掩蔽扩散Transformer是一种强大的图像合成器

Shanghua Gao[1,2 *] Pan Zhou[2 †] Ming-Ming Cheng[1 †] Shuicheng Yan[2]
[1]Nankai University [2]Sea AI Lab
{shanghuagao,shuicheng.yan}@gmail.com zhoupan@sea.com cmm@nankai.edu.cn

Abstract

尽管扩散概率模型（*DPMs*）在图像合成方面取得了成功，但我们观察到它通常缺乏上下文推理能力，无法学习图像中物体部分之间的关系，从而导致学习过程缓慢。为了解决这个问题，我们提出了一种掩蔽扩散*Transformer*（*MDT*），引入了一个掩蔽隐空间建模方案，明确增强了*DPMs*在图像中物体语义部分之间上下文关系学习的能力。在训练过程中，*MDT*在隐空间上操作，以掩蔽某些标记。然后，设计了一个非对称的掩蔽扩散*Transformer*，从未掩蔽的标记中预测被掩蔽的标记，同时保持扩散生成过程。我们的*MDT*能够从不完整的上下文输入中重建图像的完整信息，从而使其能够学习图像标记之间的关联关系。实验结果表明，*MDT*在图像合成方面取得了优越的性能，例如在*ImageNet*数据集上获得了新的*SoTA*（*State-of-the-Art*）*FID*分数，并且比之前的*SoTA DiT*具有约*3*倍的更快学习速度。源代码已在以下链接发布：*https://github.com/sail-sg/MDT*。

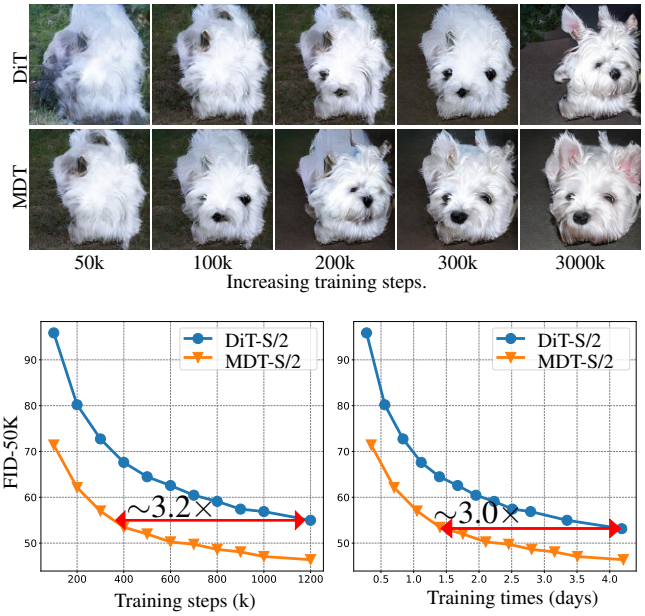

图 1. 上半部分：展示了MDT/DiT [32]随训练步骤的可视化示例。下半部分：对比了DiT和MDT在使用8个A100 GPU进行训练的过程中的学习进展。MDT的学习速度大约是DiT的3倍，同时获得了更优越的FID分数。

1. 介绍

扩散概率模型（DPMs） [10, 35] 已经成为近期图像级生成模型方面的先驱，在很多情况下超越了之前的最先进生成对抗网络（GANs [4,16,34,51]。此外，DPMs还在许多其他应用中展现了成功，包括文本到图像生成 [35] 和语音生成 [22]。DPMs采用反向的随机微分方程（SDE）来逐步将高斯噪声通过多个时间步骤逐渐映射成样本，每个步骤对应于一次网络评估。在实际操作中，由于SDE需要数千个时间步骤才能收敛，生成一个样本是非常耗时的。为了解决这个问题，各种生成样本策略 [19,29,38]已经被提出，以加速推理速度。然而，改善DPMs的训练速度尚未得到充分探索，但却备受期望。DPMs的训练也不可避免地需要大量的时间步骤来确保SDE的收敛，使其在计算上非常昂贵，尤其是在当前使用大规模模型 [10, 32] 和数据 [8, 13, 41] 来提高生成性

*This work was done while S. Gao was a research intern at Sea AI Lab.
†Pan Zhou and Ming-Ming Cheng are joint corresponding authors.

能的时代。

在这项工作中，我们首先观察到DPMs通常难以学习图像中物体部分之间的关联关系，导致训练过程缓慢。具体而言，在图 Fig. 1 所示的例子中，使用DiT [32]作为骨干网络的经典DPM，DDPM [19]，在训练的第5万个步骤中已经学会了狗的整体形状，然后在第20万个步骤中逐渐学会了其中的一个眼睛和嘴巴，但仍然错过了另一个眼睛。而且，甚至在训练的前30万个步骤中，两只耳朵的相对位置也不是非常准确。这个学习过程表明，DPMs独立地学习每个语义部分，未能学习到语义部分之间的关联关系。该现象的原因在于，DPMs通过最小化每像素预测损失来最大化真实数据的对数概率，而忽视了图像中物体部分之间的关联关系，从而导致了它们的学习进展缓慢。

受到上述观察的启发，我们提出了一种有效的掩蔽扩散Transformer（MDT），以提高DPMs的训练效率。MDT提出了一种专门设计用于基于Transformer的DPMs的掩蔽隐空间建模方案，以明确增强上下文学习能力，并改善图像中语义部分之间的关联关系学习。具体来说，类似于 [32, 35]，MDT在隐空间中执行扩散过程，以节省计算成本。MDT掩蔽了某些图像标记，并设计了一个非对称的扩散Transformer结构，以扩散生成的方式从未掩蔽的标记中预测被掩蔽的标记。为此，这个非对称结构包含一个编码器、一个侧插值器和一个解码器。编码器和解码器是通过修改DiT [32]中的Transformer块，在其中插入全局和局部的标记位置信息得到的，从而有助于预测被遮蔽的标记。编码器在训练期间仅处理未掩蔽的标记，而在推理期间处理所有标记，因为推理时没有掩膜。因此，为了确保解码器始终在训练预测或推理生成时处理所有标记，由一个小型网络实现的侧插值器在训练期间利用编码器的输出预测被掩蔽的标记，在推理中，侧插值器则被移除。

通过这种掩蔽隐空间建模方案，我们的MDT能够从上下文不完整的输入中重建图像的完整信息，学习图像中语义部分之间的关联关系。如图 Fig. 1 所示，MDT通常在几乎同一的训练步骤中生成了狗

图 2. MDT-XL/2生成的图像的可视化结果。

的两只眼睛（和两只耳朵），表明它通过利用掩蔽隐空间建模方案正确学习了图像的关联语义。相比之下，DiT [32]难以轻松合成具有正确语义关系的狗。这个比较显示了MDT相比于DiT优越的关系建模和更快的学习能力。实验结果表明，MDT在图像合成任务上取得了优越的性能，并在ImageNet数据集上的类条件图像合成任务中创造了新的SoTA，如图 Fig. 2 和表 Tab. 1 所示。MDT在训练过程中的学习进展也比SoTA的DPMs（即DIT）快了约3倍，如图 Fig. 1 和表 Tab. 2 所示。我们希望我们的工作能够激发更多关于如何通过统一的表示学习来加速扩散训练过程的研究。

主要贡献总结如下：

- 通过引入一种高效的掩蔽隐空间建模机制，我们提出了一种掩蔽扩散Transformer方法，它首次显著增强了DPMs的上下文学习能力。
- 实验结果表明，我们的方法更好地合成了图像，同时比SOTA使用了更少的训练时间。

2. 相关工作

2.1. 扩散概率模型

扩散概率模型（DPM） [10, 19]，也被称为基

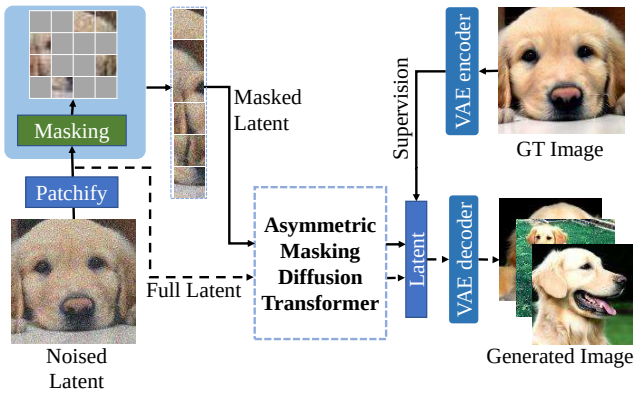

图 3. 掩蔽扩散Transformer（MDT）的总体框架。实线/虚线表示每个时间步的训练/推理过程。掩蔽和侧插值器仅在训练期间使用，并在推理期间移除。

于分数的模型 [45, 46]，是一种有竞争力的图像合成方法。 DPMs首先使用不断演化的随机微分方程（SDE）逐步将高斯噪声添加到真实数据中，将复杂的数据分布转化为高斯分布。然后，它采用反向的SDE，经过多个步骤，逐步将高斯噪声映射成为样本。在每个生成样本时间步骤中，一个也被称为分数函数 [47]的网络被用于沿着对数概率的梯度生成样本。扩散模型的迭代性质可能导致高训练和推断成本。为了降低推理成本，高效的采样策略 [19, 21, 29, 38, 43]、隐空间扩散 [35, 48]以及多分辨率级联生成 [20]已经被提出。此外，一些训练方案 [2, 11]被引入来改进扩散模型的训练，例如近似最大似然训练 [25, 31, 44]，训练损失加权 [23, 24]。与这些优化扩散训练过程的方法不同，我们发现扩散模型在上下文建模能力方面存在不足。为了解决这个问题，我们提出了掩蔽隐空间建模方案作为一种补充方法，以增强扩散模型的上下文表示能力，这与现有的扩散训练方案是互不相关的。

2.2. 扩散模型的网络结构

通过空间自注意力 [39, 49]和组归一化 [52]进行增强的类似UNet的网络结构 [36]，被首先用于扩散模型 [19]。在 [10]中提出了一些设计改进，例如增加更多的注意力头、BigGAN [4]残差块和自适应组归一化，以进一步增强UNet的生成能力。最近，由于Transformer网络的广泛适用性，一些研究尝试将视觉Transformer（ViT）结构用于扩散模型 [1, 32, 53]。 GenViT [53]证明了ViT可以进行图像生成，但性能较UNet略逊一筹。U-ViT [1]通过添加长跳连接和卷积层改进了ViT，在性能上与UNet相媲美。DiT [32]验证了ViT在大模型尺寸和特征分辨率上的扩展能力。我们的MDT与这些扩散网络互不相关，因为它专注于上下文表示学习。此外，MDT中的位置感知设计揭示了掩蔽隐空间建模方案从更强的扩散网络中获益。我们将进一步探索如何在MDT中释放这些网络的潜力。

2.3. 掩蔽建模

掩蔽建模在识别学习 [9, 14, 17] 和生成建模 [7, 33] 领域都被证明是有效的。在自然语言处理（NLP）领域，掩蔽建模首先被引入用于表示预训练 [9, 33] 和语言生成 [5]。随后，它也被证明在视觉识别 [3] 和生成 [7, 15, 54] 任务中是可行的。在视觉识别中，利用掩蔽建模的预训练方案可以实现良好的表示质量 [55]、可扩展性 [17] 和更快的收敛速度 [14]。在生成建模中，继NLP中的双向生成建模之后，MaskGIT [7] 和 MUSE [6] 使用了掩蔽生成Transformer来预测随机遮蔽的图像标记以进行图像生成。类似地，VQ-Diffusion [15] 提出了一种遮蔽替换扩散策略来生成图像。相比之下，我们的MDT旨在通过掩蔽隐空间建模来增强去噪扩散变换器 [32] 的上下文表示。通过在推理过程中保持扩散过程，它保留了去噪扩散模型的细节细化能力。为了确保MDT中的掩蔽隐空间建模专注于表示学习而不是重建，我们在掩蔽建模训练中提出了一个不对称的结构。额外的好处是，与掩蔽生成模型相比，它使训练成本更低，因为它在训练中跳过了掩蔽的区域，而不是用掩蔽标记替掩蔽的输入区域。

3. 掩蔽扩散Transformer

对扩散概率模型的重新审视 对于扩散概率模型 [10, 42]，例如DDPM [19]和DDIM [43]，训练包括前向加噪过程和反向去噪过程。在前向加噪过程中，通过离散的SDE方程 $q(x_t, x_0) = \sqrt{\bar{\alpha}t}x_0 + \epsilon\sqrt{1-\bar{\alpha}t}$，将高斯噪声$\epsilon \sim \mathcal{N}(0, \mathbf{I})$逐步添加到真实样本$x0$，其中$\bar{\alpha}t$表示噪声大小。如果时间步长$t$很大，$x_t$将成

为高斯噪声。类似地，反向去噪过程是一个逐步将高斯噪声映射成为样本的离散SDE。在每个时间步长，给定x_t，通过网络可以预测下一个反向步骤 $p\theta(xt-1|x_t) = \mathcal{N}(x_{t-1};\mu_\theta(x_t,t),\Sigma_\theta(x_t,t))$。该网络通过优化$p_\theta(x_0)$的变分下界 L_{vlb} 进行训练 [42]，其中 $L_{\text{vlb}} = -\log p_\theta(x_0|x_1) + \sum_t D_{KL}(q(x_{t-1}|x_t,x_0)||p_\theta(x_{t-1}|x_t))$。

根据 [31,32], 网络被通过优化对数概率$p_\theta(x_0)$的变分下界L_{vlb}而被训练 [42]。在推理过程中，它可以采样一个高斯噪声，然后逐步地反向映射到一个样本 x_0。

与 [31,32]相同，我们训练需要类别标签 c 作为条件的扩散模型，即 $p_\theta(x_{t-1}|x_t,c)$。在我们的实验中，默认使用类条件的图像生成。

3.1. 概述

如Fig. 1所示，用DiT做骨干网络的DPM由于在图像中的语义关联学习缓慢，导致训练收敛较慢。为了缓解这个问题，我们提出了掩蔽扩散Transformer（MDT），引入了一种掩蔽隐空间建模方案，以明确增强上下文学习能力，提高在图像中建立不同语义之间关联的能力。为此，如Fig. 3所示，MDT包括：1）隐空间掩蔽操作，用于在隐空间中对输入图像进行掩蔽。2）一个非对称掩蔽扩散Transformer结构，执行与DPMs相同的基础扩散过程，但输入为掩蔽后的图像。为了减少计算成本，MDT遵循LatentDiffusion [35]，在隐空间而不是原始像素空间中进行生成学习。

在训练阶段，MDT首先使用预训练的VAE编码器 [35]将图像编码为隐空间中的表示。然后，MDT向图像隐空间表示中加入高斯噪声。MDT中的隐空间掩蔽操作随后将产生的带有噪声的隐空间表示划分为一系列标记，并对某些标记进行掩蔽。剩余的未掩蔽标记被馈送到非对称掩蔽扩散Transformer中，它包含编码器、侧插值器和解码器，用于从未掩蔽的标记中预测掩蔽的标记。在推理过程中，MDT使用额外的位置嵌入替换侧插值器。MDT将高斯噪声的隐空间表示作为输入，生成去噪的隐空间表示，然后将其传递给预训练的VAE解码器 [35]进行图像生成。

上述的训练阶段中的掩蔽隐空间建模方案迫使扩散模型从其上下文不完整的输入中重构图像的完整信息。从而，模型被鼓励学习图像隐空间标记之间的关系，特别是图像中语义之间的关联关系。例如，如Fig. 3所示，模型应首先正确理解狗图像中的小图像部分（标记）之间的关联关系。然后，它应该通过使用其他未被掩蔽的标记作为上下文信息，来生成掩蔽的“眼睛”标记。此外，Fig. 1显示，MDT通常会以几乎相似的速度学习生成图像的相关语义，比如几乎在同一训练步骤中生成狗的两只眼睛（两只耳朵）。而DiT [32]（带有Transformer骨干网络的DDPM）开始只学习生成一只眼睛（一只耳朵），然后在大约10万次训练步骤后学习生成另一只眼睛（耳朵）。这证明了MDT在图像语义的相关关系学习方面优于DiT的学习能力。

在接下来的部分中，我们将介绍MDT的两个关键组成部分：1）隐空间掩蔽操作，和2）非对称掩蔽扩散Transformer结构。

3.2. 隐空间掩蔽

在掩蔽扩散Transformer（MDT）中，类似于隐空间扩散模型（LDM） [35]，我们将生成学习从原始像素空间转移到隐空间中，以减少计算成本。接下来，我们将简要回顾一下LDM，然后介绍我们在隐空间输入上的隐空间掩蔽操作。

隐空间扩散模型（LDM）。 LDM使用一个预训练的VAE编码器 $\mathbf{E}$ 来将图像 $v \in \mathbb{R}^{3\times H\times W}$ 编码为隐空间表示 $z = \mathbf{E}(v) \in \mathbb{R}^{c\times h\times w}$。它在前向过程中逐渐向 z 添加噪声，然后在反向过程中对其进行去噪以预测 z。最后，LDM使用一个预训练的VAE解码器 $\mathbf{D}$ 将 z 解码为高分辨率图像 $v = \mathbf{D}(z)$。在训练和推理过程中，VAE编码器和解码器都保持固定。由于 h 和 w 比 H 和 W 较小，因此在低分辨率潜在空间中进行扩散过程比在像素空间中更高效。

隐空间掩蔽操作。 现在我们介绍在隐空间输入上的掩蔽方案。在训练过程中，我们首先向图像的隐空间表示 z 添加高斯噪声。然后，遵循 [32]，我们将带

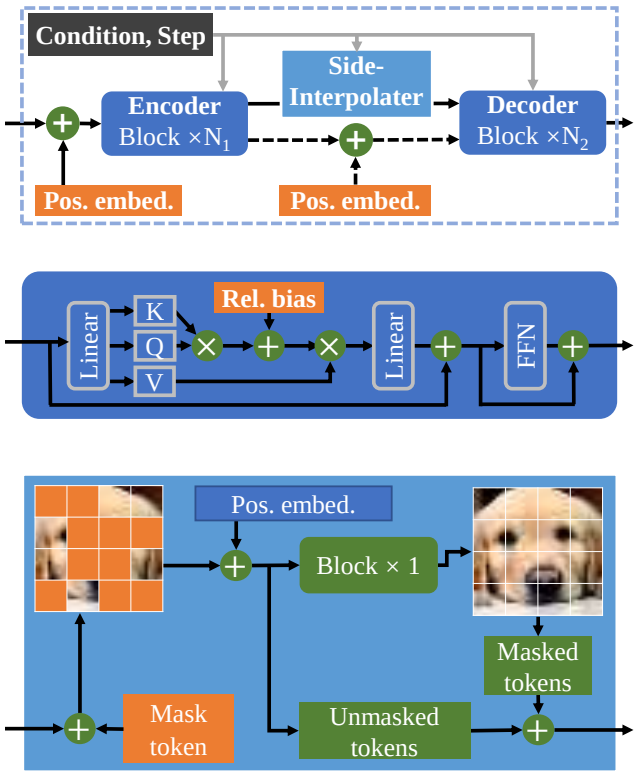

图 4. MDT中的非对称掩蔽扩散Transformer。我们通过添加侧插值器、局部相对位置偏移和可学习的全局位置嵌入，对DiT [32]进行了修改。为了简单起见，我们省略了条件化方案。

有噪声的嵌入 z 划分为一系列大小为 $p \times p$ 的标记，并将它们连接成一个矩阵 $u \in \mathbb{R}^{d \times N}$，其中 d 是通道数，N 是标记数量。接下来，我们随机掩蔽一定比例 ρ 的标记，并将剩余的标记连接成 $\hat{u} \in \mathbb{R}^{d \times \hat{N}}$，其中 $\hat{N} = \rho N$。因此，我们可以建立一个二进制掩码 $M \in \mathbb{R}^{N}$，其中 1（0）表示掩蔽（未掩蔽）的标记。最后，我们将标记 $\hat{u}$ 输入到我们的扩散模型进行处理。我们只使用标记 $\hat{u}$，有两点原因。

1）模型应专注于学习语义，而不是预测掩蔽的标记。如 Sec. 4.3中所示，与像 [3,6,7] 那样用可学习的掩蔽标记替换被掩蔽的标记并处理所有标记相比，它实现了更好的性能；2）与处理所有 N 个标记相比，它节省了训练成本。

3.3. 非对称掩蔽扩散Transformer

我们引入了非对称掩蔽扩散Transformer，用于掩蔽隐空间建模和扩散过程的联合训练。如Fig. 4所示，它由三个组件组成：编码器、侧插值器和解码器，下面将详细描述每个组件。

位置感知编码器和解码器。在MDT中，从未被掩蔽的标记中预测掩蔽的隐空间标记需要所有标记的位置关系。为了增强模型中的位置信息，我们提出了一个位置感知编码器和解码器，有助于学习掩蔽的隐空间标记。具体而言，编码器和解码器通过添加两种类型的标记位置信息来修改标准的DiT块，并且各自包含了 N_1 和 N_2 个定制的块。

首先，如Fig. 4所示，编码器将常规的可学习的全局位置嵌入添加到加噪声的隐空间嵌入输入中。类似地，在训练和推理阶段，解码器也将可学习位置嵌入引入到其输入中，但在两个阶段中采用不同的方法。在训练过程中，侧插值器已经使用了下面介绍的可学习全局位置嵌入，这可以将全局位置信息传递给解码器。在推理过程中，由于侧插值器被丢弃（见下文），解码器显式地将位置嵌入添加到其输入中，以增强位置信息。

其次，如Fig. 4所示，在计算自注意力 [50]的注意分数时，编码器和解码器在每个块的每个头部中添加了局部相对位置偏移 [27]：

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}} + B_r\right)V,$$

其中 Q、K 和 V 分别表示自注意模块中的查询、键和值，d_k 是键的维度，$B_r \in \mathbb{R}^{N \times N}$ 是相对位置偏移，通过第 i 个位置与其他位置之间的相对位置差异选择。B_r 在训练过程中被更新。局部相对位置偏移有助于捕捉标记之间的相对关系，从而促进掩蔽隐空间建模。

编码器接受我们的隐空间掩蔽操作提供的未被掩蔽的噪声隐空间嵌入，并在训练/推理期间将其输出馈送到侧插值器/解码器中。对于解码器，其输入为侧插值器的输出（训练时）或编码器输出和可学习位置嵌入的组合（推理时）。由于在训练过程中，编码器和解码器分别处理未被掩蔽的标记和全部的标记，我们将我们的模型称为“非对称”模型。

侧插值器。 如Fig. 3所示，在训练期间，为了提高效

率和性能，编码器仅处理未被掩蔽的标记 $\hat{u}$。然而，在推理阶段，由于没有掩膜，编码器处理所有标记 u。这意味着在训练和推理过程中，至少在标记数量上，编码器的输出（即解码器的输入）存在很大差异。为了确保解码器始终能够在训练预测或推理生成时处理所有标记，侧插值器由一个小型网络实现，在训练期间从编码器输出中预测掩蔽的标记，并在推理过程中被移除。

在训练阶段，编码器处理未掩蔽的标记，以获取它的输出标记嵌入 $\hat{q} \in \mathbb{R}^{d \times \hat{N}}$。然后，如Fig. 3所示，侧插值器首先使用共享的可学习掩蔽标记填充由第 3.2 节中定义的掩膜 M 所指示的掩蔽位置，同时添加一个可学习的位置嵌入来得到一个嵌入 $q \in \mathbb{R}^{d \times N}$。接下来，我们使用一个基本编码器块处理 q 以预测经插值的嵌入 $\hat{k}$。$\hat{k}$ 中的标记表示预测的标记。最后，我们使用掩蔽的捷径连接将预测 $\hat{k}$ 和 q 结合为 $k = M \cdot q + (1 - M) \cdot \hat{k}$。总之，对于掩蔽的标记，我们使用侧插值器的预测。对于未掩蔽的标记，我们仍然使用 q 中的相应标记。这可以实现以下效果： 1）增强训练和推理阶段之间的一致性；2）消除解码器中的掩蔽重建过程。

由于在推理阶段没有掩膜，侧插值器被一个位置嵌入操作替代，该操作将在训练期间学习到的侧插值器的可学习位置嵌入添加到输入中。这确保解码器始终处理所有标记，并在训练预测或推理生成时使用相同的可学习位置嵌入，从而获得更好的图像生成性能。

3.4. 训练过程

在训练过程中，我们同时将完整的隐空间嵌入 u 和掩蔽的隐空间嵌入 $\hat{u}$ 都输入到扩散模型中。我们观察到，仅使用掩蔽的隐空间嵌入会使模型过于关注掩蔽区域的重建，而忽视了扩散训练。完整/掩蔽的隐空间输入是独立发送到网络的，它们的训练目标都优化了变分下界，就像在 [31,32] 中一样。由于这种非对称的掩蔽结构，使用掩蔽的隐空间嵌入所需的额外成本很小。这也可以从 Fig. 1 中证明，它显示了 MDT 在总的训练小时数方面仍然比之前的 SoTA DiT 实现了大约 3 倍快的学习进度。

Method	Cost(Iter×BS)	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
DCTrans. [30]	-	36.51	-	-	0.36	0.67
VQVAE-2 [34]	-	31.11	-	-	0.36	0.57
VQGAN [12]	-	15.78	78.3	-	-	-
BigGAN-deep [4]	-	6.95	7.36	171.4	0.87	0.28
StyleGAN [40]	-	2.30	4.02	265.12	0.78	0.53
Impr. DDPM [31]	-	12.26	-	-	0.70	0.62
MaskGIT [7]	1387k×256	6.18	-	182.1	0.80	0.51
CDM [20]	-	4.88	-	158.71	-	-
ADM [10]	1980k×256	10.94	6.02	100.98	0.69	0.63
LDM-8 [35]	4800k×64	15.51	-	79.03	0.65	0.63
LDM-4	178k×1200	10.56	-	103.49	0.71	0.62
DiT-XL/2 [32]	7000k×256	9.62	6.85	121.50	0.67	0.67
MDT	2500k×256	7.41	4.95	121.22	0.72	0.64
MDT	3500k×256	6.46	**4.92**	131.70	**0.72**	0.63
MDT	6500k×256	**6.23**	5.23	**143.02**	0.71	0.65
ADM-G [10]	1980k×256	4.59	5.25	186.70	0.82	0.52
ADM-G, U	1980k×256	3.94	6.14	215.84	0.83	0.53
LDM-8-G [35]	4800k×64	7.76	-	209.52	0.84	0.35
LDM-4-G	178k×1200	3.60	-	247.67	0.87	0.48
U-ViT-G [1]	300k×1024	3.40	-	-	-	-
DiT-XL/2-G [32]	7000k×256	2.27	4.60	278.24	0.83	0.57
MDT-G	2500k×256	2.15	4.52	249.27	0.82	0.58
MDT-G	3500k×256	2.02	**4.46**	263.77	0.82	0.60
MDT-G	6500k×256	**1.79**	4.57	**283.01**	0.81	**0.61**

表 1. 在使用 ImageNet 256x256 数据集进行的类条件图像生成任务中，与现有方法进行的比较结果。-G 表示使用无分类器指导的结果。MDT-XL/2模型的结果被给出用于比较。比较的结果是从他们的论文中获取的。

4. 实验

4.1. 实现细节

我们给出MDT的实现细节，包括模型结构、训练细节、和评估指标。

模型架构。 我们遵循 DiT [32] 的设置，来确定 MDT 的扩散Transformer的总块数（即 $N_1 + N_2$）、标记数和通道数。 DiT 表明，在使用更小的区域尺寸时，生成性能更强，因此我们默认使用区域尺寸 p=2，记为 MDT-/2。此外，我们还遵循 DiT 的参数设置，设计了适用于 MDT 的小型、基准和超大型模型，分

别记为 MDT-S/B/XL。与 LatentDiffusion [35] 和 DiT 相同，MDT 默认采用 Stable Diffusion 提供的固定 VAE[1] 来对图像/隐空间标记进行编码/解码。 VAE 编码器的下采样比率为 1/8，特征通道维度为 4，也就是说，一个尺寸为 256×256×3 的图像会被编码为尺寸为 32×32×4 的隐空间嵌入。

训练细节。 沿用 [32] 的方法，所有的模型都是在 ImageNet [8] 数据集上进行训练的，使用了 AdamW [28] 优化器，学习率为 3e-4，批大小为 256，不使用权重衰减（weight decay），图像分辨率为 256×256。我们设置了掩蔽比例0.3，$N_2 = 2$。遵循 DiT 中的训练设置，我们将训练的最大步数设定为 1000，使用线性方差调度，范围从10^{-4} 到 2×10^{-2}。其他设置也与 DiT 保持一致。

评估指标。 我们使用常用的评估指标来评估模型，包括 Fre'chet Inception Distance (FID) [18]、sFID [30]、Inception Score (IS) [37]、准确率和召回率 [26]。其中，FID 被作为主要指标，因为它能够衡量多样性和保真度，sFID 在空间级别上进行了改进。作为补充，IS 和准确率用于衡量保真度，而召回率用于衡量多样性。为了进行公平比较，我们遵循 [32] 的方法，使用 ADM [10] 中的 TensorFlow 评估套件，并报告使用 250 个 DDPM 采样步骤的 FID-50K。除非另有说明，我们报告的 FID 分数均不包括无分类器引导的结果 [21]。

4.2. 比较结果

性能对比。 Tab. 2 对我们的 MDT 与 SoTA DiT 在不同模型尺寸下进行了比较。很明显，MDT 在所有模型规模上都能以更少的训练成本实现更高的 FID 分数。 MDT 的参数和推理成本与 DiT 相似，因为正如在 Sec. 3.1 中介绍的那样，MDT 中额外的模块可以忽略。对于小模型，训练 300k 步的 MDT-S/2 在 FID 上远远优于训练 400k 步的 DiT-S/2（57.01 对 68.40）。更重要的是，训练 2000k 步的 MDT-S/2 在类似的计算预算下，实现了与使用更大模型 DiT-B/2 相当的性能。对于最大的模型，训练 1300k 步的 MDT-XL/2 在 FID 上优于训练 7000k 步的 DiT-XL/2（9.60 对 9.62），并且实现了大约 5 倍的更快训练进展。

Method	Image Res.	Training Steps (k)	FID-50K↓
DiT-S/2	256×256	400	68.40
MDT-S/2	256×256	300	57.01
MDT-S/2	256×256	400	53.46
MDT-S/2	256×256	2000	44.14
MDT-S/2	256×256	3500	**41.37**
DiT-B/2	256×256	400	43.47
MDT-B/2	256×256	400	34.33
MDT-B/2	256×256	3500	**20.45**
DiT-XL/2	256×256	400	19.47
DiT-XL/2	256×256	2352	10.67
DiT-XL/2	256×256	7000	9.62
MDT-XL/2	256×256	400	16.42
MDT-XL/2	256×256	**1300**	**9.60**
MDT-XL/2	256×256	3500	**6.65**

表 2. 在 ImageNet 256×256 数据集上，对 DiT [32] 和 MDT 在不同模型大小和训练步骤下的比较。DiT 的结果来自于 DiT 的报告结果。

我们还在Tab. 1 中将 MDT 的类条件图像生成性能与现有方法进行了比较。为了与 DiT 进行公平比较，我们在这个表格中也使用了 VAE 解码器的 EMA 权重。在类条件设置下，MDT 在一半的训练迭代次数内就远远优于 DiT，例如在 FID 上为 6.83 对 9.62。根据之前的研究 [1, 10, 32, 35]，我们利用了改进的无分类器引导 [21]，并带有幂次余弦权重缩放，以在类条件样本生成中平衡精度和召回率。MDT 在类条件图像生成方面的性能优于之前的 SoTA DiT 以及其他方法，其 FID 分数为 1.81，*在类条件图像生成方面创造了新的 SoTA*。与 DiT 类似，我们从未观察到模型在继续训练时的 FID 分数饱和。

收敛速度。 Fig. 1 在不同的训练步骤和训练时间下比较了 DiT/S-2 基准模型和 MDT/S-2 的性能，在 8×A100 GPU 上进行了测试。由于具有更强的上下文学习能力，MDT 实现了更好的性能，同时生成学习的速度更快。在训练步骤和训练时间方面，MDT

[1]该模型可在 https://huggingface.co/stabilityai/sd-vae-ft-mse 下载。

Mask Ratio	FID↓	sFID↓	IS↑	Precision↑	Recall↑
0.1	51.60	10.23	26.65	0.44	0.60
0.2	51.44	10.09	26.75	0.44	0.58
0.3	**50.26**	**10.08**	**27.61**	**0.45**	0.60
0.4	50.88	10.21	27.44	0.45	0.60
0.5	51.57	9.92	27.14	0.44	0.60
0.6	53.20	10.36	26.55	0.44	0.61
0.7	52.90	10.03	26.51	0.44	0.61
0.8	53.73	10.15	25.55	0.43	**0.61**

表 3. 不同掩蔽比率的影响。模型是进行 60 万次迭代训练MDT-S/2 。

Decoder pos.	FID↓	sFID↓	IS↑	Precision↑	Recall↑
Last0	51.05	9.97	27.31	0.44	0.60
Last1	50.96	**9.90**	**27.63**	0.45	0.60
Last2	**50.26**	10.08	27.61	**0.45**	**0.60**
Last4	51.67	10.12	26.91	0.45	0.60
Last6	52.64	10.36	26.46	0.44	0.60

表 4. 侧插值器位置的影响。MDT-S/2 模型包含 12 个块，经过 600k 次迭代训练。

的学习速度大约是 DiT 的 3 倍。例如，经过约 33 小时的训练（40 万步），MDT-S/2 达到了比经过约 100 小时训练（150 万步）的DiT-S/2更优越的性能，这表明上下文学习对于扩散模型更快速的生成学习至关重要。

4.3. 消融实验

在这一部分，我们进行了消融实验以验证 MDT 的设计。我们报告了 MDT-S/2 模型的结果，并使用 FID-50k 作为评价指标，除非另有说明。

掩蔽比例。掩蔽比例决定了训练过程中可以处理的输入区域数量。我们在Tab. 3中比较了使用不同掩蔽比例的结果。对于 MDT-S/2 模型来说，最佳的掩蔽比例是 30%，这与用于识别模型的掩蔽比例有很大不同，例如 MAE [17] 中的掩蔽比例是 75%。我们认为图像生成需要从更多的区域中学习更多细节，以实现高质量的合成，而识别模型仅需要从最基本的区域中推断语义。

侧插值器位置。为了满足扩散模型的高质量图像生成要求，侧插值器放置在网络的中间，而不是像在识别模型 [3, 17]中放置在网络的末尾。Tab. 4展示了将侧插值器放置在有12个块的MDT-S模型的不同位置的比较。结果表明，将侧插值器放置在最后两个块之前可以获得最佳的FID得分，而将其像识别模型那样放置在网络末端会降低性能。将侧插值器放置在网络的早期阶段也会损害性能，这表明掩蔽隐空间建模对于扩散模型的大多数阶段都是有益的。

非对称与对称掩蔽结构对比。不同于利用掩蔽机制生成图像的掩蔽生成工作，例如 MaskGIT [7]、MUSE [6] 等，MDT专注于通过掩蔽隐空间建模来提高扩散模型的上下文学习能力。因此，我们采用非对称结构，仅在扩散模型编码器中处理未掩蔽的标记。我们比较了 MDT 中的非对称结构和处理带有用可学习的掩蔽标记代替被掩蔽的标记的完整输入的对称结构 [7]。如Tab. 5a所示，MDT 中的非对称结构的 FID 为 50.26，优于对称结构的 FID 51.56。非对称结构进一步降低了训练成本，并允许扩散模型专注于学习上下文信息，而不是重建掩蔽标记。

侧插值器的影响。MDT中的侧插值器预测被掩蔽的标记，使得扩散模型能够学习更多的语义信息，并在训练和推理期间保持解码器输入的一致性。我们在Tab. 5b中比较了使用/不使用侧插值器的性能，发现使用side-interpolater可以获得1.34的FID提升，证明了它的有效性。

在侧插值器中使用掩蔽的捷径连接。 掩蔽的捷径连接确保了侧插值器从未被掩蔽的标记中只预测被掩蔽的标记。 Tab. 5c显示，使用掩蔽的捷径连接将FID从50.91提高到50.26，这表明限制侧插值器只预测被掩蔽的标记有助于扩散模型实现更强的性能。

完整和掩蔽的隐空间标记。在 MDT 中，完整的和掩蔽的隐空间嵌入都在训练过程中被馈送到扩散模型中。与之相比，如Tab. 5d所示，我们提供了仅使用完整或掩蔽的隐空间嵌入进行训练的结果，其中计算成本是对齐的，以进行公平比较。使用完整和掩蔽的隐空间嵌入共同用于训练明显优于另两个

Asymmetric stru.	FID-50k↓
×	51.56
✓	**50.26**

(a) 非对称掩蔽结构的影响。

Side-interpolater	FID-50k↓
×	51.60
✓	**50.26**

(b) 侧插值器的影响。

Masked shortcut	FID-50k↓
×	50.91
✓	**50.26**

(c) 掩蔽的捷径连接的影响。

Latent type	FID-50k↓
Full+Masked	**50.26**
Full	52.30
Masked	76.63

(d) 在成本对齐的情况下使用完整/掩蔽的隐表示。

Sup. parts	FID-50k↓
All	**50.26**
Masked	58.35

(e) 受监督的标记部分。

Number	FID-50k↓
1	**50.26**
2	51.77
3	51.96

(f) 侧插值器中的块数量。

IS Pos. embed.	FID-50k↓
×	51.58
✓	**50.26**

(g) 侧插值器中位置嵌入的影响。

Learnable pos.	FID-50k↓
×	50.80
✓	**50.26**

(h) 可学习的位置嵌入的影响。

Relative pos. bias	FID-50k↓
×	53.56
✓	**50.26**

(i) 相对位置偏移的影响。

表 5. MDT-S/2上的消融实验。模型经过600k次迭代的训练。

竞争方案。而仅使用掩蔽的隐空间嵌入导致收敛较慢，我们将其归因于训练和推理的不一致性，因为在 MDT 中，推理是一个扩散过程，而不是掩蔽重建过程。

在所有标记上的损失。默认情况下，我们在掩蔽和未掩蔽的隐空间嵌入上计算损失。相比之下，用于识别模型的掩蔽建模通常在掩蔽的标记上计算损失 [3, 17]。Tab. 5e显示，计算所有标记的损失要比计算掩蔽的标记的损失好得多。我们认为这是因为生成模型需要更强的区域一致性，因为细节对于高质量的图像合成至关重要，而识别模型不需要。

侧插值器中的块数。我们在Tab. 5f中比较了侧插值器中不同块数量的性能。默认设置下的1个块获得了最佳性能，随着块数的增加，FID值变差。这个结果与我们的动机一致，即侧插值器不应该学习太多关于除了插值掩蔽表示之外的信息。

位置感知增强。为了进一步释放掩蔽隐空间建模的潜力，我们通过更强的位置感知能力，即可学习的位置嵌入和基本块中的相对位置偏移，增强了DiT基准模型。Tab. 5g显示，侧插值器中的位置嵌入将FID从51.58降低到50.26，表明位置嵌入对于侧插值器至关重要。此外，启用位置嵌入的训练也在Tab. 5h中表现出在FID方面的收益。在Tab. 5i中，基本块中的相对位置偏移将FID从53.56降低到50.26，显示出相对位置建模能力对于扩散模型获得上下文表示能力和生成高质量图像至关重要。因此，扩散模型结构中的位置感知能力需要与掩蔽隐空间建模相伴而行，并对提高性能发挥关键作用。

5. 结论

本研究提出了一种掩蔽扩散Transformer，以增强上下文表示，并改善DPMs中图像语义之间的关系学习。我们将高效的掩蔽隐空间建模方案引入DPMs，并相应地设计了一个非对称的掩蔽扩散Transformer结构。实验证明，我们的掩蔽扩散Transformer在图像合成方面表现出更高的性能，并且在训练过程中极大地提升了了学习进度，从而在ImageNet数据集上实现了图像合成的新SoTA。我们希望我们对生成建模中上下文学习的初步探索能够促进更多关于统一表示学习的研究，无论是用于识别模型还是生成模型。

致谢

这项研究得到了NSFC（NO.62225604）和中央高校基本科研业务费（南开大学，070-63233089）的支持。南开大学超级计算中心提供了计算支持。

A. 模型细节

网络配置 我们遵循 DiT [32]中描述的网络配置，设置了 MDT 中的总块数（即N_1+N_2）、标记数量以及通道数。MDT 模型的配置如表 Tab. 6所示。与 DiT 一样，MDT 拥有不同规模的模型，分别用 S/B/XL 表示。

网络参数与开销 不同模型规模下的 MDT 网络参数和训练开销在表 Tab. 6中列出。与 DiT 基线相比，MDT 引入的额外推断参数和成本可忽略不计。

Size	Layers	Dim.	Head Num.	Param. (M)	FLOSs (G)
Network configurations of MDT models.					
S	12	384	6	33.1	6.07
B	12	768	12	130.8	23.02
XL	28	1152	16	675.8	118.69
Network configurations of DiT baselines.					
S	12	384	6	32.9	6.06
B	12	768	12	130.3	23.01
XL	28	1152	16	674.8	118.64

表 6. MDT 模型的网络配置如下。这些配置遵循 DiT 网络的设定。Layers包括编码器和解码器的层数，并且对于所有模型，解码器数量 N_2 都设置为 2。FLOSs 是在隐空间嵌入大小为 32x32 以及 p=2 的情况下测量的。Param 和 FLOSs 是使用推理模型测量的。

B. VAE解码器的比较结果

为确保与 DiT [32] 的公平比较，我们使用了 MSE 和 EMA 两个版本的预训练 VAE 解码器[2] 用于图像采样。如 Tab. 7 所示，EMA 版本相比 MSE 版本略微性能更好。除了原稿表1中的结果使用 EMA VAE 解码器外，我们默认使用 MSE VAE 解码器。

[2]MSE 和 EMA 版本的 VAE 模型可从 https://huggingface.co/stabilityai/sd-vae-ft-mse 和 https://huggingface.co/stabilityai/sd-vae-ft-ema 下载。

Method	Decoder	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
MDT	MSE	6.65	5.07	129.47	0.72	0.63
MDT	EMA	6.46	4.92	131.70	0.72	0.63
MDT-G	MSE	2.14	4.45	259.21	0.82	0.59
MDT-G	EMA	2.02	4.46	263.77	0.82	0.60

表 7. EMA 版本和 MSE 版本的VAE 解码器之间的比较。-G 表示使用无分类器引导的结果。

C. 使用MDT图像修复

我们通过在第一步使用侧插值器填充掩蔽的标记，然后对掩蔽的标记进行去噪扩散过程来验证 MDT 的图像修复能力。如图 Fig. 5 所示，我们在图像上使用不同的掩蔽比例，并使用 MDT 修复被掩蔽的部分。尽管 MDT 模型是使用 30% 的掩蔽比例进行训练的，但它可以轻松处理更大的掩蔽比例，如 70% 的掩蔽比例。我们将这种能力归因于我们提出的掩蔽隐空间建模与扩散模型的结合。

D. 改进的无分类器引导

无分类器引导采样 [21] 可以在样本质量和多样性之间进行权衡。它通过结合类条件和无条件估计来实现这一点：

$$\hat{\epsilon}_\theta(x_t, c) = \epsilon_\theta(x_t) + w \cdot (\epsilon_\theta(x_t, c) - \epsilon_\theta(x_t)),$$

其中，$\epsilon_\theta(x_t, c)$ 是类条件估计，$\epsilon_\theta(x_t)$ 是无条件估计，w 是引导权重。通常情况下，较大的 w 会降低多样性以提高样本质量。 MUSE [6] 在采样时使用线性增长的权重规划来替代固定的引导权重，这使得模型在早期步骤中生成更多多样化的样本，而在后期步骤中生成更高保真度的样本。受此启发，我们提出了一个在采样过程中使用的幂余弦引导权重规划：

$$w_t = \frac{1 - \cos \pi \left(\frac{t}{t_{\max}}\right)^s}{2} w,$$

其中，t 表示采样过程中的时间步，$t_{\max}$ 表示最大采样步数，w 表示最大引导权重，s 是控制引导权重增加速度的因子。如图 Fig. 6 所示，幂余弦规划在早期步骤中使用低引导权重，而在后期步骤中快

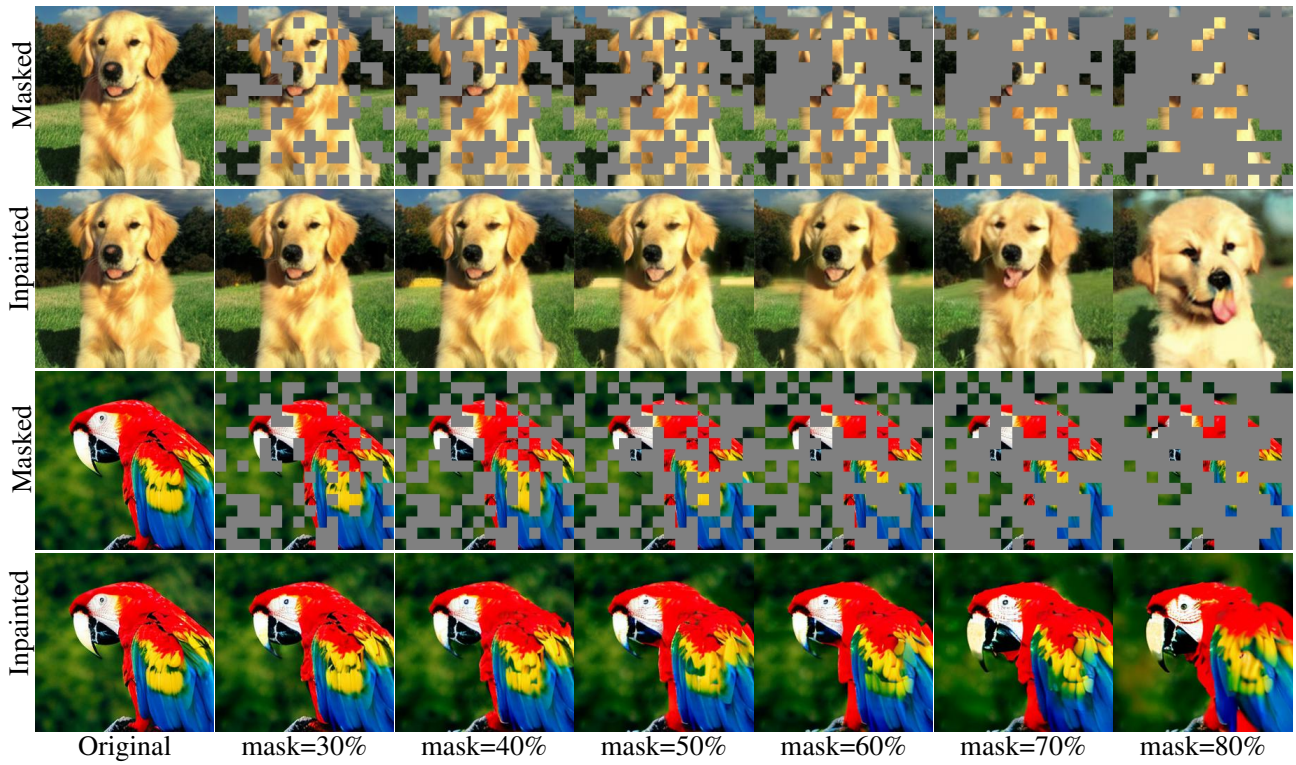

图 5. 不同掩蔽比例下使用MDT-XL/2的图像修复结果

速增加引导权重。通过增加 s，在早期步骤中引导权重缓慢增加，在后期步骤中快速增加。配备幂余弦引导权重规划的改进无分类器引导采样使模型在早期步骤中具有高多样性，在后期步骤中具有高质量。在本研究中，s 设置为 4，并相应地将 w 设置为 3.8，以确保模型在后期步骤中生成具有高保真度的图像。

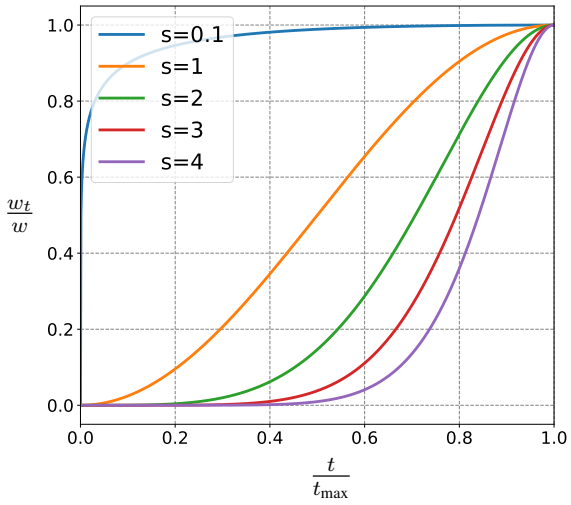

图 6. 当s不同时，无分类器引导中适用于引导权重的幂余弦权重规划。较大的 s 会导致w的增加速度在早期步骤中较慢，而在后期步骤中较快。

E. Visualization

我们在 Fig. 7中提供了更多的 MDT-XL/2 生成图像示例。在 Fig. 8中，我们展示了 MDT-S/2 随着训练过程的更多可视化示例。

参考文献

[1] Fan Bao, Chongxuan Li, Yue Cao, and Jun Zhu. All are worth words: a vit backbone for score-based diffusion models. *arXiv preprint arXiv:2209.12152*, 2022. 3, 6, 7

[2] Fan Bao, Chongxuan Li, Jiacheng Sun, Jun Zhu, and Bo Zhang. Estimating the optimal covariance with imperfect mean in diffusion probabilistic models. In *Int. Mach. Learn.*, 2022. 3

[3] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training

图 7. 由MDT-XL/2生成的图像的可视化结果

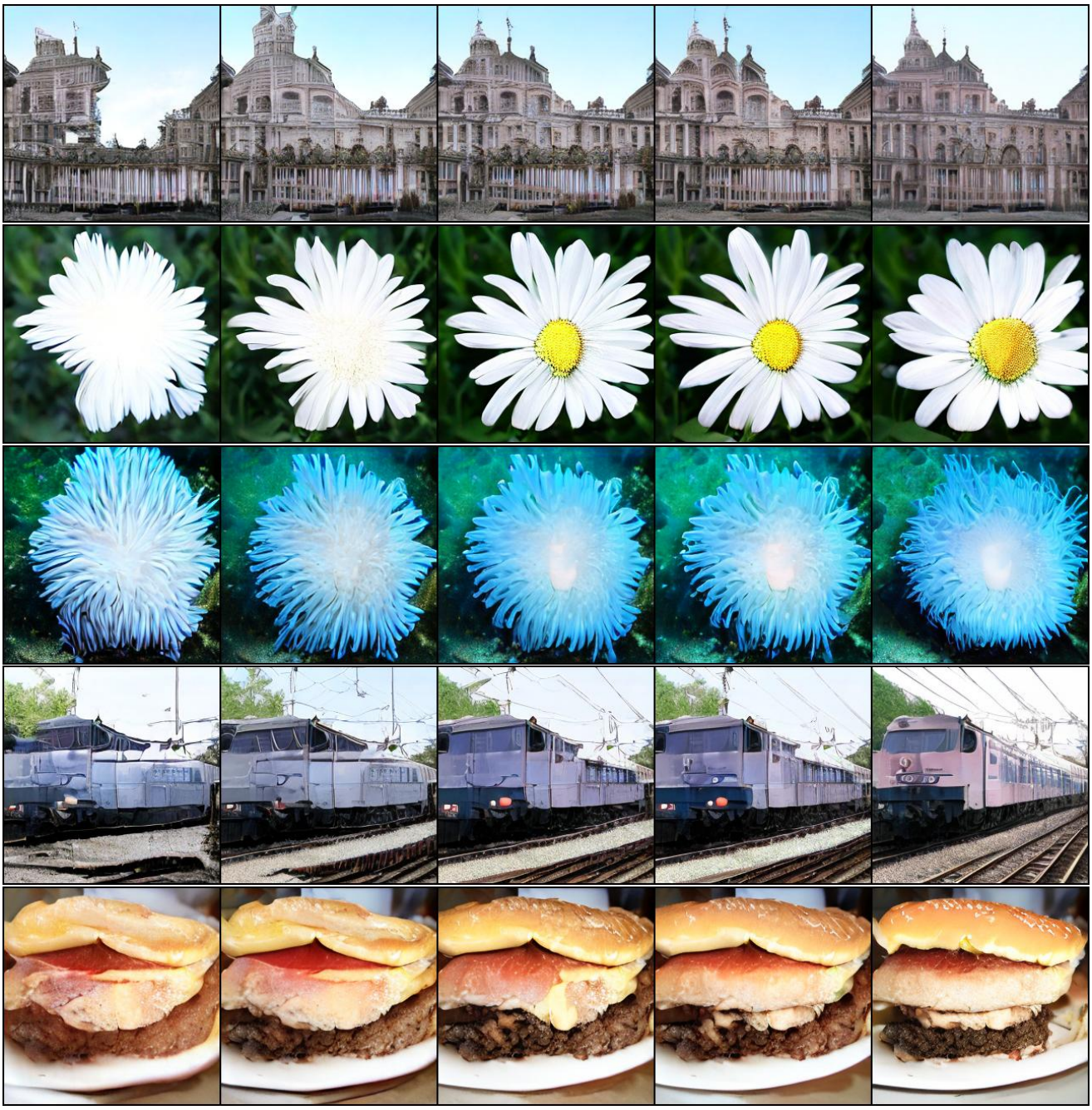

图 8. MDT-S/2随着训练过程的可视化样例

of image transformers. *Int. Conf. Learn. Represent.*, 2022. 3, 5, 8, 9

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *Int. Conf. Learn. Represent.*, 2019. 1, 3, 6

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Adv. Neural Inform. Process. Syst.*, 33:1877–1901, 2020. 3

[6] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy,

William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 3, 5, 8, 10

[7] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11315–11325, 2022. 3, 5, 6, 8

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009. 1, 7

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, 2019. 3

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Adv. Neural Inform. Process. Syst.*, 34:8780–8794, 2021. 1, 2, 3, 6, 7

[11] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. In *Int. Conf. Learn. Represent.*, 2022. 3

[12] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12873–12883, 2021. 6

[13] Shanghua Gao, Zhong-Yu Li, Ming-Hsuan Yang, Ming-Ming Cheng, Junwei Han, and Philip Torr. Large-scale unsupervised semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. 1

[14] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Towards sustainable self-supervised learning. *arXiv preprint arXiv:2210.11016*, 2022. 3

[15] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10696–10706, 2022. 3

[16] Yuchao Gu, Xintao Wang, Yixiao Ge, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Rethinking the objectives of vector-quantized tokenizers for image synthesis. *arXiv preprint arXiv:2212.03185*, 2022. 1

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16000–16009, 2022. 3, 8, 9

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Adv. Neural Inform. Process. Syst.*, 30, 2017. 7

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Adv. Neural Inform. Process. Syst.*, 33:6840–6851, 2020. 1, 2, 3

[20] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23(47):1–33, 2022. 3, 6

[21] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *NeurIPS Workshop*, 2021. 3, 7, 10

[22] Myeonghun Jeong, Hyeongju Kim, Sung Jun Cheon, Byoung Jin Choi, and Nam Soo Kim. Diff-tts: A denoising diffusion model for text-to-speech. In *INTERSPEECH*, 2021. 1

[23] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Adv. Neural Inform. Process. Syst.*, 2022. 3

[24] Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang, and Il-Chul Moon. Soft truncation: A universal training technique of score-based diffusion model for high precision score estimation. In *Int. Mach. Learn.*, 2022. 3

[25] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Adv. Neural Inform. Process. Syst.*, 34:21696–21707, 2021. 3

[26] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Adv. Neural Inform. Process. Syst.*, 32, 2019. 7

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Int. Conf. Comput. Vis.*, pages 10012–10022, 2021. 5

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *Int. Conf. Learn. Represent.*, 2019. 7

[29] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Adv. Neural Inform. Process. Syst.*, 2022. 1, 3

[30] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *Int. Mach. Learn.*, 2021. 6, 7

[31] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Int. Mach. Learn.*, pages 8162–8171. PMLR, 2021. 3, 4, 6

[32] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 1, 2, 3, 4, 5, 6, 7, 10

[33] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training, 2018. 3

[34] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Adv. Neural Inform. Process. Syst.*, 32, 2019. 1, 6

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10684–10695, 2022. 1, 2, 3, 4, 6, 7

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Int. Conf. Med Image Comput Comput Assist Interv.*, pages 234–241. Springer, 2015. 3

[37] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Adv. Neural Inform. Process. Syst.*, 29, 2016. 7

[38] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *Int. Conf. Learn. Represent.*, 2022. 1, 3

[39] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In *Int. Conf. Learn. Represent.*, 2017. 3

[40] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 6

[41] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Adv. Neural Inform. Process. Syst.*, 2022. 1

[42] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Int. Mach. Learn.*, pages 2256–2265. PMLR, 2015. 3, 4

[43] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Int. Conf. Learn. Represent.*, 2021. 3

[44] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Adv. Neural Inform. Process. Syst.*, 34:1415–1428, 2021. 3

[45] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Adv. Neural Inform. Process. Syst.*, 32, 2019. 3

[46] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Adv. Neural Inform. Process. Syst.*, 33:12438–12448, 2020. 3

[47] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *Int. Conf. Learn. Represent.*, 2021. 3

[48] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *Adv. Neural Inform. Process. Syst.*, 2021. 3

[49] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Adv. Neural Inform. Process. Syst.*, 29, 2016. 3

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Adv. Neural Inform. Process. Syst.*, 30, 2017. 5

[51] Yan Wu, Jeff Donahue, David Balduzzi, Karen Simonyan, and Timothy Lillicrap. Logan: Latent optimisation for generative adversarial networks. *Int. Conf. Learn. Represent.*, 2020. 1

[52] Yuxin Wu and Kaiming He. Group normalization. In *Eur. Conf. Comput. Vis.*, pages 3–19, 2018. 3

[53] Xiulong Yang, Sheng-Min Shih, Yinlin Fu, Xiaoting Zhao, and Shihao Ji. Your vit is secretly a hybrid discriminative-generative diffusion model. *arXiv preprint arXiv:2208.07791*, 2022. 3

[54] Zhu Zhang, Jianxin Ma, Chang Zhou, Rui Men, Zhikang Li, Ming Ding, Jie Tang, Jingren Zhou, and Hongxia Yang. M6-ufc: Unifying multi-modal controls for conditional image synthesis. *Adv. Neural Inform. Process. Syst.*, 2021. 3

[55] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *Int. Conf. Learn. Represent.*, 2022. 3