中国科学技术大学

计算机图形学前沿

3D 生成模型进展

摘　要

本文对 3D 生成模型的多种方法进行了综述，并对每类模型的原理和变种进行了简单的介绍，主要包括生成对抗网络、自回归模型、扩散模型和 NeRF 几类。在本文中，我们将重点关注这些论文中的相关算法，并对其做简要概括。

关键词：3D、生成模型、图形学

目　录

第一章　引言

3D 生成模型是计算机图形学领域的一种数学模型，用于生成三维图像。它的原理通常基于坐标系变换，通过将一个坐标系下的点 (或向量) 变换到另一个坐标系下，实现三维图像的生成。在 3D 生成模型中，通常使用两个坐标系，一个是物体自身的坐标系，另一个是观察者的坐标系。通过这两个坐标系之间的变换，可以实现物体在空间中的位置和方向的变换，从而生成逼真的三维图像。

3D 生成模型在游戏开发、虚拟现实、计算机辅助设计等领域具有广泛的应用。在游戏开发中，3D 生成模型可以用于生成游戏场景、角色和道具等；在虚拟现实中，3D 生成模型可以用于生成虚拟环境，如虚拟城市、虚拟现实体验等；在计算机辅助设计中，3D 生成模型可以用于生成三维图形界面，如三维打印、三维建模等。

3D 生成模型的原理可以简单概括为通过坐标系变换实现物体在空间中的位置和方向的变换，从而生成逼真的三维图像。这个过程需要对物体的几何形状、材质、纹理等进行建模，并通过计算机图形学技术进行渲染和显示。建模和渲染的过程需要使用多种软件和工具，如 3D 建模软件、渲染引擎、图形库等。

近年来，随着深度学习技术的发展，基于深度学习的 3D 生成模型也得到了快速的发展。基于深度学习的 3D 生成模型可以通过学习大量的三维图像数据，自动学习到物体的几何形状、材质、纹理等特征，并生成逼真的三维图像。在学习手段上，可以分为 GAN (Generative Adversial Network, 生成对抗网络)、自回归模型、扩散模型、NeRF (Neural Radiance Field, 神经辐射场) 等等，生成效率和生成质量上这些模型各有千秋，本文就对它们逐一进行讲解。

第二章　生成对抗网络

第一节　概述

生成对抗网络 (Generative Adversarial Network, 以下简称 GAN)[1] 于 2014 年由 Ian Goodfellow 等人提出后迅速地就成为了当时大火的研究课题。到目前为止，GAN 已有上千变种，除去最初的图像生成与处理领域外，其在自然语言处理、语音合成和视频生成与处理等领域上也都有着广泛的应用。对于 3D 生成模型来说，GAN 在具体的某类图像上可以生成逼真的结果，如人像或风景等。但其也有一些缺陷，比如训练过程不稳定、难以生成复杂图像等等。

第二节　原理

GAN 由两个有机整体构成——生成器和判别器。其灵感来源于博弈论中的二人零和博弈，生成器和判别器相互竞争以达到各自的目标。生成器将随机输入的高斯噪声映射成生成数据，而判别器则负责判断输入数据是否真实，两者相互竞争，来提高生成器的生成质量和判别器的判断准确性。在训练过程中，生成器和判别器不断地迭代，生成器试图生成更真实的数据以欺骗判别器，而判别器则试图更准确地判断数据是否真实。通过不断的迭代，生成器能够生成与真实数据分布相似的新数据。

以下我们对 GAN 中术语进行解释。我们将生成器记作 G，判别器记作 D，如上的描述即生成器 G 要无中生有，用生成数据骗过判别器 D。而最优状态是判别器 D 对生成器 G 生成的数据判断准确率为 0.5，此时生成的数据和真实数据几乎一致。

式 (2.1)给出了 GAN 的目标函数，也即损失函数。其中 G 代表生成器，D 代表判别器，$\boldsymbol{x}$ 代表真实数据，p_{data} 代表真实数据的概率密度分布，$\boldsymbol{z}$ 代表随机输入数据，该数据是随机高斯噪声。

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]. \quad (2.1)$$

分析上式可以看出，从判别器 D 角度来说判别器 D 希望能尽可能区分真实样本 $\boldsymbol{x}$ 和生成样本 $G(\boldsymbol{z})$，因此 $D(\boldsymbol{x})$ 要尽可能大，$D(G(\boldsymbol{z}))$ 要尽可能小，也即 $V(D,G)$ 整体尽可能大。从生成器 G 角度来说，生成器 G 希望生成的虚假数据可以尽可能骗过判别器 D，也就是希望 $D(G(\boldsymbol{z}))$ 尽可能大，也就是 $V(D,G)$ 整体尽可能

小。GAN 的两个模块在训练相互竞争，最后达到全局最优。

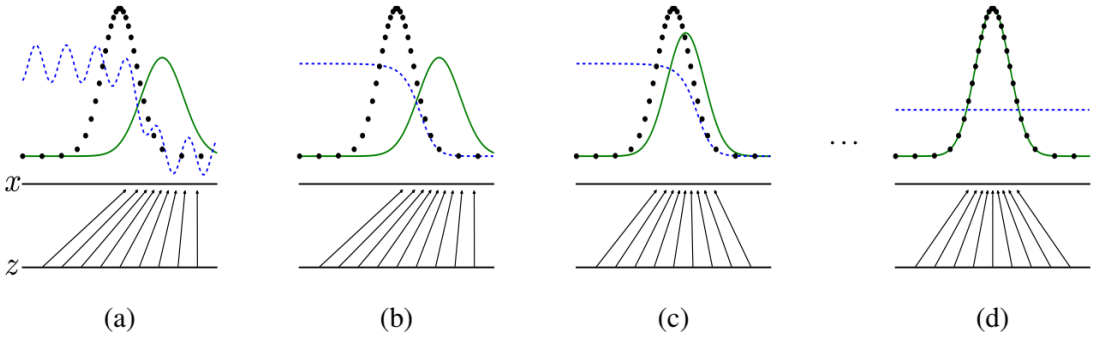

图 2.1　GAN 训练过程示意图[1]

图 2.1是原始论文给出的 GAN 训练过程示意图。两条平行线下方的一条代表噪声 $\boldsymbol{z}$，在此处服从正态分布，而上方的一条代表真实样本 $\boldsymbol{x}$，黑色圆点虚线代表真实数据分布 $p_{\boldsymbol{x}}$，绿色实线代表生成器 G 的生成数据的概率分布 p_g，蓝色虚线代表判别器 D 的输出，即对生成数据的判断正确概率。从 $\boldsymbol{z}$ 到 $\boldsymbol{x}$ 的箭头代表映射 $G(\boldsymbol{z})$ 得到对象空间中的非均匀分布 p_g，而 G 在 p_g 的高密度区域收缩，在其低密度区域扩张。

考虑到如下情况可以使得结果接近收敛：p_g 近似于 p_{data} 而判别器 D 不完全准确。在算法的内循环中，D 被训练来区分数据样本，最终收敛到 $D^*(\boldsymbol{x}) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})+p_g(\boldsymbol{x})}$。在更新 G 后，D 的梯度引导 $G(\boldsymbol{z})$ 流向更可能被分类为数据的区域。经过几步训练后，如果 G 和 D 具有足够的容量，它们将达到一个两者都无法继续改进的点，因为此时有 $p_g = p_{\text{data}}$，判别器 D 无法区分这两个分布，即

算法 2.1　小批量随机梯度下降训练生成对抗网络算法

for number of training iterations **do**
  for k steps **do**
    Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$;
    Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$;
    Update the discriminator by ascending its stochastic gradient

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

  end
  Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$;
  Update the generator by descending its stochastic gradient

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

end

$D(\boldsymbol{x}) = \frac{1}{2}$，已达最优。根据以上的分析，可以给出算法 2.1的算法过程。

第三节　变种

目前 GAN 已有上千变种，在此我们只举其中具有代表性的两种进行说明。

一、DCGAN

深度卷积生成对抗网络 (Deep Convolutional Generative Adversarial Network, 以下简称 DCGAN)[2]极大地提升了原始 GAN 训练的稳定性以及生成结果质量，奠定了之后几乎所有 GAN 的基本网络架构。

DCGAN 主要是在网络架构上改进了原始 GAN，其生成器与判别器都利用卷积神经网络 (Convolutional Neural Network, 以下简称 CNN) 的架构替换了原始 GAN 的全连接网络。

相对于 CNN，DCGAN 用卷积层替代了空间池化层，这样下采样过程不再固定地抛弃某些位置的像素值，而是让网络可以自行学习下采样方式。DCGAN 去除了全连接层，而是使用了全局均值池化，这样虽然收敛速度会变慢，但可以提高模型的稳定性。DCGAN 采用了 BN (Batch Normalization) 层，也就是将输入归一化到正态分布，BN 层可以起到加速收敛和减缓过拟合的作用。实验发现对所有层都使用 BN 会造成采样的震荡和网络不稳定，因此只对生成器 G 的输出和判别器 D 的输入使用 BN。生成器 G 输出层使用 tanh 激活函数，其余层全部使用 ReLu 激活函数。判别器所有层都使用 LeakyReLU 激活函数。

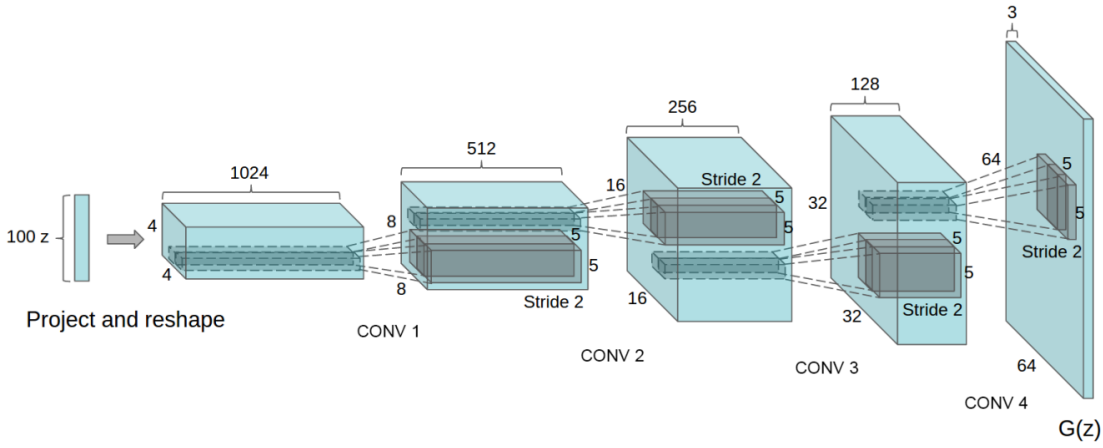

图 2.2　DCGAN 生成器结构示意图[2]

二、StyleGAN

StyleGAN[3]是英伟达于 2018 年提出的一种模型，其可以在不影响其他层级的情况下，控制某层级的视觉特征。StyleGAN 借鉴风格迁移，提出了基于样式的生成器，实现了无监督地分离高级属性 (人脸、姿势、身份) 和随机变化 (雀斑、头发、瞳孔)，实现了对生成图像中特定尺度的属性的控制。生成器从一个可学习的常量输入开始，隐码在每个卷积层调整图像的“样式”，从而直接控制不同尺度下图像特征的强度。

三、GigaGAN

GigaGAN[4]改进了 StyleGAN 架构，首次利用 GAN 实现了复杂图像的生成，其模型的参数大小达到了 10 亿。该研究通过保留一组滤波器 (filter) 并采用特定于样本的线性组合来有效地扩展了生成器的容量。该研究还采用了扩散模型中常用的技术，如将自注意力 (仅图像) 和交叉注意力 (图像-文本) 与卷积层交织在一起，这可以提高模型性能。同时还引入了多尺度训练，并提出一种新方案来改进图像-文本对齐和生成输出的低频细节。多尺度训练允许基于 GAN 的生成器更有效地使用低分辨率块中的参数，从而实现了更好的图像-文本对齐和图像质量。

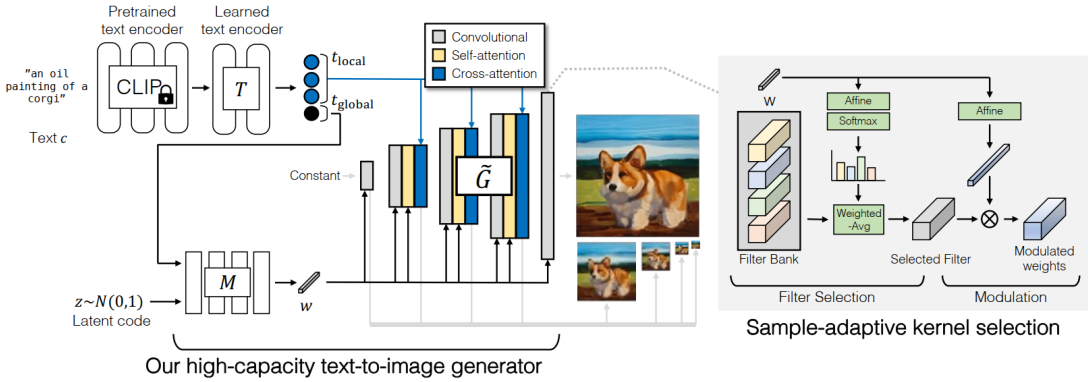

图 2.3　GigaGAN 生成器结构示意图[1]

如图 2.3所示，作者使用预训练的 CLIP 模型和学习好的文本编码器 T 提取文本嵌入。然后使用交叉注意力将本地文本描述符提供给生成器，全局文本描述符和潜在代码 z 一起被送到样式映射网络 M 以生成样式向量 w，之后样式向量 w 输入形成样本自适应核帮助调节主生成器。

第三章　自回归模型

第一节　概述

传统意义上，自回归模型是一种统计模型，基于该时间序列过去的值和一些外部变量，被用于预测一个时间序列中的未来值。该模型的核心思想是使用过去的数据来估计模型参数，然后使用这些参数来预测未来的值。在自回归模型中，时间序列被看作是一个随机过程，该过程由一些未知的参数和外部变量组成。模型的参数可以通过最小化观测值和预测值之间的离差平方和来估计。一旦参数被估计，模型就可以用于预测未来的值。自回归模型可以应用于许多领域，如经济学、金融学、气象学等。

在 3D 生成模型领域，自回归模型的应用时间并不算久。在此之前，自回归模型主要应用于自然语言处理 (Natural Language Processing, 以下简称 NLP) 领域。2017 年经典之作[5]给出了划时代意义的 Transformer 模型之后，到 2021 年才由谷歌提出了将 Transformer 引入到图像处理领域的 ViT,[6]这之后相关研究才呈井喷之势出现。同样借鉴于 GPT-3 这类语言模型，VQGAN[7]、DALL-E[8]、Parti[9]、StyleSwin[10] 也都是十分有意义的工作。

然而，自回归模型依然存在一些问题。因为自回归模型只能按一个一个 token 地去生成，其每推理一次必须等上一次图片完全推理完才能继续，这导致了其推理速度较慢。同时，按像素地扫描推理使得生成结果存在方向上的偏差。另外，因为存在犯错的可能，自回归模型会将误差累积下来，最终得到一个有较大偏差的结果。

第二节　典型范例

一、VQGAN

VQGAN 模型是个可以在多任务上实现高性能表现的视觉生成范式，绝大多数图像生成任务它都可以做到，该工作最大的亮点在于其可以生成百万像素级别的图片。VQGAN 的论文[7]直译过来是“驯服 Transformer 模型以实现高清图像合成”，可以看出该方法是在用 Transformer 生成图像。而其名称中的 GAN 则代表其使用了两阶段的图像生成方法：

- 训练时，先训练一个图像压缩模型 (包括编码器和解码器两个子模型)，再

训练一个生成压缩图像的模型。

- 生成时，先用第二个模型生成出一个压缩图像，再用第一个模型复原成真实图像。

上述第一个图像压缩模型就叫做 VQGAN，第二个压缩图像生成模型是一个基于 Transformer 的模型。这样的设计乍看起来比较复杂，实际上却大有玄机。

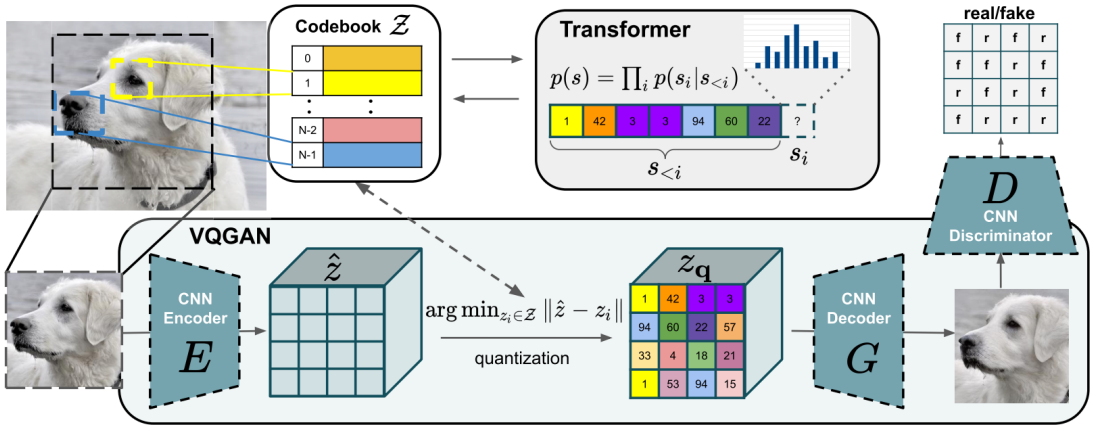

图 3.1　VQGAN 整体结构示意图[7]

在这篇工作发表的时候，Transformer 已经在文本生成领域大展身手，也在视觉任务中开始崭露头角。相比于擅长捕捉局部特征的 CNN，Transformer 的优势在于它能更好地融合图像的全局信息。可是，Transformer 的自注意力操作开销太大，只能生成一些分辨率较低的图像。因此，作者认为，可以综合 CNN 和 Transformer 的优势，先用基于 CNN 的 VQGAN 把图像压缩成一个尺寸更小、信息更丰富的小图像，再用 Transformer 来生成小图像。

VQGAN 中的 VQ 代表 Vector Quantised，来自于 VQVAE[11]这篇工作，VQGAN 参考了后者的 codebook 思想，使用了 Transformer 替代了其中的 pixelCNN，并加入了 PatchGAN 的判别器用于对抗损失。

由于 Transformer 自注意力机制的较大计算开销，作者使用的是 16×16 的压缩图像，而实验发现压缩倍数 $f = 16$ 时表现较好，也就是一次最大生成 256×256 大小的图像，这还不能称得上高清。为了生成更大的图像，作者采用了一种称作滑动窗口的方法。先训练好一个可以生成 256×256 大小图像的模型，再将待生成图像按 16×16 像素的图块作划分，每次要生成一个图块时，只根据该图块周围 16×16 图块来进行，这样大小也就控制在了 256×256 像素，又因为 Transformer 可以保证只根据已生成部分进行生成，所以不需要考虑未生成部分对待生成部分的影响。

二、DALL-E

OpenAI 的 DALL-E 模型首次展现了惊人的概念组合的图像生成能力，其可以把不同的概念很好地拼接在一起。

利用自回归模型处理图片的时候存在一个问题，即如果直接把像素拉成序列当成 image token 来处理，在图片分辨率过高时，一方面会占用过多的内存，另一方面目标函数会倾向于建模短程的像素间的关系因而会学到更多的高频细节，而非更能被人辨认的低频结构。为了解决这一问题，DALL-E 文中采用了两个阶段的方法，先用 dVAE (discrete Variational AutoEncoder) 把 256×256 的图片压缩成 32×32 的 image tokens，然后用 BPE (Byte Pair Encoding) 算法把 text 编码成 256 大小的 text tokens，接着再把 text tokens 和 image tokens 拼接起来得到 $256 + 1024$ 大小的向量，最后就可以用自回归的方式进行训练了。

上述过程可以表示为学习分布 $p_{\theta,\psi}(x, y, z) = p_\theta(x|y, z)p_\psi(y, z)$，其中 x 是图像，y 是文本，z 是 dVAE 编码后的 image tokens，p_θ 是 dVAE decoder 输出的 RGB 图片的分布，p_ψ 是 Transformer 学到的 text 和 image 的联合分布，从而有如式 (3.1)的 Evidence Lower Bound (ELB)，其中 q_ϕ 是 dVAE 编码后的 image tokens 的分布，学习目标是最大化 ELB。

$$\ln p_{\theta,\psi}(x, y) \geq \mathop{\mathbb{E}}_{z\sim q_\phi(z|x)} \left(\ln p_\theta(x|y, z) - \beta D_{\mathrm{KL}}(q_\phi(y, z|x), p_\psi(y, z))\right). \tag{3.1}$$

三、Parti

谷歌提出的 Parti (Pathways Autoregressive Text-to-Image) 证明了语言模型领域里的规模定律 (Scaling Law)，即规模越大效果越好，在自回归模型引入图像生成领域后依然适用。如图 3.2，随着参数从 350M 到 20B 增大，图像质量稳步地提升。

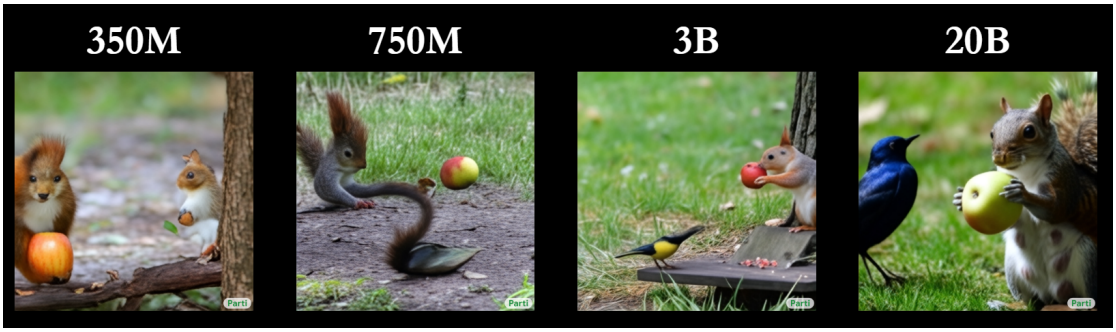

图 3.2　不同参数情况下 Parti 生成图像的情况[9]

注：使用的 prompt 是 A squirrel gives an apple to a bird

Parti 将文本到图像生成视为一个序列到序列建模问题，类似于机器翻译——这使得它能够从大型语言模型的进步中受益，特别是在通过增加数据和模型大小解锁功能方面。在这种情况下，目标输出是另一个语言中的文本 token 序列而不是图像 token 序列。Parti 使用强大的 ViT-VQGAN 将图像编码为离散 token 序列，并利用其重建这种图像 token 序列为高质量、视觉多样性的图像。

四、StyleSwin

StyleSwin 由微软亚洲研究院提出，以 SwinTransformer[12] 为基本框架，结合了 StyleGAN2 和 SwinTransformer 的主要特点。StyleSwin 使用了纯 Transformer 的生成对抗网络来生成高分辨率图像。

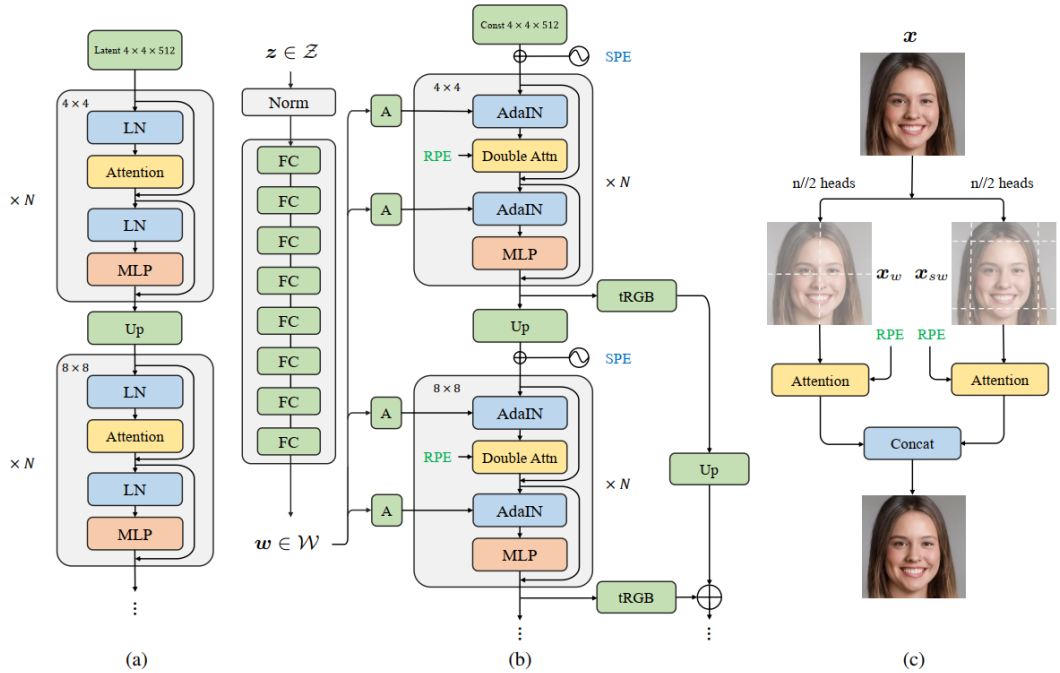

图 3.3　StyleSwin 结构示意图[10]

图 3.3 (b) 代表了 StyleSwin 的结构，输入的变量 $\boldsymbol{z} \in \mathcal{Z}$ 经过八个全连接层映射成 $\boldsymbol{w} \in \mathcal{W}$，这样减少了特征之间的相关性完成解耦，之后 $\boldsymbol{w}$ 经过可学习的放射变换 A 分别送入右侧分支进行训练。右侧分支首先输入一个 $4 \times 4 \times 512$ 大小的常量，经过 SPE (sinusoidal position encoding) 后进入 generator block，输入到 AdaIN (Adaptive Instance Normalization) 和残差连接，然后注入来自 $\boldsymbol{w}$ 的样式风格，再输入到 Double Attn 层，然后特征图进入下一层 AdaIN 和残差连接，又进行样式风格注入，经过 MLP 层输出。而此处的输出经上采样作 SPE 后作为下一个 block 的输入，这样一直叠加下去。同时每个 block 的结果都会输入到 tRGB 层中输出一张 RGB 图像累加到历史的累加 RGB 图像上，最终得到高清的图像。

第四章　扩散模型

第一节　概述

扩散模型 (Diffusion Model)[13] 理论来源于物理学中的扩散过程，于 2021 年被证明已经超越了 GAN,[14] 并且在诸多应用领域都有出色的表现，如计算机视觉、NLP、波形信号处理、多模态建模等等。此外，扩散模型与其他研究领域有着密切的联系。

扩散模型中的生成过程可以被视为一个随机扩散过程，该过程通过在空间中添加噪声逐渐破坏输入图像的结构，然后在噪声上逐渐重建图像的结构。最早的扩散模型是基于马尔可夫过程的，其核心思想是将图像看作一个状态转移矩阵，并通过在状态之间进行转移来生成图像。随着深度学习的发展，研究人员将扩散模型与神经网络相结合，提出了深度扩散模型，从而显著提高了扩散模型的生成能力。

然而，扩散模型也有其缺陷，它的采样速度慢，通常需要数千个评估步骤才能抽取一个样本；它的最大似然估计无法和基于似然的模型相比；它泛化到各种数据类型的能力较差。如今很多研究已经从实际应用的角度解决上述限制做出了许多努力，或从理论角度对模型能力进行了分析。

第二节　原理

当前大部分的扩散模型都可以追溯到 2020 年的工作 DDPM (Denoising Diffusion Probabilistic Models)[15]，其包含前向扩散过程和反向生成过程，前向扩散过程是对一张图像逐渐添加高斯噪声直至变成随机噪声，而反向生成过程是去噪声过程，从一个随机噪声开始逐渐去噪声直至生成一张图像。

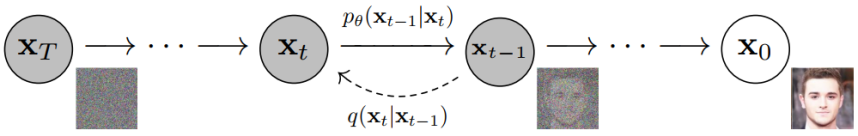

图 4.1　前向扩散过程示意图[15]

首先，前向扩散过程指的是对数据逐渐增加高斯噪声直至数据变成随机噪声的过程。对于原始数据 $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$，总共 T 步的扩散过程的每一步都是对上

一步得到的数据 $\boldsymbol{x}_{t-1}$ 按如下方式增加高斯噪声

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\boldsymbol{I}). \tag{4.1}$$

这里 $\{\beta_t\}_{t=1}^T$ 为每一步所采用的方差，它介于 0 1 之间，通常越后面的步骤会采用更大的方差，即满足 $\beta_1 < \beta_2 < \cdots < \beta_T$。在一个设计好的方差下，如果扩散步数 T 足够大，那么最终得到的 $\boldsymbol{x}_T$ 就完全丢失了原始数据而变成了一个随机噪声。扩散过程的每一步都生成一个带噪声的数据 $\boldsymbol{x}_t$，整个扩散过程也就是一个马尔卡夫链：

$$q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}). \tag{4.2}$$

如果利用原始数据 $\boldsymbol{x}_0$ 对第 t 步的 $\boldsymbol{x}_t$ 进行采样，则可以得到

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I}). \tag{4.3}$$

若 $\bar{\alpha_T}$ 接近于 0，则可以保证得到 $\boldsymbol{x}_T$ 近似为随机噪声。

算法 4.1　前向扩散过程算法

repeat
  $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$;
  t ~Uniform($\{1, \ldots, T\}$);
  $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \boldsymbol{I})$;
  Take gradient descent step on

$$\nabla_\theta ||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)||^2$$

until converged;

反向过程就是一个去噪的过程，如果我们知道反向过程的每一步的真实分布 $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$，那么从一个随机噪音 $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ 开始，逐渐去噪就能生成一个真实的样本，所以反向过程也就是生成数据的过程。

这里，我们将反向过程也定义为一个马尔卡夫链，只不过它由一系列用神经网络参数化的高斯分布组成：

$$p_\theta(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \tag{4.4}$$

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)) \tag{4.5}$$

这里 $p(\boldsymbol{x}_T) = \mathcal{N}(\boldsymbol{x}_T; \boldsymbol{0}, \boldsymbol{I})$，而 $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ 为参数化的高斯分布，它们的均值和方差由训练的网络 $\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t)$ 和 $\boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)$ 给出。实际上，扩散模型就是要得到这些训练好的网络，因为它们构成了最终的生成模型。虽然分布 $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ 是不可

直接处理的，但是加上条件 $\boldsymbol{x}_0$ 的后验分布 $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ 却是可处理的，这里待定系数

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\beta}_t \boldsymbol{I}) \tag{4.6}$$

后得到 $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$，$\tilde{\boldsymbol{\mu}}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_0$.

算法 4.2　反向生成过程算法

$\boldsymbol{x}_T \sim N(\boldsymbol{0}, \boldsymbol{I})$;
for $t = T, \ldots, 1$ **do**
　$\boldsymbol{z} \sim N(\boldsymbol{0}, \boldsymbol{I})$ if $t > 1$, else $\boldsymbol{z} = \boldsymbol{0}$;
　$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right) + \sigma_t \boldsymbol{z}$
end
return $\boldsymbol{x}_0$

第三节　变种

一、Glide

2021 年，OpenAI 继 DALL-E 之后，又发布了文本图像合成模型 GLIDE。[16] 除了能够合成更加逼真的图像，GLIDE 还加入了图像编辑功能，可以直接通过文本 prompt 对图像进行修改，包括插入新对象、阴影和反射效果。此外 GLIDE 对于零样本 (zero-shot) 生成和修复等复杂场景中的泛化能力也非常强。

GLIDE 主要研究的方向是指导扩散模型到文本 prompt 的方法。一种是无分类器引导的方式，使用无分类器引导技术允许模型在引导训练过程中充分利用自己已有的知识，而无需大量依赖其他的预训练分类模型，因此拥有更好的泛化能力和处理复杂场景的能力。另一种是 CLIP 引导的方式。CLIP 模型由独立的图像编码器 $f(x)$ 和文本编码器 $g(c)$ 构成，在模型训练阶段，通过从大规模数据集中选取图像-文本对 (x, c) 来优化对比交叉熵损失函数。因此 CLIP 模型可以输出任意一幅图像和一段文字的匹配分数，这可以用来作为引导模型嵌入到 GLIDE 中。为了将其应用到扩散模型中，作者使用图像的点积分数和文本编码相对于图像的梯度值来扰动反向过程，公式如式 (4.7)所示。

$$\hat{\mu_\theta}(x_t|c) = \mu_\theta(x_t|c) + s\sum_\theta(x_t|c)\nabla_{x_t}(f(x_t), g(c)) \tag{4.7}$$

二、DALL-E 2

DALL-E 2[17]的整体架构说起来非常简单，它用两阶段来生成图像。第一阶段用 prior 模型从文本生成图像特征，第二阶段用 decoder 生成图像。prior 模型有两种选择，自回归模型和扩散模型。作者是用了 sequence (text feature) 预

测 sequence (image feature)，没有要求前后尺寸不变。作者使用了 Transformer 将 CLIP 的文本编码、加入噪声的 CLIP 的图像编码、扩散时间步的 time embedding 等输入，去预测未加噪声的 CLIP 图像编码。

三、Imagen

谷歌的 Imagen[18]结构也不算复杂，它的重点在于把 NLP 中很强大的语言模型拿过来用作 frozen text encoder，这样就可以保证 text embedding 中不损失语义信息。之后再把 text embedding 输入到生成模型中，给模型信息，让模型基于这个信息去生成图像。第一步先成低分辨率的图像，然后再串联 2 个 super-resolution 网络，这两个网络的输入是前面的低质量图像和 text embedding，最终就可以输出高质量的图像。

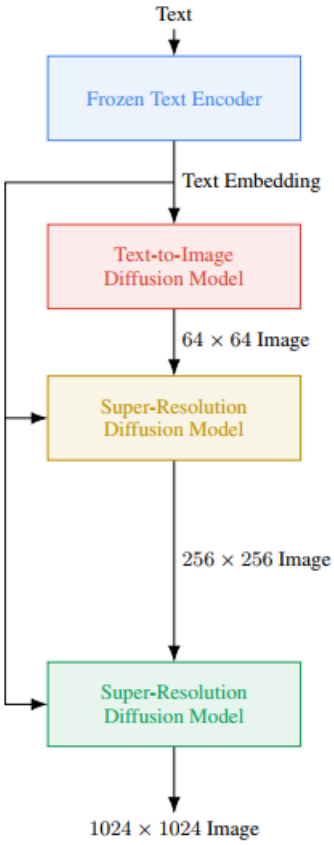

图 4.2　Imagen 结构示意图[18]

四、Stable Diffusion

Stable Diffusion[19]旨在解决速度问题。它使用的是一种隐空间扩散 (latent diffusion) 的模型。它不是在高维图像空间中操作，而是首先将图像压缩到隐空间 (latent space) 中。相比起原像素空间，隐空间要小得多，因此可以处理较少的数据，因此可以大幅提高处理速度。

Stable Diffusion 使用也是使用的 VAE 技术来实现图像隐空间压缩。这由编码器和解码器两部分组成。编码器将图像压缩为隐空间中的低维表示，解码器从隐空间恢复图像。因此，在训练过程中，它不会生成噪声图像，而是在隐空间中生成随机张量 (隐噪声)，不是用噪声破坏图像，而是用隐噪声破坏图像在隐空间中的表示。

第五章　NeRF

第一节　概述

NeRF (Neural Radiance Fields)[20]是一种用于生成高质量三维图像的方法，它使用深度神经网络估计场景中每个点的辐射率分布，从而生成逼真的三维图像。与传统的三维渲染方法不同，NeRF 不需要使用人工建模，而是通过训练神经网络来学习场景的辐射率分布。这种方法可以生成高质量的三维图像，具有较好的视觉效果和真实感。NeRF 已经被广泛应用于计算机图形学、计算机视觉和虚拟现实等领域。

NeRF 已经在三维视觉和计算机图形学领域取得了很好的效果，但仍然存在一些缺陷和局限性。首先 NeRF 需要大量的计算资源和时间来进行训练，特别是对于大型场景和高分辨率图像，因为需要对每个像素进行多次采样和计算，训练时间可能需要数小时甚至数天。在某些情况下，NeRF 渲染结果可能与真实场景存在偏差，例如对于某些材料的反射和折射率估计不准确。NeRF 的训练和渲染过程都是基于点云的，因此难以扩展到更大的场景和更复杂的物体，而且对于某些形状复杂的物体，点云的表示可能不够准确。NeRF 的训练过程是离线进行的，无法进行实时的交互和渲染，因此无法满足某些实时应用的需求。

第二节　原理

在 NeRF 中，一个连续的场景被表示为一个 5D 向量值函数，其输入是 3D 位置 $\boldsymbol{x} = (x, y, z)$ 和 2D 观察方向 $\boldsymbol{d} = (\theta, \phi)$，输出是发射颜色 $\boldsymbol{c} = (r, g, b)$ 和体积密度 σ，可以表示为 $F_\Theta : (\boldsymbol{x}, \boldsymbol{d}) \rightarrow (\boldsymbol{c}, \sigma)$。

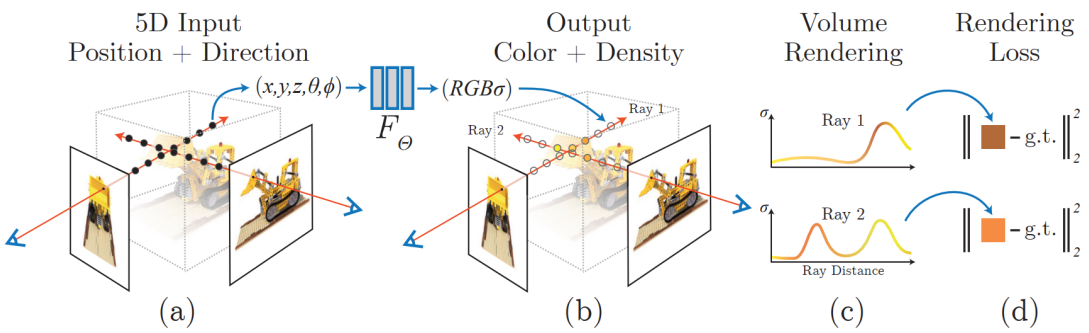

图 5.1　NeRF 结构示意图[20]

以上函数得到的是一个 3D 空间点的颜色和密度信息，但当用一个相机去对这个场景成像时，所得到的 2D 图像上的一个像素实际上对应了一条从相机出发的射线上的所有连续空间点。我们需要通过渲染算法从这条射线上的所有点得到这条射线的最终渲染颜色。同时，为了保证网络可以训练，NeRF 中需要采用可微的渲染方法。以下我们对其核心——体渲染 (Volume Rendering) 进行介绍。

体密度 $\sigma(\boldsymbol{x})$ 可以解释为射线在位置 $\boldsymbol{x}$ 终止于一个无限小的粒子的可微概率。如果计参数 t 近端和远端边界为 t_n 和 t_f，$\boldsymbol{o}$ 为射线原点，则摄像机射线 $\boldsymbol{r}(t)=\boldsymbol{o}+t\boldsymbol{d}$ 的期望颜色 $C(\boldsymbol{r})$ 为

$$C(\boldsymbol{r})=\int_{t_n}^{t_f} T(t)\sigma(\boldsymbol{r}(t))c(\boldsymbol{r}(t),\boldsymbol{d})\,\mathrm{d}t \tag{5.1}$$

这里的 $T(t)$ 代表沿着射线从 t_n 到 t 上的累积透射率，也就是射线从 t_n 到 t 而不撞击其他粒子的概率，在此用如下公式表示。

$$T(t)=\exp\left(-\int_{t_n}^{t}\sigma(\boldsymbol{r}(s))\,\mathrm{d}s\right) \tag{5.2}$$

式 (5.1)中的 $C(\boldsymbol{r})$ 只能通过近似计算，一个直观且很常用的思路是，在需要求积的区域均匀地采样 N 个点进行近似计算。但作者提出，这样的方式会导致 MLP 只需要学习一系列离散点的信息，最终会限制 NeRF 的分辨率，使得最终生成的结果不够清晰。作为替代，作者提出了一种简单有效的方法，如上图所示，首先将射线需要积分的区域分为 N 份，然后在每一个小区域中进行均匀随机采样。这样的方式能够在只采样离散点的前提下，保证采样位置的连续性。第 i 个采样点可以表示为

$$t_i\sim\mathcal{U}\left[t_n+\frac{i-1}{N}(t_f-t_n),t_n+\frac{i}{N}(t_f-t_n)\right] \tag{5.3}$$

这样就可以将原本的积分转为求和形式

$$\hat{C}(\boldsymbol{r})=\sum_{i=1}^{N}T_i(1-\exp(-\sigma_i\delta_i))\boldsymbol{c}_i \tag{5.4}$$

其中 $\delta_i=t_{i+1}-t_i$ 是邻近两个采样点之间的距离，且有 $T_i=\exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right)$。这样就可以用 NeRF 来从各种角度生成图像。

第六章　总结

3D 生成模型是深度学习领域的一个重要研究方向，其目标是从 2D 图像或其他 2D 数据中生成 3D 形状、场景或物体。近年来，GAN、自回归模型、扩散模型、NeRF，各种新式方法层出不穷，生成图像的质量越来越好，生成速度越来越快，场景也越来越复杂。当然，这不免引起人们对生成模型伦理上的思考，如对相应成果在政治或色情领域的滥用等等。当然，现有的成果依然和真实的自然界有一定差距，依然存在进步空间。相信未来的生成模型结构将更加复杂，生成的 3D 形状将更加真实，也将出现更多的无监督学习模型，减少对标注数据的依赖。结合多模态数据 (如视频、语音等) 的 3D 生成模型也是一个不错的发展方向，从而可以更好地模拟真实世界中的复杂场景。

参 考 文 献

[1] Goodfellow IJ, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks. *CoRR*, 2014, abs/1406.2661. http://arxiv.org/abs/1406.2661.

[2] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. Bengio Y, LeCun Y. *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. 2016. http://arxiv.org/abs/1511.06434.

[3] Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019: 4401-4410. https://arxiv.org/abs/1812.04948.

[4] Kang M, Zhu J, Zhang R, et al. Scaling up gans for text-to-image synthesis. *CoRR*, 2023, abs/2303.05511. https://arxiv.org/abs/2303.05511.

[5] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. Guyon I, von Luxburg U, Bengio S, et al. *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 2017: 5998-6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[6] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. https://openreview.net/forum?id=YicbFdNTTy.

[7] Esser P, Rombach R, Ommer B. Taming transformers for high-resolution image synthesis. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021: 12873-12883. https://arxiv.org/abs/2012.09841.

[8] Ramesh A, Pavlov M, Goh G, et al. Zero-shot text-to-image generation. Meila M, Zhang T. *Proceedings of Machine Learning Research: Vol. 139 Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021: 8821-

8831. https://proceedings.mlr.press/v139/ramesh21a.html.

[9] Yu J, Xu Y, Koh JY, et al. Scaling autoregressive models for content-rich text-to-image generation. *Trans. Mach. Learn. Res.*, 2022, 2022. https://arxiv.org/abs/2206.10789.

[10] Zhang B, Gu S, Zhang B, et al. Styleswin: Transformer-based GAN for high-resolution image generation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022: 11294-11304. https://doi.org/10.1109/CVPR52688.2022.01102.

[11] van den Oord A, Vinyals O, Kavukcuoglu K. Neural discrete representation learning. Guyon I, von Luxburg U, Bengio S, et al. *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 2017: 6306-6315. https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

[12] Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows. *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 2021: 9992-10002. https://doi.org/10.1109/ICCV48922.2021.00986.

[13] Yang L, Zhang Z, Song Y, et al. Diffusion models: A comprehensive survey of methods and applications. *CoRR*, 2022, abs/2209.00796. http://arxiv.org/abs/2209.00796.

[14] Dhariwal P, Nichol AQ. Diffusion models beat gans on image synthesis. Ranzato M, Beygelzimer A, Dauphin YN, et al. *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. 2021: 8780-8794. https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

[15] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. Larochelle H, Ranzato M, Hadsell R, et al. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. 2020. https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

[16] Nichol AQ, Dhariwal P, Ramesh A, et al. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. Chaudhuri K, Jegelka S, Song L, et al. *Proceedings of Machine Learning Research: Vol. 162 International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*. PMLR, 2022: 16784-16804. https://proceedings.mlr.press/v162/nichol22a.html.

[17] Ramesh A, Dhariwal P, Nichol A, et al. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, 2022, abs/2204.06125. http://arxiv.org/abs/2204.06125.

[18] Saharia C, Chan W, Saxena S, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*. 2022. https://arxiv.org/abs/2205.11487.

[19] Rombach R, Blattmann A, Lorenz D, et al. High-resolution image synthesis with latent diffusion models. *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022: 10674-10685. https://doi.org/10.1109/CVPR52688.2022.01042.

[20] Mildenhall B, Srinivasan PP, Tancik M, et al. Nerf: Representing scenes as neural radiance fields for view synthesis. Vedaldi A, Bischof H, Brox T, et al. *Lecture Notes in Computer Science: Vol. 12346 Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*. Springer, 2020: 405-421. https://arxiv.org/abs/2003.08934.