	12.61
IS ↑	17.90	18.20	26.02	19.93±0.35	26.62±0.38

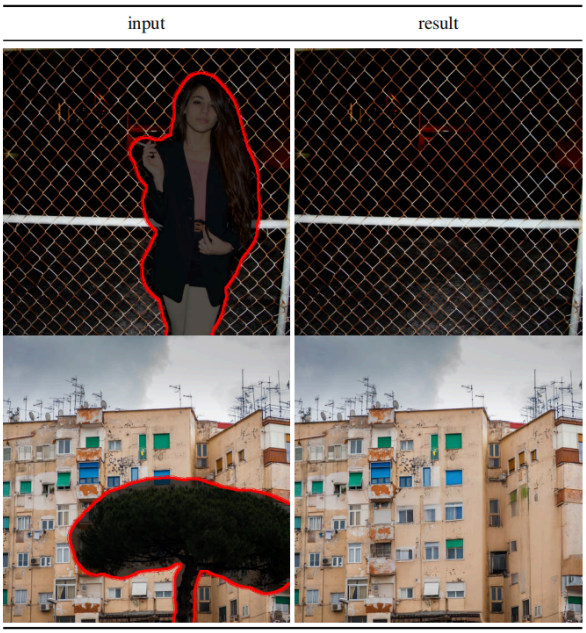
接下来，作者比较了在 ImageNet 数据集上，将基于类别条件的 ImageNet LDM 与最新的类别条件图像生成方法进行比较。如2.4.3所示：

Method	FID↓	IS↑	Precision↑	Recall↑	N_{params}	
BigGan-deep [3]	6.95	203.6±2.6	**0.87**	0.28	340M	-
ADM [15]	10.94	100.98	0.69	**0.63**	554M	250 DDIM steps
ADM-G [15]	4.59	186.7	0.82	0.52	608M	250 DDIM steps
LDM-4 (ours)	10.56	103.49±1.24	0.71	0.62	400M	250 DDIM steps
LDM-4-G (ours)	**3.60**	**247.67**±5.59	**0.87**	0.48	400M	250 steps, classifier-free guidance [31], scale 1.5

接下来，作者使用 ImageNet-Val 的 ×4 放大结果（256^2），如2.4.3所示；这项评估是基于 ImageNet 数据集的验证集进行的，并且图像被放大到 256x256 的尺寸。这是为了确保对生成图像质量的评估具有足够的细节和准确性。此外，作者还比较了模型的参数，可以看到，LDMs 在 FID 等参数指标上都领先于目前主流的合成方法，且 LDMs 的参数也显著小于其余几种方法。

Method	FID ↓	IS ↑	PSNR ↑	SSIM ↑	N_{params}	Throughput $*\left[\frac{\text{samples}}{s}\right]$
Image Regression [70]	15.2	121.1	27.9	0.801	625M	N/A
SR3 [70]	5.2	180.1	<u>26.4</u>	<u>0.762</u>	625M	N/A
$LDM-4$ (ours, 100 steps)	$2.8^{\dagger}/4.8^{\ddagger}$	166.3	$24.4_{\pm 3.8}$	$0.69_{\pm 0.14}$	169M	4.62
$LDM-4$ (ours, big, 100 steps)	$\overline{2.4}^{\dagger}/\overline{4.3}^{\ddagger}$	174.9	$24.7_{\pm 4.1}$	$0.71_{\pm 0.15}$	552M	4.5
$LDM-4$ (ours, 50 steps, guiding)	$4.4^{\dagger}/6.4^{\ddagger}$	$\overline{153.7}$	25.8 ± 3.7	$0.74_{\pm 0.12}$	184M	0.38

此外，作者还进行了图像修复，图像超分等实验，并且取得了很棒的效果，如下图所示：



3 思考与理解

在这里我们先总结一下模型各自的特点：LDMs：LDMs 是一种基于扩散过程的生成模型，通过在潜在空间中进行渐进的扩散来生成图像。LDMs 具有以下特点：

- 通过在低维空间上进行采样，提供了更高效的图像生成过程。
- 利用 UNet 架构的归纳偏差，对具有空间结构的数据特别有效，无需过于依赖质量降低的压缩级别。
- 提供通用的压缩模型，其潜在空间可用于训练多个生成模型，也可用于其他下游应用，如基于单张图像的 CLIP 引导合成。

DMs：DMs 也是一种基于扩散过程的生成模型，通过在像素空间中进行渐进的反向扩散来生成图像。DMs 与 LDMs 的区别在于采样空间的不同，DMs 采样像素空间，而 LDMs 采样潜在空间。DMs 的特点包括：

- 通过最小化与感知无关的细节的损失项，可以忽略感知上不相关的细节。
- 在像素空间中需要进行昂贵的函数评估，导致计算时间和能源资源的巨大需求。
- 具有 UNet 架构的归纳偏差，对具有空间结构的数据特别有效。

VAE：VAE 是一种基于变分推断的生成模型，通过学习潜在空间中的编码和解码过程来生成图像。VAE 具有以下特点：

- 通过学习潜在空间的分布，可以实现对图像生成过程的建模和控制。
- 通过引入 KL 散度项或其他正则化方法，可以约束潜在空间的结构和平滑性。
- 可以使用重参数化技巧有效地进行训练和推断。

GAN：GAN 是一种基于对抗训练的生成模型，通过训练一个生成器和一个判别器来实现图像生成。GAN 具有以下特点：

- 通过生成器和判别器之间的对抗过程，可以逐渐提升生成器的生成能力。
- 通过最小化生成器和判别器之间的损失函数，可以实现生成图像的多样性和质量。
- 可以通过引入条件信息或其他扩展技术，实现有条件的图像生成。

3.1 联系与区别

从我们总结出的特点来看，这几种模型的联系如下：

- 模型都是生成模型，并且它们都致力于通过学习数据分布的方式来生成新的、与训练数据相似的图像。
- 模型都采用了深度学习的方法，通过神经网络来学习图像生成的过程。
- 都能够生成高质量的图像，并在图像生成任务上取得了显著的进展。

区别如下：

- LDMs 和 DMs 是基于扩散过程的生成模型，而 VAE 和 GAN 则采用不同的生成机制。LDMs 和 DMs 通过在潜在空间或像素空间中的渐进扩散来生成图像，而 VAE 通过变分推断和解码过程，GAN 通过对抗训练的方式生成图像。
- LDMs 和 DMs 关注于高效的图像生成和压缩，通过对低维空间进行采样来提高计算效率。而 VAE 和 GAN 更注重对潜在空间的建模和多样性生成。
- 这些模型在损失函数和训练过程上有所区别。LDMs 和 DMs 通常采用感知损失和对抗目标函数进行训练，VAE 通过最小化重构损失和 KL 散度进行训练，GAN 通过最小化生成器和判别器之间的对抗损失进行训练。

3.2 尚未解决的问题

我们考虑目前领域可能存在的问题：

- 图像质量与多样性平衡：生成模型在生成高质量图像的同时，往往面临生成多样性不足的问题。生成的图像可能缺乏变化、创新和多样性，导致生成结果的局限性。
- 训练数据的需求：许多生成模型需要大量高质量的训练数据来学习数据分布。然而，获取大规模高质量的训练数据集可能是昂贵、耗时和困难的。
- 模式崩溃和模式坍缩：生成模型可能出现模式崩溃，即只生成训练数据集中的一小部分样本，而忽略其他样本。相反，模式坍缩是指生成模型生成过于类似的样本，缺乏多样性和创造力。
- 长期依赖和一致性：对于序列生成任务，如文本生成或视频生成，生成模型需要处理长期依赖关系，以便在生成过程中保持一致性和连贯性。
- 解释性和可控性：生成模型通常被视为黑盒模型，难以解释和理解其内部机制。同时，生成模型的生成过程往往缺乏可控性，难以对生成结果进行精确的控制和调整。
- 训练和推理的效率：一些生成模型在训练和推理过程中需要大量的计算资源和时间，限制了其实际应用的可行性和效率。

3.3 未来研究趋势

未来生成领域的研究方向可能包括以下几个方面：

- 提高生成模型的质量和多样性：研究人员可以探索新的网络架构、损失函数和训练方法，以改善生成模型生成图像、文本、音频等领域的质量和多样性。这可能涉及到更有效的数据增强技术、更准确的评估指标以及更鲁棒的训练算法。
- 可控性和解释性的生成：研究人员可以致力于开发可控的生成模型，使用户能够在生成过程中精确控制生成结果的某些属性，如颜色、形状、风格等。同时，提高生成模型的解释性，使生成结果的生成过程更加可解释和可理解。
- 长期依赖和一致性建模：对于序列生成任务，改善生成模型对长期依赖关系的建模能力是一个重要的研究方向。研究人员可以探索新的架构和算法来捕捉长期依赖，以生成更连贯和一致的序列结果。
- 少样本和零样本生成：在少样本和零样本情况下生成新的内容是一个具有挑战性的问题。研究人员可以探索利用先验知识、迁移学习、元学习等技术来提高少样本和零样本生成的能力，使生成模型能够在少量样本或没有样本的情况下生成有质量的结果。
- 训练和推理的效率：改善生成模型的训练和推理效率是一个重要的研究方向。研究人员可以通过剪枝、量化、并行计算等方法来减少生成模型的计算资源需求，以实现更快速、更高效的训练和推理过程。
- 多模态生成：多模态生成是指生成模型能够同时生成多种类型的内容，如图像和文本的联合生成。研究人员可以探索多模态生成的方法和算法，以实现更丰富、更多样化的生成结果。

我们总结的研究方向基于我们上面讨论过的生成模型的核心问题，如模型结构、训练算法、评估指标等。随着技术的不断发展和研究的深入，我们可以期待在未来看到更先进、更强大的生成模型出现，为图像生成领域带来更多的突破和创新。

参考文献:

[1] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In In Advances in Neural Information Processing Systems, pages 2672–2680, 2014.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.