
MSF: Efficient Diffusion Model via Multi-Scale Latent Factorization

Haohang Xu¹ * Longyu Chen^{1,2} * Yichen Zhang⁴ * Shuangrui Ding³ Zhipeng Zhang⁴ †

¹ Huawei Inc. ² Huazhong University of Science & Technology

³ The Chinese University of Hong Kong

⁴ School of Artificial Intelligence, Shanghai Jiao Tong University

Abstract

While diffusion-based generative models have made significant strides in visual content creation, conventional approaches face computational challenges, especially for high-resolution images, as they denoise the entire image from noisy inputs. This contrasts with signal processing techniques, such as Fourier and wavelet analyses, which often employ hierarchical decompositions. Inspired by such principles, particularly the idea of signal separation, we introduce a diffusion framework leveraging multi-scale latent factorization. Our framework uniquely decomposes the denoising target, typically latent features from a pretrained Variational Autoencoder, into a low-frequency base signal capturing core structural information and a high-frequency residual signal that contributes finer, high-frequency details like textures. This decomposition into base and residual components directly informs our two-stage image generation process, which first produces the low-resolution base, followed by the generation of the high-resolution residual. Our proposed architecture facilitates reduced sampling steps during the residual learning stage, owing to the inherent ease of modeling residual information, which confers advantages over conventional full-resolution generation techniques. This specific approach of decomposing the signal into a base and a residual, conceptually akin to how wavelet analysis can separate different frequency bands, yields a more streamlined and intuitive design distinct from generic hierarchical models. Our method, MSF (Multi-Scale Factorization), demonstrates its effectiveness by achieving FID scores of 2.08 (256×256) and 2.47 (512×512) on class-conditional ImageNet benchmarks, outperforming the DiT baseline (2.27 and 3.04 respectively) while also delivering a $4\times$ speed-up with the same number of sampling steps. Thus, our MSF effectively balances generation quality with computational efficiency through its innovative base-residual decomposition strategy.

1 Introduction

Diffusion models have recently demonstrated impressive performance across a range of generative tasks, including image [44, 40, 29, 10], video [18, 56, 47, 3, 2], and audio synthesis [27, 41, 23, 8]. These models iteratively transform random Gaussian noise into high-quality outputs by progressively applying denoising steps. Despite their effectiveness, conventional diffusion models operate directly on high-resolution images throughout the denoising process, which results in significant computational demands due to the multi-step reverse sampling. This issue is especially pronounced in architectures with $O(n^2)$ computational complexity, such as DiT [39] and FiT [34, 52].

*Equal contributions.

†Corresponding author

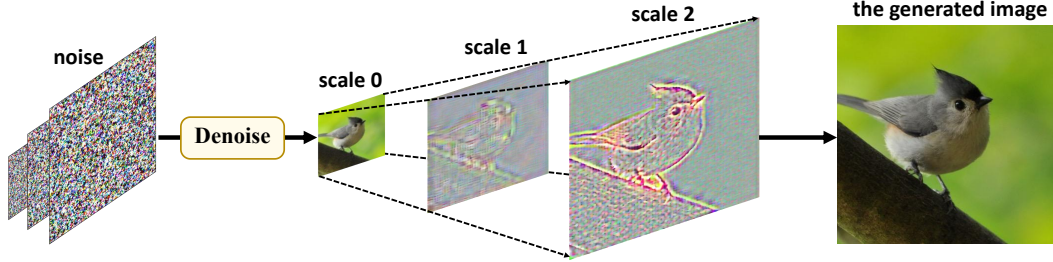


Figure 1: An illustration of MSF, where diffusion models generate hierarchical representations at multiple scales. The multi-scale representations are progressively accumulated to produce the final output. At each scale, the denoising target (scale 0~2) is derived from latents extracted by an off-the-shelf, pretrained Variational Autoencoder (VAE).

To accelerate diffusion models, significant efforts have recently been devoted. One line of research [32, 57, 58, 24, 33] focuses on expediting the reverse sampling process by employing deterministic differential equation solvers. These methods reformulate the reverse diffusion process as either a Stochastic Differential Equation (SDE) [49] or an Ordinary Differential Equation (ODE) [30], thereby enabling the use of various high-order solvers to speed up sampling. However, while these solver-based sampling methods accelerate the generation process, they typically do not address the computational overhead inherent in the diffusion model’s architecture itself. Recently, another alternative branch, namely step distillation techniques [45, 37, 28, 53], has also emerged to expedite sampling. Nevertheless, the distillation process inherently introduces substantial additional training costs and significantly increases GPU memory demands, as it necessitates the concurrent utilization of both student and teacher models during training. These considerations naturally lead to the question that *“Is it possible to innovate the architecture of diffusion models to simultaneously enhance both speed and performance without incurring significant additional costs?”*

Our answer is affirmative. Rather than concentrating solely on sampling strategies, we address the acceleration challenge from a novel perspective by decomposing the overall image generation task into several sub-tasks, each of which can be addressed with a simpler architecture. Specifically, we decompose the generation process into multiple scales, with each scale representing a distinct level of information. The initial scale, which has the lowest resolution, contains the most fundamental information, similar to the low-frequency component in wavelet analysis. Compared to full-resolution generation, these low-resolution components can be trained or sampled more efficiently. Subsequent scales capture increasingly finer residual information between the full-resolution input and the lower-resolution components at each scale, as shown in Fig. 1. As a common wisdom, residual information is often easier to model, allowing us to use fewer sampling steps to process these high-resolution residual components. Based on this, we propose a multi-scale training and sampling pipeline, employing a single DiT backbones to generate images.

To sum up, we decompose the complex full-resolution generation task into two simplified stages: (i) generating a low-resolution base component, and (ii) generating multiple high-resolution residual components. Through experiments, MSF achieves an excellent balance between the generation quality and computational overhead, showing that the combined cost of these two stages is significantly lower than that of the original full-resolution generation task under the premise of better generation quality. In particular, MSF achieves an FID of 2.47 on ImageNet 512×512 , outperforming the baseline DiT [39] with an FID of 3.04, while delivering a speedup of over $4\times$ for the same sampling steps. Moreover, the proposed approach is orthogonal to, and potentially complementary with, existing techniques for accelerating diffusion models. For instance, MSF can effectively integrate with step distillation, further boosting the sampling speed, as detailed in Sec. 4.

The key contributions of this paper are as follows:

- We propose a hierarchical target decomposition strategy for diffusion models, which systematically factorizes the input into a foundational low-resolution base component and multiple high-resolution residual components, each capturing distinct levels of image detail.
- We design a corresponding multi-scale training and sampling pipeline, wherein a unified Diffusion Transformer (DiT) backbone is employed to generate these decomposed target components at their respective spatial scales.

- Extensive experiments validate the effectiveness of MSF, demonstrating notable improvements in computational efficiency, *i.e.*, delivering a speedup of over $4\times$ in sampling than the baseline DiT, while achieving superior generation quality, especially at higher resolutions such as 512×512 .

2 Related Work

Diffusion-based Image Generation. Diffusion-based models [19, 48, 50, 38] emerge as a prominent class of generative methods. These models function by learning to reverse a predefined forward noising process, progressively transforming a noisy input into a clean sample through iterative denoising steps. While this iterative refinement mechanism enables the generation of high-fidelity images and offers superior training stability compared to Generative Adversarial Networks (GANs) [15, 46, 22, 36, 9], it also introduces considerable computational overhead during the sampling phase. To address this challenge, Latent Diffusion Models (LDMs) [43] are introduced, which operate by performing the diffusion process within the compressed latent space of a Variational Autoencoder (VAE [26]). The significant spatial compression provided by the VAE encoder, typically an 8×8 downsampling, leads to substantial improvements in both training and sampling efficiency. This enhanced computational efficiency is instrumental in scaling diffusion models to larger and more complex architectures. Following the LDM paradigm, a subsequent body of work [29, 10, 40, 7, 5, 6] focuses on further advancing these models to produce images of even higher quality and resolution, thereby expanding the frontiers of diffusion-based visual synthesis.

Diffusion Backbone. Diffusion-based generative models predominantly utilize two primary architectural frameworks. Initial methodologies, exemplified by works such as [19, 43], employ traditional U-Net structures as their backbone. To enhance the scalability of these models, DiT [39] introduces the use of Transformers, specifically a Vision Transformer, as the backbone for Latent Diffusion Models (LDMs), replacing the conventional U-Net. Subsequent research further explores Transformer based architectures. For instance, MDT [14] proposes a masked latent modeling technique to improve the learning of internal contextual relationships within images by Diffusion Probabilistic Models (DPMs). U-ViT [1] adopts a unified token-based representation for all inputs and incorporates long skip connections. Building upon the DiT framework, SiT [35] introduces an interpolation scheme and investigates various rectified flow configurations. FiT [34] explores the generation of images at arbitrary resolutions and aspect ratios, and its successor, FiTv2 [52], incorporates additional engineering optimizations to further improve convergence speed and performance. Nevertheless, these aforementioned approaches generate full resolution images at a single scale, which incurs significant computational overhead. In this work, drawing inspiration from frequency decomposition principles and building upon the established DiT architecture, we propose MSF, an efficient diffusion model achieved through multi-scale latent factorization.

Multi-stage Diffusion Models. Cascaded diffusion models, also termed multi-stage diffusion models, are introduced and widely adopted for generating high-quality visual content [20, 44, 18, 56, 47]. This process typically commences with a base diffusion model producing low-resolution images and subsequently cascades to generate high-resolution visual content through a series of supplementary diffusion models and interpolation networks [3, 2]. However, in these frameworks, each stage undertakes a complete generation task, thereby incurring considerable computational costs. PDD [31] applies pyramidal discrete diffusion models to synthesize large-scale 3D scenes under resource constraints, utilizing a coarse-to-fine approach for progressive upscaling and employing scene subdivision techniques. DoD [55] proposes a multi-stage generation framework that enhances diffusion models with visual priors, wherein the initial stage performs standard class-conditional generation, and subsequent stages extract visual priors by encoding the outputs of the preceding stage to guide the diffusion model. Nevertheless, DoD’s multi-stage diffusion model maintains a uniform resolution across all stages, and its initial stage encounters challenges in generating high-resolution images from coarse and limited class information. In contrast, our approach partitions the generation process into multiple scales and progressively refines residual details between the full-resolution input and its lower-resolution counterparts at each stage. This multi-scale residual information is more tractable to model compared to direct full-resolution generation, allowing for a simpler architecture that improves computational efficiency for both training and sampling.

3 Method

In this section, we present a comprehensive description of the proposed MSF method. First, we provide an overarching overview of MSF to contextualize its components in Sec. 3.1. Subsequently, we detail the key modules of MSF as follows: (i) the extraction of multi-scale latent representations using a pretrained variational autoencoder (VAE) in Sec. 3.2; (ii) the multi-scale diffusion network that generates latent representations at each scale in Sec. 3.3; and (iii) the sampling process, which aggregates the residual latent representations from all levels to produce the final output in Sec. 3.4.

3.1 An Overview of MSF

Given an input image $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$, a multi-scale VAE encodes it into a set of tokens $\{\mathbf{f}_0, \mathbf{f}_1, \dots, \mathbf{f}_N\}$, where each token $\mathbf{f}_i \in \mathbb{R}^{h_i \times w_i \times C}$ represents a distinct scale. After we obtain the multi-scale latents $\{\mathbf{f}_0, \mathbf{f}_1, \dots, \mathbf{f}_N\}$, a straightforward approach is using all preceding scales to predict the ground truth latent of the current scale:

$$p(\mathbf{f}_0, \mathbf{f}_1, \dots, \mathbf{f}_N) = \prod_{i=0}^N p(\mathbf{f}_i | \mathbf{f}_0, \dots, \mathbf{f}_{i-1}) \quad (1)$$

where $p(\cdot)$ indicates a diffusion-based next-scale prediction network, conditioned on the preceding scale inputs. In this paper, each $\mathbf{f}_i$ essentially represents a decomposition of the visual signal. To model Eq. 1 more effectively, we scale all latents from previous scales to a unified scale and aggregate them as a condition for predicting the latent at the current scale, or as a prior. We will delve into this process for the priors in Sec. 3.3.

3.2 Multi-scale Variational Autoencoder

Given a pretrained VAE encoder $\mathcal{E}$, we can derive the latent representation $\hat{\mathbf{f}} \in \mathbb{R}^{h \times w \times C}$ via $\hat{\mathbf{f}} = \mathcal{E}(\mathbf{x})$. Various methods are available to factorize the single-scale latent into multi-scale latents:

Scaling Latent A straightforward method for obtaining multi-scale latents involves scaling the latent representation via down-sampling interpolation. This operation can be directly applied to either the VAE latents $\hat{\mathbf{f}}$ or raw RGB inputs $\mathbf{x}$. We formulate these two methods as follows, respectively:

$$\mathbf{f}_i = \text{BILINEAR}(\hat{\mathbf{f}}, h_i, w_i) \quad (2)$$

$$\mathbf{f}_i = \mathcal{E}(\text{BILINEAR}(\mathbf{x}, H_i, W_i)) \quad (3)$$

where $h_i(H_i)$ and $w_i(W_i)$ denote the target height and width.

Scaling Residual Latent. The aforementioned scaling methods generate multi-scale latents from the original single-scale latent through direct interpolation. Though these approaches allow each latent at different scales to represent information from images of varying resolutions, they do not establish an explicit relationship between the generated multi-scale latents. As a result, the multi-scale latents may independently capture different types of information, without any coherent interaction.

Algorithm 1 Multi-scale Residual Latents Extraction

- 1: **Inputs:** high and low resolution image pair $(\mathbf{I}_h, \mathbf{I}_l)$, resolutions $(h_i, w_i)_{i=0}^N$;
 - 2: **Encode:** $\hat{\mathbf{f}} = \mathcal{E}(\mathbf{I}_h)$, $\mathbf{f}_0 = \mathbf{f}_{resi}^{h_0} = \mathcal{E}(\mathbf{I}_l)$;
 - 3: **Initialize:** $\mathbf{R} = \{\mathbf{f}_{resi}^{h_0}\}$;
 - 4: **for** $i = 0$ to $N - 1$ **do**
 - 5: $\hat{\mathbf{f}} = \hat{\mathbf{f}} - \text{Up}(\mathbf{f}_{resi}^{h_i}, h_N, w_N)$;
 - 6: $\mathbf{f}_{resi}^{h_{i+1}} = \text{Down}(\hat{\mathbf{f}}, h_{i+1}, w_{i+1})$;
 - 7: $\mathbf{R} = \mathbf{R} \cup \{\mathbf{f}_{resi}^{h_{i+1}}\}$;
 - 8: **end for**
 - 9: **Return:** multi-scale residual latents $\mathbf{R}$;
-

To ensure the hierarchical relationship of the extracted multi-scale latents, we propose “scaling of residual latents”, which leverages the concept of residual learning, a fundamental technique in the field of deep learning [17]. Within our VAE framework employing residual learning, the first step involves extracting the base component $\mathbf{f}_0$ from the latent representation $\hat{\mathbf{f}}$. The base component should capture the most fundamental, low-level information of $\hat{\mathbf{f}}$, akin to the low-frequency components

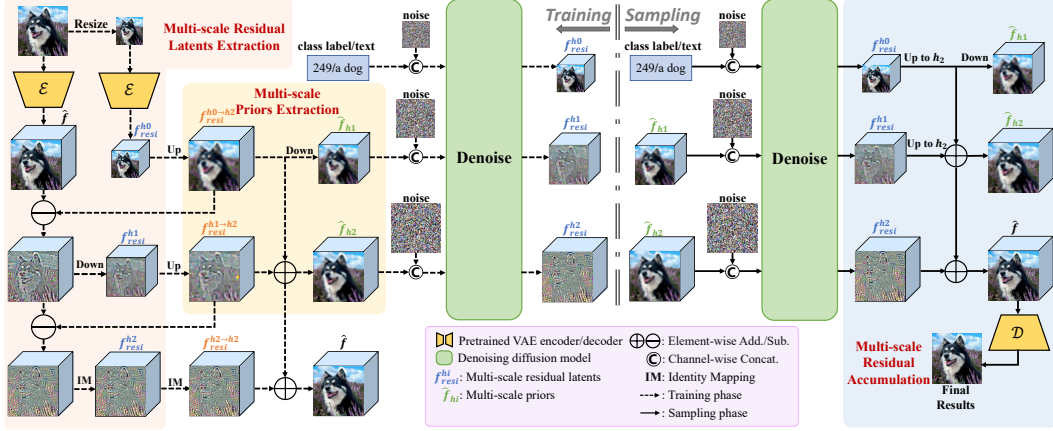


Figure 2: An overview of the proposed MSF. **Left:** The figure on the left illustrates the process of extracting multi-scale residual latents $f_{resi}^{h_i}$ using the pretrained VAE. **Middle:** The middle depicts the construction process of multi-scale priors $\hat{f}_{h_i}$ and how the diffusion model takes $\hat{f}_{h_i}$ as input to generate the ground-truth residual latents $f_{resi}^{h_i}$. **Right:** During sampling, we first generate the residual latents $f_{resi}^{h_i}$, and then perform residual accumulation to obtain the $\hat{f}_{h_i}$, which serves as input for the next-scale prediction.

in Fourier decomposition. We evaluate two methods for obtaining the base component: (1) directly downsampling $\hat{f}$ in the latent space, and (2) first downsampling the input image and then generating its VAE latent. We find that the first method, downsampling in the latent space, results in significant information loss. In contrast, the second method, downsampling the input image before VAE encoding, preserves most of the information contained in $\hat{f}$. Consequently, MSF adopts the second method.

Multi-scale Residual Latents Extraction. Given the original latent representation $\hat{f}$ and its primary component f_0 (denoted as $f_{resi}^{h_0}$ in Fig. 2), we first compute the residual latent at the initial scale as $\hat{f} - \text{Up}(f_0)$, where $\text{Up}(\cdot)$ represents the upsampling operation employed to match the spatial dimensions of $\hat{f}$ and f_0 . To further decompose the first-scale residual latent $\hat{f} - \text{Up}(f_0)$, we derive its fundamental component by applying downsampling $f_{resi}^{h_1} = \text{Down}(\hat{f} - \text{Up}(f_0))$, where $\text{Down}(\cdot)$ represents the downsampling operation. The subsequent steps follow a similar procedure. A detailed description of this process is provided in Alg. 1. During training, the extracted multi-scale residual latents $\mathbf{R} = \{f_{resi}^{h_i}\}_{i=0}^N$ serve as the ground truth for each scale prediction.

3.3 Multi-scale Diffusion Network

Multi-scale Priors Extraction. In our framework, the prior $\hat{f}_{h_i}$ represents the cumulative latent representations from scales 0 to $i - 1$. Notably, the initial scale has no prior and directly produces a low-resolution primary signal from class labels c . It is natural for the model to execute sequentially in the sampling phase. Whereas during the training phase, we utilize teacher-forcing [16], a strategy that trains a model by using ground truth data as input at each step instead of its own predictions, to guide the model in parallel training. The process of extracting multi-scale priors reflects the inverse of the decomposition performed during multi-scale residual latents extraction on the original latent representation $\hat{f}$. Beginning at scale 0, $\hat{f}$ accumulates residuals $f_{resi}^{h_i}$ progressively across scales. It is important to note that these residuals must be upsampled to $f_{resi}^{h_i \rightarrow h_N}$ to match the spatial dimensions of $\hat{f}$. The prior $\hat{f}_{h_i}$ is

Algorithm 2 Multi-scale Priors Extraction

- 1: **Inputs:** multi-scale residual latents $\mathbf{R} = \{f_{resi}^{h_i}\}_{i=0}^N$;
 - 2: **Initialize:** $\hat{f} = 0$, $\mathbf{P} = \{\}$;
 - 3: **for** $i = 0$ to $N - 1$ **do**
 - 4: $f_{resi}^{h_i \rightarrow h_N} = \text{Up}(f_{resi}^{h_i}, h_N, w_N)$;
 - 5: $\hat{f} = \hat{f} + f_{resi}^{h_i \rightarrow h_N}$;
 - 6: $\hat{f}_{h_{i+1}} = \text{Down}(\hat{f}, h_{i+1}, w_{i+1})$;
 - 7: $\mathbf{P} = \mathbf{P} \cup \{\hat{f}_{h_{i+1}}\}$;
 - 8: **end for**
 - 9: **Return:** multi-scale priors $\mathbf{P}$;
-

obtained by downsampling $\hat{\mathbf{f}}$, which has integrated information from scales 0 through $i - 1$, to match the dimensions of the current scale. As illustrated in Fig. 2, the aggregation of latent representations across all scales enables the reconstruction of $\hat{\mathbf{f}}$. The detailed procedure for obtaining the multi-scale priors $\mathbf{P} = \{\hat{\mathbf{f}}_{h_i}\}_{i=1}^N$ during the training phase is elaborated in Alg. 2.

Residual-based Next-scale Prediction. The priors, obtained via Alg. 2, serve as the condition to generate the residual target at the current scale. Specifically, we use the rectified flow [30] model due to its stability during training and fast sampling speed. Rectified flow aims to learn a velocity field v , which transforms random noise $\mathbf{z}_1 \sim \mathcal{N}(0, I)$ into the ground truth distribution $\mathbf{z}_0 = \mathbf{f}_{resi}^{h_i} \sim \pi_{data}$:

$$\mathcal{L} = \sum_{i=0}^N \int_0^1 \mathbb{E}[\|(\mathbf{z}_1 - \mathbf{f}_{resi}^{h_i}) - v_\theta(\mathbf{z}_t, t|c, \mathbf{f}_{resi}^{h_{i-1}})\|^2] dt \quad (4)$$

where N is the number of total scales, c is the class label, and $\mathbf{z}_t = t \cdot \mathbf{z}_1 + (1 - t) \cdot \mathbf{f}_{resi}^{h_i}$. The latents at each scale are encoded with a segment embedding e_i in the transformer to differentiate information across scales. After training, an ODE function is used to sample $\mathbf{f}_{resi}^{h_i}$ at different scales:

$$d\mathbf{z}_t = v_\theta(\mathbf{z}_t, t|c, \mathbf{f}_{resi}^{h_{i-1}})dt \quad (5)$$

3.4 Multi-scale Accumulate Sampling

The sampling process begins with the class label c and computes the residual latent representation $\mathbf{f}_{resi}^{h_i}$ for each scale i in a manner similar to autoregressive methods. The residual latents are accumulated to obtain the final latent representation $\hat{\mathbf{f}}$. Specifically, the initial scale generates a low-resolution primary signal $\mathbf{f}_{resi}^{h_0}$ based on c , while subsequent scales produce high-resolution residual signals conditioned on the class label c and prior $\hat{\mathbf{f}}_{h_i}$. Each computed residual signals $\mathbf{f}_{resi}^{h_i}$ is upsampled to (h_N, w_N) and accumulated into $\hat{\mathbf{f}}$. And then $\hat{\mathbf{f}}$, containing the cumulative results of all current scales, is downsampled to (h_{i+1}, w_{i+1}) and serves as the prior for the next scale. After incorporating the residual latent of the last scale, the VAE decoder generates the image from $\hat{\mathbf{f}}$. The complete procedure for sampling is detailed in Alg. 3.

Algorithm 3 Autoregressive sampling

- 1: **Inputs:** class label c , resolutions $(h_i, w_i)_{i=0}^N$;
 - 2: **Initialize:** $\hat{\mathbf{f}} = 0$;
 - 3: **for** $i = 0$ to N **do**
 - 4: **if** $i == 0$ **then**
 - 5: $\mathbf{f}_{resi}^{h_i} \sim v_\theta(\mathbf{z}_t, t|c)$;
 - 6: **else**
 - 7: $\mathbf{f}_{resi}^{h_i} \sim v_\theta(\mathbf{z}_t, t|c, \hat{\mathbf{f}}_{h_i})$;
 - 8: **end if**
 - 9: $\hat{\mathbf{f}} = \hat{\mathbf{f}} + \text{Up}(\mathbf{f}_{resi}^{h_i}, h_N, w_N)$;
 - 10: **if** $i < N$ **then**
 - 11: $\hat{\mathbf{f}}_{h_{i+1}} = \text{Down}(\hat{\mathbf{f}}, h_{i+1}, w_{i+1})$;
 - 12: **end if**
 - 13: **end for**
 - 14: **Decode:** $\hat{\mathbf{I}}_h = \mathcal{D}(\hat{\mathbf{f}})$;
 - 15: **Return:** generated high-resolution image $\hat{\mathbf{I}}_h$;
-

4 Experiments

In this section, we conduct experiments on ImageNet [11] at the resolutions of 256×256 and 512×512 . We first evaluate our method with both the Fréchet Inception Distance (FID) and Inception Score (IS) using standard evaluation protocols. Subsequently, we compare the sampling time of MSF with the baseline method DiT, demonstrating the effectiveness of MSF. We also analyze the performance of MSF with several detailed ablation studies in Sec. 5.

4.1 Experimental Setup

We train diffusion models using the ImageNet 2012 training dataset. Following Stable Diffusion [43], we employ a VAE encoder with a downsampling ratio of 8 to extract latent representations of input images. We set the number of scales as 2 for both 256×256 and 512×512 generation: the first scale is fixed at 192×192 and the second is determined by the target resolution, either 256×256 or 512×512 . We provide two variants of MSF with different model sizes, MSF-B (Base) with 19 DiT-transformer blocks and MSF-L (Large) with 28 DiT-transformer blocks. The entire training process consists of two stages. In stage 0, we first train scale 0 independently until convergence.

Table 1: Quantitative Comparison of Class-Conditional Generative Methods on ImageNet 256×256 and 512×512 . In this comparison, “ $\downarrow$ ” indicates that lower values are preferable, while “ $\uparrow$ ” indicates that higher values are better. We report results on Fréchet Inception Distance (FID), Inception Score (IS), Precision (Pre), and Recall (Rec). Note that * represents MSF being trained with a patch size of 4 at scale 1, which enables faster training and sampling compared with the default patch size of 2.

Model	Iters (K)	Para	FID $\downarrow$	IS $\uparrow$	Pre $\uparrow$	Rec $\uparrow$
ImageNet 256×256						
VQVAE-2 [42]	2207	13.5B	31.11	~ 45	0.36	0.57
VDM++ [25]	—	2B	2.12	267.7	—	—
ViTVQ [54]	950	1.7B	4.17	175.1	—	—
VQGAN [13]	—	1.4B	15.78	74.3	—	—
ADM [12]	1980	554M	10.94	100.98	0.69	0.63
ADM-G [12]	1980	554M	4.59	186.7	0.82	0.52
LDM-8-G [43]	4800	506M	7.76	209.52	0.84	0.35
LDM-4-G [43]	178	400M	3.60	247.67	0.87	0.48
GIVT-Causal [51]	—	304M	5.67	—	0.75	0.59
MaskGIT [4]	1500	227M	6.18	182.1	0.80	0.51
CDM [20]	1600	—	4.88	158.7	—	—
DiT-L [39]	—	458M	5.02	167.2	0.75	0.57
MSF-B	1100 + 125	368M	2.76	218.3	0.84	0.55
DiT-XL [39]	7000	675M	2.27	278.2	0.83	0.57
MSF-L	1025 + 200	680M	2.08	245.5	0.83	0.57
ImageNet 512×512						
VDM++ [25]	—	2B	2.65	278.1	—	—
ADM [12]	1940	554M	23.24	58.06	0.73	0.60
ADM-G [12]	1940	554M	7.72	172.71	0.87	0.42
DiT-XL [39]	3000	675M	3.04	240.82	0.84	0.54
MSF-L*	1025 + 450	680M	2.58	252.8	0.83	0.54
MSF-L	1025 + 200	680M	2.47	249.1	0.83	0.53

Once converged, we proceed to stage 1, where scale 1 is trained jointly with scale 0. During this stage, the training losses from both scales are simply summed and backpropagated together. The iteration counts reported in Tab. 1 refer to the total number of iterations across both stages.

Due to GPU memory limitations, we set the batch size to 1024 for scale 0 and 512 for scale 1. The learning rate is fixed at 1×10^{-4} . As shown in Tab. 1, the number of training iterations required for scale 1 is approximately 1/4 of that for scale 0, since scale 1 only models the residual information.

4.2 Comparison with State-of-the-arts.

Results on ImageNet. This section presents a comparison of our method against other works on the ImageNet dataset, evaluated at 256×256 and 512×512 resolutions. We evaluate MSF using several standard metrics, including Fréchet Inception Distance (FID), Inception Score (IS), Precision, and Recall. For a fair comparison, we adopt evaluation scripts from ADM [12] and report results on 50K generated samples across 1000 ImageNet categories. We set the sampling steps for the two scales to 100 and 20, and the classifier-free guidance scales to 1.3 and 1.0 (*i.e.*, without CFG). As shown in Tab. 1, MSF achieves an excellent balance between the generation quality and computational overhead, with FID scores of 2.08 and 2.47 on ImageNet 256×256 and 512×512 , respectively, clearly outperforming DiT-L and DiT-XL baselines. We also train the MSF-L* variant for 512×512 generation, which uses a transformer patch size of 4 at scale 1. In this setting, the number of default image tokens at scale 1 is reduced from $(512/8/2)^2 = 1024$ to $(512/8/4)^2 = 256$ (where 8 is the VAE downsample size). Remarkably, even with the token count reduced to 1/4, MSF maintains strong generation performance, which significantly improves both training and sampling speed. We also provide the visualizations of samples generated by MSF in Fig. 3.

Sampling Speed Compared with DiT Beyond superior generation quality, MSF also accelerates sampling significantly. For high-resolution (512×512) image synthesis, we first perform an efficient sampling stage at scale 0 (192×192 resolution), followed by only a few steps to generate the residual information at scale 1. Notably, we observe that classifier-free guidance is not necessary at scale 1, eliminating the need to double the batch size for conditional and unconditional generations.

As shown in the top section of Tab. 2, MSF-L generates 64 images at 512×512 resolution in 76 seconds, whereas DiT requires 326 seconds with similar sampling steps, showing our MSF is more than $4\times$ faster than DiT. Even increasing the sampling steps of DiT to 250, our model still demonstrates better performance and runs at $11\times$ faster. Notably, the sampling steps at scale 1 can be significantly reduced. For example, using only 4 steps at scale 1 still achieves FID comparable to DiT.

As mentioned earlier, in the original MSF model, the majority of sampling cost lies in the scale 0 generation. As previously discussed, our approach is complementary to other acceleration techniques. To further improve efficiency, we integrate piecewise rectified flow (PeRFlow) [53] as a plug-and-play accelerator for scale 0 sampling (see "step distillation" in the Tab. 2). Following PeRFlow’s methodology, we divide the scale 0 sampling trajectory into 4 time windows and apply a reflow operation to straighten the flow trajectories within each window. After applying step distillation at scale 0, MSF-L achieves a sampling speed of 16 *imgs/sec* while maintaining competitive generation performance. Visualizations of the distilled MSF outputs are provided in the supplementary material.

Table 2: The generation results with few sampling steps of MSF-L on ImageNet 512×512 . The sampling steps of MSF is measured as $s_0 + s_1$, where s_0 is the steps of scale 0 and s_1 is the steps of scale 1. The sampling time is measured on a A800 GPU with batch size of 64.

Model	Steps	FID-5K	FID-50K	Time(s)
DiT-XL	250	—	3.04	816
DiT-XL	100	—	—	326
MSF-L	100 + 20	5.07	2.47	76
MSF-L	100 + 10	5.23	2.58	70
MSF-L	100 + 4	6.54	3.19	68
MSF-L	50 + 20	5.32	2.51	48
MSF-L	50 + 4	6.67	3.24	36
MSF-L*	100 + 20	5.33	2.58	67
Step Distillation				
MSF-L	10 + 10	6.79	3.91	11
MSF-L	5 + 10	7.37	4.35	8
MSF-L	5 + 4	8.16	4.90	4

5 Analysis and Discussion

Effects of Classifier-Free Guidance Scale. Classifier-free guidance [21] (CFG) is a core technique to enhance the generative quality of diffusion models by interpolating between conditional and unconditional outputs. We evaluate the outputs of each scale under varying CFG individually, as shown in Tab. 3a. With the CFG scale of scale 1 fixed at 1.0, we evaluate FID-5K for scale 0 under different CFG, finding the best result at 1.3 and the performance of scale 0 generation primarily impacted by CFG. Then, fixing the CFG scale of scale 0 at 1.3, we execute the same operation for scale 1, observing that scale 1 achieves satisfactory results even without CFG (Tab. 3b). Since CFG requires doubling the forward passes of the inputs and introduces additional sampling costs, we omit CFG for scale 1 generation (*i.e.*, CFG=1.0 for scale 1) to achieve a $2\times$ speedup in sampling.

Table 3: FID-5K vs. CFG scale. (a) CFG at scale 1 is fixed to 1.0 while varying at scale 0. (b) CFG at scale 0 is fixed to 1.3 while varying at scale 1.

(a)			(b)		
CFG		FID-5K	CFG		FID-5K
scale 0	scale 1		scale 0	scale 1	
1.0	1.0	9.77	1.3	1.0	5.23
1.1	1.0	6.88	1.3	1.1	5.23
1.2	1.0	5.51	1.3	1.2	5.23
1.3	1.0	5.23	1.3	1.3	5.22
1.4	1.0	5.48	1.3	1.4	5.22
1.5	1.0	6.16	1.3	1.5	5.22

Effects of the Number of Scales. In MSF, we propose dividing the image generation into multiple stages, each addressing a specific scale. To explore how the number of scales affects the model’s generation capability, we conduct experiments with three scales ($192 \rightarrow 384 \rightarrow 512$) besides the two-scale experiments. Tab. 4 shows CFG of the first scale 0, with other scales set to CFG = 1, and



Figure 3: Some generated 256×256 and 512×512 samples by MSF trained on ImageNet.

the sampling steps of all scales. The results indicate that the generation quality degrades compared to the two-scale setting. We hypothesize that this is due to error accumulation as the number of scales increases. Therefore, we recommend using a two-scale configuration for MSF.

6 Conclusion

In this paper, we propose a versatile and efficient image generation framework that decomposes the full-resolution image generation process into a multi-scale and residual-based, progressive hierarchy. Each scale within the hierarchy captures distinct image details, with the framework learning both a fundamental low-resolution component and several residual high-resolution components. This structure enables to model the multi-scale information pro-

Table 4: Performance comparison across different numbers of scales, The patch size of scale 1 and scale 2 are set as 4, the same as the MSF-L* setting in Tab. 1

Model	CFG	Steps	Scales	FID-5K
MSF-L*	1.3	100 + 10	192 $\rightarrow$ 512	5.44
MSF-L*	1.2	100 + 10 + 10	192 $\rightarrow$ 384 $\rightarrow$ 512	9.44
MSF-L*	1.3	100 + 10 + 10	192 $\rightarrow$ 384 $\rightarrow$ 512	8.29
MSF-L*	1.4	100 + 10 + 10	192 $\rightarrow$ 384 $\rightarrow$ 512	7.81
MSF-L*	1.5	100 + 10 + 10	192 $\rightarrow$ 384 $\rightarrow$ 512	7.84
MSF-L*	1.6	100 + 10 + 10	192 $\rightarrow$ 384 $\rightarrow$ 512	8.09

gressively, leading to substantial efficiency gains without compromising output quality. Through extensive experiments, we demonstrated that such multi-scale factorization approach enhances the generative quality, achieving higher fidelity with fewer computational resources than traditional methods, validating the effectiveness and efficiency of MSF in high-resolution image synthesis. Currently, MSF is exclusively evaluated on class-conditional image generation tasks. In future work, we plan to extend its framework to text-to-image generation, enabling multimodal conditional synthesis.

References

- [1] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22669–22679, 2023.
- [2] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.
- [3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023.
- [4] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11305–11315, 2022.
- [5] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart- σ : Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv preprint arXiv:2403.04692*, 2024.
- [6] Junsong Chen, Yue Wu, Simian Luo, Enze Xie, Sayak Paul, Ping Luo, Hang Zhao, and Zhenguo Li. Pixart- δ : Fast and controllable image generation with latent consistency models. *arXiv preprint arXiv:2401.05252*, 2024.
- [7] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.
- [8] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, Najim Dehak, and William Chan. Wavegrad 2: Iterative refinement for text-to-speech synthesis. *ArXiv*, abs/2106.09660, 2021.
- [9] Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.
- [10] 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.
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- [13] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021.
- [14] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23164–23173, 2023.
- [15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

- [16] Sebastian Goodman, Nan Ding, and Radu Soricut. TeaForN: Teacher-forcing with n-grams. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8704–8717, Online, November 2020. Association for Computational Linguistics.
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [18] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.
- [19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [20] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.
- [21] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [22] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10124–10134, 2023.
- [23] Heeseung Kim, Sungwon Kim, and Sungroh Yoon. Guided-tts: A diffusion model for text-to-speech via classifier guidance. In *International Conference on Machine Learning*, 2021.
- [24] Sanghwan Kim, Hao Tang, and Fisher Yu. Distilling ode solvers of diffusion models into smaller steps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9410–9419, 2024.
- [25] Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36, 2024.
- [26] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [27] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*.
- [28] Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation, 2024.
- [29] Bingchen Liu, Ehsan Akhgari, Alexander Visheratin, Aleks Kamko, Linmiao Xu, Shivam Shrirao, Joao Souza, Suhail Doshi, and Daiqing Li. Playground v3: Improving text-to-image alignment with deep-fusion large language models. *arXiv preprint arXiv:2409.10695*, 2024.
- [30] Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*.
- [31] Yuheng Liu, Xinke Li, Xueting Li, Lu Qi, Chongshou Li, and Ming-Hsuan Yang. Pyramid diffusion for fine 3d large scene generation. *arXiv preprint arXiv:2311.12085*, 2023.
- [32] 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.
- [33] 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.
- [34] Zeyu Lu, Zidong Wang, Di Huang, Chengyue Wu, Xihui Liu, Wanli Ouyang, and Lei Bai. Fit: Flexible vision transformer for diffusion model. *arXiv preprint arXiv:2402.12376*, 2024.

- [35] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. *arXiv preprint arXiv:2401.08740*, 2024.
- [36] Mehdi Mirza. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- [37] Thuan Hoang Nguyen and Anh Tran. Swiftbrush: One-step text-to-image diffusion model with variational score distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7807–7816, June 2024.
- [38] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.
- [39] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.
- [40] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.
- [41] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail A. Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In *International Conference on Machine Learning*, 2021.
- [42] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Neural Information Processing Systems*, 2019.
- [43] 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.
- [44] 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.
- [45] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.
- [46] 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.
- [47] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*.
- [48] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [49] 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.
- [50] 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.
- [51] Michael Tschannen, Cian Eastwood, and Fabian Mentzer. Givit: Generative infinite-vocabulary transformers. In *European Conference on Computer Vision*, pages 292–309. Springer, 2025.
- [52] ZiDong Wang, Zeyu Lu, Di Huang, Cai Zhou, Wanli Ouyang, et al. Fitv2: Scalable and improved flexible vision transformer for diffusion model. *arXiv preprint arXiv:2410.13925*, 2024.

- [53] Hanshu Yan, Xingchao Liu, Jiachun Pan, Jun Hao Liew, qiang liu, and Jiashi Feng. PeRFlow: Piecewise rectified flow as universal plug-and-play accelerator. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [54] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *ArXiv*, abs/2110.04627, 2021.
- [55] Xiaoyu Yue, Zidong Wang, Zeyu Lu, Shuyang Sun, Meng Wei, Wanli Ouyang, Lei Bai, and Luping Zhou. Diffusion models need visual priors for image generation. *arXiv preprint arXiv:2410.08531*, 2024.
- [56] David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *International Journal of Computer Vision*, pages 1–15, 2024.
- [57] Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. *Advances in Neural Information Processing Systems*, 36:55502–55542, 2023.
- [58] Zhenyu Zhou, Defang Chen, Can Wang, and Chun Chen. Fast ode-based sampling for diffusion models in around 5 steps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7777–7786, 2024.