Efficient Diffusion Training via Min-SNR Weighting Strategy

Tiankai Hang[1], Shuyang Gu[2]*, Chen Li[3], Jianmin Bao[2], Dong Chen[2],
Han Hu[2], Xin Geng[1], Baining Guo[1]*
[1]Southeast University, [2]Microsoft Research Asia,
[3]National Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
National Engineering Research Center for Visual Information and Applications,
and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
{tkhang,xgeng,307000167}@seu.edu.cn,{shuyanggu,t-chenli1,jianmin.bao,doch,hanhu}@microsoft.com

Abstract

Denoising diffusion models have been a mainstream approach for image generation, however, training these models often suffers from slow convergence. In this paper, we discovered that the slow convergence is partly due to conflicting optimization directions between timesteps. To address this issue, we treat the diffusion training as a multi-task learning problem, and introduce a simple yet effective approach referred to as Min-SNR-γ. This method adapts loss weights of timesteps based on clamped signal-to-noise ratios, which effectively balances the conflicts among timesteps. Our results demonstrate a significant improvement in converging speed, 3.4$\times$ faster than previous weighting strategies. It is also more effective, achieving a new record FID score of 2.06 on the ImageNet 256 $\times$ 256 benchmark using smaller architectures than that employed in previous state-of-the-art. The code is available at https://github.com/TiankaiHang/Min-SNR-Diffusion-Training.

1. Introduction

In recent years, denoising diffusion models [51, 20, 62, 38] have emerged as a promising new class of deep generative models due to their remarkable ability to model complicated distributions. Compared to prior Generative Adversarial Networks (GANs), diffusion models have demonstrated superior performance across a range of generation tasks in various modalities, including text-to-image generation [42, 46, 44, 18], image manipulation [28, 36, 4, 61], video synthesis [19, 50, 24], text generation [30, 17, 64], 3D avatar synthesis [41, 58], etc. A key limitation of present denoising diffusion models is their slow convergence rate, requiring substantial amounts of GPU hours for training [44, 43]. This constitutes a considerable challenge for researchers seeking to effectively experiment with these models.

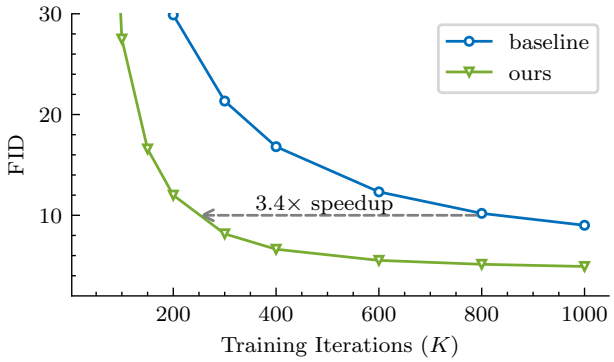

Figure 1: By leveraging a non-conflicting weighting strategy, our method can converge 3.4 times faster than baseline, resulting in superior performance.

In this paper, we first conducted a thorough examination of this issue, revealing that the slow convergence rate likely arises from conflicting optimization directions for different timesteps during training. In fact, we find that by dedicatedly optimizing the denoising function for a specific noise level can even harm the reconstruction performance for other noise levels, as shown in Figure 2. This indicates that the optimal weight gradients for different noise levels are in conflict with one another. Given that current denoising diffusion models [20, 12, 38, 44] employ shared model weights for various noise levels, the conflicting weight gradients will impede the overall convergence rate, if without careful consideration on the balance of these noise timesteps.

To tackle this problem, we propose the **Min-SNR-γ loss weighting strategy**. This strategy treats the denoising process of each timestep as an individual task, thus diffusion

*Corresponding authors.

training can be considered as a multi-task learning problem. To balance various tasks, we assign loss weights for each task according to their difficulty. Specifically, we adopt a clamped signal-to-noise ratio (SNR) as loss weight to alleviate the conflicting gradients issue. By organizing various timesteps using this new weighting strategy, the diffusion training process can converge much faster than previous approaches, as illustrated in Figure 1.

Generic multi-task learning methods usually seek to mitigate conflicts between tasks by adjusting the loss weight of each task based on their gradients. One classical approach [11, 49], Pareto optimization, aims to seek a gradient descent direction to improve all the tasks. However, these approaches differ from our Min-SNR-γ weighting strategy in three aspects: 1) **Sparsity**. Most previous studies in the generic multi-task learning field have focused on scenarios with a small number of tasks, which differs from the diffusion training where the number of tasks can be up to thousands. As in our experiments, Pareto optimal solutions in diffusion training tend to set loss weights of most timesteps as 0. In this way, many timesteps will be left without any learning, and thus harm the entire denoising process. 2) **Instability**. The gradients computed for each timestep in each iteration are often noisy, owing to a limited number of samples for each timestep. This hampers the accurate computation of Pareto optimal solutions. 3) **Inefficiency**. The calculation of Pareto optimal solutions is time-consuming, significantly slowing down the overall training.

Our proposed Min-SNR-γ strategy is a predefined global step-wise loss weighting setting, instead of run-time adaptive loss weights for each iteration as in the original Pareto optimization, thus avoiding the sparsity issue. Moreover, the global loss weighting strategy eliminates the need for noisy computation of gradients and the time-consuming Pareto optimization process, making it more efficient and stable. Though suboptimal, the global strategy can be also almost as effective: Firstly, the optimization dynamics of each denoising task are largely shaped by the task's noise level, without the need to account for individual samples too much. Secondly, after a moderate number of iterations, the gradients of the majority subsequent training process become more stable, thus it can be approximated by a stationery weighting strategy.

To validate the effectiveness of the Min-SNR-γ weighting strategy, we first compute its Pareto objective value and compare it with the optimal step-wise loss weights obtained by directly solving the Pareto problem. Together, we also compare it with several conventional loss weighting strategies, including constant weighting, SNR weighting, and SNR with an lower bound. Figure 4 shows that our Min-SNR-γ weighting strategy produces Pareto objective values almost as low as the optimal one, significantly better than other existing works, indicating a significant alleviation of the gradient conflicting issue. As a result, the proposed weighting strategy not only converges much faster than previous approaches, but is also effective and general for various generation scenarios. It achieves a new record of FID score 2.06 on the ImageNet 256×256 benchmark, and proves to also improve models using other prediction targets and network architectures.

Our contributions are summarized as follows:

- We have uncovered a compelling explanation for the slow convergence issue in diffusion training: a conflict in gradients across various timesteps.
- We have proposed a new loss weighting strategy for diffusion model training, which greatly mitigates the conflicting gradients across timesteps and results in a marked acceleration of convergence speed.
- We have established a new FID score record on the ImageNet 256×256 image generation benchmark.

2. Related Works

Denoising Diffusion Models. Diffusion models [20, 53, 12] are strong generative models, particularly in the field of image generation, due to their ability to model complex distributions. This advantage has led to superiority over previous GAN models in terms of both high-fidelity and diversity of generated images [12, 26, 37, 42, 44, 46]. Besides, diffusion models also show great success in text-to-video generation [19, 50, 56], 3D Avatar generation [41, 58], image to image translation [39], image manipulation [4, 28], music generation [25], and even drug discovery [60]. The most widely used network structure for diffusion models in the field of image generation is UNet [20, 12, 37, 38]. Recently, researchers have also explored the use of Vision Transformers [14] as an alternative, with U-ViT [2] borrowing the skip connection design from UNet [45] and DiT [40] leveraging Adaptive LayerNorm and discovering that the zero initialization strategy is critical for achieving state-of-the-art class-conditional ImageNet generation results.

Improved Diffusion Models. Recent studies have tried to improve the diffusion models from different perspectives. Some works aim to improve the quality of generated images by guiding the sampling process [13, 23]. Other studies propose fast sampling methods that require only a dozen steps [52, 31, 34, 26] to generating high-quality images. Some works have further distilled the diffusion models for even fewer steps in the sampling process [47, 35]. Meanwhile, some researchers [20, 26, 6] have noticed that the noise schedule is important for diffusion models. Other works [38, 47] have found that different predicting targets from denoising networks affect the training stability and final performance. Finally, some works [15, 1] have proposed using the Mixture of Experts (MoE) approach to han-

dle noise from different levels, which can boost the performance of diffusion models, but require a larger number of parameters and longer training time.

Multi-task Learning. The goal of Multi-task learning (MTL) is to learn multiple related tasks jointly so that the knowledge contained in a task can be leveraged by other tasks. One of the main challenges in MTL is negative transfer [9], means the joint training of tasks hurts learning instead of helping it. From an optimization perspective, it manifests as the presence of conflicting task gradients. To address this issue, some previous works [63, 59, 8] try to modulate the gradient to prevent conflicts. Meanwhile, other works attempt to balance different tasks through carefully design the loss weights [7, 27]. GradNorm [7] considers loss weight as learnable parameters and updates them through gradient descent. Another approach MTO [11, 49] regards the multi-task learning problem as a multi-objective optimization problem and obtains the loss weights by solving a quadratic programming problem.

3. Method

3.1. Preliminary

Diffusion models consist of two processes: a forward noising process and a reverse denoising process. We denote the distribution of training data as $p(\mathbf{x}_0)$. The forward process is a Gaussian transition, gradually adds noise with different scales to a real data point $\mathbf{x}_0 \sim p(\mathbf{x}_0)$ to obtain a series of noisy latent variables $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \alpha_t \mathbf{x}_0, \sigma_t^2 \mathbf{I}) \tag{1}$$

$$\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon} \tag{2}$$

where $\boldsymbol{\epsilon}$ is the noise sampled from Gaussian distribution $\mathcal{N}(0, \mathbf{I})$. The noise schedule σ_t denotes the magnitude of noise added to the clean data at t timestep. It increases monotonically with t. In this paper, we adopt the standard variance-preserving diffusion process, where $\alpha_t = \sqrt{1-\sigma_t^2}$.

The reverse process is parameterized by another Gaussian transition, gradually denoises the latent variables and restores the real data $\mathbf{x}_0$ from a Gaussian noise:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \hat{\mu}_\theta(\mathbf{x}_t), \hat{\Sigma}_\theta(\mathbf{x}_t)). \tag{3}$$

$\hat{\mu}_\theta$ and $\hat{\Sigma}_\theta$ are predicted statistics. Ho et al. [20] set $\hat{\Sigma}_\theta(\mathbf{x}_t)$ to the constant $\sigma_t^2 \mathbf{I}$, and $\hat{\mu}_\theta$ can be decomposed into the linear combination of $\mathbf{x}_t$ and a noise approximation model $\hat{\epsilon}_\theta$. They find using a network to predict noise ϵ works well, especially when combined with a simple re-weighted loss function:

$$\mathcal{L}^t_{\text{simple}}(\theta) = \mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\|\epsilon - \hat{\epsilon}_\theta(\alpha_t \mathbf{x}_0 + \sigma_t \epsilon)\|_2^2\right]. \tag{4}$$

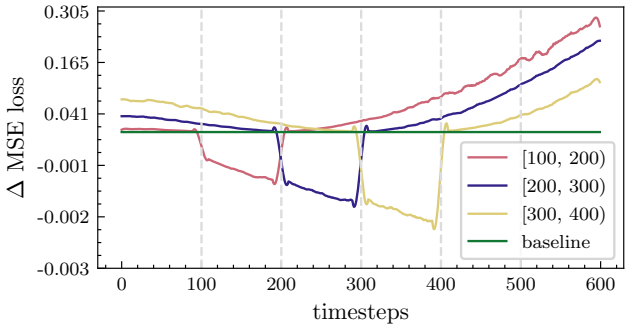

Figure 2: We finetune the diffusion model in specific ranges of timesteps:[100, 200), [200, 300), and [300, 400), then we investigate how it affects the loss in different timesteps. The surrounding timesteps may derive benefit from it, while others may experience adverse effects.

Most previous works [38, 12, 37] follow this strategy and predict the noise. Later works [18, 47] use another reparameterization that predicts the noiseless state x_0:

$$\mathcal{L}^t_{\text{simple}}(\theta) = \mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\|\mathbf{x}_0 - \hat{\mathbf{x}}_\theta(\alpha_t \mathbf{x}_0 + \sigma_t \epsilon)\|_2^2\right]. \tag{5}$$

And some other works [47, 44] even employ the network to directly predict velocity v. Despite their prediction targets being different, we can derive that they are mathematically equivalent by modifying their loss weights.

3.2. Diffusion Training as Multi-Task Learning

To reduce the number of parameters, previous studies [20, 38, 12] often share the parameters of the denoising models across all steps. However, it's important to keep in mind that different steps may have vastly different requirements. At each step of a diffusion model, the strength of the denoising varies. For example, easier denoising tasks (when $t \to 0$) may require simple reconstructions of the input in order to achieve lower denoising loss. This strategy, unfortunately, does not work as well for noisier tasks (when $t \to T$). Thus, it's extremely important to analyze the correlation between different timesteps.

In this regard, we conduct a simple experiment. We begin by clustering the denoising process into several separate bins. Then we finetune the diffusion model by sampling timesteps in each bin. Lastly, we evaluate its effectiveness by looking at how it impacted the loss of other bins. As shown in Figure 2, we can observe that finetuning specific steps benefited those surrounding steps. However, it's often detrimental for other steps that are far away. This inspires us to consider whether *we can find a more efficient solution that benefits all timesteps simultaneously*.

We re-organized our goal from the perspective of multitask learning. The training process of denoising diffusion models contains T different tasks, each task repre-

sents an individual timestep. We denote the model parameters as θ and the corresponding training loss is $\mathcal{L}^t(\theta), t \in \{1, 2, \ldots, T\}$. Our goal is to find a update direction $\delta \neq 0$, that satisfies:

$$\mathcal{L}^t(\theta + \delta) \leq \mathcal{L}^t(\theta), \forall t \in \{1, 2, \ldots, T\}. \tag{6}$$

We consider the first-order Taylor expansion:

$$\mathcal{L}^t(\theta + \delta) \approx \mathcal{L}^t(\theta) + \left\langle \delta, \nabla_\theta \mathcal{L}^t(\theta) \right\rangle. \tag{7}$$

Thus, the ideal update direction is equivalent to satisfy:

$$\left\langle \delta, \nabla_\theta \mathcal{L}^t(\theta) \right\rangle \leq 0, \forall t \in \{1, 2, \ldots, T\}. \tag{8}$$

3.3. Pareto optimality of diffusion models

Theorem 1 *Consider a update direction δ^*:*

$$\delta^* = -\sum_{t=1}^{T} w_t \nabla_\theta \mathcal{L}^t(\theta), \tag{9}$$

of which w_t is the solution to the optimization problem:

$$\min_{w^t} \left\{ \| \sum_{t=1}^{T} w^t \nabla_\theta \mathcal{L}^t(\theta) \|^2 | \sum_{t=1}^{T} w^t = 1, w^t \geq 0 \right\} \tag{10}$$

If the optimal solution to the Equation 8 exists, then δ^* should satisfy it. Otherwise, it means that we must sacrifice a certain task in exchange for the loss decrease of other tasks. In other words, we have reached the Pareto Stationary and the training has converged.

A more general form of this theorem was first proposed in [11] and we leave a succinct proof in the appendix. Since diffusion models are required to go through all the timesteps when generating images. So any timestep should not be ignored during training. Consequently, a regularization term is included to prevent the loss weights from becoming excessively small. The optimization goal in Equation 10 becomes:

$$\min_{w_t} \left\{ \| \sum_{t=1}^{T} w_t \nabla_\theta \mathcal{L}^t(\theta) \|_2^2 + \lambda \sum_{t=1}^{T} \| w_t \|_2^2 \right\} \tag{11}$$

where λ controls the regularization strength.

To solve Equation 11, [49] leverages the Frank-Wolfe [16] algorithm to obtain the weight $\{w_t\}$ through iterative optimization. Another approach is to adopt Unconstrained Gradient Descent(UGD). Specifically, we re-parameterize w_t through β_t:

$$w_t = \frac{e^{\beta_t}}{Z}, Z = \sum_t e^{\beta_t}, \beta_t \in \mathbb{R}. \tag{12}$$

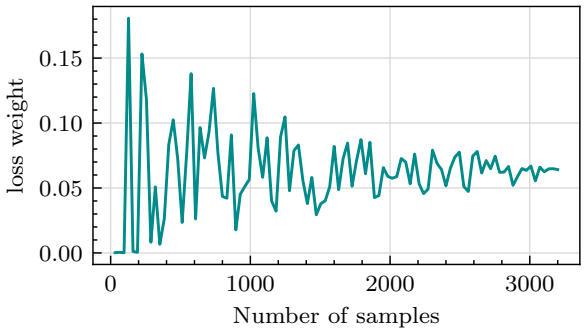

Figure 3: Demonstration of the instability of optimization-based weighting strategy. As the number of samples increases, the loss weight becomes stable, while the computation cost increases.

Combined with Equation 11, we can use gradient descent to optimize each term independently:

$$\min_{\beta_t} \frac{1}{Z^2} \| \sum_{t=1}^{T} e^{\beta_t} \nabla_\theta \mathcal{L}_t(\theta) \|_2^2 + \frac{\lambda}{Z^2} \sum_{t=1}^{T} \| e^{\beta_t} \|_2^2 \tag{13}$$

However, whether leveraging the Frank-Wolfe or the UGD algorithm, there are two disadvantages: 1) *Inefficiency*. Both of these two methods need additional optimization at each training iteration, it greatly increases the training cost. 2) *Instability*. In practice, by using a limited number of samples to calculate the gradient term $\nabla_\theta \mathcal{L}^t(\theta)$, the optimization results are unstable(as shown in Figure 3). In other words, the loss weights for each denoising task vary greatly during training, making the entire diffusion training inefficient.

3.4. Min-SNR-γ Loss Weight Strategy

In order to avoid the inefficiency and instability caused by the iterative optimization in each iteration, one possible attempt is to adopt a stationery loss weight strategy.

To simplify the discussion, we assume that the network is reparametered to predict the noiseless state $\mathbf{x}_0$. However, it's worth noting that different prediction objectives can be transformed into one another, we will delve into it in Section 4.2. Now, we consider the following alternative training loss weights:

- Constant weighting. $w_t = 1$. Which treats different tasks as equally weighted and has been used in both discrete diffusion models [18, 55] and continuous diffusion models [5].
- SNR weighting. $w_t = \text{SNR}(t)$, where $\text{SNR}(t) = \alpha_t^2 / \sigma_t^2$. It's the most widely used weighting strategy [35, 24, 12, 44]. By combining with Equation 2, we can find it's numerically equivalent to the constant weighting strategy when the predicting target is noise.

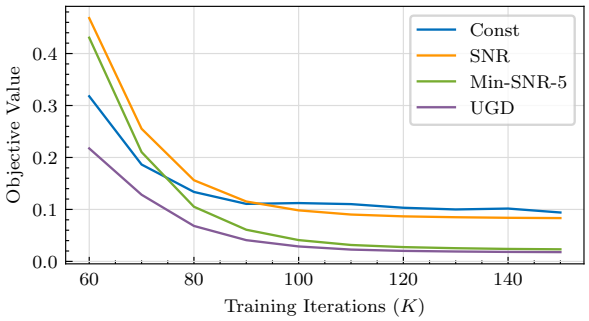

Figure 4: Comparison of the objective values in Equation 11 on different weighting strategies.

- Max-SNR-γ weighting. $w_t = \max\{\text{SNR}(t), \gamma\}$. This modification of SNR weighting is first proposed in [47] to avoid a weight of zero with zero SNR steps. They set $\gamma = 1$ as their default setting. However, the weights still concentrate on small noise levels.

- Min-SNR-γ weighting. $w_t = \min\{\text{SNR}(t), \gamma\}$. We propose this weighting strategy to avoid the model focusing too much on small noise levels.

- UGD optimization weighting. w_t is optimized from Equation 13 in each timestep. Compared with the previous setting, this strategy changes during training.

First, we combine these weighting strategies into Equation 11 to validate whether they are approach to the Pareto optimality state. As shown in Figure 4, the UGD optimization weighting strategy can achieve the lowest score on our optimization target. In addition, the Min-SNR-γ weighting strategy is the closest to the optimum, demonstrating it has the property to optimize different timesteps simultaneously.

In the following section, we present experimental results to demonstrate the effectiveness of our Min-SNR-γ weighting strategy in balancing diverse noise levels. Our approach aims to achieve faster convergence and strong performance.

4. Experiments

In this section, we first provide an overview of the experimental setup. Subsequently, we conduct comprehensive ablation studies to show that our method is versatile and suitable for various prediction targets and network architectures. Finally, we compare our approach to the state-of-the-art methods across multiple image generation benchmarks, demonstrating not only its accelerated convergence but also its superior capability in generating high-quality images.

4.1. Setup

Datasets. We perform experiments on both unconditional and conditional image generation using the CelebA dataset [32] and the ImageNet dataset [10]. The CelebA dataset, which comprises 162,770 human faces, is a widely-used resource for unconditional image generation studies. We follow ScoreSDE [62] for data pre-processing, which involves center cropping each image to a resolution of 140×140 and then resizing it to 64×64. For the class conditional image generation, we adopt the ImageNet dataset [10] with a total of 1.3 million images from 1000 different classes. We test the performance on both 64×64 and 256×256 resolutions.

Training Details. For low resolution (64×64) image generation, we follow ADM [12] and directly train the diffusion model on the pixel-level. For high-resolution image generation, we utilize LDM [44] approach by first compressing the images into latent space, then training a diffusion model to model the latent distributions. To obtain the latent for images, we employ VQ-VAE from Stable Diffusion[1], which encodes a high-resolution image ($256 \times 256 \times 3$) into $32 \times 32 \times 4$ latent codes.

In our experiments, we employ both ViT and UNet as our diffusion model backbones. We adopt a vanilla ViT structure without any modifications [14] as our default setting. we incorporate the timestep t and class condition $\mathbf{c}$ as learnable input tokens to the model. Although further customization of the network structure may improve performance, our focus in this paper is to analyze the general properties of diffusion models. For the UNet structure, we follow ADM [12] and keep the FLOPs similar to the ViT-B model, which has $1.5\times$ parameters. Additional details can be found in the appendix.

For the diffusion settings, we use a cosine noise scheduler following the approach in [38, 12]. The total number of timesteps is standardized to $T = 1000$ across all datasets. We adopt AdamW [29, 33] as our optimizer. For the CelebA dataset, we train our model for 500K iterations with a batch size of 128. During the first 5,000 iterations, we implement a linear warm-up and keep the learning rate at 1×10^{-4} for the remaining training. For the ImageNet dataset, the default learning rate is fixed at 1×10^{-4}. The batch size is set to 1024 for 64^2 resolution and 256 for 256^2 resolution.

Evaluation Settings. To evaluate the performance of our models, we utilize an Exponential Moving Average (EMA) model with a rate of 0.9999. During the evaluation phase, we generate images with the Heun sampler from EDM [26]. For conditional image generation, we also implement the classifier-free sampling strategy [22] to achieve better results. Finally, we measure the quality of the generated images using the FID score calculated on 50K images.

4.2. Analysis of the Proposed Min-SNR-γ

Comparison of Different Weighting Strategies. To demonstrate the significance of the loss weighting strategy, we conduct experiments with different loss weight set-

[1]https://huggingface.co/stabilityai/sd-vae-ft-mse-original

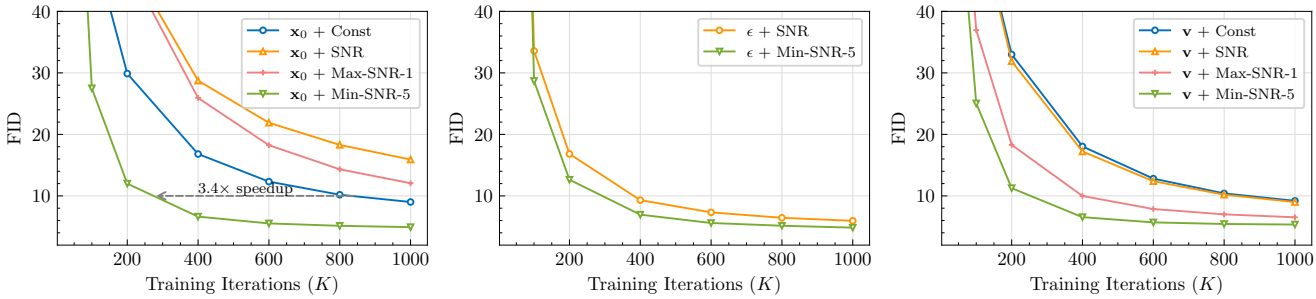

Figure 5: Comparing different loss weighting designs on predicting $\mathbf{x}_0$, ϵ, $\mathbf{v}$. Taking the neural network output as noise with const or Max-SNR-γ strategy lead to divergence. Min-SNR-γ strategy converges the fastest under all these settings.

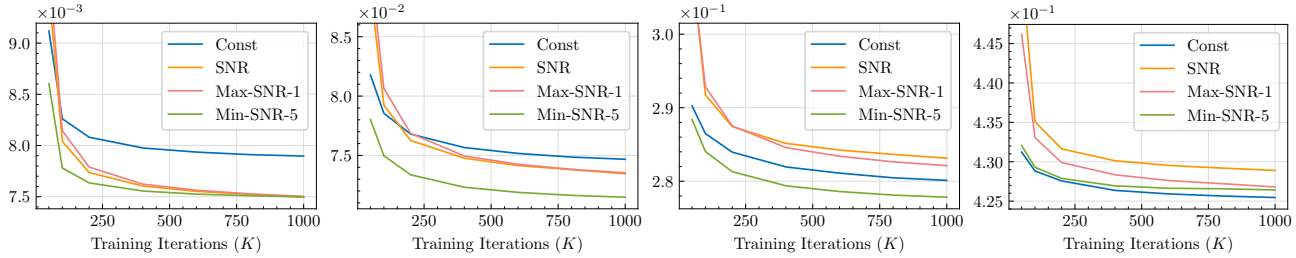

Figure 6: Unweighted loss in different ranges of timesteps. From left to right, each figure represents a specific range of timesteps: $[0, 100)$, $[200, 300)$, $[600, 700)$, $[800, 900)$. The y-axis represents the Mean Squared Error (MSE), averaged over each range of timesteps.

tings for predicting $\mathbf{x}_0$. These settings include: 1) constant weighting, where $w_t = 1$, 2) SNR weighting, with $w_t = \text{SNR}(t)$, 3) truncated SNR weighting, with $w_t = \max\{\text{SNR}(t), \gamma\}$ (following [47] with a set value of $\gamma = 1$), and 4) our proposed Min-SNR-γ weighting strategy, with $w_t = \min\{\text{SNR}(t), \gamma\}$, we set $\gamma = 5$ as the default value.

The ViT-B serves as our default backbone and experiments are performed on ImageNet 256×256. As illustrated in Figure 5, we observe that all results improve as the number of training iterations increases. However, our method demonstrates a significantly faster convergence compared to other methods. Specifically, it exhibits a $3.4\times$ speedup in reaching an FID score of 10. It is worth mentioning that the SNR weighting strategy performed the worst, which could be due to its disproportionate focus on less noisy stages.

For a deeper understanding of the reasons behind the varying convergence rates, we analyzed their training loss at different noise levels. For a fair comparison, we exclude the loss weight term by only calculating $\|\mathbf{x}_0 - \hat{\mathbf{x}}_\theta\|_2^2$. Considering that the loss of different noise levels varies greatly, we calculate the loss in different bins and present the results in Figure 6. The results show that while the constant weighting strategy is effective for high noise intensities, it performs poorly at low noise intensities. Conversely, the SNR weighting strategy exhibits the opposite behavior. In contrast, our proposed Min-SNR-γ strategy achieves a lower training loss across all cases, and indicates quicker convergence through the FID metric.

Furthermore, we present visual results in Figure 7 to demonstrate the fast convergence of the Min-SNR-γ strategy. We apply the same random seed for noise to sample images from training iteration 50K, 200K, 400K, and 1M with different loss weight settings. Our results show that the Min-SNR-γ strategy generates a clear object with only 200K iterations, which is significantly better in quality than the results obtained by other methods.

Min-SNR-γ for Different Prediction Targets. Instead of predicting the original signal $\mathbf{x}_0$ from the network, some recent works have employed alternative re-parameterizations, such as predicting noise ϵ, or velocity $\mathbf{v}$ [47]. To verify the applicability of our weighting strategy to these prediction targets, we conduct experiments comparing the four aforementioned weighting strategies across these different re-parameterizations.

As we discussed in Section 3.4, predicting noise ϵ is mathematically equivalent to predicting $\mathbf{x}_0$ by intrinsically involving Signal-to-Noise Ratio as a weight factor, thus we divide the SNR term in practice. For example, the Min-SNR-γ strategy in predicting noise can be expressed as $w_t = \frac{\min\{\text{SNR}(t), \gamma\}}{\text{SNR}(t)} = \min\{\frac{\gamma}{\text{SNR}(t)}, 1\}$. And the SNR strategy in predicting noise is equivalent to a "constant strategy". For simplicity and consistency, we still refer to them as Min-SNR-γ and SNR strategies. Similarly, we can derive that when predicting velocity $\mathbf{v}$, the loss weight factor must be divided by $(\text{SNR} + 1)$. These strategies are still referred to by their original names for ease of reference.

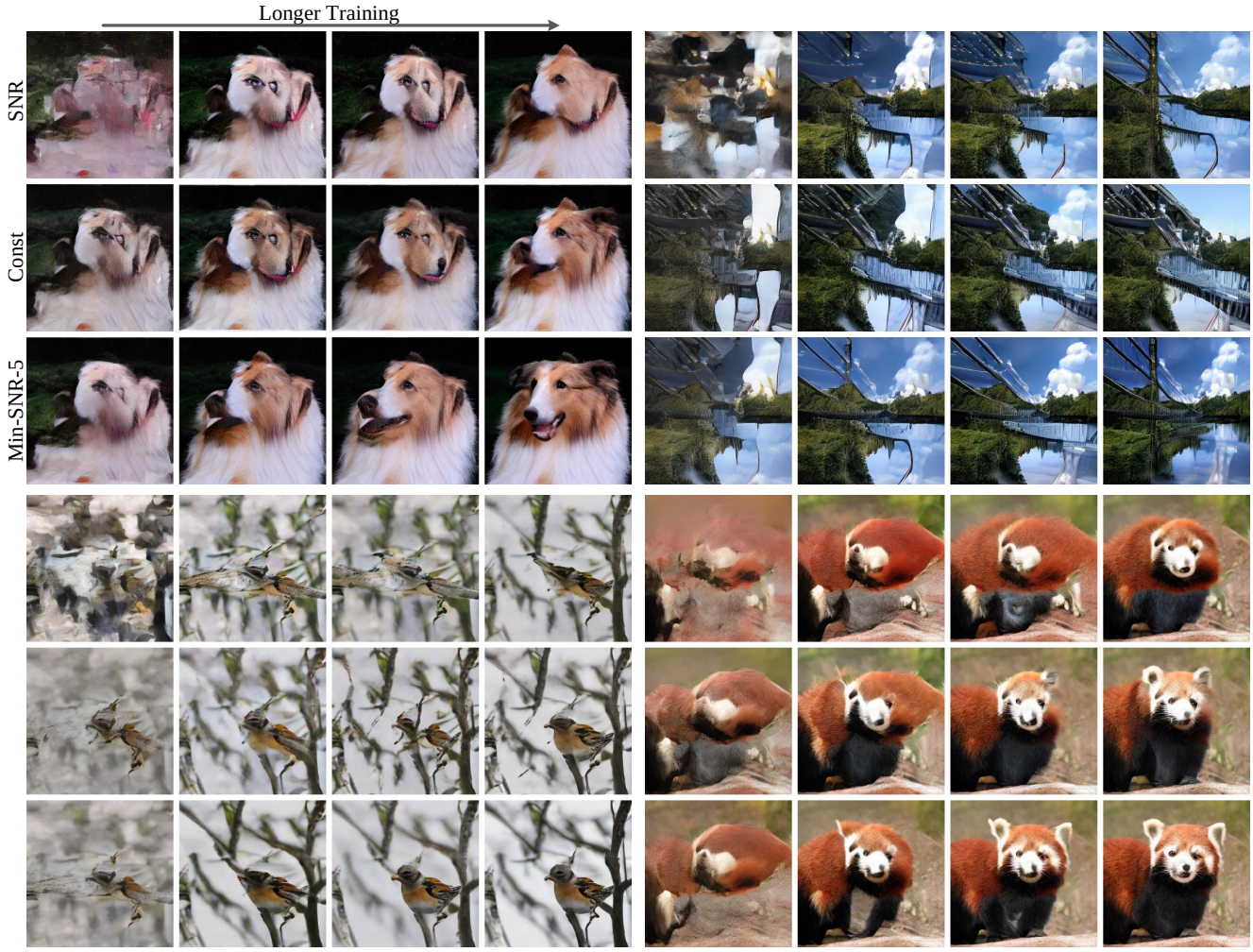

Figure 7: Qualitative comparison of the generation results from different weighting strategies on ImageNet-256 dataset. Images in each column are sampled from 50K, 200K, 400K, and 1M iterations. Our Min-SNR-5 strategy yields significant improvements in visual fidelity from the same iteration.

We conduct experiments on these two variants and present the results in Figure 5. Taking the neural network output as noise with const or Max-SNR-γ setting leads to divergence. Meanwhile, our proposed Min-SNR-γ strategy converges faster than other loss weighting strategies for both prediction noise and predicting velocity. These demonstrate that balancing the loss weights for different timesteps is intrinsic, independent of any re-parameterization.

Min-SNR-γ on Different Network Architectures. The Min-SNR-γ strategy is versatile and robust for different prediction targets and network structures. We conduct experiments on the widely used UNet and keep the number of parameters close to the ViT-B model. For each experiment, models were trained for 1 million iterations and their FID scores were calculated at multiple intervals. The results in Table 1 indicate that the Min-SNR-γ strategy converges significantly faster than the baseline and provides better performance for both predicting $\mathbf{x}_0$ and predicting noise.

Training Iterations	200K	400K	600K	800K	1M
Baseline ($\mathbf{x}_0$)	25.93	15.41	11.54	9.52	8.33
+ Min-SNR-5	**7.99**	**5.34**	**4.69**	**4.41**	**4.28**
Baseline (ϵ)	8.55	5.43	4.64	4.35	4.21
+ Min-SNR-5	**7.32**	**4.98**	**4.48**	**4.24**	**4.14**

Table 1: Ablation studies on the UNet backbone. Whether the network predicts $\mathbf{x}_0$ or ϵ, the Min-SNR-5 weighting design converges faster and achieves better FID score.

Robustness Analysis. Our approach utilizes a single hyperparameter, γ, as the truncate value. To assess its robustness, we conducted a thorough robustness analysis in various settings. Our experiments were performed on the ImageNet-256 dataset using the ViT-B model and the prediction target of the network is $\mathbf{x}_0$. We varied the truncate value γ by set-

γ	1	5	10	20
ViT ($\mathbf{x}_0$)	4.98	**4.92**	5.34	5.45
ViT (ϵ)	4.89	**4.84**	4.94	5.41
UNet ($\mathbf{x}_0$)	4.49	**4.28**	4.32	4.37
UNet (ϵ)	4.30	4.14	4.14	**4.12**

Table 2: Ablation study on γ. The results are robust to the hyper-parameter γ in different settings.

Method	#Params	FID
DDIM [52]	79M	3.26
Soft Truncation [52]	62M	1.90
Our UNet	59M	**1.60**
U-ViT-Small [2]	44M	2.87
ViT-Small (ours)	43M	**2.14**

Table 3: FID results of unconditional image generation on CelebA 64×64 [32]. We conduct experiments with both UNet and ViT backbone.

ting it to 1, 5, 10, and 20 and evaluated their performance. The results are shown in Table 2. We find there are only minor variations in the FID score when γ is smaller than 20. Additionally, we conducted more experiments by modifying the predicting target to the noise ϵ, and modifying the network structure to UNet. We find that the results were also consistently stable. Our results indicate that good performance can usually be achieved when γ is set to 5, making it the established default setting.

4.3. Comparison with state-of-the-art Methods

CelebA-64. We conduct experiments on the CelebA 64×64 dataset for unconditional image generation. Both UNet and ViT are used as our backbones and are trained for 500K iterations. During the evaluation, we use the EDM sampler [26] to generate 50K samples and calculate the FID score. The results are summarized in Table 3. Our ViT-Small [14] model outperforms previous ViT-based models with an FID score of 2.14. It is worth mentioning that no modifications are made to the naive network structure, demonstrating that the results could still be improved further. Meanwhile, our method using the UNet [12] structure achieves an even better FID score of 1.60, outperforming previous UNet methods.

ImageNet-64. We also validate our method on class-conditional image generation on the ImageNet 64×64 dataset. During training, the class label is dropped with the probability 0.15 for classifier-free inference [22]. The model is trained for 800K iterations and images are synthesized using classifier-free guidance with a scale of cfg $=$ 1.5 and the EDM sampler for image generation. For a fair comparison, we adopt a 21-layer ViT-Large model without additional architecture designs, which has a similar number of parameters to U-ViT-Large [2]. The results presented in Table 4 show that our method achieves an FID score of 2.28, significantly improving upon the U-ViT-Large model.

Method	#Params	FID
BigGAN-deep [3]		4.06
StyleGAN-XL [48]		1.51
IDDPM (small) [38]	100M	6.92
IDDPM (large) [38]	270M	2.92
CDM [21]		1.48
ADM [12]	296M	2.61
U-ViT-Mid [2]	131M	5.85
U-ViT-Large [2]	287M	4.26
ViT-L (ours)	269M	**2.28**

Table 4: FID results on ImageNet 64×64. We conduct experiments using the ViT-L backbone which significantly improves upon previous methods.

Method	#Params	FID
BigGAN-deep [3]	340M	6.95
StyleGAN-XL [48]		2.30
Improved VQ-Diffusion [18]	460M	4.83
IDDPM [38]	270M	12.26
CDM [21]		4.88
ADM [12]	554M	10.94
ADM-U [12]	608M	7.49
ADM-G [12]	554M	4.59
ADM-U, ADM-G [12]	608M	3.94
LDM [44]	400M	3.60
UNet (ours)	395M	**2.81**†
U-ViT-L [2]	287M	3.40
DiT-XL-2 [40]	675M	9.62
DiT-XL-2 (cfg=1.50) [40]	675M	2.27
ViT-XL (ours)	451M	8.10
ViT-XL (ours, cfg=1.50)	451M	**2.06**

Table 5: FID results on ImageNet 256×256. † denotes only train 1.4M iterations. Our model with a ViT-XL backbone achieves a new record FID score of 2.06.

ImageNet-256. We also apply diffusion models for higher-resolution image generation on the ImageNet 256×256 benchmark. To enhance training efficiency, we first compress $256 \times 256 \times 3$ images into $32 \times 32 \times 4$ latent codes using the encoder from LDM [44]. During the sampling process, we employ the EDM sampler and the classifier-free guidance to generate images. The FID comparison is presented in Table 5. Under the setting of predicting ϵ with Min-SNR-5, our ViT-XL model achieves the FID of 2.08 for only 2.1M iterations, which is $3.3\times$ faster than DiT

and outperforms the previous state-of-the-art FID record of 2.27. Moreover, with longer training (about 7M iterations as in [40]), we are able to achieve the FID score of 2.06 by predicting $\mathbf{x}_0$ with Min-SNR-5. Our UNet-based model with 395M parameters is trained for about 1.4M iterations and achieves FID score of 2.81.

5. Conclusion

In this paper, we point out that the conflicting optimization directions between different timesteps may cause slow convergence in diffusion training. To address it, we regard the diffusion training process as a multi-task learning problem and introduce a novel weighting strategy, named Min-SNR-γ, to effectively balance different timesteps. Experiments demonstrate our method can boost diffusion training several times faster, and achieves the state-of-the-art FID score on ImageNet-256 dataset.

Acknowledgments

We sincerely thank Yixuan Wei, Zheng Zhang, and Stephen Lin for helpful discussion. This research was partly supported by the National Key Research & Development Plan of China (No. 2018AAA0100104), the National Science Foundation of China (62125602, 62076063).

References

[1] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediff-i: Text-to-image diffusion models with ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 2

[2] 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. *arXiv preprint arXiv:2209.12152*, 2022. 2, 8, 13

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *International Conference on Learning Representations*, 2019. 8

[4] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022. 1, 2

[5] Hanqun Cao, Cheng Tan, Zhangyang Gao, Guangyong Chen, Pheng-Ann Heng, and Stan Z Li. A survey on generative diffusion model. *arXiv preprint arXiv:2209.02646*, 2022. 4

[6] Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023. 2

[7] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR, 2018. 3

[8] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *Advances in Neural Information Processing Systems*, 33:2039–2050, 2020. 3

[9] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020. 3

[10] 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. 5

[11] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012. 2, 3, 4

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 1, 2, 3, 4, 5, 8, 13

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 2

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, and S. Gelly. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2, 5, 8

[15] Zhida Feng, Zhenyu Zhang, Xintong Yu, Yewei Fang, Lanxin Li, Xuyi Chen, Yuxiang Lu, Jiaxiang Liu, Weichong Yin, Shikun Feng, et al. Ernie-vilg 2.0: Improving text-to-image diffusion model with knowledge-enhanced mixture-of-denoising-experts. *arXiv preprint arXiv:2210.15257*, 2022. 2

[16] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956. 4

[17] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Sequence to sequence text generation with diffusion models. In *International Conference on Learning Representations*, 2023. 1

[18] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022. 1, 3, 4, 8

[19] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey A. Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *ArXiv*, abs/2210.02303, 2022. 1, 2

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1, 2, 3

[21] 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, 2022. 8

[22] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 5, 8

[23] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2

[24] Jonathan Ho, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 1, 4

[25] Qingqing Huang, Daniel S Park, Tao Wang, Timo I Denk, Andy Ly, Nanxin Chen, Zhengdong Zhang, Zhishuai Zhang, Jiahui Yu, Christian Frank, et al. Noise2music: Text-conditioned music generation with diffusion models. *arXiv preprint arXiv:2302.03917*, 2023. 2

[26] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 2, 5, 8, 13, 14

[27] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018. 3

[28] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2426–2435, 2022. 1, 2

[29] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2014. 5, 13

[30] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *arXiv preprint arXiv:2205.14217*, 2022. 1

[31] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *ArXiv*, abs/2202.09778, 2022. 2

[32] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 5, 8

[33] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 5, 13

[34] 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. *ArXiv*, abs/2206.00927, 2022. 2, 14

[35] Chenlin Meng, Ruiqi Gao, Diederik P Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. *arXiv preprint arXiv:2210.03142*, 2022. 2, 4

[36] Chenlin Meng, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 1

[37] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, 2021. 2, 3

[38] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 1, 2, 3, 5, 8

[39] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. *arXiv preprint arXiv:2302.03027*, 2023. 2

[40] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 2, 8, 9, 13

[41] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1, 2

[42] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1, 2

[43] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1

[44] 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. 1, 2, 3, 4, 5, 8, 13

[45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 2

[46] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 1, 2

[47] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. 2, 3, 5, 6, 13

[48] 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. 8

[49] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018. 2, 3, 4

[50] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2

[51] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 1

[52] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 2, 8

[53] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 2

[54] Jianlin Su. Talking about multi-task learning (2): By the way of gradients, Feb 2022. 12

[55] Zhicong Tang, Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. Improved vector quantized diffusion models. *arXiv preprint arXiv:2205.16007*, 2022. 4

[56] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual descriptions. In *International Conference on Learning Representations*, 2023. 2

[57] John Von Neumann and Oskar Morgenstern. Theory of games and economic behavior, 2nd rev. 1947. 12

[58] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. *arXiv preprint arXiv:2212.06135*, 2022. 1, 2

[59] Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. *arXiv preprint arXiv:2010.05874*, 2020. 3

[60] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022. 2

[61] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. *arXiv preprint arXiv:2211.13227*, 2022. 1

[62] S. Yang, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 1, 5

[63] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020. 3

[64] Zixin Zhu, Yixuan Wei, Jianfeng Wang, Zhe Gan, Zheng Zhang, Le Wang, Gang Hua, Lijuan Wang, Zicheng Liu, and Han Hu. Exploring discrete diffusion models for image captioning. *arXiv preprint arXiv:2211.11694*, 2022. 1

In the appendix, we first provide the proof of Theorem 1 in Section A. Then we derive the relationship between loss weights of different predicting targets in Section B. In Section C, we provide more details on the network architecture, training and sampling settings. Finally, we present more visual results in Section D.

A. Proof for Theorem 1

First, we introduce the Pareto Optimality mentioned in the paper. Assume the loss for each task is $\mathcal{L}^t(\theta), t \in \{1, 2, \dots, T\}$ and the respective gradient to θ is $\nabla_\theta \mathcal{L}^t(\theta)$. For simplicity, we denote $\mathcal{L}^t(\theta)$ as $\mathcal{L}^t$. If we treat each task with equal importance, we assume each loss item $\mathcal{L}^1, \mathcal{L}^2, \dots, \mathcal{L}^T$ is decreasing or kept the same. There exists one point θ^* where any change of the point will leads to the increase of one loss item. We call the point θ^* "Pareto Optimality". In other words, we cannot sacrifice one task for another task's improvement. To reach Pareto Optimality, we need to find an update direction δ which meet:

$$\begin{cases} \left\langle \nabla_\theta \mathcal{L}_\theta^1, \delta \right\rangle & \leq 0 \\ \left\langle \nabla_\theta \mathcal{L}_\theta^2, \delta \right\rangle & \leq 0 \\ \vdots & \\ \left\langle \nabla_\theta \mathcal{L}_\theta^T, \delta \right\rangle & \leq 0 \end{cases} \tag{14}$$

$\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors. It is worth noting that $\delta = 0$ satisfies all the above inequalities. We care more about the non-zero solution and adopt it for updating the network parameter θ. If the non-zero point does not exist, it may already achieve the "Pareto Optimality", which is referred as "Pareto Stationary".

For simplicity, we denote the gradient for each loss item $\nabla_\theta \mathcal{L}^t$ as $\mathbf{g}_t$. Suppose we have a gradient vector $\mathbf{u}$ to satisfy that all $\langle \mathbf{g}_t, \mathbf{u} \rangle \geq 0, t \in \{1, 2, \dots, T\}$. Then $-\mathbf{u}$ is the updating direction ensuring a lower loss for each task.

As proposed in [54], $\langle \mathbf{g}_t, \mathbf{u} \rangle \geq 0, \forall t \in \{1, 2, \dots, T\}$ is equivalent to $\min_t \langle \mathbf{g}_t, \mathbf{u} \rangle \geq 0$. And it could be achieved when the minimal value of $\langle \mathbf{g}_t, \mathbf{u} \rangle$ is maximized. Thus the problem is further converted to:

$$\max_{\mathbf{u}} \min_t \langle \mathbf{g}_t, \mathbf{u} \rangle$$

There is no constraint for the vector $\mathbf{u}$, so it may become infinity and make the updating unstable. To avoid it, we add a regularization term to it

$$\max_{\mathbf{u}} \min_t \langle \mathbf{g}_t, \mathbf{u} \rangle - \frac{1}{2} \|\mathbf{u}\|_2^2. \tag{15}$$

And notice that the max function ensures the value is always greater than or equal to a specific value $\mathbf{u} = 0$.

$$\begin{aligned} & \max_{\mathbf{u}} \min_t \langle \mathbf{g}_t, \mathbf{u} \rangle - \frac{1}{2} \|\mathbf{u}\|_2^2 \\ \geq & \min_t \langle \mathbf{g}_t, \mathbf{u} \rangle - \frac{1}{2} \|\mathbf{u}\|_2^2 \Big|_{\mathbf{u}=0} \\ = & \, 0, \end{aligned}$$

which also means $\max_{\mathbf{u}} \min_t \langle \mathbf{g}_t, \mathbf{u} \rangle \geq \frac{1}{2} \|\mathbf{u}\|_2^2 \geq 0$. Therefore, the solution of Equation 15 satisfies our optimization goal of $\langle \mathbf{g}_t, \mathbf{u} \rangle \geq 0, \forall t \in \{1, 2, \dots, T\}$.

We define $\mathcal{C}^T$ as a set of n-dimensional variables

$$\mathcal{C}^T = \left\{ (w_1, w_2, \dots, w_T) | w_1, w_2, \dots, w_T \geq 0, \sum_{t=1}^{T} w_t = 1 \right\}, \tag{16}$$

It is easy to verify that

$$\min_t \langle \mathbf{g}_t, \mathbf{u} \rangle = \min_{w \in \mathcal{C}^T} \left\langle \sum_t w_t \mathbf{g}_t, \mathbf{u} \right\rangle. \tag{17}$$

We can also verify the above function is concave with respect to $\mathbf{u}$ and α. According to Von Neumann's Minmax theorem [57], the objective with regularization in Equation 15 is equivalent to

$$\max_{\mathbf{u}} \min_{w \in \mathcal{C}^T} \left\{ \left\langle \sum_t w_t \mathbf{g}_t, \mathbf{u} \right\rangle - \frac{1}{2} \|\mathbf{u}\|_2^2 \right\} \tag{18}$$

$$= \min_{w \in \mathcal{C}^T} \max_{\mathbf{u}} \left\{ \left\langle \sum_t w_t \mathbf{g}_t, \mathbf{u} \right\rangle - \frac{1}{2} \|\mathbf{u}\|_2^2 \right\} \tag{19}$$

$$= \min_{w \in \mathcal{C}^T} \left\{ \left\langle \sum_t w_t \mathbf{g}_t, \mathbf{u} \right\rangle - \frac{1}{2} \|\mathbf{u}\|_2^2 \right\} \Big|_{\mathbf{u} = \frac{1}{2} \sum_t w_t \mathbf{g}_t} \tag{20}$$

$$= \min_{w \in \mathcal{C}^T} \frac{1}{2} \left\| \sum_t w_t \mathbf{g}_t \right\|_2^2. \tag{21}$$

Finally, we achieved Theorem 1 in the main paper.

B. Relationship between Different Targets

The most common predicting target is in ϵ-space. Loss for prediction in $\mathbf{x}_0$-space and ϵ-space can be transformed by the SNR loss weight.

$$\begin{aligned} \mathcal{L}_\theta &= \|\epsilon - \hat{\epsilon}_\theta(\mathbf{x}_t)\|_2^2 \\ &= \left\| \frac{1}{\sigma_t}(\mathbf{x}_t - \alpha_t \mathbf{x}_0) - \frac{1}{\sigma_t}(\mathbf{x}_t - \alpha_t \hat{\mathbf{x}}_\theta(\mathbf{x}_t)) \right\|_2^2 \\ &= \frac{\alpha_t^2}{\sigma_t^2} \|\mathbf{x}_0 - \hat{\mathbf{x}}_\theta(\mathbf{x}_t))\|_2^2 \\ &= \mathrm{SNR}(t) \|\mathbf{x}_0 - \hat{\mathbf{x}}_\theta(\mathbf{x}_t))\|_2^2, \end{aligned}$$

where $\hat{\epsilon}_\theta$ is the network to predict the noise and $\hat{\mathbf{x}}_\theta$ is to predict the clean data.

Prediction target $\mathbf{v} = \alpha_t \epsilon - \sigma_t \mathbf{x}_0$ is proposed in [47], we can derive the related loss

$$
\begin{aligned}
\mathcal{L}_\theta &= \|\mathbf{v}_t - \mathbf{v}_\theta(\mathbf{x}_t)\|_2^2 \\
&= \|(\alpha_t \epsilon - \sigma_t \mathbf{x}_0) - (\alpha_t \hat{\epsilon}_\theta(\mathbf{x}_t) - \sigma_t \hat{\mathbf{x}}_\theta(\mathbf{x}_t))\|_2^2 \\
&= \|\alpha_t \left(\epsilon - \hat{\epsilon}_\theta(\mathbf{x}_t)\right) - \sigma_t \left(\mathbf{x}_0 - \hat{\mathbf{x}}_\theta(\mathbf{x}_t)\right)\|_2^2 \\
&= \left\| \alpha_t \frac{\alpha_t}{\sigma_t} \left(\hat{\mathbf{x}}_\theta(\mathbf{x}_t) - \mathbf{x}_0\right) - \sigma_t \left(\mathbf{x}_0 - \hat{\mathbf{x}}_\theta(\mathbf{x}_t)\right) \right\|_2^2 \\
&= \left\| \frac{\alpha_t^2 + \sigma_t^2}{\sigma_t} \left(\mathbf{x}_0 - \hat{\mathbf{x}}_\theta(\mathbf{x}_t)\right) \right\|_2^2 \\
&= \frac{1}{\sigma_t^2} \|(\mathbf{x}_0 - \hat{\mathbf{x}}_\theta(\mathbf{x}_t))\|_2^2 \\
&= \frac{\alpha_t^2 + \sigma_t^2}{\sigma_t^2} \|(\mathbf{x}_0 - \hat{\mathbf{x}}_\theta(\mathbf{x}_t))\|_2^2 \\
&= (\mathrm{SNR}(t) + 1) \|(\mathbf{x}_0 - \hat{\mathbf{x}}_\theta(\mathbf{x}_t))\|_2^2
\end{aligned}
$$

C. Hyper-parameter

Here we list more details about the architecture, training and evaluation setting.

C.1. Architecture Settings

The ViT setting adopted in the paper are as follows,

Model	Layers	Hidden Size	Heads	Params
ViT-Small	13	512	6	43M
ViT-Base	12	768	12	88M
ViT-Large	21	1024	16	269M
ViT-XL	28	1152	16	451M

Table 6: Configurations of our used ViTs.

We use ViT-Small for face generation on CelebA 64×64. Besides, we adopt ViT-Base as the default backbone for the ablation study. To make relative fair comparison with U-ViT, we use a 21-layer ViT-Large for ImageNet 64×64 benchmark. To compare with former state-of-the-art method DiT [40] on ImageNet 256×256, we adopt the similar setting ViT-XL with the same depth, hidden size, and patch size.

In the paper, we also evaluate our method's robustness to model architectures using the UNet backbone. For ablation study, we adjust the setting based on ADM [12] to make the parameters and FLOPs close to ViT-B. The setting is

- Base channels: 192
- Channel multipliers: 1, 2, 2, 2
- Residual blocks per resolution: 3
- Attention resolutions: 8, 16
- Attention heads: 4

We also conduct experiments with the same architecture (296M) in ADM [12] on ImageNet 64×64. After 900K training iterations with batch size 1024, it could achieve an FID score of 2.11.

For high resolution generation on ImageNet 256×256. We use the 395M setting from LDM [44], which operates on the $32 \times 32 \times 4$ latent space.

C.2. Training Settings

The training iterations and learning rate have been reported in the paper. We use AdamW [33, 29] as our default optimizer. (β_1, β_2) is set to $(0.9, 0.999)$ for UNet backbone. Following [2], we set (β_1, β_2) to $(0.99, 0.99)$ for ViT backbone.

C.3. Sampling Settings

If not otherwise specified, we only use EDM's [26] Heun sampler. We only adjust the sampling steps for better results. For ablation study with ViT-B and UNet, we set the number of steps to 30. For ImageNet 64×64 in Table 4, the number of steps is set to 20. For ImageNet 256×256 in Table 5, the number of sampling steps is set to 50.

D. Additional Results

D.1. Ablation Study on Pixel Space

In the paper, most of the ablation study is conducted on ImageNet 256×256's latent space. Here, we present the results on ImageNet 64×64 pixel space. We adopt a ViT-B model as our backbone and train the diffusion model for 800K iterations with batch size 512. Our predicting targets are $\mathbf{x}_0$ and ϵ and they are equipped with our proposed simple Min-SNR-γ loss weight ($\gamma = 5$). We adopt the pre-trained noisy classifier at 64×64 from ADM [12] as conditional guidance. We can see that the loss weighting strategy contributes to the faster convergence for both $\mathbf{x}_0$ and ϵ.

D.1.1 Min-SNR-γ on EDM

We also apply our Min-SNR-γ weighting strategy on the SoTA "denoiser" framework EDM. We find that our strategy can also help converge faster in such framework in Figure 9. The specific implementation is to multiply $\frac{\min\{\mathrm{SNR}, 5\}}{\mathrm{SNR}}$ in `EDMLoss` from official code[2]. We keep the

[2] https://github.com/NVlabs/edm.git

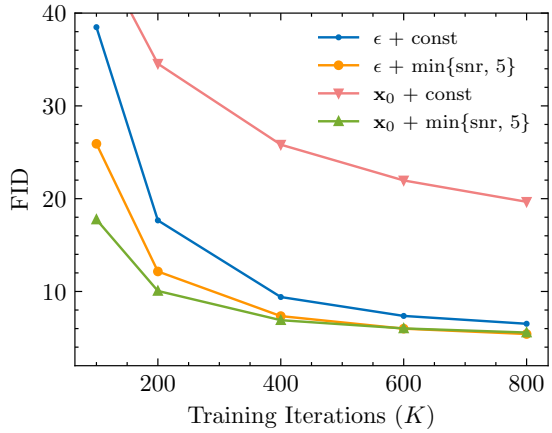

Figure 8: Ablate loss weight design in pixel space (ImageNet 64×64). We adopt DPM Solver [34] to sample $50k$ images to calculate the FID score with classifier guidance.

same setting as official ImageNet-64 training setting, including batch size and optimizer. Due to the limit of compute budget, we did not train the model as long as that in EDM [26] (about 2k epochs on ImageNet). We use 2^{nd} Heun approach with 18 steps (NFE=35). The curve in Figure 9 reflects the FID's changing with training images.

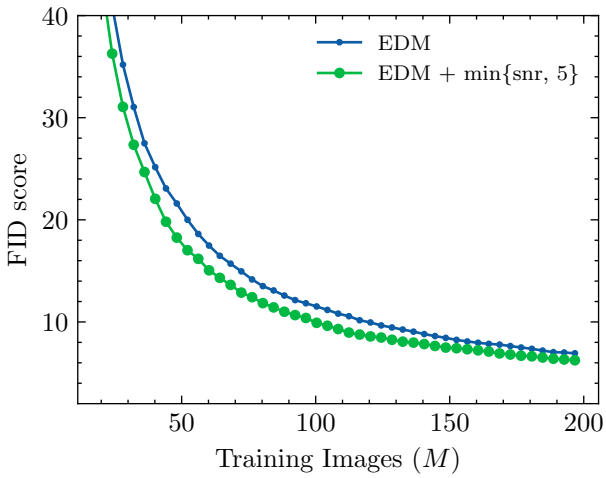

Figure 9: Effect of Min-SNR-γ on EDM [26].

D.2. Visual Results on Different Datasets

We provide additional generated results in Figure 10-13. Figure 10 shows the generated samples with UNet backbone on CelebA 64×64. Figure 11 and Figure 12 demonstrate the generated samples on conditional ImageNet 64×64 benchmark with ViT-Large and UNet backbone respectively. The visual results on CelebA 64×64 and ImageNet 64×64 are randomly synthesized without cherry-pick.

We also present some visual results on ImageNet 256×256 with our model which can achieve the FID 2.06 in Figure 13.

Figure 10: Additional generated samples on CelebA 64×64. The samples are from UNet backbone with 1.60 FID.

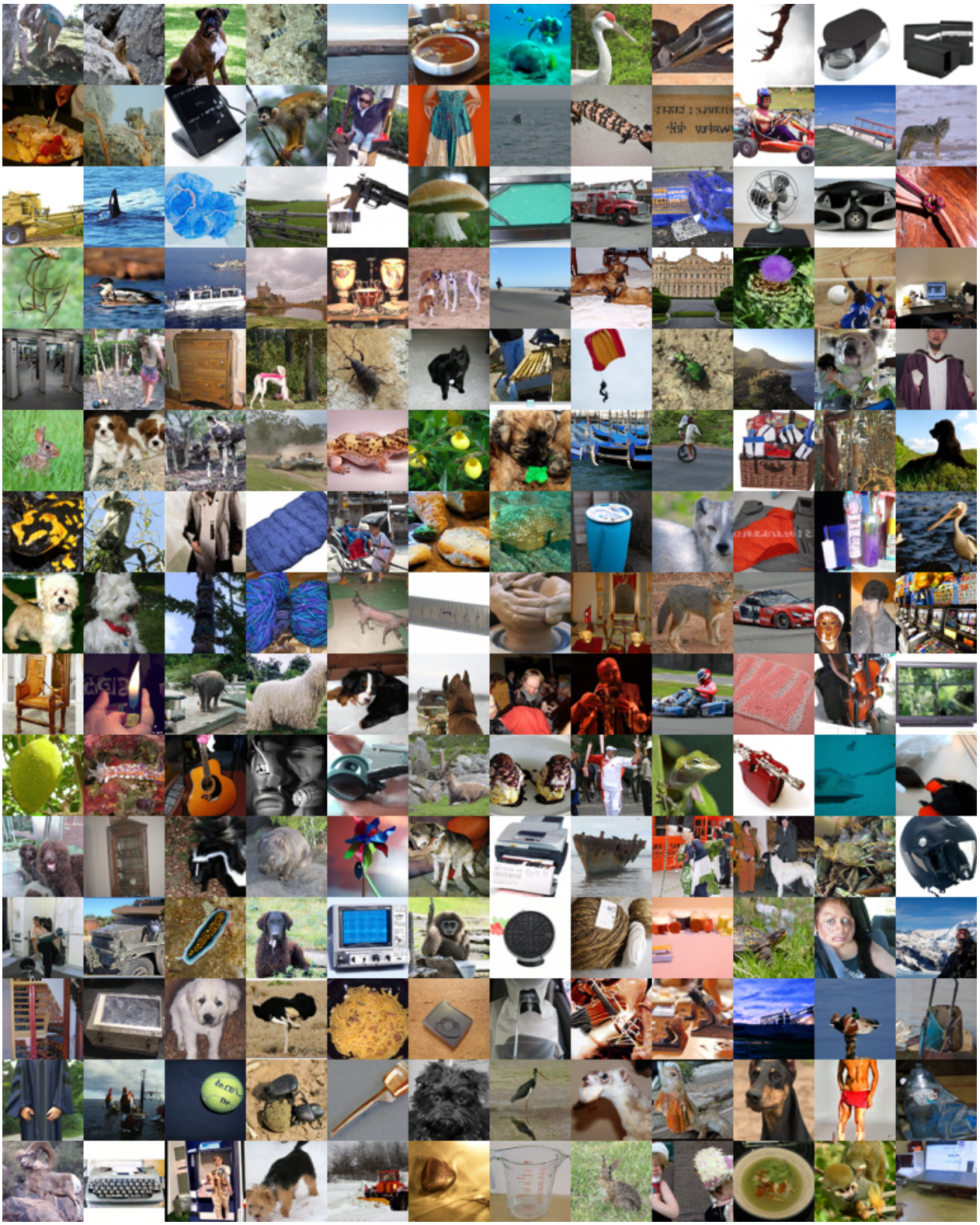

Figure 11: Additional generated samples on ImageNet 64×64. The samples are from ViT backbone with 2.28 FID.

Figure 12: Additional generated samples on ImageNet 64×64. The samples are from UNet backbone with 2.14 FID.

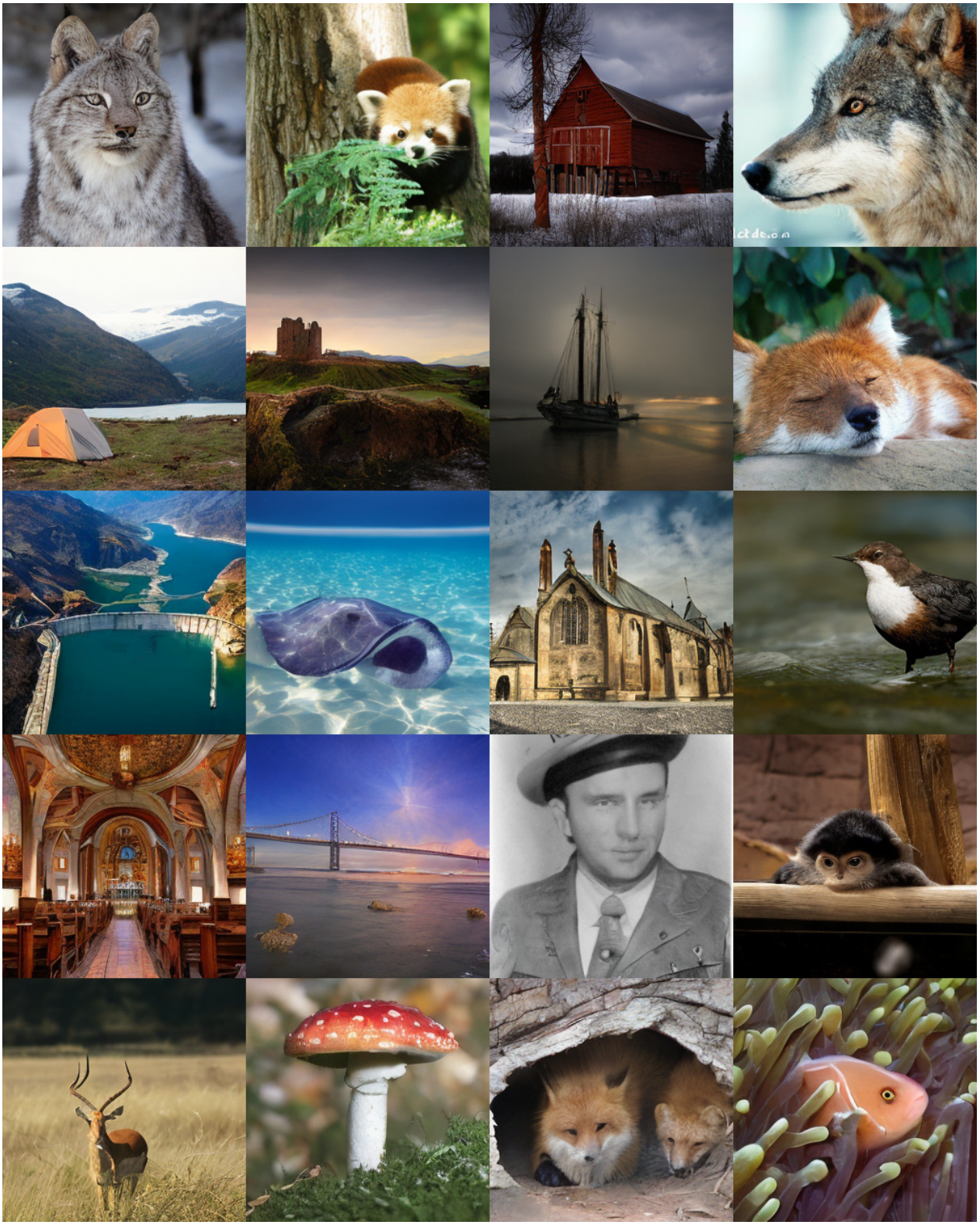

Figure 13: Additional generated samples on ImageNet 256 $\times$ 256. The samples are from ViT backbone with 2.06 FID.