Unmasking Bias in Diffusion Model Training

Hu Yu[†,1], Li Shen[2], Jie Huang[1], Hongsheng Li[3], and Feng Zhao[‡,1]

[1]MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition
University of Science and Technology of China
[2]Alibaba Group [3]The Chinese University of Hong Kong
yuhu520@mail.ustc.edu.cn, lshen.lsh@gmail.com
fzhao956@ustc.edu.cn

Abstract. Denoising diffusion models have emerged as a dominant approach for image generation, however they still suffer from slow convergence in training and color shift issues in sampling. In this paper, we identify that these obstacles can be largely attributed to bias and suboptimality inherent in the default training paradigm of diffusion models. Specifically, we offer theoretical insights that the prevailing constant loss weight strategy in ϵ-prediction of diffusion models leads to biased estimation during the training phase, hindering accurate estimations of original images. To address the issue, we propose a simple but effective weighting strategy derived from the unlocked biased part. Furthermore, we conduct a comprehensive and systematic exploration, unraveling the inherent bias problem in terms of its existence, impact and underlying reasons. These analyses contribute to advancing the understanding of diffusion models. Empirical results demonstrate that our method remarkably elevates sample quality and displays improved efficiency in both training and sampling processes, by only adjusting loss weighting strategy. The code is released publicly at https://github.com/yuhuUSTC/Debias

Keywords: Diffusion model training · Bias issue · Efficient training

1 Introduction

Diffusion models [13, 35] have emerged as powerful generative models that garner significant attention recently. Their popularity stems from the remarkable ability to generate diverse and high-quality samples [7, 26, 28, 29] as well as the training-stable loss form, compared to the adversarial training paradigms used in Generative Adversarial Networks (GANs) [9]. Diffusion models often serve as a fundamental block and have exhibited impressive success on numerous tasks [30, 31, 40, 41]. While, it is usually employed as a black-box component in these works.

There have been some attempts to delve into the methodology of diffusion models. The works in [22, 23, 25, 32, 36] target on the acceleration of the reverse

† Work done during internship at Alibaba DAMO Academy.
‡Corresponding Author.

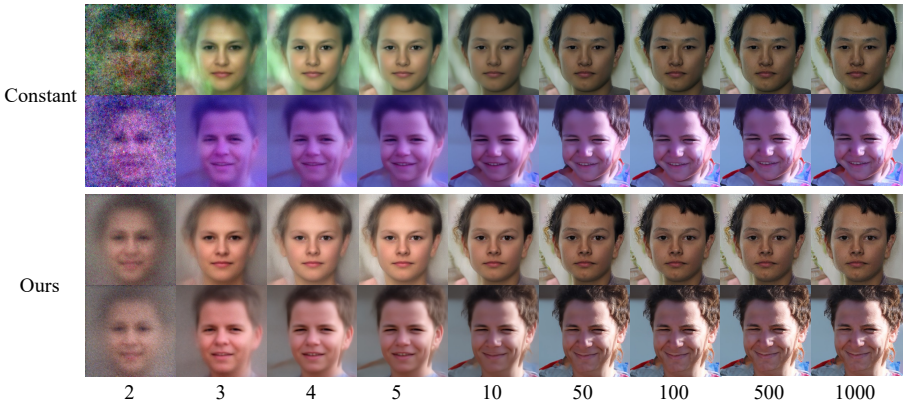


Fig. 1: Examples for the bias problem in ϵ-prediction with constant weighting. Images are generated with different total sampling steps T. The upper two rows showcase samples obtained through constant weighting, exhibiting color shift and poor details. The bottom ones display samples generated using our method.

sampling process. An alternative line of research has directed its attention towards the training objective, traditionally characterized by an elegantly simple loss function, i.e., the pixel-wise loss with a constant weight between the Gaussian noise and the predicted outcome as follows:

$$L = \sum_t \mathbb{E}_{x_0,\epsilon} \left[||\epsilon - \epsilon_\theta(x_t, t)||^2 \right]. \tag{1}$$

Prior works find that this loss formulation is less effective for training diffusion models, and alternative training objectives and weighting strategies are thus proposed. For instance, ϵ-prediction with a range of customized weighting strategies [4, 10, 24], or combining the strength of ϵ-prediction and x_0-prediction to get new training targets [16, 32] can enhance model performance. However, a comprehensive examination of the underlying reasons and issues within the basic ϵ-prediction in Eq. 1 is still lacking.

In this paper, we aim to fill this gap by conducting a detailed analysis to elucidate the bias and flaws associated with the basic ϵ-prediction with constant weighting. Specifically, we provide a theoretical demonstration of its suboptimality, revealing its potential to introduce biased estimations during training and consequently diminish the overall performance of the model (as shown in Fig. 1). To address the issue, we propose a simple but effective loss weighting strategy, termed the inverse of the Signal-to-Noise Ratio (SNR)'s square root, which is motivated from the uncovered biased part. Furthermore, we figure out several pivotal questions essential for systematically understanding the bias problem in conventional diffusion models, covering aspects of its existence, impact and underlying reasons. Firstly, we demonstrate the existence of a biased estimation problem during the training process. The denoising network estimation may closely approach the target Gaussian noise at every step t, while, the corresponding estimated $\hat{x}_0$ may significantly deviate from the true x_0, with this

deviation amplifying as t increases. Next, we analyse the influence of this biased estimation problem on the sampling process, termed as *biased generation*. Biased generation primarily contributes to chaos and inconsistency in the early few sampling steps (left column of Fig. 1), further affecting the final generation with error propagation effect. We further uncover the root causes of biased estimation, elucidating that the importance and optimization difficulty of the denoising network vary significantly at different step t.

We empirically show that the proposed method is capable of addressing the above problems and substantially elevates sample quality. The method can achieve superior performance to constant-weighting strategy with much less training iterations and sampling steps. Through comprehensive analyses and comparison, we also provide a unified prospective on existing weighting strategies [4,10,24], highlighting the benefit of employing appropriate weights for loss penalties at different timesteps.

2 Background and Related Work

2.1 Preliminary of Diffusion models

Definition. Diffusion models [13,35] transform complex data distribution into simple noise distribution and learn to recover data from noise. The *forward diffusion process* starts from a clean data sample x_0 and repeatedly injects Gaussian noise according to the transition kernel $q(x_t|x_{t-1})$ as follows:

$$q(x_t|x_{t-1}) = N(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I). \tag{2}$$

We can further derive closed-form expressions of distribution $q(x_t|x_0)$.

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon, \tag{3}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and $\alpha_t := \prod_{s=1}^{t}(1-\beta_s)$.

The *reverse denoise process* is trained to reverse the forward diffusion process in Eq. 2 by learning the denoise network. Kingma et al. [20] further proposed the use of *signal-to-noise ratio* (SNR) to simplify the representation the noise schedules in diffusion models, which is expressed as:

$$\mathrm{SNR}(t) = \alpha_t/(1-\alpha_t). \tag{4}$$

Training objectives. Diffusion models are trained by optimizing a variational lower bound (VLB). For each step t, the denoising score matching loss L_t is the distance between two Gaussian distributions, rewritten as:

$$\begin{aligned} L_t &= D_{KL}(q(x_{t-1}|x_t, x_0) \;||\; p_\theta(x_{t-1}|x_t)), \\ &= \mathbb{E}_{x_0,\epsilon}\left[\frac{1}{2\sigma_t^2}\left\|C_1 x_0 + C_2 x_t - \boldsymbol{x}_\theta\left(\mathbf{x}_t, t\right)\right\|^2\right], \\ &= \mathbb{E}_{x_0,\epsilon}\left[\frac{\beta_t^2}{(1-\beta_t)(1-\alpha_t)}||\epsilon - \epsilon_\theta(x_t, t)||^2\right]. \end{aligned} \tag{5}$$

The expression of C_1 and C_2 as well as full derivation are available in the supplementary material. The denoising network is indeed optimized to approach x_0, while ϵ can also be employed as training target with a deterministic relationship to x_0. Ho et al. [13] empirically demonstrated that ϵ-prediction outperforms x_0-prediction. Additionally, they observed that the simplified objective (Eq. 1) with constant weight yields better sample quality, which subsequently becomes the default training objective of diffusion models.

2.2 Related Work

Different training objectives. Many existing works adhere to the prevailing training objective in Eq. 1. Recent methods [4, 10, 16, 24, 24, 32] find Eq. 1 less effective in performance and investigate improved training objectives and weighting strategies. They can be categorized into two types. One is ϵ-prediction with various weighting strategies. Particularly, P2 [4] proposed a weighting strategy that prioritizes higher noise levels for recovering content information. Min-SNR [10] interpreted the training goal from the perspective of multitask learning and studied the weighting strategy of Min-SNR. The other is combining the strengths of ϵ-prediction and x_0-prediction to get new training targets [16, 32]. Salimans et al. [32] presented v-prediction for distilling diffusion models. EDM [16] also realizes that directly predicting the Gaussian noise induces error amplification. While EDM resorts to precondition technique requiring network inputs and training targets to have unit variance, which is in the same spirit as previous reparameterization method like v-prediction to adaptively mix signal and noise.
Induced color shift issues. Generated images of diffusion models suffer from errors in their spatial means, i.e., color shift. Song et al. [37] observed this issue in the images generated by diffusion models, especially at higher image resolutions. They employ the exponential moving average strategy to alleviate this problem. Deck et al. [37] proposed a nonlinear bypass connection in the network to predict the mean of the score function. P2 [4] suggests weighting the loss function to mitigate color shift, based on the intuition that crucial spatial features are generated early in the sampling process. Although the color shift issue can be alleviated by using these methods, they still face the challenges in interpreting the root cause of this phenomenon. In this paper, we unveil that the utilization of constant weights during the training stage plays a crucial role in causing the color shift problem, and this issue can be effectively addressed with our strategy.

3 Theoretical Exploration of the Inherent Bias

3.1 Constant Weighting Induces Bias in Training

We treat ϵ as the explicit and direct target, and x_0 as the implicit but intrinsic target. Given the predicted noise $\epsilon_\theta(x_t, t)$ of the denoising network, we can easily derive the corresponding $\hat{x}_0$ from Eq. 3 as follows:

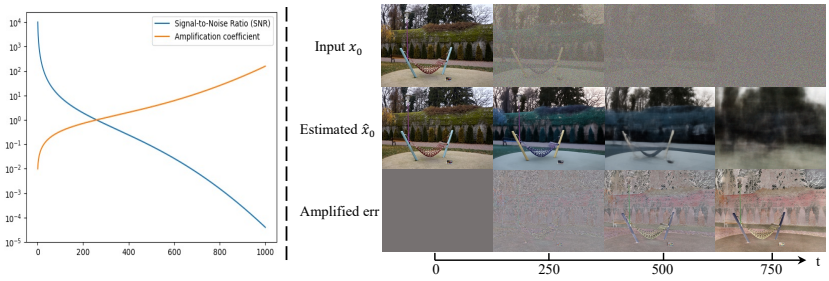


Fig. 2: Left: The visualization of SNR(t) and amplification coefficient $\frac{1}{\sqrt{\text{SNR}(t)}}$ at different timesteps. Right: The upper row is the input x_t at different timesteps. We employ the diffusion model [7] pretrained on ImageNet dataset to obtain the *estimated* $\hat{x}_0$ part and *amplified error* part of each input x_t. The second row is the *estimated* $\hat{x}_0$. The bottom row is the corresponding *amplified error* part. Apparently, as step t gets larger, the *estimated* $\hat{x}_0$ severely deviates from x_0 and the *amplified error* part gradually approaches x_0.

$$\begin{aligned}\hat{x}_0 &= \frac{1}{\sqrt{\alpha_t}}x_t - \frac{\sqrt{1-\alpha_t}}{\sqrt{\alpha_t}}\epsilon_\theta(x_t,t)\\ &= \frac{1}{\sqrt{\alpha_t}}(\sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon) - \frac{\sqrt{1-\alpha_t}}{\sqrt{\alpha_t}}\epsilon_\theta(x_t,t)\\ &= x_0 + \frac{1}{\sqrt{SNR(t)}}(\epsilon - \epsilon_\theta(x_t,t)).\end{aligned} \tag{6}$$

It is noticeable that while certain prior methods may reach the same derivation [24], they conclude the exploration at this point. Besides, instead of proceeding to train diffusion models for image generation, they apply it to other use cases. In stark contrast, this derivation is the start of our paper. We conduct comprehensive analyses and studies to thoroughly unlock the problems behind this formulation, and propose a simple but effective solution.

Further, we can rewrite Eq. 6 to express x_0 in terms of two components: the *estimated* $\hat{x}_0$ part and the *amplified error* part.

$$x_0 = \underbrace{\hat{x}_0}_{estimated\ \hat{x}_0} + \underbrace{\frac{1}{\sqrt{SNR(t)}}(\epsilon_\theta(x_t,t) - \epsilon)}_{amplified\ error}. \tag{7}$$

Although the difference between the predicted $\epsilon_\theta(x_t,t)$ and the target Gaussian noise ϵ may be very small at every step, the amplification coefficient $\frac{1}{\sqrt{\text{SNR}(t)}}$ is expected to be significantly larger as the step t increases (as shown in Fig. 2), which would result in a substantial deviation of the estimated $\hat{x}_0$ from the target x_0. We also visualize the estimated $\hat{x}_0$ and the amplified error at different timesteps via feeding $x_t = \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon$ into the denoising network once. The *estimated* $\hat{x}_0$ increasingly deviates from the ground-truth x_0 when t grows

larger, meanwhile the *amplified error* becomes larger and even gradually approaches x_0. In this regard, we can find that the constant training weight is indeed biased, and optimizing the explicit target ϵ uniformly across different timesteps cannot guarantee approaching the implicit target x_0 exactly.

3.2 Improved Training Strategy

The above theoretical analysis provides a principled guidance on coping with the biased estimation problem and designing the loss weighting strategy (existing weighting strategies can be covered from unified perspective under the proposed principle in Subsec. 5.2). Concretely, besides expecting the loss function to reach the explicit target ϵ, which is relatively simple, more importantly, we desire to encourage the estimated $\hat{x}_0$ to approach the implicit target x_0. Therefore, it is essential to consider the varying impact of noise prediction at different steps t when designing the loss weight. In this regard, we adopt the amplification coefficient in the amplified error part of Eq. 7 as the loss weighting coefficient:

$$L = \sum_t \mathbb{E}_{x_0,\epsilon}\left[\frac{1}{\sqrt{SNR(t)}}||\epsilon - \epsilon_\theta(x_t, t)||^2\right]. \tag{8}$$

In other word, we assign higher weight as the step t increases (i.e., when adding more noise to x_0), thereby compelling the noise error $(\epsilon_\theta(x_t, t) - \epsilon)$ to decrease more significantly at larger step t. Note that the key of this paper is the comprehensive exploration of the bias issue in diffusion models. Grounded in our unlocked bias problem, a simple loss weight design can still achieve substantial performance improvement (refer to the analyses and experiments in the following sections). Besides, we also provide more discussions on the weight selection in the Sec. 5.2 and the supplementary material.

4 Comprehensive Understanding the Bias Problem

In this section, we aim to address several key questions crucial for achieving a systematical understanding of the bias problem in conventional diffusion models: Why is the bias problem important? What are its effects? And what is the underlying cause? We believe answering these questions is essential for unraveling the black box of diffusion models.

4.1 Biased Estimation in Training Process

First, we illustrate the one-step estimation $\hat{x}_0$ in Fig. 3 to compare the results obtained using the original constant weighting and our variant. There is a general tendency for the estimated $\hat{x}_0$ of both weighting strategies to gradually deviate from the original x_0 as the step t increases, which is inevitable due to the increasing noise in the input x_t. However, when utilizing the constant weighting loss for training, noticeable color shifts and inferior arrangement of human faces can be

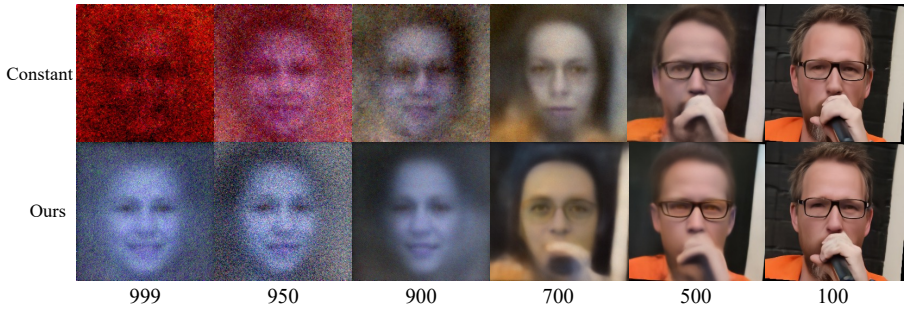


Fig. 3: We present the one-step estimation results of $\hat{x}_0$ using different input samples x_t, where the diffusion models are pretrained on the FFHQ dataset [18] with different loss weighting strategies. One-step estimation: start from a clean image and add noise to get x_t according to Eq. 3. Then put x_t into the denoising network once to get the estimated noise $\hat{\epsilon}$, and the corresponding $\hat{x}_0$. The top row displays the results obtained using a well-trained constant weighting model, while the bottom row depicts the results achieved with our well-trained improved weighting model.

observed in the early steps ($t = 999$ and $t = 950$), severely deviating from the target x_0. In contrast, our strategy effectively reduces the bias, achieving greater consistency with the targets across various timesteps, even under relatively high noise levels (e.g., at $t = 999$ and $t = 950$). These findings indicate that the proposed weighting strategy facilitates training in a more appropriate direction. More analyses are available in Sec. 5.3.

4.2 Biased Generation on Sampling Process

We further analyse the detrimental effects of the biased estimation problem introduced by the constant weighting loss for model inference, i.e., *biased generation* on the sampling process. As seen in the first two rows in Fig. 1, biased generation primarily attributes to the chaos and inconsistency in the early few sampling steps, which substantially affects the final generation through error propagation. We particularly observe pronounced color shifting in biased generation when employing a small number of sampling steps (e.g., $T = 2$), which remains challenging to correct even with an extended sampling process (e.g., $T = 1000$). In contrast, training with our strategy can essentially prevent the issue (e.g., the shown images with $T = 2$), eliminating the need for a lengthy correction process. Moreover, generated images using our strategy show enhanced details and global consistency compared to the baseline method. More visual results and analyses are available in supplementary material.

4.3 Underlying Causes of Biased Estimation

Finally, we take one step further to unravel the underlying causes of biased estimation. Specifically, we find that the optimization difficulty and importance of the denoising network is vastly different across step t.

Different optimization difficulty. Intuitively, the input x_t is closer to the target as step t becomes larger. Consequently, the network encounters varying levels of fitting difficulty across different values of t, with larger values of t being relatively easier. To verify this, we plot the Mean squared error (MSE)-step curve under several settings in Fig. 4. In the "Initial" setting, the MSE value is directly computed between the network input x_t and the target Gaussian noise. The remaining two settings compute the MSE value between the network output and the target Gaussian noise, with "Constant" representing the constant weighting method and "Ours" representing the proposed weighting strategy. The distribution of MSE value under "Initial" mode is extremely unbalanced, in which the MSE value is negligible when $t > 600$. Consequently, this imbalance endows different optimization difficulty across step t and renders the constant weighting strategy suboptimal. Specifically, when t is sufficiently large, the MSE value between the network input and the target becomes extremely small, allowing the network to "do nothing" while still maintaining a low MSE loss.

The above analysis is verified in the right part of Fig. 4. For $t > 950$, the MSE value in constant weight mode surpasses that of the "Initial" mode, indicating the output deviates even further from the target than the input. **This observation illustrates that the denoising network in constant weight setting fails to identify the noise pattern in the input at large step t and, therefore, cannot effectively handle the denoising task.** In contrast, our weight strategy consistently yields MSE values lower than those of the "Initial" mode, demonstrating its exceptional denoising capability, particularly for highly noisy inputs.

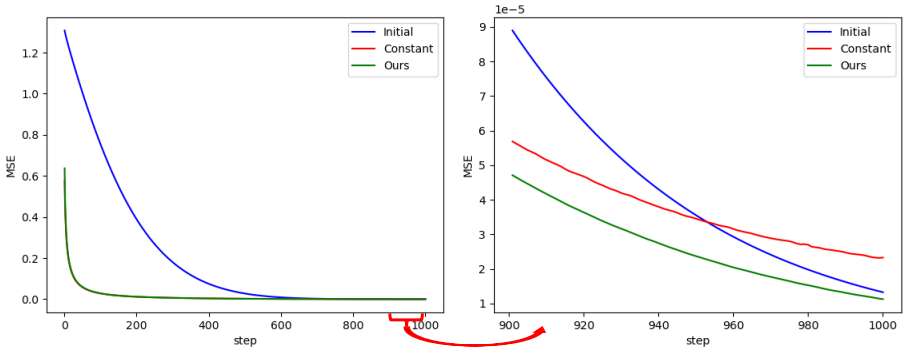


Fig. 4: MSE-step curve under several settings. "Initial" mode is calculated between input and target. Obviously, the optimization difficulty is vastly different across step t. "Constant" and "Ours" modes are calculated between network output and target, and "Constant" denotes constant weight strategy and "Ours" stands for our proposed weight strategy. **Note that the green and red curve visually overlap in the left figure due to large scale.**

Different importance. Lastly, we reveal that the importance of the denoising network also varies across step t. Intuitively, initial steps are important for

both training and sampling process. For training, the initial steps pose greater difficulty due to high noise levels in the input. For sampling, the initial steps serve as the foundation for subsequent steps, contributing to error propagation. Theoretically, we have verified that initial steps should be emphasized to reach the implicit target x_0 in Sec. 3. Additionally, we also find evidence supporting the crucial role of initial steps in diffusion models [26, 39]. For example, Nichol et al. [26] demonstrated that the first few steps contribute the most to the variational lower bound. Wang et al. [39] found that reusing update directions from initial steps with adaptive momentum sampler can generate images with enhanced details. The constant weighting strategy assumes equal importance across all steps. While, our method assigns higher weights to the initial steps, which is consistent with both intuition and theory.

5 Experiments

5.1 Setup

Datasets. We perform experiments on unconditional image generation using the FFHQ [18], CelebA-HQ [15], AFHQ-dog [5], and MetFaces [17] datasets. These datasets contain approximately 70k, 30K, 50k, and 1k images respectively. Besides, we conduct class-conditional generation on CIFAR-10 [21] and ImageNet [6] datasets. We resize and center-crop data to 256×256, following the pre-processing performed by ADM [7].

Training details. We set T = 1000 for all experiments. We implement the proposed approach on top of ADM [7], which offers well-designed architecture. We train our model for 500K iterations with a batch size of 8.

Evaluation settings. Following the common practice [37], we utilize an Exponential Moving Average (EMA) model with a rate of 0.9999 for all experiments. Besides, we generate 50K samples for each trained model and use the full training set to compute the reference distribution statistics, following [4, 13]. During inference, we obtain results with fewer sampling steps than T by employing the respacing technique. For quantitative evaluations, we employ the Fréchet Inception Distance (FID) [12].

5.2 Comparison to Existing Weighting Strategies

Unified perspective on existing weighting strategies. Some methods explore various weighting strategies in ϵ-prediction mode, including P2 [4] and Min-SNR [10]. We present these distinct weighting strategies in Fig. 5. Most methods modify the weight on the basis of the conventional constant weighting with intuition or observation. Compared to using constant weights, they only lower the weights for small t, maintaining the weights unchanged for the remaining substantial portion of the steps. Besides, they encounter difficulties in establishing general principles for guiding the design of the weighting strategy. In contrast, we can take a unified perspective on these weighting strategies within

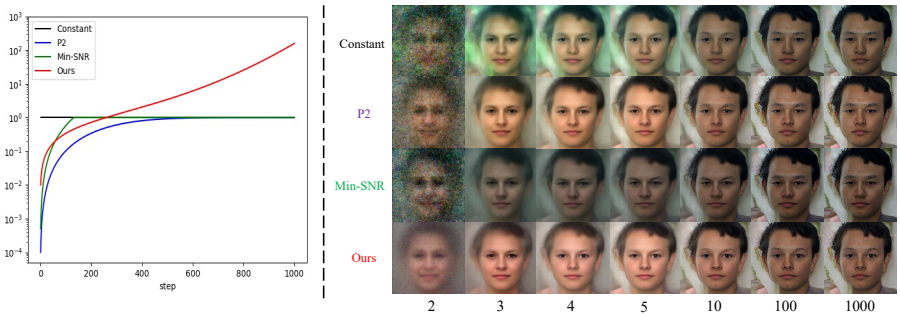


Fig. 5: Left: Visualization of various weighting strategies. P2 and Min-SNR start from the basis of constant weight and lower the weight down for small t. Right: Sampling results with different total sampling steps T. From top to bottom, they are constant, P2, Min-SNR, and our method. Evidently, P2 and Min-SNR still suffer from bias and artifacts during the initial generation stage.

the framework of the unlocked bias analyses. Specifically, the amplification coefficient in Eq. 7 serves as a general principle on the loss weight design. Our theoretically principle elucidates that the weight should monotonically increase as t increases, as depicted with the red curve in Fig. 5. Prior methods [4, 10] tend to assign lower weights to small t values, adhering to the principle overall. These observations can also substantiate the rationale behind their superior performance in comparison to constant weighting. We can also demonstrate that using our improved formulation can further enhance performance, leveraging insights gained from the analysis of bias issues.

The related works in [16,32] employ different training objectives, and detailed experimental comparison and discussions on these works can be found in the supplementary material.

Quantitative comparison. Tab. 1 presents a quantitative performance comparison of various weighting strategies across different sampling steps T. Our method substantially lifts the performance limit across multiple datasets and sampling steps. These datasets are of various scales ranging from 1k to 70k images, which indicates the generalization and robustness of our method. Besides, our method can effectively elevate the performance of the constant weight baseline on all the possible total sampling steps. It is worth noting that the performance gain is particularly pronounced with smaller datasets and shorter sampling steps, which matches our theoretical derivation and extensive analyses.

Tab. 2 presents a quantitative performance comparison on class-conditional generation of various weighting strategies. Besides FID, we also adopt Inception Score (IS) to measure the generation diversity. Obviously, our method surpasses previous methods with better FID score and higher diversity.

Qualitative comparison. Fig. 6 presents the qualitative results. As anticipated, the biased constant weighting strategy produces images with inferior global structure and color alignment. P2 and Min-SNR enhance the sample quality by building upon the constant weight foundation. However, they still produce images with inferior global structure. This is due to their significant bias and

Table 1: Quantitative comparison on unconditional generation. The experimental results are reported in terms of FID under a fair setting, with the only distinction being the loss weighting strategy. * denotes the results reported in the original paper. However, as certain essential training details of P2* (e.g., training iterations) are unknown, its reported values are used for reference only.

Dataset	Step T	Constant	P2	P2*	Min-SNR	Ours
FFHQ	1000	10.864	6.517	6.92	6.501	6.354
	500	11.027	6.792	6.97	6.873	6.706
	250	11.780	7.478	-	7.722	7.385
	100	15.671	10.855	-	11.391	10.815
	50	22.375	16.538	-	17.328	15.345
	20	41.270	34.399	-	34.652	29.380
CelebA-HQ	1000	9.374	7.258	6.91	6.322	5.980
	500	10.236	7.718		6.923	6.572
	250	11.097	8.433	-	8.016	7.604
	100	12.006	9.297	-	9.385	8.836
AFHQ-dog	1000	18.300	17.068	11.55	17.342	14.928
	500	18.606	17.474	-	17.639	14.946
	250	19.104	17.759	11.66	17.922	15.033
	100	20.446	18.344	-	18.421	15.821
MetFaces	1000	41.418	14.204	-	30.876	9.168
	500	42.115	14.448	-	31.168	9.429
	250	42.324	14.738	36.80	31.340	9.849
	100	42.624	14.994	-	31.626	10.388

chaotic behavior during the initial sampling steps, as depicted in Fig. 5. In contrast, our method is totally free of the dilemma of color shift.

5.3 More Analyses

High efficiency. Fig. 7 illustrates the FID-training iterations curve and the FID-sampling steps curve for the FFHQ dataset. The training curve clearly demonstrates the superior efficiency and potential of our method. For instance, our weighting strategy matches the performance of 1000k iterations of constant weight training with only 400k iterations. In terms of sampling, our method surpasses all existing weight strategies across all sampling steps. Moreover, consistent with the analysis in Sec. 4.2, the performance gains are more pronounced with fewer sampling steps.

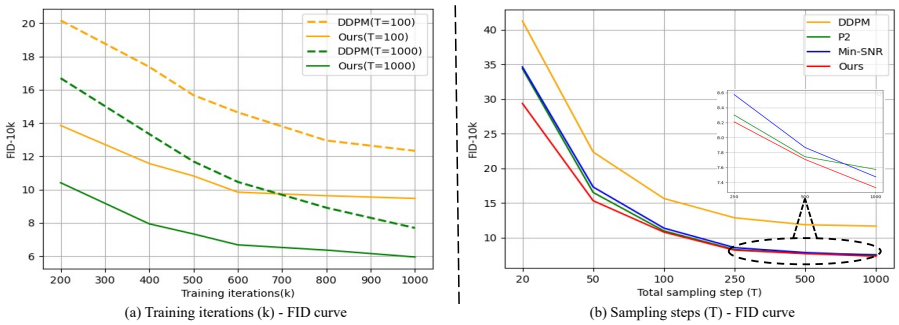
Different samplers. Our weighting strategy is orthogonal to samplers. We conduct additional DDIM sampler [36] to validate this conclusion. As shown in Fig. 8, we depict the generated samples of DDIM sampler under four different weighting strategies on FFHQ dataset. Similar to the conclusion in DDPM sampler in Fig. 6, our method achieves the highest performance with DDIM sampler among these four weighting strategies.

Table 2: Quantitative comparison on class-conditional generation. We employ FID metric to evaluate the distribution distance and IS metric to evaluate the sampling diversity. Our method achieves better results on both metrics.

Dataset	Step T	Metrics	Constant	P2	Min-SNR	Ours
CIFAR-10	1000	FID ↓	11.97	11.71	9.02	8.45
		IS ↑	8.07	8.11	8.14	8.13
ImageNet	100	FID ↓	99.00	95.47	94.81	93.32
		IS ↑	11.96	13.02	13.21	13.37

Fig. 6: Visual results of different weighting strategies on different datasets. We randomly choose the first nine generated images without cherry-pick. The first row is trained on FFHQ dataset and the second row is on AFHQ-dog dataset.

More analyses of the bias in training Process. In this part, we give more analyses and visualization to validate the existence of the bias problem in the training process. Previous image editing methods [11, 38] find that the intermediate feature maps can reflect the structures underlying the noisy samples. We follow this practice with the intuition that the bias problem may also lead to poor structure generation. Concretely, we present the intermediate feature maps at various step t in Fig. 9. Consistent with the conclusion in Fig. 3, the constant weight mode generates poor structures with relatively large noise scale. For example, it struggles to generate clear facial structure when step t >500 (nearly half of the range field). On the contrary, our method has clear facial layout across all timesteps. Especially, the facial structure is visible with the most noisy x_{999}.
Comparison to prior literature. Performance comparison between our method and existing generative methods on the FFHQ dataset [18] is presented in Tab. 3. Previous methods [1, 2, 8, 14, 18, 29, 33] achieve exceptional results with meticulously designed architectures and methodologies. In contrast, our method achieves competitive performance with a simple loss weight strategy. Besides,



(a) Training iterations (k) - FID curve (b) Sampling steps (T) - FID curve

Fig. 7: (a): FID-training iterations curve. (b): FID-sampling steps curve. These two curves are obtained on FFHQ dataset. Our method is more efficient and high-performing. Note that, we use DDPM to denote the constant weighting strategy.

Constant P2 Min-SNR Ours

Fig. 8: The generated samples of DDIM sampler under four different weighting strategies on FFHQ dataset.

our method is a general strategy for diffusion models and can further elevates their performance. For instance, we achieved substantial improvements by solely adjusting the loss weight on top of ADM [7], reducing the FID score from 10.86 to 6.35. Moreover, our method offers the capability to achieve even higher performance. Firstly, we can extend the training duration. For instance, with 500k iterations, our method achieves a FID of 6.35, while with 1000k iterations, it achieves a FID of 4.97. Additionally, we have the flexibility to replace the codebase ADM with a stronger model, such as stable diffusion.

5.4 Discussions

We unlock the biased training problem of diffusion models, which lies as the key of our method. Grounded in our unlocked bias problem, the proposed simple loss weight design can achieve substantial performance improvement. Given that diffusion models usually serve as fundamental building blocks for various application-oriented works, our method provides valuable inspiration and insights for these endeavors. Additionally, we identify several potential avenues for future research. (1) The elucidated mechanism behind the biased problem offers valuable insights for downstream tasks, such as editing and restoration, facilitating the integration of the bias issue into specific tasks. (2) The biased problem

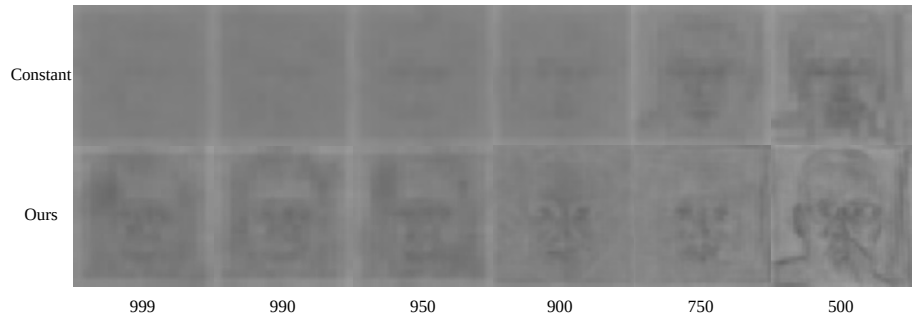


Fig. 9: The intermediate feature maps at different steps t. Intermediate feature maps are correlated with image structure underlying the noisy samples. Obviously, constant strategy struggles to generate clear facial architecture with noisy x_t as input (t >900). In contrast, our method can generate clear facial layout even with the most noisy x_{999} as input.

Table 3: Quantitative comparison to prior generative models on FFHQ dataset. Our method is on top of ADM with only one additional line of code, yet achieving substantial performance lift.

Dataset	Method	Type	FID
FFHQ	BigGAN [2]	GAN	12.4
	UNet GAN [34]	GAN	10.9
	StyleGAN [18]	GAN	4.16
	StyleGAN2 [19]	GAN	3.73
	VQGAN [8]	GAN+AR	9.6
	LDM [29]	Diffusion model	4.98
	ADM (Baseline) [7]	Diffusion model	10.86
	Ours (500k iterations)	Diffusion model	6.35
	Ours (1000k iterations)	Diffusion model	4.97

can be investigated from other perspectives, such as noise schedule [3, 27]. It is encouraging to discuss the defects of diffusion models from a unified perspective.

6 Conclusion

This paper provides theoretical analyses and comprehensive studies to demonstrate that the traditional uniform weighting loss function is suboptimal, by examining the existence, impact, and underlying reasons behind this issue. To mitigate this problem, we employ a simple yet highly effective weighting strategy. Empirical studies conducted on multiple datasets, along with comparisons with existing weight methods, further validate the effectiveness of our approach. We also believe these analyses contribute to a deeper understanding of the underlying mechanism of diffusion models.

Acknowledgements. This work was supported by the Anhui Provincial Natural Science Foundation under Grant 2108085UD12. We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

1. Bao, F., Nie, S., Xue, K., Cao, Y., Li, C., Su, H., Zhu, J.: All are worth words: A vit backbone for diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22669–22679 (2023) 12
2. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018) 12, 14
3. Chen, T.: On the importance of noise scheduling for diffusion models. arXiv preprint arXiv:2301.10972 (2023) 14
4. Choi, J., Lee, J., Shin, C., Kim, S., Kim, H., Yoon, S.: Perception prioritized training of diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11472–11481 (2022) 2, 3, 4, 9, 10
5. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8188–8197 (2020) 9
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) 9
7. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in Neural Information Processing Systems **34**, 8780–8794 (2021) 1, 5, 9, 13, 14
8. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021) 12, 14
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014) 1
10. Hang, T., Gu, S., Li, C., Bao, J., Chen, D., Hu, H., Geng, X., Guo, B.: Efficient diffusion training via min-snr weighting strategy. arXiv preprint arXiv:2303.09556 (2023) 2, 3, 4, 9, 10
11. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022) 12
12. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017) 9
13. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020) 1, 3, 4, 9
14. Ho, J., Saharia, C., Chan, W., Fleet, D.J., Norouzi, M., Salimans, T.: Cascaded diffusion models for high fidelity image generation. The Journal of Machine Learning Research **23**(1), 2249–2281 (2022) 12
15. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017) 9
16. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. Advances in Neural Information Processing Systems **35**, 26565–26577 (2022) 2, 4, 10

17. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. Advances in neural information processing systems **33**, 12104–12114 (2020) 9
18. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4401–4410 (2019) 7, 9, 12, 14
19. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8110–8119 (2020) 14
20. Kingma, D., Salimans, T., Poole, B., Ho, J.: Variational diffusion models. Advances in Neural Information Processing Systems **34**, 21696–21707 (2021) 3
21. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009) 9
22. Liu, L., Ren, Y., Lin, Z., Zhao, Z.: Pseudo numerical methods for diffusion models on manifolds. In: International Conference on Learning Representations (2021) 1
23. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: 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) 1
24. Mardani, M., Song, J., Kautz, J., Vahdat, A.: A variational perspective on solving inverse problems with diffusion models. arXiv preprint arXiv:2305.04391 (2023) 2, 3, 4, 5
25. Meng, C., Rombach, R., Gao, R., Kingma, D., Ermon, S., Ho, J., Salimans, T.: On distillation of guided diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14297–14306 (2023) 1
26. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning. pp. 8162–8171. PMLR (2021) 1, 9
27. Ning, M., Sangineto, E., Porrello, A., Calderara, S., Cucchiara, R.: Input perturbation reduces exposure bias in diffusion models. International Conference on Machine Learning (2023) 14
28. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022) 1
29. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022) 1, 12, 14
30. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22500–22510 (2023) 1
31. Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D.J., Norouzi, M.: Image super-resolution via iterative refinement. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(4), 4713–4726 (2022) 1
32. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. In: International Conference on Learning Representations (2022) 1, 2, 4, 10
33. Sauer, A., Chitta, K., Müller, J., Geiger, A.: Projected gans converge faster. Advances in Neural Information Processing Systems **34**, 17480–17492 (2021) 12
34. Schonfeld, E., Schiele, B., Khoreva, A.: A u-net based discriminator for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8207–8216 (2020) 14

35. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International Conference on Machine Learning. pp. 2256–2265. PMLR (2015) 1, 3
36. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (2020) 1, 11
37. Song, Y., Ermon, S.: Improved techniques for training score-based generative models. Advances in neural information processing systems **33**, 12438–12448 (2020) 4, 9
38. Tumanyan, N., Geyer, M., Bagon, S., Dekel, T.: Plug-and-play diffusion features for text-driven image-to-image translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1921–1930 (2023) 12
39. Wang, X., Dinh, A.D., Liu, D., Xu, C.: Boosting diffusion models with an adaptive momentum sampler. arXiv preprint arXiv:2308.11941 (2023) 9
40. Yu, H., Huang, J., Zheng, K., Zhou, M., Zhao, F.: High-quality image dehazing with diffusion model. arXiv preprint arXiv:2308.11949 (2023) 1
41. Yu, H., Luo, H., Wang, F., Zhao, F.: Uncovering the text embedding in text-to-image diffusion models. arXiv preprint arXiv:2404.01154 (2024) 1