Beta-Tuned Timestep Diffusion Model

Tianyi Zheng[1,2], Peng-Tao Jiang[2], Ben Wan[1], Hao Zhang[2], Jinwei Chen[2], Jia Wang[1] ✉, and Bo Li[2] ✉

[1] Shanghai Jiao Tong University, Shanghai, China
{tyzheng, burn-w, jiawang}@sjtu.edu.cn
[2] vivo Mobile Communication Co., Ltd, China
{pt.jiang,haozhang,jinwei.chen,libra}@vivo.com

Abstract. Diffusion models have received a lot of attention in the field of generation due to their ability to produce high-quality samples. However, several recent studies indicate that treating all distributions equally in diffusion model training is sub-optimal. In this paper, we conduct an in-depth theoretical analysis of the forward process of diffusion models. Our findings reveal that the distribution variations are non-uniform throughout the diffusion process and the most drastic variations in distribution occur in the initial stages. Consequently, simple uniform timestep sampling strategy fail to align with these properties, potentially leading to sub-optimal training of diffusion models. To address this, we propose the Beta-Tuned Timestep Diffusion Model (B-TTDM), which devises a timestep sampling strategy based on the beta distribution. By choosing the correct parameters, B-TTDM aligns the timestep sampling distribution with the properties of the forward diffusion process. Extensive experiments on different benchmark datasets validate the effectiveness of B-TTDM.

Keywords: Generative Models · Diffusion Models · Beta Distribution

1 Introduction

Diffusion models have gained widespread application across various domains in recent years, such as text-to-image generation [5, 43, 50], text generation [31, 52], video synthesis [21] and AI security [6, 12–14, 32], primarily due to their ability to produce high-quality samples. During the training stage, diffusion models introduce varying levels of Gaussian noise into the data distribution, creating a range of altered distributions. The model is then trained by minimizing denoising score-matching losses across these diverse distributions. In the inference stage, the trained model reverses this noise addition process to generate new samples.

While DDPM [22], iDDPM [37] and ADM [9] have achieved significant success in image generation tasks, several studies [7, 17, 19, 48] identify that treating the distribution equally across different timesteps in the training stage could be

✉ Corresponding authors.
This work was done during Tianyi Zheng's internship at vivo.

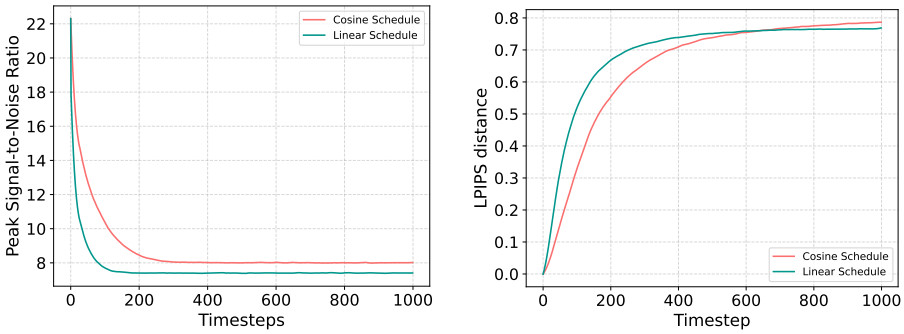
sub-optimal. For instance, the P2-Weight [7] method discover through empirical analysis that distributions at various timesteps in forward diffusion encapsulate distinct semantic information. Building on this insight, P2-Weight propose a re-weighted loss function to elevate the quality of generated samples. Meanwhile, Min-SNR [19] and ANT [17] find inconsistencies in optimization gradients across various timesteps and recommend a re-weighted loss strategy to mitigate these issues. Recently, E-TSDM [48] uncover that the issue of numerical instability caused by high Lipschitz constants near the zero point. Therefore, E-TSDM assigns a weight of 0 to certain timesteps near zero to mitigate the issue of numerical instability. Despite the improvement in sample quality achieved by these methods through empirical adjustments to the training objective's weight, they still utilize DDPM's uniform timestep sampling strategy. As another influencing factor, the choice of timestep sampling strategy is also closely related to the training objective but has been overlooked in previous work. In this paper, we conduct a theoretical analysis of DDPM's forward diffusion process and design a novel strategy for timestep sampling, aiming to address this overlooked aspect and further improve the quality of generated samples.

Our theoretical analysis in Theorem 2 demonstrates that distribution variations throughout the diffusion process are non-uniform, with the most significant variations occurring at the initial stages. This is experimentally supported by Figure 1, which shows non-uniform variations in both Peak Signal-to-Noise Ratio (PSNR) and Learned Perceptual Image Patch Similarity (LPIPS) [51] metrics, especially pronounced in the initial stages. Both our theoretical insights from Theorem 2 and the experimental evidence in Figure 1 underscore the non-uniform properties of forward diffusion, indicating that a uniform timestep sampling strategy may not be the most effective approach in the training stage.

Building on these findings, we propose a novel timestep sampling strategy that utilizes the beta distribution, named the Beta-Tuned Timestep Diffusion Model (B-TTDM). By choosing the correct parameters of the beta distribution, B-TTDM aligns the timestep sampling distribution with the observed properties of the diffusion process and is specifically tuned to avoid singularities at $t = 0$ timesteps. Extensive experiments on various datasets demonstrates that B-TTDM not only improves the quality of the generated samples but also expedites the training process. For instance, B-TTDM achieves similar FID scores on the ImageNet dataset with about 20% of the training iterations needed by the original approach and significantly improves generation quality after convergence. Moreover, B-TTDM can be compatible with other advanced diffusion model training methods, such as P2-Weight [7], Min-SNR [19], Debias [19], and DDPM-IP [39]. Through integration with these methods, B-TTDM can combine different advantages and generate higher quality samples. In summary, our contributions are:

- Through theoretical and experimental verification, we confirm that distribution variations at different timesteps during forward diffusion are non-uniform, with the most significant shifts occurring in the initial stages. This indicates that the uniform time sampling in training is sub-optimal.

– We design a novel timestep sampling strategy leveraging the Beta distribution to align with the non-uniform distribution shifts observed during forward diffusion. Furthermore, our B-TTDM is compatible with other enhancement methods, allowing for flexible integration into existing improved methods.
– Extensive experiments on different benchmark datasets demonstrate that the B-TTDM method significantly improves generation quality and drastically speeds up training of the diffusion models.



(a) PSNR at each forward diffusion timestep. **(b)** LPIPS at each forward diffusion timestep.

Fig. 1: The PSNR and LPIPS metrics between the distribution at timestep t and the original distribution in forward diffusion under different noise schedules.

2 Related Work

Diffusion Models. Diffusion models [15,22,37] are a family of generative models that generate samples from Gaussian noise via a learned denoising process. Recently, diffusion models have outperformed flow models [10,27] and variational autoencoders [16, 28] in generating higher quality samples. Furthermore, the diffusion models also outperform GANs [18, 47] in certain cases [9]. This superior performance has led to the widespread adoption of diffusion models across diverse applications. For example, the Diffusion model [43] implements the generation of images based on textual inputs [4, 11], image to image translation [35, 40] and 3D Avatar generation [41]. Despite their success, the speed of sample generation by diffusion models is unsatisfactory. Therefore, DPM-Solver [34] and DDIM [44] design new generation methods based on the framework of ordinary differential equation (ODE), which have greatly improved the generation speed of the diffusion model. Meanwhile, EDM [25] design a higher order Runge-Kutta sampling method to further improve the generation speed and quality. These methods significantly enhance the speed of sample generation in diffusion models, with a primary emphasis on the generative aspect. Yet, they rely on pre-trained diffusion models without addressing training optimizations. Our proposed method presents an

orthogonal strategy, aiming to improve the training stage and thereby further elevate the generate quality of diffusion models.

Improved Diffusion Models. To further improve the quality of generative samples in the training stage, ADM [9] and U-ViT [2] design new advanced architectures for diffusion models. Based on these advanced architectures, the quality of generation is significantly improved. Additionally, P2-Weight [7], Min-SNR [19] and Debias [49] methods design the re-weight loss function based on SNR in the training stage. These can reduce the bias problem in the diffusion models. To make the neural network smoother, the E-TSDM [48] sets the weight of the loss function corresponding to timesteps close to the zero to 0. Additional, ANT [17] propose multi-task learning methods to set the loss weight of different timestep in the training stage. Although these improved methods can produce better samples, they overlook the non-uniform distribution variations in the diffusion process and continue to use uniform timestep sampling during training. Our theoretical analysis indicates that this uniform time sampling strategy in the training stage is not optimal, suggesting that a more tailored timestep sampling approach could yield significant improvements in generated samples.

3 Method

3.1 Background

Let $q(x_0)$ represent the data distribution over $\mathbb{R}^n$. During the forward diffusion, a sample x_0 is drawn from $q(x_0)$ and subsequently perturbed using a stochastic differential equation (SDE) [46].

$$\mathrm{d}x_t = f(x_t, t)\mathrm{d}t + g(t)\mathrm{d}w. \tag{1}$$

The purpose of the forward diffusion is transform the data distribution into simple normal Gaussian distribution $\mathcal{N}(0, \boldsymbol{I})$. Meanwhile, we define $q_t(x_t)$ to represent the distribution at each time t during the forward diffusion process. For any time t, the conditional distribution $q_t(x_t|x_0)$ is [9, 22, 37]

$$q_t\left(x_t \mid x_0\right) = \mathcal{N}\left(x_t \mid \alpha_t x_0, \sigma_t^2 \mathbf{I}\right). \tag{2}$$

In Equation (2), α_t and σ_t satisfy $\alpha_t^2 + \sigma_t^2 = 1$. This forward diffusion process is known as the Ornstein-Uhlenbeck (OU) process. For more general expressions, we have the following conditional distribution

$$q_t\left(x_t \mid x_0\right) = \mathcal{N}\left(x_t \mid e^{-\frac{t}{2}} x_0, (1 - e^{-t})\mathbf{I}\right). \tag{3}$$

The reverse diffusion transforms normal Gaussian distribution into the data distribution $q(x_0)$. This process satisfies the following reverse SDE equation [1].

$$\mathrm{d}x_t = \left(f\left(x_t, t\right) - g(t)^2 \nabla_x \log q_t\left(x_t\right)\right)\mathrm{d}t + g(t)\mathrm{d}w. \tag{4}$$

Specially, $\nabla_x \log q_t(x_t)$ is the score function of the probability density $q_t(x_t)$. Motivated by this correspondence, diffusion models train a neural network to

estimate the $-\sigma_t \nabla_x \log q_t(x_t)$. In the inference stage, we can simulate Equation (4) for sampling and generate new samples.

3.2 Non-uniform variation in forward diffusion

We first start by introducing and discussing our main assumptions. The only assumption we consider is that in the vicinity of high-density data, the score of the data distribution is bounded. Based on the definition of score [23, 45, 46], this assumption is quite common used [3, 30].

Assumption 1 *For the diffusion process described by Equation* (3)*, assume there exists a constant* $\delta > 0$*. For any* $t \in [0, T]$ *and any point* x_t *in high-density regions, the score* $\|\nabla_x \log q_t(x_t)\|$ *is bounded by* δ*, i.e.,*

$$\|\nabla_x \log q_t(x_t)\| \leq \delta.$$

This ensures that the score of the data distribution is bounded in the vicinity of high-density data throughout the diffusion process.

Next, we examine the Fokker-Planck equation [42] corresponding to the Ornstein-Uhlenbeck process, as depicted in Equation (5). The parameter θ is drift coefficient and the parameter D is diffusion coefficient.

$$\frac{\partial}{\partial t} q(x_t) = \theta \nabla_{x_t} q(x_t) + D \nabla^2_{x_t} q(x_t). \tag{5}$$

Since the θ and D are both constants independent of x_t, to simplify the analysis that follows, we can let both of them be 1. Therefore, we have the following simplified Fokker-Planck equation.

$$\frac{\partial}{\partial t} q(x_t) = \nabla_{x_t} q(x_t) + \nabla^2_{x_t} q(x_t). \tag{6}$$

The Fokker-Planck equation (6) sheds light on how a distribution's evolution is influenced by both time and spatial variables in the forward diffusion stage of DDPM. Our analysis further reveals that in the initial stages of diffusion, the changes in the distribution are primarily driven by the second-order derivative term, indicating a significant diffusion effect at this stage. Therefore, we have the following Lemma:

Lemma 1. *Consider the forward diffusion process as described by Equation (3). When* $t \to 0^+$*, the time derivative of the distribution* $q_t(x_t)$ *is predominantly governed by the second-order derivative term, encapsulated in the relationship:*

$$\frac{\partial}{\partial t} q_t(x_t) = \nabla^2_{x_t} q(x_t), \tag{7}$$

where $\nabla^2_{x_t}$ *denotes the Laplacian operator with respect to* x_t*, indicating the diffusion term's dominance in the evolution of* $q_t(x_t)$ *when* $t \to 0^+$*.*

The detailed proof of the Lemma (1) is shown in Appendix Section 2.2. Lemma (1) suggests that the alteration in the distribution primarily stems from the $\nabla^2_{x_t} q(x_t)$. Expanding on this insight, we can establish the following theorem, providing an upper bound on the alteration in distribution.

Theorem 2. *Consider the forward diffusion process described by Equation (7). The norm of the time derivative of the distribution $q_t(x_t)$ is bounded by the following inequality:*

$$\left\| \frac{\partial}{\partial t} q_t(x_t) \right\| \leq \left\| \frac{q_t(x_t)}{2t} \right\|. \tag{8}$$

As $t \to 0^+$, the norm of the derivative of the distribution $q_t(x_t)$ may become unbounded. This is indicated by the limit superior:

$$\limsup_{t \to 0^+} \left\| \frac{q_t(x_t)}{2t} \right\| = \infty. \tag{9}$$

The detailed proof of the Theorem (2) is shown in Appendix Section 2.3. Theorem (2) suggests that the bound on the norm of the time derivative of $q_t(x_t)$ grows indefinitely as $t \to 0^+$, indicating that the norm of the derivative could be unbounded. We present a straightforward example to illustrate this behavior.

Simple case illustration. Let $q_t(x_t) = \frac{1}{\sqrt{4\pi t}} e^{-\frac{x^2}{4t}}$ be a solution to differential equation (7). Meanwhile, we notice that $\int_{-\infty}^{+\infty} q_t(x_t) dx = 1$ and $q_t(x) > 0$, which means $q_t(x)$ is a probability density function. For every $t > 0$, the norm of the time derivative of $q_t(x)$ is given by:

$$\left\| \frac{\partial}{\partial t} q_t(x_t) \right\| = \left\| \frac{\sqrt{t}(-2t + x^2) e^{-\frac{x^2}{4t}}}{8\sqrt{\pi} t^3} \right\|. \tag{10}$$

This expression leads to the conclusion that:

$$\limsup_{t \to 0^+} \left\| \frac{\partial}{\partial t} q_t(x_t) \right\| = \infty, \tag{11}$$

indicating that the norm of the time derivative of $q_t(x)$ becomes unbounded as $t \to 0^+$. This result indicates an infinite rate of variation for the distribution $q_t(x_t)$ when $t \to 0^+$.

Theorem (2) highlights the non-uniform properties of distribution variations during diffusion, with the initial stages experiencing the most significant shifts. Furthermore, Proposition (1) clarifies that the distribution variations $q_t(x_t)$ are non-uniform across the diffusion process, and these variations tend to flatten as the process progresses to later stages.

Proposition 1. *Consider the forward diffusion process in Equation (3). The upper bound of the KL divergence between the distribution $q_t(x_t)$ and the standard Gaussian distribution converges exponentially. i.e.,*

$$\mathrm{KL}\left(q_t(x_t)\|\mathcal{N}(0,1)\right) \leq e^{-t}\mathrm{KL}\left(q_0(x_0)\|\mathcal{N}(0,1)\right). \tag{12}$$

Based on the Theorem (2) and Proposition (1), we notice that the variations in the distribution of forward diffusion is non-uniform, with the sharpest change at the beginning of the diffusion and converging to a Gaussian distribution at an exponential rate. These theoretical insights guide the design of our timestep sampling strategy.

3.3 Beta distribution timestep sampling

To align the non-uniform trend of variation, we introduce a time sampling strategy that utilizes the Beta distribution for training the diffusion model. This strategy leverages the Beta density function's capacity to take on a diverse array of shapes, controlled by the parameters α and β. The probability density function (pdf) of the Beta distribution is defined as:

$$f(t;\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} t^{\alpha-1}(1-t)^{\beta-1}, \tag{13}$$

Where $\Gamma(z)$ is the gamma function. Setting $\alpha = 1$ and $\beta = 1$ simplifies the Beta distribution to a uniform distribution, illustrating the flexibility of Beta distribution. By adjusting the values of α and β, we can tailor the time sampling distribution within the diffusion model training to better align the desired variation properties.

Since the beta distribution is a family of continuous probability distributions defined on the interval $[0, 1]$. We can use $\frac{t}{T}$ to map the diffusion timestep into the $[0, 1]$ interval of the beta distribution. Then we compare the probability density function of beta distribution for different α and β in Figure 2. We find that when we set $\alpha > 1$ and $\beta > 1$, The probability density function is U-shaped. Meanwhile, we notice that $\alpha = 1$ and $\beta > 1$ and $\alpha < 1$ and $\beta = 1$, the probability density function conforms to our expectation of a non-uniformly decreasing. However, when $\alpha = 1$ and $\beta > 1$, the probability density function exhibits a positively skewed shape, lacking alignment with the most significant varies in the initial diffusion stage. We provide ablation study of different α and β in the section 4.4.

To align with the previously identified properties, we choose to set $\alpha < 1$ and $\beta = 1$. With these parameters, the distribution adopts a reverse J-shaped profile. Consequently, the probability density function under these conditions becomes a strictly decreasing function, exhibiting a sharp change as it approaches 0. This specific shape and behavior of the pdf are instrumental in aligning the rapid initial changes and the convergence properties observed in the forward diffusion process. Therefore, we replace the uniform time sampling in the original diffusion model with this time sampling method based on the beta distribution. In addtion, to prevent the probability density function from reaching infinite values as time t approaches 0, we adopt $\frac{1}{T}$ as the time sampling weight at the initial

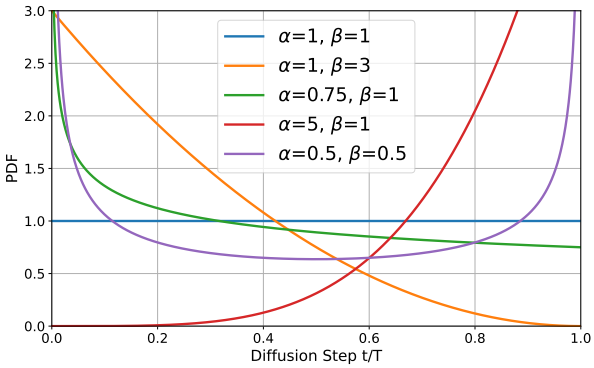


Fig. 2: Probability density function of beta distribution with different parameters.

moment. Therefore, our method is called Beta-Tuned Timestep Diffusion Model (B-TTDM). Next, we analyse the errors introduced by this tuned approach.

Proposition 2. *Given a timestep sampling method utilizing the B-TTDM with a Beta distribution parameterized by $\alpha < 1$ and $\beta = 1$, the tuned error at the initial time position t is $\left(\frac{1}{T}\right)^{\alpha}$, where T is the total diffusion steps.*

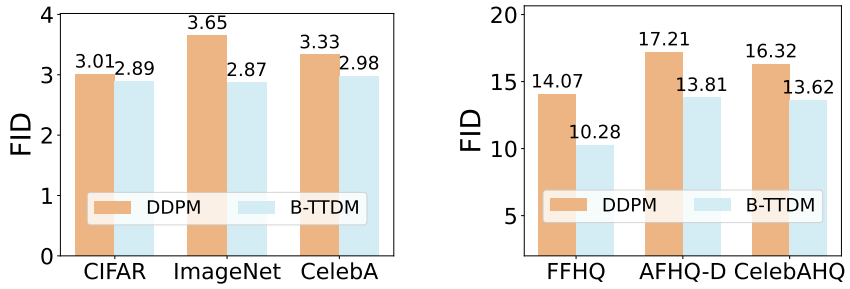
The detailed proof of the Proposition (2) is shown in Appendix Section 2.4. Since the value of T is usually large in the training of diffusion models (e.g.,T=1000), this implies that the tuned error is quite small. Therefore, we can ignore it and still consider this as a beta distribution. Finally, the corresponding training loss can be written as

$$\mathcal{L}\left(\epsilon_\theta\right) := \mathbb{E}_{t\sim\text{Beta}(\alpha,\beta),\mathbf{x}_0\sim q(\mathbf{x}_0),\epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[\left\|\epsilon_\theta\left(\alpha_t\mathbf{x}_0+\sigma_t\epsilon,t\right)\right)-\epsilon\right\|_2^2\right]. \tag{14}$$

Furthermore, we present a comprehensive outline of the B-TTDM method in Algorithm (1). The key difference between B-TTDM and DDPM is the strategy to timestep sampling in the training stage. Thus, B-TTDM is adaptable to most fast sampling methods and improved training methods designed for DDPM without the need for specific modifications.

Algorithm 1 Training of B-TTDM

Require: Beta distribution parameter α and β. Total diffusion timestep T.
1: Generate a beta distribution with parameters α and β.
2: Tuned $f(0;\alpha,\beta) = f(\frac{1}{T};\alpha,\beta)$
3: **repeat**
4: $\quad x_0 \sim q(x_0)$
5: $\quad t \sim \text{Beta}(\alpha,\beta), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\boldsymbol{I})$
6: $\quad$ Take a gradient descent step on $\nabla_{\boldsymbol{\theta}}\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t,t\right)\right\|^2$
7: **until** converged



(a) Comparison of FID on low-resolution. **(b)** Comparison of FID on hign-resolution.

Fig. 3: Quantitative Comparison. Comparing FID of different resolution datasets.

4 Experiment

In this section, we first provide an implementation details of the experimental setup in section 4.1. Then, we provide quantitative comparison of B-TTDM with other state-of-the-art methods in section 4.2. In Section 4.3, we assess the robustness of B-TTDM to different fast sampling algorithms and its training efficiency in comparison to DDPM. Subsequent ablation studies are discussed in Section 4.4. Lastly, qualitative comparisons and visualization results are showcased in Section 4.5.

4.1 Experiment Setup

Implementation details. All of our experiments utilize the same settings of DDPM [22]. Meanwhile we employ the same model architecture as ADM [9] since it is more advanced compared to DDPM baseline. Throughout the training stage, we maintain $T = 1,000$ for all experiments. We conduct comparative experiments using the CIFAR-10 (32×32), ImageNet (32×32), CelebA (64×64) [33], FFHQ (128×128) [26], AFHQ-D (256×256) [8] and CelebAHQ (256×256) [24] datasets for unconditional image generation. To evaluate the sampling quality, we use the Frechet inception distance (FID) [20], sFID [36], improved Precision and Recall [29] as our evaluation metrics. To be more specific, we generate 50K samples for CIFAR-10, ImageNet, CelebA, FFHQ and 10K samples for AFHQ-D, CelebAHQ. In the inference stage, we use 1000 steps for CIFAR-10 and ImageNet, 300 steps for CelebA and FFHQ, and 100 steps for AFHQ-D and CelebAHQ. More details can be found in the Appendix Section 3.

4.2 Quantitative Comparison

Comparison to Baseline. We compare our method with DDPM baseline using various benchmark datasets. The results of the experiments are shown in Figure. 3, where we find that the FID of our method significantly outperforms the DDPM

on datasets of different resolutions. For instance, on the FFHQ dataset, B-TTDM achieved a FID improvement of 3.79. This underscores our method's efficacy and suggests that the uniform time sampling approach previously employed for training diffusion models is not the most efficient. Notably, B-TTDM reaches an FID of 2.87 on the ImageNet dataset with a larger data size, significantly enhancing DDPM's performance and highlighting B-TTDM's effectiveness.

Table 1: Quantitative Comparison. Comparison of FID and sFID with other state-of-the-art methods on different resolution datasets.

Method	CIFAR 32×32		FFHQ 128×128		AFHQ-D 256×256		CelebAHQ 256×256	
	FID	sFID	FID	sFID	FID	sFID	FID	sFID
DDPM	3.01	4.69	14.07	12.89	17.21	49.03	16.32	25.20
Debias	3.17	4.56	11.86	11.65	15.91	46.77	14.23	21.98
Min-SNR	3.23	4.80	12.46	12.85	17.03	47.94	14.37	24.87
P2-Weight	3.19	4.78	11.96	12.74	16.18	47.88	14.38	22.14
B-TTDM	**2.89**	**4.22**	**10.28**	**11.38**	**13.81**	**45.25**	**13.62**	**21.93**

Comparison to other methods. Since the B-TTDM method is a non-uniform treatment of different time intervals, we contrast it with other start-of-the-art methods that employ non-uniformly re-weighted loss functions in the training stage. The result shown in Table 1 indicates that our B-TTDM perform best across multiple different resolution datasets. In addition, we find that most of these re-weighted loss methods have negative effect on low-resolution CIFAR datasets. Despite this, B-TTDM consistently achieves the best performance, underscoring its broad applicability and generalizability.

Plug-and-Enhance. Our B-TTDM is designed to align the time sampling and the non-uniform variation observed in the data distribution during forward diffusion, setting it apart from other techniques aimed at enhancing diffusion model performance. Therefore, our method is complementary and can be integrated as a plug-in enhancement within existing strategies for optimizing the training of diffusion models. We combine our B-TTDM with other different improved training methods during the training stage following the same setting. The results are shown in Table 2, we find that our B-TTDM is beneficial for other improved methods like P2-Weight, Min-SNR and DDPM-IP. Notably, the integration of B-TTDM with DDPM-IP leads to a significant achievement, such as an FID score of 2.46 for the CIFAR dataset and a 6.69 FID score for FFHQ.

4.3 Fast Sampling and Training

Fast Sampling. While diffusion models produce high-quality samples, their slow inference speed constrains their applicability in real-world settings. To improve this, various methods like DDIM [44], Time Respacing [9], DPM-Slover [34] and Exposure Bias [38] have been introduced. Fortunately, our B-TTDM method maintains the reliance of these acceleration techniques on the training process,

Table 2: Quantitative Comparison. Comparison of FID and sFID when combining B-TTDM with other improved training methods.

Method	CIFAR		FFHQ	
	FID	sFID	FID	sFID
DDPM-IP	2.72	4.54	7.07	8.35
+ B-TTDM	**2.46**	**3.95**	**6.69**	**8.10**

Method	AFHQ-D	
	FID	sFID
Min-SNR	17.03	47.94
+ B-TTDM	**13.34**	**46.13**

Method	AFHQ-D	
	FID	sFID
P2-Weight	16.18	47.88
+ B-TTDM	**12.45**	**46.93**

allowing for their straightforward integration into our approach. The fast sampling outcomes of B-TTDM and DDPM models are contrasted in Table 3. We find that the B-TTDM method significantly outperforms the DDPM method under each different fast samplers and different NFE. Thus, B-TTDM can be seamlessly applied with various fast samplers for DDPM without requiring specific tuning.

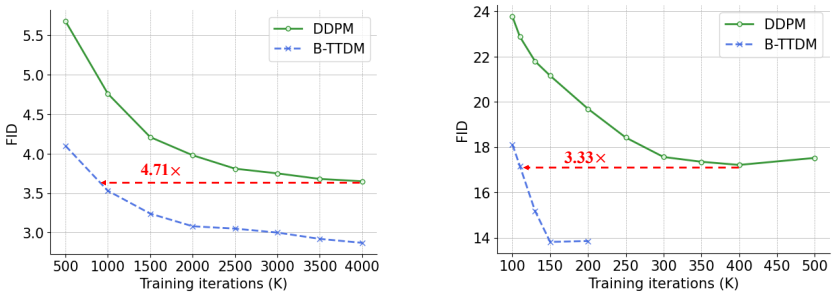
Fast Training. In addition to yielding higher-quality samples, B-TTDM also accelerates the training process of diffusion models. In Figure 4, we present a comparison between B-TTDM and DDPM in terms of performance and the training iterations on the large scale ImageNet and high resolution AFHQ-D datasets. Specifically, for ImageNet, B-TTDM achieves a 4.71× acceleration in training speed to attain equivalent FID scores. Once training converges, B-TTDM's FID score significantly exceeds that of DDPM. Similarly, on the large resolution AFHQ-D dataset, B-TTDM also achieves a 3.33× training speed-up and perform better than DDPM upon convergence. We attribute this acceleration to our beta-tuned timestep sampling, which align the timestep sampling distribution with the properties of the diffusion process. This alignment not only speed up the training of diffusion models but also improve the generated samples quality.

Table 3: Quantitative comparison. Comparison of FID score across various Fast Samplers on AFHQ-D dataset. NFE means the number of function evaluations.

Fast Samplers		**Epsilon Scaling**		**DDIM**		**Respacing**		**DPM-Solver**	
NFE		25	50	25	50	25	50	25	50
Method	DDPM	23.63	18.50	24.22	18.25	23.64	19.83	18.49	17.49
	B-TTDM	**20.81**	**15.34**	**21.76**	**16.77**	**21.79**	**16.63**	**15.82**	**14.79**

4.4 Ablation Study

Parameters of the beta distribution. The B-TTDM utilizes the Beta-tuned distribution to devise a novel time sampling strategy. This distribution is defined by two parameters, α and β, where α predominantly influences the left side of the distribution's shape. As α increases, the bulk of the probability density function shifts towards the right. Conversely, β primarily affects the right side of the distribution's shape. To further explore their impact on diffusion model training, we experiment with various α and β combinations, with the experimental



(a) ImageNet (32×32) **(b)** AFHQ-D (256×256)

Fig. 4: FID scores concerning the number of training iterations on ImageNet (32×32) and AFHQ-D (256×256) dataset .

outcomes presented in Table 4. We find that beta-tuned timestep sampling leads to significant improvements in all metrics and achieves the best performance at $\alpha = 0.8, \beta = 1$. This also implies that beta-tuned timestep sampling is beneficial for generating quality improvements in diffusion models. More experimental results of different parameters can be found in the Appendix Section 3.

Noise schedules. To ensure the robustness of the B-TTDM method to different noises schedules, we try two different commonly used noise schedules, cosine and linear noise schedules [9] on the CIFAR10 dataset. We follow the previous setting and each FID and sFID score is computed using $T' = 1000$ inference steps. The results of the different noise schedules experiments are presented in Table 5. The B-TTDM is significantly better than the Baseline in both FID and sFID scores, which indicates that the B-TTDM method is robust to different noise schedules.

Table 4: Ablation study on the parameters of Beta Distribution.

Beta Distribution		AFHQ-D (256×256)			
α	β	FID	sFID	Recall	Precision
1	1	17.21	49.03	0.553	0.738
0.95	1	14.12	46.46	0.562	0.805
0.90	1	13.85	45.47	0.612	0.752
0.80	1	**13.81**	**45.25**	**0.612**	**0.819**
0.70	1	14.78	47.56	0.564	0.817

Table 5: Ablation study on the Noise Schedule on CIFAR.

Noise Schedule		DDPM	B-TTDM
Linear	FID	3.53	**3.28**
	sFID	4.78	**4.41**
Cosine	FID	3.01	**2.89**
	sFID	4.69	**4.22**

Other timestep adjust method. In this section, we compare various methods for adjusting the timestep on CIFAR and AFHQ datasets. For instance, the ANT-UW [17] method employs Uncertainty Weighting for different timestep based on multi-task learning with varying variances. Additionally, we compare timestep sampling methods based on alternative non-uniform distributions, such as exponential distributions (DDPM-E) and those where $\beta > 1$ and $\alpha = 1$ (B-TTDM$^{\alpha}$). The experimental results are presented in Table 6. It is evident that

Table 6: Quantitative comparison. Ablation study on different non-uniform timestep sampling method in the training stage of diffusion model.

Method	CIFAR (32×32)				AFHQ (256×256)			
	FID	sFID	Recall	Precision	FID	sFID	Recall	Precision
DDPM	3.01	4.69	0.600	0.682	17.21	49.03	0.553	0.738
DDPM-E	2.95	4.66	0.589	0.695	15.12	48.38	0.509	0.703
ANT-UW	2.93	4.45	0.596	0.687	14.96	47.02	0.598	0.794
B-TTDM$^{\alpha}$	2.98	4.53	0.600	0.692	15.00	46.98	0.602	0.768
B-TTDM	**2.89**	**4.22**	**0.602**	**0.695**	**13.81**	**45.25**	**0.612**	**0.819**

all these enhanced methods exhibit some degree of improvement in generation quality compare to DDPM. However, the B-TTDM method demonstrates the most significant enhancement in all evaluation metrics, underscoring its effectiveness.

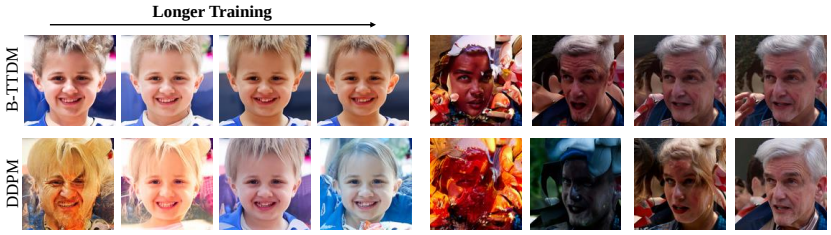


Fig. 5: Qualitative comparison. The generation results from DDPM and B-TTDM on FFHQ dataset. Images in each column are sampled from 50K, 100K, 300K and 500K training iterations. B-TTDM generates significantly better quality than DDPM with the same number of training iterations.

4.5 Qualitative Comparison

Generation results. We present a comparison of the generation results between various DDPMs and B-TTDMs in Figure 5. As depicted, images generated by B-TTDM exhibit markedly superior detail compared to those generated by DDPM. Furthermore, B-TTDMs manage to produce clear images by the 100K training iterations, whereas the quality of DDPM generated images at this stage remains relatively poor. Furthermore, we find that after 500K training iterations, B-TTDM still have better generation quality than DDPM in details.

Reconstruct results. Building upon previous analyses, which suggest that the distribution exhibits greater variability in the initial stages of diffusion, we compare the generation capabilities of B-TTDM and DDPM for this significant difference. To achieve this, we input identical noise-containing samples to models trained for the same iterations and reconstructed the outputs. The results in

Fig. 6: Qualitative comparison. Reconstruction results based on the same noisy inputs on FFHQ dataset.

Figure 6 indicate that the reconstructions generated by B-TTDM closely resemble the original samples, whereas those generated by DDPM exhibit more differences.

Unconditional generation results. Figure 7 provides a qualitative comparison of the unconditional image generation capabilities of B-TTDM and DDPM across the different datasets. The Figure 7 reveals that samples generated by DDPM often exhibit noticeable color biases, whereas samples generated by B-TTDM are free from such biases and show a marked improvement in quality compared to DDPM. This underscores the effectiveness of B-TTDM. More unconditional generation results can be found in the Appendix Section 4.

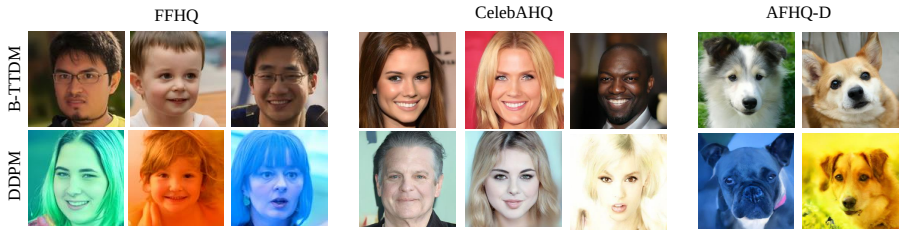


Fig. 7: Qualitative comparison. Unconditional generation results.

5 Conclusion

In this paper, through both experimental exploration and theoretical analysis, we conclude that the distribution varies non-uniformly in forward diffusion. Moreover, the most drastic variations occur in the initial stages of the forward diffusion process. These insights suggest that the uniform timestep sampling is sub-optimal for training diffusion models. Building on these findings, we design the B-TTDM method, which aligns timestep sampling with these inherent properties. Our extensive experiments demonstrate that B-TTDM significantly enhances the quality of model generation while also accelerating the training process.

Acknowledgments

This work was supported in part by NSFC under Grant 61927809 and in part by STCSM under Grant 22DZ2229005.

References

1. Anderson, B.D.: Reverse-time diffusion equation models. Stochastic Processes and their Applications **12**(3), 313–326 (1982)
2. 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: CVPR. pp. 22669–22679. IEEE (2023)
3. Block, A., Mroueh, Y., Rakhlin, A., Ross, J.: Fast mixing of multi-scale langevin dynamics under the manifold hypothesis. CoRR **abs/2006.11166** (2020)
4. Cai, R., Song, Z., Guan, D., Chen, Z., Luo, X., Yi, C., Kot, A.: Benchlmm: Benchmarking cross-style visual capability of large multimodal models. ECCV (2024)
5. Chen, Z., Li, B., Wu, S., Jiang, K., Ding, S., Zhang, W.: Content-based unrestricted adversarial attack. NeurIPS **36** (2024)
6. Chen, Z., Li, B., Xu, J., Wu, S., Ding, S., Zhang, W.: Towards practical certifiable patch defense with vision transformer. In: CVPR. pp. 15148–15158 (2022)
7. Choi, J., Lee, J., Shin, C., Kim, S., Kim, H., Yoon, S.: Perception prioritized training of diffusion models. In: CVPR. pp. 11462–11471. IEEE (2022)
8. Choi, Y., Uh, Y., Yoo, J., Ha, J.: Stargan v2: Diverse image synthesis for multiple domains. In: CVPR. pp. 8185–8194. Computer Vision Foundation / IEEE (2020)
9. Dhariwal, P., Nichol, A.Q.: Diffusion models beat gans on image synthesis. In: NeurIPS. pp. 8780–8794 (2021)
10. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real NVP. In: ICLR. OpenReview.net (2017)
11. Fan, K., Tang, J., Cao, W., Yi, R., Li, M., Gong, J., Zhang, J., Wang, Y., Wang, C., Ma, L.: Freemotion: A unified framework for number-free text-to-motion synthesis. ECCV (2024)
12. Fang, B., Li, B., Wu, S., Yi, R., Ding, S., Ma, L.: Re-thinking data availablity attacks against deep neural networks. CVPR (2024)
13. Fang, B., Li, B., Wu, S., Zheng, T., Ding, S., Yi, R., Ma, L.: Towards generalizable data protection with transferable unlearnable examples. CoRR **abs/2305.11191** (2023)
14. Ge, X., Liu, X., Yu, Z., Shi, J., Qi, C., Li, J., Kälviäinen., H.: Difffas: Face anti-spoofing via generative diffusion models. In: ECCV (2024)
15. Geng, C., Han, T., Jiang, P.T., Zhang, H., Chen, J., Hauberg, S., Li, B.: Improving adversarial energy-based model via diffusion process. ICML (2024)
16. Geng, C., Wang, J., Gao, Z., Frellsen, J., Hauberg, S.: Bounds all around: training energy-based models with bidirectional bounds. NeurIPS (2021)
17. Go, H., Lee, Y., Lee, S., Oh, S., Moon, H., Choi, S.: Addressing negative transfer in diffusion models. NeurIPS **36** (2024)
18. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: NIPS. pp. 2672–2680 (2014)
19. Hang, T., Gu, S., Li, C., Bao, J., Chen, D., Hu, H., Geng, X., Guo, B.: Efficient diffusion training via min-snr weighting strategy. In: ICCV. pp. 7407–7417. IEEE (2023)

20. 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. In: NeurIPS. pp. 6626–6637 (2017)
21. Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A.A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., Salimans, T.: Imagen video: High definition video generation with diffusion models. CoRR **abs/2210.02303** (2022)
22. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
23. Hyvärinen, A.: Estimation of non-normalized statistical models by score matching. J. Mach. Learn. Res. **6**, 695–709 (2005)
24. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. CoRR **abs/1710.10196** (2017)
25. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. In: NeurIPS (2022)
26. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. PAMI **43**(12), 4217–4228 (2021)
27. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. In: NeurIPS. pp. 10236–10245 (2018)
28. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
29. Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T.: Improved precision and recall metric for assessing generative models. In: NeurIPS. pp. 3929–3938 (2019)
30. Lee, H., Lu, J., Tan, Y.: Convergence for score-based generative modeling with polynomial complexity. In: NeurIPS (2022)
31. Li, X., Thickstun, J., Gulrajani, I., Liang, P., Hashimoto, T.B.: Diffusion-lm improves controllable text generation. In: NeurIPS (2022)
32. Liu, Y., Chen, Y., Dai, W., Gou, M., Huang, C.T., Xiong, H.: Source-free domain adaptation with domain generalized pretraining for face anti-spoofing. PAMI (2024)
33. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV. pp. 3730–3738. IEEE Computer Society (2015)
34. 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. In: NeurIPS (2022)
35. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations. In: ICLR (2022)
36. Nash, C., Menick, J., Dieleman, S., Battaglia, P.W.: Generating images with sparse representations. In: ICML. vol. 139, pp. 7958–7968. PMLR (2021)
37. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: ICML. vol. 139, pp. 8162–8171. PMLR (2021)
38. Ning, M., Li, M., Su, J., Salah, A.A., Ertugrul, I.Ö.: Elucidating the exposure bias in diffusion models. ICLR (2024)
39. Ning, M., Sangineto, E., Porrello, A., Calderara, S., Cucchiara, R.: Input perturbation reduces exposure bias in diffusion models. In: ICML (2023)
40. Parmar, G., Singh, K.K., Zhang, R., Li, Y., Lu, J., Zhu, J.: Zero-shot image-to-image translation. In: SIGGRAPH (Conference Paper Track). pp. 11:1–11:11. ACM (2023)
41. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. In: ICLR (2023)
42. Risken, H., Risken, H.: Fokker-planck equation. Springer (1996)
43. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10674–10685. IEEE (2022)

44. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: ICLR. OpenReview.net (2021)
45. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. In: NeurIPS. pp. 11895–11907 (2019)
46. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: ICLR. OpenReview.net (2021)
47. Tao, S., Wang, J.: Alleviation of gradient exploding in gans: Fake can be real. In: CVPR. pp. 1188–1197. Computer Vision Foundation / IEEE (2020)
48. Yang, Z., Feng, R., Zhang, H., Shen, Y., Zhu, K., Huang, L., Zhang, Y., Liu, Y., Zhao, D., Zhou, J., Cheng, F.: Eliminating lipschitz singularities in diffusion models. ICLR (2024)
49. Yu, H., Shen, L., Huang, J., Zhou, M., Li, H., Zhao, F.: Debias the training of diffusion models. CoRR **abs/2310.08442** (2023)
50. Zeng, W., Yan, Y., Zhu, Q., Chen, Z., Chu, P., Zhao, W., Yang, X.: Infusion: Preventing customized text-to-image diffusion from overfitting (2024)
51. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
52. Zhu, Z., Wei, Y., Wang, J., Gan, Z., Zhang, Z., Wang, L., Hua, G., Wang, L., Liu, Z., Hu, H.: Exploring discrete diffusion models for image captioning. CoRR **abs/2211.11694** (2022)