Faster Diffusion：重新思考编码器在扩散模型推理中的作用

李森茂 [1*], **胡泰航** [1*], **Joost van de Weijer**[2], **Fahad Shahbaz Khan**[3,4], **刘涛** [1]
李林轩 [1], **杨诗琪** [5], **王亚星** [1†], **程明明** [1], **杨健** [1]
[1]VCIP, 计算机学院, 南开大学, [2] 计算机视觉中心, 巴塞罗那自治大学
[3] 穆罕默德 · 本 · 扎耶德人工智能大学, [4] 林雪平大学, [5] 独立研究员, 东京
{senmaonk, hutaihang00, ltolcy0, linxuanli520, shiqi.yang147.jp}@gmail.com
joost@cvc.uab.es, fahad.khan@liu.se, {yaxing,cmm,csjyang}@nankai.edu.cn

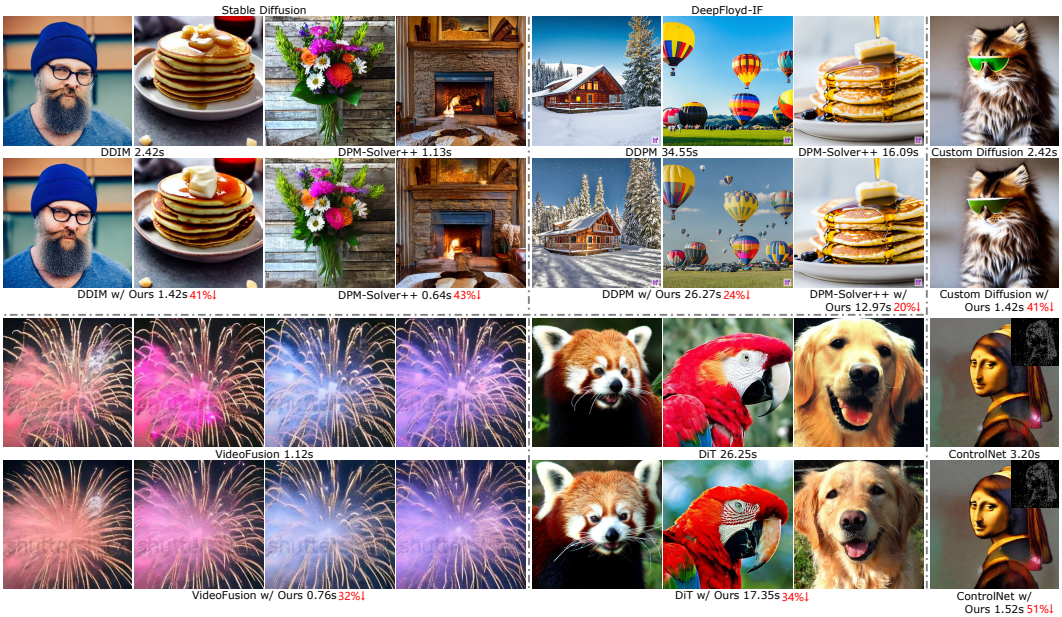

图 1: 我们的方法在多种生成任务中的结果。我们的方法显著提高了图像生成速度（秒/图）。

摘要

扩散模型的主要缺点之一是图像生成的推理时间较慢。解决这一问题的最成功方法之一是蒸馏技术。然而，这些方法需要消耗大量的计算资源。在本文中，我们提出了一种不同的扩散模型加速方法。我们对 UNet 编码器进行了全面的研究，并实证分析了编码器特征的行为变化。这项研究揭示了推理过程中编码器特征变化较小，而解码器特征在不同时间步之间的变化显著。这一发现促使我们在某些相邻时间步省略编码器的计算，并重复利用先前时间步的编码器特征作为解码器的输入，从而在多个时间步执行解码器的并行计算，进一步加速去噪过程。此外，我们引入了一种先验噪声注入方法，以提升生成图像的纹理细节表现。除了标准的文本生成图像任务外，我

*同等贡献。
†通讯作者。

38th 神经信息处理系统会议 (NeurIPS 2024).

们还在其他任务中验证了我们的方法，包括文本生成视频、个性化生成和参考引导生成。在不使用任何知识蒸馏技术的情况下，我们的方法分别将稳定扩散 (SD) 和 DeepFloyd-IF 模型的采样速度提升了 41% 和 24%，以及 DiT 模型的采样速度提升了 34%，同时保持了高质量的生成性能。项目主页: https://sen-mao.github.io/FasterDiffusion/。

1 引言

扩散模型（Diffusion Models, DMs）[1, 2, 3] 是图像生成领域的一种流行范式，近年来在多个领域取得了显著突破，包括文本生成视频 [4, 5, 6]、个性化图像生成 [7, 8, 9] 和参考引导图像生成 [10, 11, 12]。尽管扩散模型能够生成具有卓越视觉质量的图像，但其主要缺点在于推理时间较长。例如，与生成对抗网络（GANs）相比，原始扩散模型的推理时间慢了几个数量级。加速扩散模型的一大难题在于其固有的逐步去噪过程，这种序列特性限制了高效并行化的可能性。

为了提高扩散模型的推理速度，已经开发了多种方法，这些方法大致可以分为两类：首先是步数减少方法，其目标是通过减少扩散模型推理中的采样步数来加速生成过程，例如 DDIM [13] 和 DPM-Solver [14]，这些方法显著减少了采样步数的数量。其次是知识蒸馏方法，该类方法通过逐步将一个慢速（多步）教师模型蒸馏为一个快速（少步）学生模型 [15, 16]。尽管一些最新的研究 [17, 18, 19] 在少步采样场景下能够生成高保真的图像，但在单步采样中仍面临保持图像质量和多样性的挑战。知识蒸馏方法的主要缺点在于，它需要对模型进行重新训练以完成蒸馏过程，从而将其转化为更快的扩散模型。

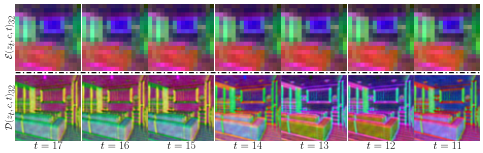

图 2: 可视化分层特征 [1]。我们参考 PnP [20]，对分层特征应用主成分分析（PCA），并使用前三个主成分生成 RGB 图像以进行可视化。从结果可以看出，编码器特征在多个时间步之间变化较小且具有较高的相似性（顶部），而解码器特征在不同时间步之间表现出显著的变化（底部）。

与上述方法不同，我们深入研究了扩散模型逐步去噪过程的序列特性，重点关注预训练扩散模型（例如 SD 和 DiT [21][2]）中编码器的特性。有趣的是，基于我们在 Sec 3.2中所呈现的分析，我们发现编码器特征的变化非常小（如 Fig. 3a 所示），并且具有高度的相似性（如 Fig. 2（顶部）所示），而解码器特征在不同时间步之间的变化则非常显著（如 Fig. 3a 和 Fig. 2（底部）所示）。这一发现至关重要，因为它使我们能够在多个时间步中绕过编码器的重复计算。由于基于相同编码器输入的解码器计算可以并行进行，我们可以将一个时间步中计算出的编码器特征（由于变化极小）重复用于后续时间步的解码器输入。与最近的 DeepCache [22] 和 CacheMe [23] 方法不同，这些方法虽然也利用了特征的相似性来实现加速，但仍依赖于序列化去噪，并且 CacheMe 需要对模型进行微调。我们的方法支持并行处理，从而显著加快了推理速度（如 Tab. 2所示）。

我们证明了所提出的传播方案将 SD 的采样速度提升了 24%，DeepFloyd-IF 的采样速度提升了 18%，DiT 的采样速度提升了 27%。此外，由于相同的编码器特征（来自之前时间步）可以作为解码器在多个后续时间步中的输入，这使得多个时间步的解码过程可以并行进行。通过这种并行处理，SD 的采样速度进一步加速到 41%，DeepFloyd-IF 达到 24%，DiT 提升到 34%。为了缓解生成质量的下降，我们引入了一种先验噪声注入策略，以保留生成图像中的纹理细节。凭借这些创新点，我们的方法在提高采样效率的同时保持了高质量的图像生成性能。更重要的是，我们的方法可以与现有的多种加速扩散模型的方法相结合。与基于蒸馏的

[1]请参阅 Appendix A 中关于所有块分层特征的可视化结果。
[2]我们将 DiT 的前几个 Transformer 块定义为编码器，其余部分定义为解码器。

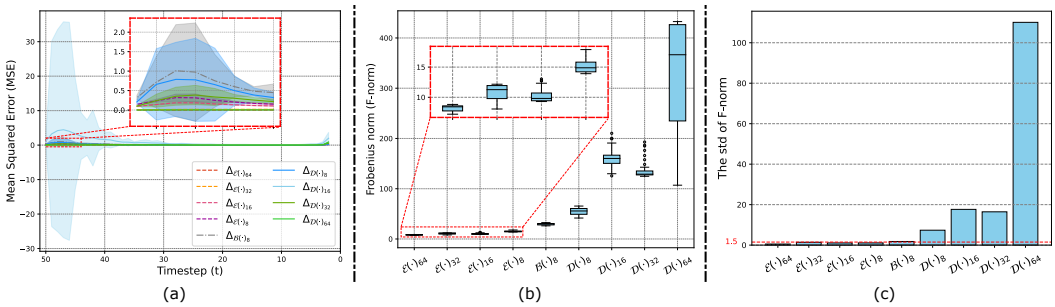

图 3: **分析扩散模型中的 UNet 特性**。(a) 使用均方误差（MSE）衡量相邻时间步特征的变化。(b) 提取 UNet 各层在每个时间步的分层特征输出，沿通道维度进行平均以获得二维分层特征，并计算其 *Frobenius* 范数。(c) UNet 编码器的分层特征表现出较低的标准差，而解码器的分层特征则表现出较高的标准差。

方法相比，我们的主要优势在于该方法可以直接应用于推理阶段，无需重新训练一个更快的蒸馏模型，从而避免了蒸馏过程所需的高昂计算成本，对于计算资源有限的用户尤其有利。最后，我们在多种条件扩散任务中验证了方法的有效性，包括文本生成视频（如 Text2Video-zero [4] 和 VideoFusion [5]）、个性化图像生成（如 Dreambooth [7]）以及参考引导图像生成（如 ControlNet [10]）。

总结而言，我们的主要贡献如下：

- 我们对扩散模型中 UNet 的特征进行了全面的实证研究，发现编码器特征变化极小（而解码器特征变化显著）。
- 我们提出了一种针对扩散模型在相邻时间步进行采样的并行策略，大幅加速了去噪过程。重要的是，我们的方法无需任何训练或微调。
- 此外，我们还提出了一种先验噪声注入方法，以提升图像质量（主要改善高频纹理的质量）。
- 我们的方法可以与现有方法（如 DDIM 和 DPM-solver）相结合，以进一步加速扩散模型的推理时间。

2 相关工作

去噪扩散模型。 近年来，文本生成图像的扩散模型 [1, 24, 25, 26] 取得了显著进展。其中，Stable Diffusion 和 DeepFloyd-IF 脱颖而出，成为当前开源社区中最成功的扩散模型之一。这些模型基于 UNet 架构，具有高度的通用性，可应用于广泛的任务，例如图像编辑 [27, 28]、超分辨率 [29, 30]、分割 [31, 32] 和目标检测 [33, 34]。鉴于 Transformer 网络的强大扩展性，DiT [21] 探索了基于 Transformer 的扩散模型骨干网络。

扩散模型加速。 扩散模型通过基于 UNet 的迭代去噪进行图像生成，但这一过程耗时较长。为了解决这一问题，已有大量研究工作提出了不同的优化策略。一种策略是采用高效的扩散模型求解器，例如 DDIM [13] 和 DPM-Solver [14]，这些方法显著减少了采样步骤。此外，ToMe [35] 利用 token 冗余来减少注意力操作 [36] 中的计算开销。与此相对，知识蒸馏方法（如学生模型的逐步简化 [15, 16]）旨在简化现有模型。一些最新研究将模型压缩与蒸馏相结合，以实现更快的采样 [37, 38]。与这些方法正交，我们提出了一种全新的方法来提升扩散模型推理过程中的采样效率。我们进一步证明了，我们的方法可以与现有多种加速方法结合，从而实现更进一步的加速效果。

DeepCache [22] 和 CacheMe [23] 是最近两种利用特征相似性实现加速的研究方法。DeepCache [22] 采用了一种直接重用前一步缓存特征的策略，但需要迭代去噪过程。此外，

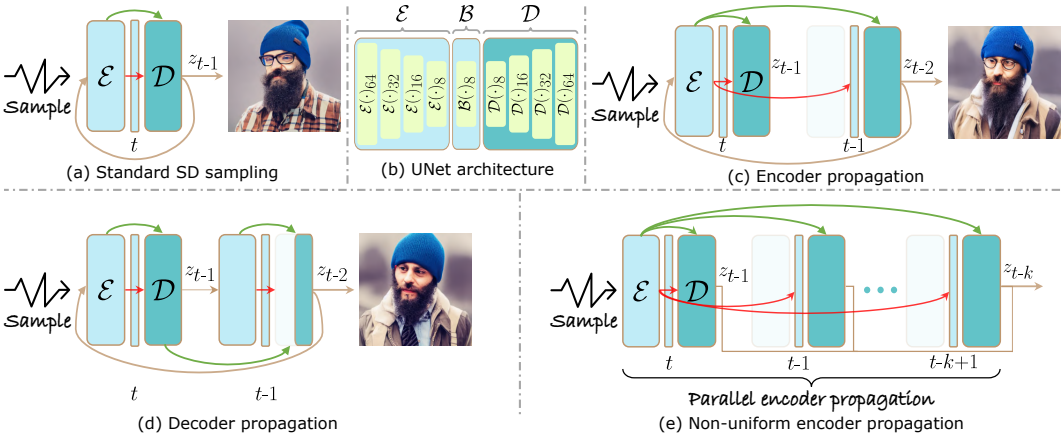

图 4: (a) 标准的 SD 采样流程。(b) UNet 架构。(c) 编码器传播。我们在某些相邻时间步中省略编码器的计算，并行复用前一时间步的编码器特征作为解码器的输入。在每两次迭代中应用统一的编码器传播策略。需要注意的是，在时间步 $t-1$ 中预测噪声时，并不需要 z_{t-1}（即 Eq. 1: $z_{t-2} = \sqrt{\frac{\alpha_{t-2}}{\alpha_{t-1}}} z_{t-1} + \sqrt{\alpha_{t-2}} \left(\sqrt{\frac{1}{\alpha_{t-2}} - 1} - \sqrt{\frac{1}{\alpha_{t-1}} - 1} \right) \cdot \epsilon_\theta(\cancel{z_{t-1}}, t-1, \boldsymbol{c})$）。(d) 解码器传播。生成的图像通常未能完全覆盖文本提示中的某些特定对象。例如，对于提示词“一个戴着眼镜和毛线帽、留着胡子的男人”，该方法未能生成与“眼镜”相关的内容。量化评估详见 Appendix F。(e) 应用非统一策略的编码器传播。得益于我们的传播方案，我们能够在某些相邻时间步中并行执行解码器操作。

CacheMe [23] 需要额外的微调以获得更好的性能。与这些方法相比，我们的方法支持并行处理，从而显著加快了推理速度。

3 方法

我们首先简要回顾了 SD 的架构（见 Sec. 3.1），然后对 UNet 的分层特征进行了全面分析（见 Sec. 3.2）。我们的分析表明，扩散模型的去噪过程可以部分实现并行化。因此，我们提出了一种新方法来加速扩散采样，同时在很大程度上保持生成质量和保真度（见 Sec. 3.3）。

3.1 潜在扩散模型

在扩散推理阶段，去噪网络 ϵ_θ 以文本嵌入 $\boldsymbol{c}$、潜在编码 $\boldsymbol{z_t}$ 和时间嵌入为输入，预测噪声并通过 DDIM 调度器 [13] 生成潜在变量 $\boldsymbol{z_{t-1}}$：

$$\boldsymbol{z_{t-1}} = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} \boldsymbol{z_t} + \sqrt{\alpha_{t-1}} \left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) \cdot \epsilon_\theta(\boldsymbol{z_t}, t, \boldsymbol{c}), \tag{1}$$

其中，α_t 是时间步 t ($t = T, ..., 1$) 的预定义标量函数。典型的去噪网络采用基于 UNet 的架构，分别由编码器 $\mathcal{E}$、瓶颈层 $\mathcal{B}$ 和解码器 $\mathcal{D}$ 组成（如 Fig. 4b 所示）。编码器 $\mathcal{E}$ 提取的分层特征通过跳跃连接注入到解码器 $\mathcal{D}$ 中（如 Fig. 4a 所示）。为便于描述，我们将 UNet 分为特定的块：$\mathcal{E} = \{\mathcal{E}(\cdot)_s\}$、$\mathcal{B} = \{\mathcal{B}(\cdot)_8\}$ 和 $\mathcal{D} = \{\mathcal{D}(\cdot)_s\}$，其中 $\mathtt{s} \in \{8, 16, 32, 64\}$（详见 Fig. 4b）。$\mathcal{E}(\cdot)_s$ 和 $\mathcal{D}(\cdot)_s$ 分别表示编码器和解码器中输入分辨率为 $\mathtt{s}$ 的块层 [3]。

Diffusion Transformer (DiT) [21] 是一种面向扩散模型的新型架构。它用 Transformer 替代了传统的 UNet 骨干网络，由 28 个块组成。根据我们的观察，我们将前 18 个块定义为编码器，将剩余的 10 个块定义为解码器（详见 Appendix A.3）。

[3] 一旦我们将具体输入替换为 $\mathcal{E}(\cdot)_s$ 中的 $\cdot$，则定义它表示 $\mathcal{E}(\cdot)_s$ 的特征。

3.2 分析扩散模型中的 UNet

在本节中，我们以基于 UNet 的扩散模型为例，分析预训练扩散模型的特性。我们深入研究了由编码器 $\mathcal{E}$、瓶颈层 $\mathcal{B}$ 和解码器 $\mathcal{D}$ 组成的 UNet，以更深入地理解 UNet 的不同部分。需要注意的是，下述观察到的特性同样存在于 DiT 中（详见 Appendix A.3）。

特征在时间步之间的演化。 我们通过实验观察到，编码器特征在相邻时间步的变化非常微小，而解码器特征在不同时间步之间的变化显著（见 Fig. 3a 和 Fig. 2）。具体来说，给定一个预训练的扩散模型，我们迭代生成潜在编码 $\boldsymbol{z_t}$（见 Eq. 1），以及相应的分层特征：$\{\mathcal{E}(z_t, c, t)_s\}$、$\{\mathcal{B}(z_t, c, t)_8\}$ 和 $\{\mathcal{D}(z_t, c, t)_s\}$，其中 $\mathtt{s} \in \{8, 16, 32, 64\}$ [4]，如 Fig. 4b 所示。我们在此分析分层特征在相邻时间步的变化情况。为了实现这一目标，我们通过以下公式量化分层特征的变化：

$$\Delta_{\mathcal{E}(\cdot)_s} = \frac{1}{d \times s^2} \|\mathcal{E}(z_t, c, t)_s - \mathcal{E}(z_{t-1}, c, t-1)_s\|_2^2, \tag{2}$$

其中，d 表示 $\mathcal{E}(z_t, c, t)_s$ 的通道数量。同样地，我们也计算 $\Delta_{\mathcal{B}(\cdot)_8}$ 和 $\Delta_{\mathcal{D}(\cdot)_s}$ 的变化量。

如 Fig. 3a 所示，对于编码器 $\mathcal{E}$ 和解码器 $\mathcal{D}$，特征变化曲线呈现相似趋势：在初期增长后，变化达到一个平台期，然后下降，最后在推理阶段末端再次增长。然而，$\Delta_{\mathcal{E}(\cdot)_s}$ 和 $\Delta_{\mathcal{D}(\cdot)_s}$ 的变化幅度在数量上有显著差异。例如，$\Delta_{\mathcal{E}(\cdot)_s}$ 的最大值和方差分别小于 0.4 和 0.05（见 Fig. 3a 中的放大区域），而 $\Delta_{\mathcal{D}(\cdot)_s}$ 的对应值约为 5 和 30（见 Fig. 3a）。此外，我们发现解码器最后一层的变化 $\Delta_{\mathcal{D}(\cdot)_{64}}$ 接近于零。这是因为去噪网络的输出在相邻时间步之间较为相似 [39]。综上所述，在整个推理阶段，编码器特征变化 $\Delta_{\mathcal{E}(\cdot)_s}$ 总体上小于解码器特征变化 $\Delta_{\mathcal{D}(\cdot)_s}$。

层间特征的演化。 我们通过实验观察到，在所有时间步中，编码器和解码器的特征性质存在显著差异。对于编码器 $\mathcal{E}$，特征变化的强度较小，而对于解码器 $\mathcal{D}$，特征变化则非常剧烈。具体来说，我们计算了在所有时间步上的分层特征 $\mathcal{E}(z_t, c, t)_s$ 的 *Frobenius* 范数，并将其定义为 $\mathcal{F}_{\mathcal{E}(\cdot)_s} = \{\mathcal{F}_{\mathcal{E}(z_T, c, T)_s}, ..., \mathcal{F}_{\mathcal{E}(z_1, c, 1)_s}\}$。同样地，我们分别计算了 $\mathcal{F}_{\mathcal{B}(\cdot)_8}$ 和 $\mathcal{F}_{\mathcal{D}(\cdot)_s}$。

如 Fig. 3b 所示，图中通过箱线图展示了各层特征的演化情况 [5]。具体来说，对于 $\{\mathcal{F}_{\mathcal{E}(\cdot)_s}\}$ 和 $\{\mathcal{F}_{\mathcal{B}(\cdot)_8}\}$，箱体相对紧凑，第一个四分位数和第三个四分位数之间的范围较窄。例如，这些特征的最大箱体高度（$\mathcal{F}_{\mathcal{E}}(.)_{32}$）小于 5（见 Fig. 3b 的放大区域）。这表明编码器 $\mathcal{E}$ 和瓶颈层 $\mathcal{B}$ 的特征变化幅度较小。相比之下，解码器 $\{\mathcal{D}(\cdot)_s\}$ 的箱体高度则相对较大。例如，$\mathcal{D}(.)_{64}$ 的箱体高度在第一个四分位数和第三个四分位数之间超过 150（见 Fig. 3b）。此外，我们还提供了标准差（见 Fig. 3c），其结果与 Fig. 3b 表现出类似的现象。这些结果表明，编码器特征在所有层之间的差异较小，并具有较高的相似性。然而，解码器特征则发生了显著的演化。

我们可以在某些时间步省略编码器吗？ 正如之前的实验分析所示，在去噪过程中，解码器特征发生显著变化，而编码器 $\mathcal{E}$ 的特征变化幅度很小，并且在某些相邻时间步之间具有高度相似性。因此，如 Fig. 4c 所示，我们提出在某些时间步省略编码器的计算，并在多个解码器步骤中复用相同的编码器特征。这种方法使我们能够并行计算这些解码器步骤。

具体而言，我们在时间步 $t-1$（$t-1 < T$）省略编码器的计算，对应的解码器（包括跳跃连接）使用来自前一时间步 t 的编码器 $\mathcal{E}$ 的分层输出作为输入，而不是像标准的 SD 采样那样使用当前时间步 $t-1$ 的编码器输出（更多细节见 Sec. 3.3）。

在某些时间步省略编码器时，我们依然能够生成与标准 SD 采样（如 Fig. 4a 所示）相似的图像（如 Fig. 4c 所示），相关结果也体现在 Tab. 1（第一行和第二行）以及 Appendix F 中的附

[4] 在编码器中，特征分辨率为上一层的一半；在解码器中，特征分辨率为上一层的两倍。需要注意的是，在 SD 模型中，$\mathcal{E}(.)_8$、$\mathcal{B}(.)_8$ 和 $\mathcal{D}(.)_{64}$ 的特征分辨率保持不变。

[5] 每个箱线图包含特征 Frobenius 范数的最小值（0 百分位）、最大值（100 百分位）、中位数（50 百分位）、第一个四分位数（25 百分位）和第三个四分位数（75 百分位），例如 $\{\mathcal{F}_{\mathcal{E}(z_T, c, T)_s}, ..., \mathcal{F}_{\mathcal{E}(z_1, c, 1)_s}\}$。

加结果。相比之下，如果对解码器采用类似策略（即 *解码器传播*），我们发现生成的图像通常无法完全覆盖文本提示中的某些特定对象（如 Fig. 4d 所示）。例如，当提示词为“一个留着胡子、戴着眼镜和毛线帽的男人”时，在应用解码器传播时，SD 模型未能生成“眼镜”。这是因为语义信息主要包含在解码器的特征中，而非编码器的特征中 [40]。

*编码器传播*使用前一时间步的编码器输出作为当前解码器的输入，可以在推理阶段加速扩散模型的采样过程。在接下来的 Sec. 3.3 中，我们将对编码器传播进行进一步详细阐述。

3.3 编码器传播

扩散采样结合了迭代去噪和 Transformer 的架构，因此计算过程耗时较长。为此，我们提出了一种新颖且实用的扩散采样加速方法。在扩散采样过程中 t=$\{T, ..., 1\}$，我们将执行*编码器传播*的时间步称为 *非关键* 时间步，记为 $t^{non\text{-}key}$=$\left\{t_0^{non\text{-}key}, ..., t_{N-1}^{non\text{-}key}\right\}$。其余时间步被称为 *关键* 时间步，记为 t^{key}=$\left\{t_0^{key}, t_1^{key}, ..., t_{T-1-N}^{key}\right\}$。换句话说，我们在时间步 $t^{non\text{-}key}$ 中省略编码器的计算，而使用来自时间步 t^{key} 的编码器分层特征。同时，在初始时间步（$t_0^{key} = T$）中，我们仍然使用编码器 $\mathcal{E}$。因此，扩散推理时间步可以重新表述为 $\left\{t^{key}, t^{non\text{-}key}\right\}$，其中 $t^{key} \cup t^{non\text{-}key} = \{T, ..., 1\}$ 且 $t^{key} \cap t^{non\text{-}key} = \varnothing$。接下来，我们将分别介绍 *统一编码器传播* 和 *非统一编码器传播*策略。

如 Fig. 3a 所示，在推理过程中，编码器特征的变化在初始阶段相较于后续阶段更为显著。因此，我们在推理的初始阶段选择更多的 *关键* 时间步，而在后续阶段选择较少的 *关键* 时间步。在实验中，对于使用 DDIM 的 SD 模型，我们将 *关键* 时间步定义为 t^{key}=$\{50, 49, 48, 47, 45, 40, 35, 25, 15\}$。对于 DeepFloyd-IF 的三个阶段，*关键* 时间步分别定义为 t^{key}=$\{100, 99, 98, \ldots, 92, 91, 90, 85, 80, \ldots, 25, 20, 15, 14, 13, \ldots, 2, 1\}$[6]、$\{50, 49, \ldots, 2, 1\}$ 和 $\{75, 73, 70, 66, 61, 55, 48, 40, 31, 21, 10\}$。关于 *关键* 时间步选择的更多细节，详见 Appendix F.2。

剩余的时间步被归类为 *非关键* 时间步。我们将这一策略称为非均匀编码器传播（见 Fig. 4e）。如 Fig. 4c 所示，我们还探索了固定步长（例如 2）的时间步选择策略，并将其称为均匀编码器传播。

需要注意的是，我们的方法并未减少采样步骤的数量。在编码器传播期间，解码器仍需要对所有时间步进行计算，这需要为每个时间步的解码器提供时间嵌入输入，以保持时间上的一致性（详见 Appendix D）。

Tab. 5 报告了关于不同 *关键* 和 *非关键* 时间步组合的消融实验结果。这些结果表明，非均匀的 *关键* 时间步设置在图像生成中表现更优。

并行非均匀编码器传播。 在应用非均匀编码器传播策略时，对于时间步 $t \in t^{non-key}$，解码器的输入不依赖于时间步 t 的编码器输出（见 Fig. 4e）。相反，它依赖于前一个最近的 *关键* 时间步的编码器输出。这使我们能够在 $t^{non-key}$ 的相邻时间步中执行并行非均匀编码器传播。我们在 t 到 $t-k+1$ 的时间步内并行执行解码操作。由于解码器在多个时间步的前向传播可以同时进行，这项技术进一步提高了推理效率。我们将这种方法称为并行批处理 *非关键* 时间步。如 Fig. 5（右）所示，对于 SD 模型，这种方法将评估时间进一步减少了 41%。

图 5: 与标准 SD（左）相比，编码器传播将采样时间减少了 24%（中）。此外，并行编码器传播进一步将采样时间减少了 41%（右）。

[6]时间步中的省略号表示包含两侧时间步之间的所有时间步。例如，80...25 表示从 80 到 25 的所有时间步均被包括在内。

先验噪声注入。 尽管编码器传播可以提高推理阶段的效率，但我们观察到它会导致生成结果中的纹理信息略有丢失（见 Fig. 6 左图和中图）。受到相关研究 [41, 42] 的启发，我们提出了一种先验噪声注入策略。该策略在后续时间步（即 z_t）的生成过程中将初始潜在编码 z_T 引入，遵循公式 $z_t = z_t + \alpha \cdot z_T$，if $t < \tau$，其中 $\alpha = 0.003$ 是用于控制 z_T 影响的比例参数。我们从 $\tau = 25$ 的时间步开始应用这一注入机制。这种策略性融合成功改善了纹理信息，且几乎不需要额外的计算资源。我们发现纹理信息的丢失发生在频域的所有频率上（见 Fig. 6 右图中的红色和蓝色曲线）。该方法保证了生成结果在频域上与标准 SD 和 z_T 注入保持接近（见 Fig. 6 右图中的红色和绿色曲线），同时生成的图像保持了理想的保真度（见 Fig. 6 左图底部）。

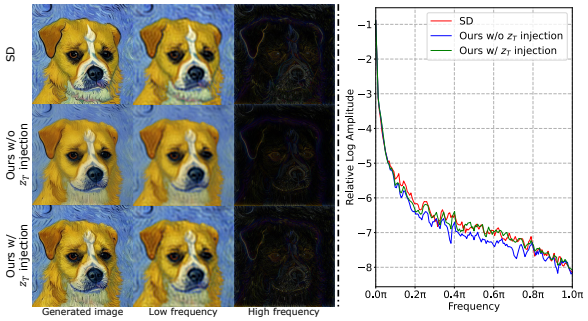

图 6: (左)通过 z_T 注入，我们保留了图像内容，稍微弥补了由于编码器传播导致的纹理信息损失。(右)通过 z_T 注入生成的图像振幅与标准 SD 生成的图像非常相似。

4 实验

在我们的实验中，我们评估了与其他推理加速方法相比，我们方法的加速效果，同时也探讨了将我们的方法与这些方法结合的可能性。我们未直接将我们的方法与蒸馏方法进行比较，尽管蒸馏方法可以提供更优的结果，但其需要耗费大量计算资源进行重新训练。

数据集与评估指标。 我们从 MS-COCO2017 验证集 [44] 中随机选择 10K 条文本提示，输入文本生成图像扩散模型以生成 10K 张图像。对于基于 Transformer 架构的扩散模型，我们从 1000 个 ImageNet [45] 类别标签中随机生成 50K 张图像。对于其他任务，我们使用与基线相同的设置（例如 Text2Video-zero [4]、VideoFusion [5]、Dreambooth [7] 和 ControlNet [10]）。我们使用 Fréchet Inception Distance (FID) [46] 评估生成图像的视觉质量，并使用 Clipscore [43] 衡量图像内容与文本提示的一致性。此外，我们报告了单张图像的平均计算工作量（GFLOPs/image）和采样时间（s/image），以表征生成单张图像所需的资源需求。更详细的实现信息请见 Appendix A。

表 1: 针对 SD 和 DeepFloyd-IF 扩散模型的定量评估[7]。

DM	Sampling Method	T	FID↓	Clip-score↑	GFLOPs/image↓	s/image ↓ Unet of DM	s/image ↓ DM
Stable Diffusion	DDIM	50	21.75	0.773	37050	2.23	2.42
	DDIM w/ Ours	50	21.08	0.783	**27350**$_{27\%\downarrow}$	**1.21**$_{45\%\downarrow}$	**1.42**$_{41\%\downarrow}$
	DPM-Solver	20	21.36	0.780	14821	0.90	1.14
	DPM-Solver w/ Ours	20	21.25	0.779	**11743**$_{21\%\downarrow}$	**0.46**$_{48\%\downarrow}$	**0.64**$_{43\%\downarrow}$
	DPM-Solver++	20	20.51	0.782	14821	0.90	1.13
	DPM-Solver++ w/ Ours	20	20.76	0.781	**11743**$_{21\%\downarrow}$	**0.46**$_{48\%\downarrow}$	**0.64**$_{43\%\downarrow}$
	DDIM + ToMe	50	22.32	0.782	35123	2.07	2.26
	DDIM + ToMe w/ Ours	50	20.73	0.781	**26053**$_{26\%\downarrow}$	**1.15**$_{44\%\downarrow}$	**1.33**$_{41\%\downarrow}$
DeepFloyd-IF	DDPM	225	23.89	0.783	734825	33.91	34.55
	DDPM w/ Ours	225	23.73	0.782	**626523**$_{15\%\downarrow}$	**25.61**$_{25\%\downarrow}$	**26.27**$_{24\%\downarrow}$
	DPM-Solver++	100	20.79	0.784	370525	15.19	16.09
	DPM-Solver++ w/ Ours	100	20.85	0.785	**313381**$_{15\%\downarrow}$	**12.02**$_{21\%\downarrow}$	**12.97**$_{20\%\downarrow}$

4.1 文本生成图像

我们首先在潜在空间（即 SD）和像素空间（即 DeepFloyd-IF）的扩散模型上评估了所提出的编码器传播方法在标准文本生成图像任务中的表现。如 Tab. 1 所示，我们的方法在几乎无性能下降的情况下显著加速了扩散采样。具体而言，与标准 DDIM 采样在 SD 中的结果相比，我们的方法大幅减少了计算负担（GFLOPs）27%，并将采样时间减少到 41%。同样地，在

[7]我们使用 Clipscore 的官方实现 [43]，获得的分数约为 0.75，而非约 0.3。详见 Appendix C。

表 2: 与 DeepCache 和 CacheMe 的比较。

Sampling Method	T	Parallel	FID ↓	Clipscore ↑	s/image
DDIM	50	×	21.75	0.773	2.42
DDIM w/ DeepCache	50	×	21.53	0.770	1.05 56%↓
DDIM w/ CacheMe	50	×	–	–	1.30 44%↓
DDIM w/ Ours	50	✓	21.62	0.775	0.56 77%↓

表 3: DiT 定量评估。

Sampling Method	T	Image Res.	FID ↓	sFID ↓	IS ↑	Precision ↑	Recall ↑	s/image
DiT	250	256	2.27	4.60	278.24	0.83	0.57	5.13
DiT w/ Ours	250	256	**2.31**	**4.55**	**276.05**	**0.82**	**0.57**	**3.62** 29%↓
DiT	250	512	3.04	5.02	240.82	0.84	0.54	26.25
DiT w/ Ours	250	512	**3.25**	**5.05**	**245.13**	**0.83**	**0.51**	**17.35** 34%↓

图 7: 在不同时间步生成的图像。

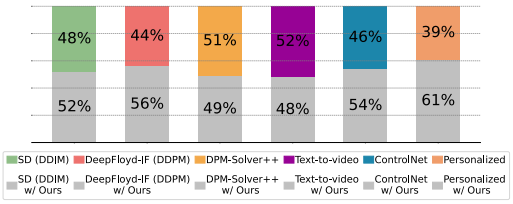

图 8: 用户研究结果。

DeepFloyd-IF 模型中，计算负担和采样时间的减少分别达到了 15% 和 24%。此外，我们的方法可以与最新的采样技术（如 DPM-Solver [14]、DPM-Solver++ [47] 和 ToMe [35]）相结合。我们的方法在保持良好模型性能的同时提高了采样效率，FID 和 Clipscore 的变化均可以忽略不计（见 Tab. 1 第三到第八行）。在不同采样步数下，我们的方法也表现出良好的性能（见 Fig. 7 和 Appendix D 的定量结果）。重要的是，这些结果表明我们的方法与这些加速技术是正交且兼容的。如 Fig. 1 所示，我们可视化了采用不同采样技术生成的图像。我们的方法仍能生成高质量的结果（更多结果请参见 Appendix F）。

我们的方法支持使用多 GPU 并行生成单张图像。在多 GPU 并行的情况下，我们的方法进一步将 SD 的采样速度提升了 77%，而 DeepCache [22] 和 CacheMe [23] 分别实现了 56% 和 44% 的加速（详见 Tab. 2）。这些结果表明，与 DeepCache [22] 和 CacheMe [23] 相比，我们的方法在加速效果上具有显著优势。

4.2 扩散 Transformer

我们的方法同样在 DiT 模型上进行了评估。如 Tab. 3 所示，对于分辨率为 256 和 512 的 DiT 采样，我们分别实现了约 29% 和 34% 的加速，同时保持了高质量的生成结果（见 Figs. 1 和 18）。

4.3 文本引导扩散模型的其他任务

除了标准的文本生成图像任务外，我们还在其他任务上验证了我们提出的方法，包括：文本生成视频、个性化生成以及参考引导图像生成。

文生视频 为了评估我们的方法，我们将其与 Text2Video-zero [4] 和 VideoFusion [5] 相结合。如 Tab. 4（第二行和第四行）所示，当结合我们的方法时，这两种方法的计算负担和生成时间分别减少了约 22% 到 33%。这些结果表明，我们能够在文本生成视频任务中提高生成过程的效率，同时保持视频的保真度（见 Fig. 1（左下））。以生成提示词 “烟花在夜空中绽放” 的视频为例，VideoFusion 模型生成 16 帧视频耗时 17.92 秒（1.12 秒/帧）。当结合我们的方法后，仅需 12.27 秒（0.76 秒/帧）即可生成高质量视频（见 Fig. 1（左下））。

个性化图像生成 Dreambooth [7] 和 Custom Diffusion [8] 是通过微调文本生成图像扩散模型来定制任务的两种方法。如 Tab. 4（第九到第十二行）所示，我们的方法结合这两种定制化方法，可以加速图像生成并降低计算需求。从视觉效果来看，我们的方法仍然能够根据参考图像生成具有特定上下文关系的图像（见 Fig. 1（右））。

表 4: 文本生成视频、个性化生成和参考引导生成任务的定量评估。† 和 ‡ 分别表示“边缘”和“涂鸦”条件。

Method	T	FID↓	Clip-score↑	GFLOPs/image↓	s/image↓ Unet of SD	s/image↓ SD
Text2Video-zero	50	-	0.732	39670	12.59/8	13.65/8
Text2Video-zero w/ Ours	50	-	0.731	**30690**$_{22\%\downarrow}$	**9.46/8**$_{25\%\downarrow}$	**10.54/8**$_{23\%\downarrow}$
VideoFusion	50	-	0.700	224700	16.71/16	17.93/16
VideoFusion w/ Ours	50	-	0.700	**148680**$_{33\%\downarrow}$	**11.1/16**$_{34\%\downarrow}$	**12.2/16**$_{32\%\downarrow}$
ControlNet (†)	50	13.78	0.769	49500	3.09	3.20
ControlNet (†) w/ Ours	50	14.65	0.767	**31400**$_{37\%\downarrow}$	**1.43**$_{54\%\downarrow}$	**1.52**$_{51\%\downarrow}$
ControlNet (‡)	50	16.17	0.775	56850	3.85	3.95
ControlNet (‡) w/ Ours	50	16.42	0.775	**35990**$_{37\%\downarrow}$	**1.83**$_{53\%\downarrow}$	**1.93**$_{51\%\downarrow}$
Dreambooth	50	-	0.640	37050	2.23	2.42
Dreambooth w/ Ours	50	-	0.660	**27350**$_{27\%\downarrow}$	**1.21**$_{45\%\downarrow}$	**1.42**$_{41\%\downarrow}$
CustomDiffusion	50	-	0.640	37050	2.21	2.42
CustomDiffusion w/ Ours	50	-	0.650	**27350**$_{27\%\downarrow}$	**1.21**$_{45\%\downarrow}$	**1.42**$_{41\%\downarrow}$

表 5: 在 MS-COCO 2017 10K 子集上，不同传播策略的定量评估。FTC=FID×Time/Clipscore。

Propagation strategy		FID ↓	Clipscore ↑	GFLOPs /image ↓	s/image ↓ Unet of SD	s/image ↓ SD	FTC ↓
SD		21.75	0.773	37050	2.23	2.42	68.1
Uniform	Ⅰ	$t^{key}=\{50,48,46,44,42,40,38,36,34,32,30,28,26,24,22,20,18,16,14,12,10,8,6,4,2\}$					
		21.55	0.775	31011$_{16\%\downarrow}$	1.62$_{27\%\downarrow}$	1.81$_{25\%\downarrow}$	50.3
	Ⅱ	$t^{key}=\{50,44,38,32,26,20,14,8,2\}$					
		21.54	0.773	27350$_{27\%\downarrow}$	1.26$_{43\%\downarrow}$	1.46$_{40\%\downarrow}$	40.7
	Ⅲ	$t^{key}=\{50,38,26,14,2\}$					
		24.61	0.766	26370$_{29\%\downarrow}$	1.12$_{50\%\downarrow}$	1.36$_{44\%\downarrow}$	43.7
Non-uniform	Ⅰ	$t^{key}=\{50,40,39,38,30,25,20,15,5\}$					
		22.94	0.776	27350$_{27\%\downarrow}$	1.26$_{43\%\downarrow}$	1.42$_{41\%\downarrow}$	41.9
	Ⅱ	$t^{key}=\{50,30,25,20,15,14,5,4,3\}$					
		35.25	0.742	27350$_{27\%\downarrow}$	1.25$_{43\%\downarrow}$	1.42$_{41\%\downarrow}$	67.4
	Ⅲ	$t^{key}=\{50,41,37,35,22,21,18,14,5\}$					
		22.14	0.778	27350$_{27\%\downarrow}$	1.22$_{45\%\downarrow}$	1.42$_{41\%\downarrow}$	40.4
	Ⅳ (Ours)	$t^{key}=\{50,49,48,47,45,40,35,25,15\}$					
		21.08	0.783	27350$_{27\%\downarrow}$	1.21$_{45\%\downarrow}$	1.42$_{41\%\downarrow}$	**38.2**

参考引导图像生成 ControlNet [10] 结合了一个可训练的编码器，能够成功生成文本引导的图像，并保留与条件信息相似的内容。我们的方法可以同时应用于 ControlNet 的两个编码器。在本文中，我们验证了提出的方法在两种条件控制下的效果：边缘和涂鸦。Tab. 4（第五到第八行）报告了定量结果。我们观察到该方法显著减少了生成时间和计算负担。此外，Fig. 1（中间下方）从定性角度展示了我们的方法能够成功保留给定的结构信息，并实现与 ControlNet 类似的生成结果。

用户研究 我们进行了用户研究，如 Fig. 8 所示，邀请受试者对生成结果进行选择。研究采用成对比较（强制选择）的方式，共有 18 名用户参与，每位用户评估 35 对图像或视频。结果表明，我们的方法与基线方法的表现同样出色。

4.4 消融研究

我们对均匀和非均匀编码器传播的不同选择进行了消融实验。如 Tab. 5 所示，非均匀设置在 FID 和 Clipscore 方面的性能均优于均匀设置（见 Tab. 5 第三行和第八行）。此外，我们探讨了非均匀策略中的不同配置。使用我们设定的 关键 时间步集合的策略在生成过程中表现出更好的结果（见 Tab. 5 第八行）。我们进一步展示了基于上述选择的定性结果。如 Fig. 9 所示，在相同数量的 关键 时间步下，九步非均匀策略

表 6: 先验噪声注入的定量评估。

Sampling Method	SD (DDIM)	SD (DDIM) + Ours w/o z_T injection	SD (DDIM) + Ours w/ z_T injection
FID ↓	21.75	21.71	21.08
Clipscore ↑	0.773	0.779	0.783

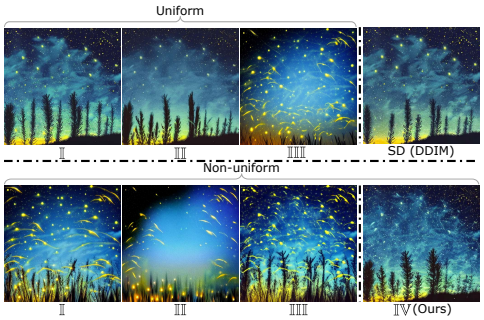

图 9: 使用均匀和非均匀编码器传播生成图像时，均匀策略 Ⅲ 的结果相较于 SD 更加平滑，但丢失了文本相关性。均匀策略 Ⅲ 和非均匀策略 Ⅰ、Ⅲ、Ⅲ 生成的图像均表现出不自然的饱和度水平。

Ⅱ、Ⅲ 和 Ⅲ 的生成结果未能与提示词 “萤火虫点缀夜空” 对齐。尽管两步设置的生成图像在视觉质量上令人满意，但其采样效率低于我们选择的设置（见 Tab. 5 第二行和第八行）。

先验噪声注入的有效性。 我们评估了注入初始 z_T 的有效性。如 Tab. 6 所示，与 DDIM 和我们的方法（第二列和第四列）相比，不注入 z_T（第三列）的 FID 和 Clipscore 差异约为 0.01%，可以认为是可以忽略不计的。然而，这种差异在生成图像的视觉表现上有所体现。我们观察到，输出图像虽然具有完整的语义信息，但纹理较为平滑（参见 Fig. 6（左图第二行））。注入 z_T 有助于在编码器传播期间保持生成结果的保真度（参见 Fig. 6（左图第三行）和 Fig. 6（右图中红色和绿色曲线））。

5 结论

在这项工作中，我们探索了文本生成图像扩散模型中 UNet 的编码器和解码器的特性，发现编码器的特征在多个时间步之间变化极小，而解码器在所有时间步中都发挥着重要作用。基于这一发现，我们提出了编码器传播方法以实现高效的扩散采样，加速了基于 UNet 和基于 Transformer 的扩散模型在多种生成任务中的推理过程。我们进行了广泛的实验，验证了该方法能够在保持图像质量的同时显著提升采样效率。**局限性：** 尽管我们的方法实现了高效的扩散采样，但在使用较少采样步数（例如 5 步）时，生成质量仍面临一定的挑战。此外，尽管我们提出的并行化方法也可以应用于网络蒸馏方法 [18, 17, 19]，但本文未对这一方向进行探索，留待未来研究。

参考文献

[1] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[3] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[4] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *arXiv preprint arXiv:2303.13439*, 2023.

[5] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10209–10218, 2023.

[6] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arxiv:2307.10373*, 2023.

[7] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[8] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023.

[9] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[10] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[11] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

[12] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22511–22521, 2023.

[13] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[14] 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. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

[15] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[16] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023.

[17] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

[18] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.

[19] Thuan Hoang Nguyen and Anh Tran. Swiftbrush: One-step text-to-image diffusion model with variational score distillation. *arXiv preprint arXiv:2312.05239*, 2023.

[20] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023.

[21] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

[22] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. *arXiv preprint arXiv:2312.00858*, 2023.

[23] Felix Wimbauer, Bichen Wu, Edgar Schoenfeld, Xiaoliang Dai, Ji Hou, Zijian He, Artsiom Sanakoyeu, Peizhao Zhang, Sam Tsai, Jonas Kohler, et al. Cache me if you can: Accelerating diffusion models through block caching. *arXiv preprint arXiv:2312.03209*, 2023.

[24] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[25] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[26] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023.

[27] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[28] Senmao Li, Joost van de Weijer, Taihang Hu, Fahad Shahbaz Khan, Qibin Hou, Yaxing Wang, and Jian Yang. Stylediffusion: Prompt-embedding inversion for text-based editing. *arXiv preprint arXiv:2303.15649*, 2023.

[29] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *arXiv preprint arXiv:2305.07015*, 2023.

[30] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12413–12422, 2022.

[31] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*, 2021.

[32] Julia Wolleb, Robin Sandkühler, Florentin Bieder, Philippe Valmaggia, and Philippe C Cattin. Diffusion models for implicit image segmentation ensembles. In *International Conference on Medical Imaging with Deep Learning*, pages 1336–1348. PMLR, 2022.

[33] Walter HL Pinaya, Mark S Graham, Robert Gray, Pedro F Da Costa, Petru-Daniel Tudosiu, Paul Wright, Yee H Mah, Andrew D MacKinnon, James T Teo, Rolf Jager, et al. Fast unsupervised brain anomaly detection and segmentation with diffusion models. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 705–714. Springer, 2022.

[34] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19830–19843, 2023.

[35] Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4598–4602, 2023.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[37] Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. On architectural compression of text-to-image diffusion models. *arXiv preprint arXiv:2305.15798*, 2023.

[38] Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. *arXiv preprint arXiv:2306.00980*, 2023.

[39] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. *arXiv preprint arXiv:2305.16807*, 2023.

[40] Guanqi Zhan, Chuanxia Zheng, Weidi Xie, and Andrew Zisserman. What does stable diffusion know about the 3d scene? *arXiv preprint arXiv:2310.06836*, 2023.

[41] Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11472–11481, 2022.

[42] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. *arXiv preprint arXiv:2210.10960*, 2022.

[43] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

[44] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[45] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[46] 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. *Advances in neural information processing systems*, 30, 2017.

[47] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

[48] Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. Ablating concepts in text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22691–22702, 2023.

[49] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022.

[50] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[51] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation, 2023.

[52] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. 2023.

[53] Ziqi Huang, Tianxing Wu, Yuming Jiang, Kelvin C.K. Chan, and Ziwei Liu. ReVersion: Diffusion-based relation inversion from images. *arXiv preprint arXiv:2303.13495*, 2023.

[54] Enshu Liu, Xuefei Ning, Zinan Lin, Huazhong Yang, and Yu Wang. Oms-dpm: Optimizing the model schedule for diffusion probabilistic models. In *International Conference on Machine Learning*, pages 21915–21936. PMLR, 2023.

[55] Lijiang Li, Huixia Li, Xiawu Zheng, Jie Wu, Xuefeng Xiao, Rui Wang, Min Zheng, Xin Pan, Fei Chao, and Rongrong Ji. Autodiffusion: Training-free optimization of time steps and architectures for automated diffusion model acceleration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7105–7114, 2023.

[56] Shuai Yang, Yukang Chen, Luozhou Wang, Shu Liu, and Yingcong Chen. Denoising diffusion step-aware models. *arXiv preprint arXiv:2310.03337*, 2023.