Improved Immiscible Diffusion: Accelerate Diffusion Training by Reducing Its Miscibility

Yiheng Li[1], Feng Liang[2], Dan Kondratyuk[3], Masayoshi Tomizuka[1], Kurt Keutzer[1], and Chenfeng Xu[2*]

[1] UC Berkeley, Berkeley CA 94720, USA
[2] UT Austin, Austin TX 78712, USA
[3] Rekursiv.ai, Mountain View CA, 94041
{yhli, tomizuka, keutzer}@berkeley.edu, {jeffliang, cxu}@utexas.edu, dan@rekursiv.ai

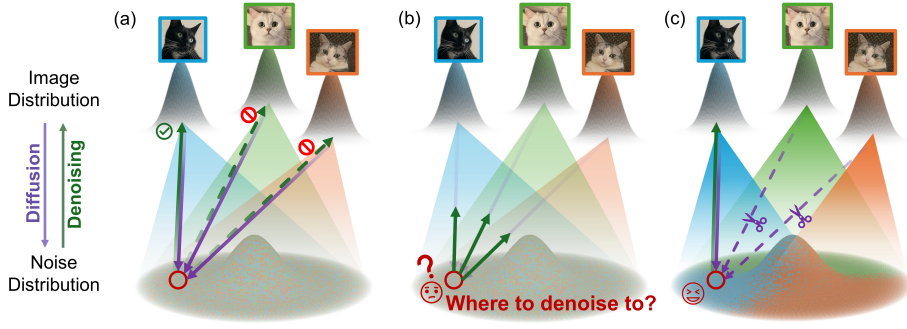

Fig. 1: Improved Immiscible Diffusion. (a) While vanilla diffusion trajectories (flows) are mixed (miscible), *each noise point is stably correlated to a specific generated image*, making many crossed diffusion trajectories irreversible. **(b)** The irreversible trajectories confuse the denoising process. **(c)** We extend immiscible diffusion to cut mixed (miscible) diffusion trajectories during diffusion training for accelerating it.

Abstract. The substantial training cost of diffusion models hinders their deployment. Immiscible Diffusion [20] showed that reducing the mixing of images' diffusion destination in the noise space via linear assignment can accelerate diffusion training. However, concerns regarding limited image diversity and exploding execution times under large batch sizes limit its feasibility for large-scale training. In this work, we start from thoroughly tackling these limitations: For the image diversity, we demonstrate the bijective nature of the denoising process of vanilla diffusion, underlying that being immiscible cannot hurt the diversity. Moving beyond the noise layer, we refine immiscible diffusion's concept to a broader miscibility reduction at any layer, which enables us to propose a new family of its implementations much more efficient to execute under high batch sizes, including K-nearest neighbor (KNN) noise selection and image scaling. Overall, the immiscible diffusion family achieves up to 4× faster training across diverse models and tasks, including unconditional/conditional generation, image editing, and robotics planning.

* Corresponding author.

Extensive analysis shows step-by-step on how immiscibility eases denoising and improves efficiency. Besides, our analysis of immiscibility offers a novel perspective on how optimal transport (OT) enhances diffusion training. By identifying trajectory miscibility as a fundamental bottleneck, we believe this work establishes a potentially new direction for future research in high-efficiency diffusion training.

Keywords: Diffusion-based Models · Manifold Learning

1 Introduction

Training diffusion-based models is costly and time-consuming, and such training costs are continuously growing. For example, Stable Diffusion V1.1 [28] was trained for more than 24 days on 256 GPUs, while its V2's training time grew to 32 days with the same GPUs. Consequently, the training efficiency problem attracts intensive explorations from diverse aspects [23,27,28,34].

Recent works [27,34] found that applying batch-wise optimal transport (OT) to pair the images and the noises can boost the training efficiency of flow matching [23] by shortening flow trajectories and lowering variance in denoising. However, immiscible diffusion [20] shows that such approximate OT reduces the image-noise distance by only $\approx 2\%$, and multisample flow matching [27] shows that the standard deviation (std) of the denoising function reduces only $\approx 4\%$. On the other hand, immiscible diffusion [20] assigns each image to a relatively separated noise area to boost the training efficiency, attributing such performance improvements to better denoising performances in noisy layers. However, its implementation is limited to image-noise linear assignment, which not only reduces image-noise distance for just $\approx 2\%$ as well, but also has a time complexity of $O(n^3)$, questioning its efficiency optimality. Additionally, the concept of immiscible diffusion itself raises questions on the diversity of its generated images, as it stops images from being diffused to some far-away noise areas.

In this work, we first refine immiscible diffusion to a broader concept: a diffusion process with reduced mixtures of diffusion trajectories (flows) from different images. We find that generated images are stably correlated to their noise origins, which makes the mixing of diffusion trajectories unnecessary. This resolves the diversity concerns of immiscible diffusion. We then explore immiscible diffusion's benefits step-by-step, demonstrating how the miscibility reduction eventually leads to boosts on diffusion models. With the refined definition, we accordingly offer improved immiscible diffusion implementations including KNN and image scaling, which satisfy the improved concept but either do not qualify for an assignment or even involve no image-noise pairings.

Extensive experiments are performed to examine the performance of the new immiscible diffusion family, where we observe consistent training efficiency boosts on various baseline models including consistency models [32], flow matching [23] and DDIM [30]. Further experiments see similar boosts across diverse image generation tasks including unconditional and conditional ones, different stages (training and fine-tuning), and various image datasets such as CIFAR-10 [17], ImageNet [6] and MSCOCO [22]. Moreover, the immiscible diffusion

family is subsequently extended to tasks outside image generation such as image in-painting and out-painting and robotics planning [3], where the coherent performance enhancements support its robustness. Thorough discussion is provided to distinct immiscible diffusion to other training efficiency improvement methods, and to compare between the implementations. Our contributions are summarized as follows,

- We extend immiscible diffusion [20] to an implementation-agnostic concept, characterized by reduced mixture (miscibility) of diffusion trajectories from different images. Our experiments show that generated images are stably correlated with their noise origins, relieving concerns on immiscible diffusion's generation diversity. Systematic feature analysis clarifies how immiscible diffusion enhances diffusion training.
- Based on the improved immiscible diffusion concept, we design a family of implementations, including the KNN, image scaling. These methods are more efficient to linear assignment, and feature analysis shows that they both effectively reduce the miscibility of diffusion trajectories.
- The immiscible diffusion family is applied to various image generation tasks, including unconditional and conditional image generation training and fine-tuning, and across diverse datasets and baseline methods. Unanimous effectiveness of immiscible diffusion is observed, and additional experiments extend its benefits to applications such as image editing and robotics planning. The miscibility problem we established points out a potential direction for future research in efficient diffusion training.

2 Related Works

2.1 Training Efficiency of Diffusion-based Models

As training efficiency limits the large-scale deployment of diffusion-based models, actions are taken to trigger diverse abundant parts inside them. For the feature dimensions, Stable Diffusion [28] shrinks image sizes with VAE [15], downsizing the image dimension by 16-64 times. [36] introduces a novel patch strategy to control the ease of diffusion training, achieving both training and data efficiency. For the noise space, [10] modifies the noise used in diffusion models to boost the performance. For training dynamics, [12] discovers that the magnitude of activation and the magnitude of neural weights can impact the diffusion's training speed. [31] demonstrates the importance of training dynamics on training efficiency, including the usage of exponential moving average (EMA), training loss and the noise schedule. [35, 41] notices that denoising some noisy steps helps little, so focusing more on other steps can improve training efficiency. However, these works hardly alter the diffusion trajectory, therefore cannot solve the miscibility problem immiscible diffusion aims to tackle.

2.2 Diffusion Paths and Training Efficiency

Recent works redesign diffusion trajectories for faster training. Notably, based on the rectified flow [24], [25] improves training efficiency by making diffusion

trajectories deterministic and straight. [23] further straightens the diffusion trajectory by making the image-noise mixture linear. Probing more deeply, several studies began exploring alternatives to mapping each image to the full noise space. [19] pointed out the curvature problem in the ODE paths caused by the collapse of the denoising trajectories pointing to the average direction. However, they replace the noise sampling with a VAE encoder-style structure to eliminate such curvatures, which destroys the strict Gaussian of the noise. In order to make diffusion trajectory paths even straighter and shorter, [27,34] applies batch-wise OT to assign noise to closer images before performing flow matching, followed up by a few further improvements on them [2,7,11,16,37]. However, [20] shows that the OT only reduces average image-noise distance by $\approx 2\%$, and [27] claims that the standard deviation of denoising reduces only $\approx 4\%$ after applying OT, which doubts the contributions of OT in the training efficiency boosts. [38] trains diffusion models to denoise towards the weighted average of a large image batch, to avoid large denoising variance. However, the weighted average calculations introduced place additional costs to the training. More recently, immiscible diffusion [20] assigns noise to some preferred noise areas, aiming to avoid the difficulty of denoising in noisy layers. However, its implementation of batch-wise linear assignment is still similar to OT, so its theory needs to be further justified versus OT.

2.3 Image-Noise Correlation in Diffusion Models

While most diffusion-based models diffuse each image to the whole noise space, some diffusion inversion works indicate that learned diffusion model's inversion does not follow this. [33] indicates that the diffusion and denoising is not symmetric, and [39] demonstrates that nearly the same image would be generated with the same noise but diverse diffusion models. These imply that generated images and their noise origins are somehow correlated. [1] further illustrates some similarity between the generated images and their noise origin, so as [26]. Quantitatively, [13,18,40] suggests that DDPM's inversion is an L_2 OT process. However, the correlation strength between generated images and their noise origin, *i.e.* how much perturbation can such correlation resist, was not thoroughly discussed. Our work proves a **stable** relation exists between the generated images and their noise origins, which supports the generation diversity of immiscible diffusion.

3 Preliminaries

For vanilla diffusion models, during diffusion training, we add random Gaussian noise to images,

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1-\alpha_t}\epsilon, \epsilon \in N(0, I) \tag{1}$$

where x_0 represents the image, ϵ is the added Gaussian noise, and α_t is the noise schedule for each diffusion timestep $t \in [0, T]$ controlling the mixing

weights of x_0 and ϵ. x_t is the noisy image we get, and we train the model to predict $\epsilon_\theta(x_t, t)$ with ϵ as supervision.

However, such a method can mix different image's diffusion paths at a same point in the image space. For example, we have two images $x_{0,1}$ and $x_{0,2}$. Since ϵ is randomly sampled, for some t's (especailly when $t \to T$) there exists ϵ_1 and ϵ_2, so that,

$$x_t = \sqrt{\alpha_t} x_{0,1} + \sqrt{1-\alpha_t}\epsilon_1 = \sqrt{\alpha_t} x_{0,2} + \sqrt{1-\alpha_t}\epsilon_2 \tag{2}$$

We call this *miscible diffusion*, where different x_0's diffusion path mix at a same x_t.

Such miscible diffusion can cause difficulties in denoising. For a specific point x_t, the denoising function $\epsilon_\theta(x_t, t)$ can only provide one single prediction. However, it is trained to predict both ϵ_1 and ϵ_2, so it ends up predicting

$$\epsilon_\theta(x_t, t) \to \frac{1}{2}\Sigma_{i=1}^{2}\epsilon_i \tag{3}$$

which cannot be used to denoise to either x_0's. Most significantly, when $t \to T$, the noise schedule $\alpha_t \to 0$, so $x_t \to \epsilon$. In this case, any specific point x_t can be reached by the diffusion paths originated from all images x_0's. Therefore, the denoising function will converge to

$$\begin{aligned} \epsilon_\theta(x_t, t) &\to \frac{1}{N}\Sigma_{i=1}^{N}\epsilon_i \\ &= \frac{1}{N\sqrt{1-\alpha_t}}\Sigma_{i=1}^{N}(x_t - \sqrt{\alpha_t} x_{0,i}) \\ &= \frac{1}{\sqrt{1-\alpha_t}}(x_t - \sqrt{\alpha_t}\bar{x_0}) \end{aligned} \tag{4}$$

where N is the size of the image dataset. This training goal converges to constant regardless of x_0, as $x_0 \to const$ when $N \to +\infty$, which does not help denoise to any specific image.

Therefore, based on [20]'s theory, we extend immiscible diffusion, which now represents a set of methods to avoid miscible diffusion, i.e. avoid the mixing of images' diffusion paths discussed above, during diffusion training. A straightforward method to achieve so is assign x_t to closest x_0 in terms of pixel values, by minimizing the image-noise $L2$ distance $\|\epsilon - x_0\|_2$. In this work, we introduce two example implementations based on this in Section 4.3. Note that while these methods can achieve immiscible diffusion, they are not the only methods. Another simple yet effective demonstration for immiscible diffusion is the image scaling method detailed in Section 4.3.

4 Improved Immiscible Diffusion

[20] borrows immiscible diffusion, a physics concept describing solutes which cannot mix homogeneously, to assign images with near noise points via batch-wise linear assignment. While this accelerates diffusion training, the linear assignment is weak with only $\approx 2\%$ image-noise distance drop afterwards, and its computational complexity is $O(n^3)$, which scales up quickly with the batch size.

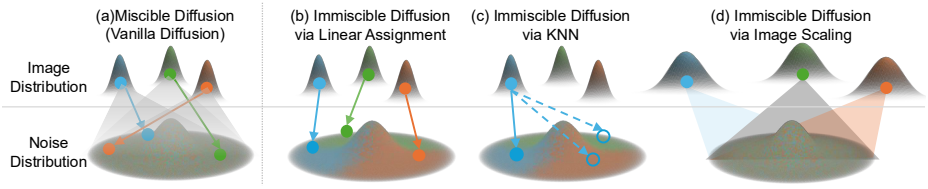

Fig. 2: Implementations of Immiscible Diffusion. (a) Miscible Diffusion pairs the batch of images and noises randomly before adding noise to images. **(b)(c)(d) Immiscible Diffusion** tries to reduce the miscibility of diffusion by (b) L_2 linear assignment between the batch of images and noises and (c) sampling k noises and pick the nearest one (KNN) to use. (d) scaling images by multiplying their pixel values with a constant > 1, which reduces overlaps between diffuse-able areas of different images.

Therefore, to improve immiscible diffusion, we need to break the limit of using linear assignments. However, involving image-noise correlation generally triggers concerns on the diversity of generated images, as images are not diffused image to uncorrelated noises. Consequently, we need to take a deeper look into the generation diversity, to justify building correlations between images and noises.

4.1 Image-noise Correlation in Denoising

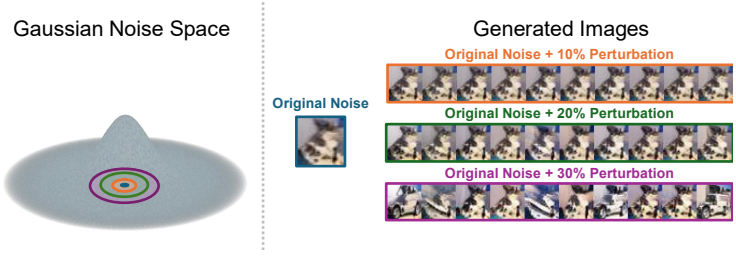

Fig. 3: Stable correlation between generated images and its noise origins. Here perturbation means another Gaussian noise added to the fixed original Gaussian noise. Note that even with 20% perturbation, images changes are nearly unnoticeable. Only with 30% perturbation, a image object change happens. These demonstrate stable correlation from a noise area to a specific generated image.

To evaluate the image-noise correlation during denoising, we generate images on a trained vanilla DDIM [30] with a specific noise point and its ambient noise area. We first sample a specific noise point N_{orig}. Then we add diverse perturbations N_{pert} onto it, respectively. Specifically, we sample 10 independent Gaussian N_{pert}'s. The perturbed total noise can be expressed as,

$$N_{tot} = N_{orig} + W \cdot N_{pert} \tag{5}$$

Where W is the weight of the perturbation. We present images generated by the original and perturbed total noise in Figure 3. For example, we see N_{orig} generate a cat image. Surprisingly, with $W = 10\%$ on all 10 perturbation, we see that the generated images are still all cats without noticeable differences.

Further, we see that $W = 20\%$ of perturbation only results in very slight image differences (like the background difference in the $1st$ and $2nd$ images from left), while no noticeable modal changes are observed. Only with 30% weight of perturbation, we observe image object changes in a few generated images. However, we can clearly find pixel-level similarities between these images despite modal differences. These results suggest that generated images are stably correlated with the sampled noise. As a result, while vanilla (miscible) diffusion models diffuse each image equally to the whole noise area, hoping to see that every image can be generated from a noise point or a small noise area, this goal might not be fully achieved, significantly weakening the motivation of miscible diffusion in generation diversity, so as the concerns on immiscible diffusion's generation diversity from a specific noise point or area. In Section 5.3, we further quantitatively show that immiscible diffusion does not negatively impact the diversity of generated images.

4.2 Step-by-step Feature-level Benefits Analysis of Immiscible Diffusion

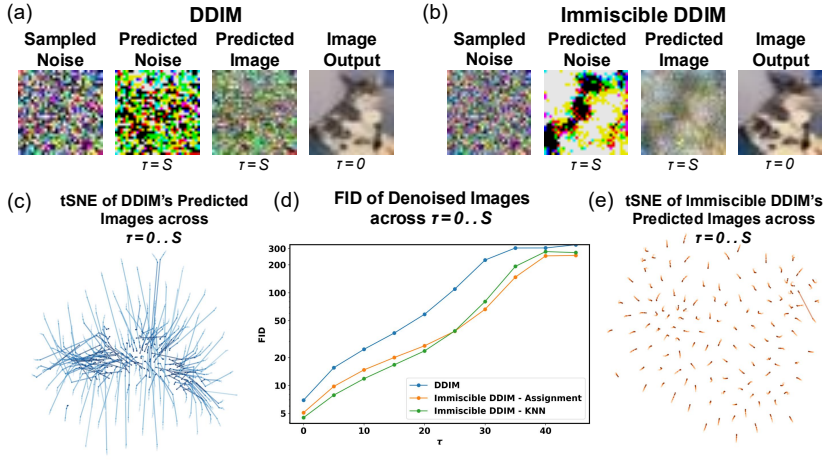

Fig. 4: Feature analysis of vanilla (miscible) and immiscible DDIM. Referring to [30], $\tau = S$ represents the layer denoising from the pure noise. We show that immiscible diffusion activates the noisiest ($\tau \to S$) layers' denoising functions by clarifying their denoising goals, as shown in the $tSNE$ of denoised images across τ's. Such activation results in FID improvements on the denoised images from large τ's, which leads to better performance and faster convergence of diffusion models.

We now dive deeper to understand more clearly on how immiscible diffusion accelerates diffusion training, in order to help to accordingly design more implementations of it. Notations used mostly follow [30], and are listed in Table 1.

[20] suggests and mathematically proves that miscible diffusion causes difficulty in denoising at noisy layers (larger τ). As shown in Figure 4 (a), the noisiest denoising layer, referred as $\tau = S$ in DDIM [30], does not effectively predict the noise added. To confirm the ubiquity of such denoising difficulty, in Figure 4 (c)

Table 1: Notations used in diffusion and denoising.

Symbol	Definition
T	Total *diffusion* steps, $T = 1000$.
S	Total *denoising* steps, $S \in [20, 50, \dots]$.
t	*Diffusion* step, $t \in [0, T]$, $t = 0$ is image.
τ	*Denoising* step, $\tau \in [0, S]$, $\tau = 0$ is image.

we show the $tSNE$ of the predicted images from each denoising layer $\tau = 0..S$. We collect 128 noise points, generating images with them, and logging the predicted images ($x_{pred,0}$'s) from each denoising layer, which are computed with the layer's feature x_τ and the predicted noise $n_{pred,\tau}$ according to noise schedule α_τ:

$$x_{pred,0} = \frac{1}{\sqrt{\alpha_\tau}} x_\tau - \frac{\sqrt{1-\alpha_\tau}}{\sqrt{\alpha_\tau}} n_{pred,\tau} \tag{6}$$

Points on the same line represent $x_{pred,0}$'s from the same initial noise on different τ's. Apparently, though different lines start from different noise points, they are chaotically tangled, suggesting that the same denoising goals are frequently shared between different denoising paths, which implies that the denoising difficulty commonly happens in the denoising. That is not surprising, as the vanilla DDIM takes miscible diffusion. To quantitatively express the miscibility, we stat the average L_2 distance between noise clusters (i.e. $10,000$ noise points) assigned to each image. The distance for vanilla DDIM is only 0.92 ± 0.06, proving that the diffusion process is truly miscible.

However, immiscible diffusion substantially overturns these difficulties. With the linear assignment implementation, we find that the average distance between noise clusters assigned to each image is increased significantly to 4.11 ± 0.37, suggesting that the noise clusters are more separate than the miscible diffusion, and thus is more immiscible by the definition. Figure 4 (b) illustrates that under immiscible diffusion, even the noisiest layer $\tau = S$ can effectively predict the noise and can denoise the image, and Figure 4 (e) shows that with immiscible diffusion, the $tSNE$ figure shows much less intersections, implying that each noise point has its own stable denoising goal, which corresponds to the goal of easing denoising. As a result, as shown in Figure 4 (d), the FID of the denoised images from noisy layers exhibit significant improvement, which finally leads to the performance and training efficiency boost of immiscible diffusion.

The analysis above clearly figures out that the reduced miscibility helps to clarify the denoising goal, easing the denoising and helping it to be effective, and finally improve the performance of the miscible layers, and the final outputs. These step-by-step benefits of immiscible diffusion help us to extract the essence of it, and to design additional implementations in the following section.

4.3 Improved Immiscible Diffusion

As the benefits of immiscible diffusion can be traced back to the reduction of miscibility in diffusion, we naturally propose that the concept of immiscible diffusion should also reflect only the reduction of miscibility in diffusion, without unnecessary bounds to image-noise pairing. Under this improved concept, we argue that such assignment is only *one* way to achieve immiscible diffusion. In

this work, we additionally propose two new immiscible diffusion implementations, which do not need linear assignment, or even image-noise correlations. Nevertheless, both methods exhibit excellent immiscibility, and therefore boost the training efficiency significantly.

Batch-wise Linear Assignment Batch-wise linear assignment in [20] still qualifies immiscible diffusion. As shown in Figure 2 (b), this method performs a linear assignment [5] between the batch of images and noise points sampled. Such an assignment preferably assigns noise to nearby images while keeping the general distribution of all noises Gaussian. However, this trades off the running speed for each step when the batch size scales up, as the linear assignment is an $O(n^3)$ operation.

KNN Noise Selection To avoid the scaling-up of execution time with larger batch sizes, we provide the second implementation of immiscible diffusion, the KNN method, where we sample k Gaussian noise points for each image and pick the one L_2-closest to the image, as illustrated in Figure 2 (c). This method is very efficient - its execution time is only $0.2ms$ for a batch size of 256 and would not scale up quickly when larger batches are used, as it is an $O(n)$ operation. As shown in Table 2, the KNN is much faster than the linear assignment, demonstrating its efficiency in achieving immiscibility. The algorithm is shown in Algorithm 1.

Table 2: Execution Time (ms) of Immiscible Diffusion on a single A5000 GPU.

Batch Size	128	256	512	1024
Linear Assignment [20]	5.4	6.7	8.8	22.8
KNN	0.2	0.2	0.3	0.7
t_{assign}/t_{knn}	27.0x	33.5x	29.3x	32.6x

Algorithm 1 Immiscible Diffusion Implementation - KNN Sampling

1: **Input:** Image x, k random noises $\{n_1, \ldots, n_k\}$, noise schedule α_t
2: $n \leftarrow \arg\min_{n_j \in \{n_1, n_2, \ldots, n_k\}} \text{dist}(x, n_j)$
3: $x_t \leftarrow \sqrt{\alpha_t} x + \sqrt{1 - \alpha_t} \cdot n$
4: **Output:** Diffused image batch x_t

While the distribution of overall noise points used in training KNN-implemented immiscible diffusion is not guaranteed to be Gaussian, as some sampled noise points are selectively dropped, we argue that such discrepancy is negligible. To prove this, we sample $50k$ Gaussian noise points in the size of CIFAR-10 [17] images, i.e. $3 \times 32 \times 32$. Comparing them with the Gaussian distribution results in a KL divergence of 48.25. We then collect another $50k$ noise points which are *selected* in the KNN Immiscible Diffusion ($k = 8$), finding that their KL divergence to Gaussian is 48.60, which is only very slightly higher than noise points from a strict Gaussian distribution. Therefore, our KNN implementation doesn't significantly alter the Gaussian noise distribution.

Image Scaling Though image-noise correlation is effective in building immiscible diffusion, there are also ways to reduce miscibility without it. A typical

example is image scaling, *i.e.* to multiply images' all pixel values by a constant factor greater than 1, like 2 or 4. Such action has no influence on the noise space. However, the L_2 distance between each image is farther after the scaling, making the centers of diffused areas ($t < t_{max}$) of different images farther away from each other. Considering the noise amplitude at each diffusion step t is kept consistent, the diffused areas for different images at every step $t < t_{max}$ are naturally having less intersections, which constitutes immiscible diffusion. We will defer the immiscibility and performance experiments of this method to Section 5.6.

5 Effectiveness of Immiscible Diffusion

With the improved immiscible diffusion concept and the new and existing implementations, we perform a large series of experiments to systematically examine the benefits and the generalization ability of immiscible diffusion across various image generation tasks, models, and datasets, as well as extended tasks including image in-painting, out-painting, and robotics planning.

5.1 Experiment Setups

We implement Immiscible Diffusion on diverse diffusion-based methods, including Consistency Models [32], DDIM [30], Stable Diffusion [28], and Flow Matching [23]. We train these implementations on a variety of popular datasets, including CIFAR-10 [17], ImageNet-1k [6] and MS-COCO [22]. The training hyperparameters are discussed in Section 8.1 and Table 9 of the Supplementary Material.

5.2 Unconditional Image Generation Training

We compare the unconditional image generation training steps necessary to reach the best FID of the vanilla diffusion-based models in Figure 5. Astonishing and consistent training efficiency enhancement is observed across all diffusion-based models, despite their significant differences in diffusion trajectories (flows), denoising solvers and sampling step picking strategies. Across all experiments, we generally observe that the baseline diffusion-based models need a maximum of **2.5 to 4.5** times of training steps to achieve the same performance of their immiscible diffusion counterparts. Besides, we offer the converged FIDs in Tab. 3, confirming that immiscible models achieve better performance in most cases. These results strongly support the robust ability of immiscible diffusion in improving the training efficiency of diverse diffusion-based models. Furthermore, as shown in Tab. 4, it is parallel to many popular diffusion training acceleration methods, as the improved immiscible diffusion modifies only the goal diffusion path for training.

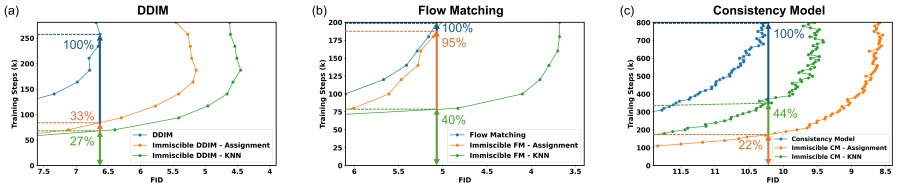

Fig. 5: Immiscible diffusion boosts training efficiency. We show the training steps required to reach the best FID for vanilla models across **three** diverse diffusion-based architectures. Results consistently show that immiscible diffusion trains significantly faster.

Table 3: Final Performances of Immiscible Diffusion.

Model	DDIM	Flow Matching	Consistency Models
Vanilla	6.63	4.92	10.22
Immiscible (KNN)	4.45 (-2.18)	3.56 (-1.36)	9.16 (-1.06)
Immiscible (Assignment)	5.14 (-1.49)	5.02 (+0.10)	8.54 (-1.68)

5.3 Conditional Image Generation Training and Fine-tuning

Class-conditional Image Generation from Scratch. We conduct this on Stable Diffusion [28] and ImageNet-1k [6], whose result is shown in Figure 6 (a). Similar to the unconditional generations, immiscible diffusion exhibits much faster training. However, since immiscible diffusion does not diffuse each image equally to the noise space, questions on whether immiscible diffusion models can still follow the prompts as good as vanilla models can be raised. Simply put, the answer is *Yes, they can*. In Section 4.1, we have shown that vanilla diffusion-based models yet have strong image-noise correlations, so the diversity of generated images should not be influenced solely by the miscibility of the diffusion process. We further confirm this by evaluating the diversity of the generated images with CLIPScore [8], which shows that both the immiscible and the baseline models generate images with CLIPScores of 28.55, with a standard deviation of 0.01 and 0.02 respectively, indicating that Immiscible Diffusion does not hurt the image-prompt correspondence in complicated ImageNet dataset.

Class-conditional Image Generation Fine-tuning. Immiscible diffusion can also benefit the fine-tuning. We confirm this intuition with a class-conditional image generation fine-tuning experiment, which use ImageNet to fine-tunes Stable Diffusion which is pre-trained on LAION [29] by [28]. Results in Figure 6 show significant and consistent performance enhancements of immiscible diffusion compared to vanilla baselines. Note that our class-conditional generation uses class names as prompts instead of the class number, so class-conditional fine-tuning and conditional pre-training don't conflict in the form of conditions.

Table 4: Superpositioning Immiscible Diffusion (ID) to Other Accelerative Methods.

	FM Base	ID	FM + [14]	FM + [14] + ID	FM+ [4]	FM + [4] + ID
FID 100k	6.10	4.28	6.43	4.75	5.58	4.26
FID 200k	5.07	3.68	5.38	3.93	5.41	3.86

Table 5: FID evaluations of vanilla and Immiscible Diffusion in Image-to-image Tasks.

Models	Vanilla SD	Immiscible SD KNN
In-painting	18.35	**17.32**
Out-painting	29.34	**27.57**

Free-prompt Conditional Image Generation from Scratch. To test immiscible diffusion's effects on free prompts other than limited classes of prompts, we train Stable Diffusion [28] from scratch on MSCOCO [22]. Results also suggest a performance boost with immiscible diffusion.

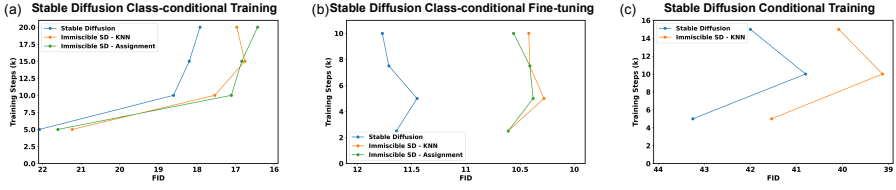

Fig. 6: Immiscible diffusion stays effective across diverse image generation tasks. We demonstrate this with figures showing the training steps *v.s.* FID of **(a)(b)** Stable Diffusion class-conditional training and fine-tuning with ImageNet, and **(c)** conditional training with MS-COCO.

5.4 Image Editing

We perform in-painting and out-painting, two classic image-to-image tasks, with vanilla and immiscible diffusion respectively. Both models are Stable Diffusion (SD) models trained on ImageNet dataset. For the in-painting, we load images from ImageNet, replacing its center with Gaussian noise, and we let the model to repair the missing parts of the images. The out-painting task is similar, but everything other than the center part is replaced with Gaussian noise. We compare the completed images with the ImageNet dataset to calculate the FID, as shown in Table 5. We see that in both tasks the immiscible models exhibit better completed images. We further qualitatively provide a few comparisons in Figure 9, where immiscible diffusion enjoys a better overall output which we ascribe to its better preserving of source information.

5.5 Robotics Planning

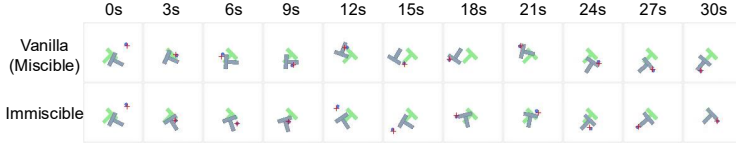

Fig. 7: Immiscible diffusion boosts the performance of diffusion policy. The robot (circle) needs to push the T-shaped object (gray) into the desired place (green).

Table 6: Immiscible diffusion in robotics planning.

Experiment	Vanilla (Miscible)	Immiscible KNN k=2
Ave. Coverage for Last 10 Ckpts (%)	79.56%	82.83%
Max Coverage (%)	85.71%	86.74%

To further demonstrate the generalization ability of immiscible diffusion, we apply it onto PushT, a task in diffusion policy [3] for robotics, which let the robot to push a T-shaped object to a desired place. Our experiments are performed on the simulated PushT task explained in Figure 7, and with the data provided in [3]. Each experiment is trained for $3,000$ epochs with 3 seeds, where the averages are taken as the results. We take the average area coverage by the T-shaped object onto the desired destination as the metric [3].

The experiment results are shown in Table 6. We see that area coverage is significantly improved with immiscible diffusion. To illustrate this improvement more directly, we show a typical improvement case in Figure 7. We observe a more accurate pushing process without errors for immiscible diffusion policy where the vanilla one fails due to errors during the pushing.

5.6 Immiscible Diffusion Beyond Image-noise Correlations

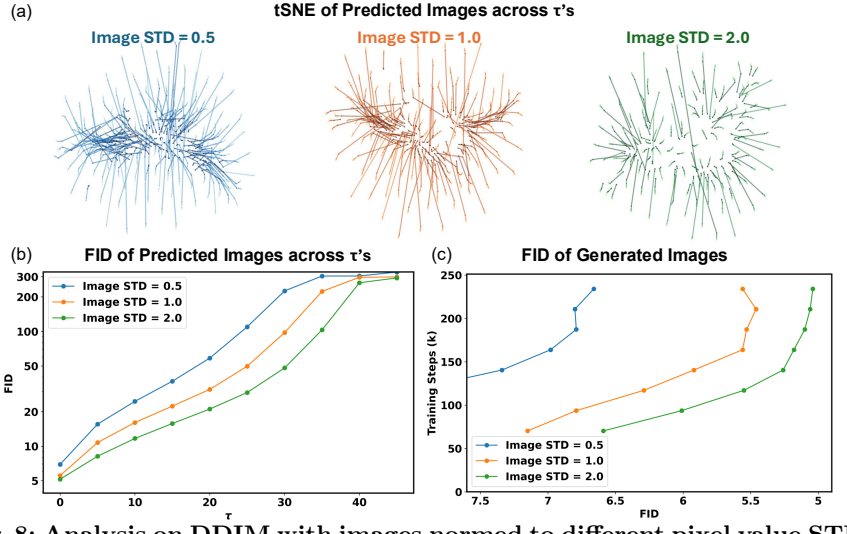

Fig. 8: Analysis on DDIM with images normed to different pixel value STDs.

As discussed in Section 4.3, image scaling can intuitively achieve immiscible diffusion without enforcing image-noise correlations. Indeed, in Figure 8, we compare scaling pixel values of normed images to different STD's before performing diffusion with DDIM, whose default is to norm image to a pixel value $STD = 0.5$. Indeed, Figure 8 (a) shows that larger STD helps to reduce the confusion in denoising caused by miscible diffusion. Consequently in Figure 8 (b), we observe that experiments with larger image STD enjoys lower FIDs of predicted images during denoising, and finally in Figure 8 performs better with faster training and

better convergence performance. These results are particularly interesting as the scaling can increase the SNR of the final diffusion step, which was shown harmful to the diffusion [21], further suggesting immiscible diffusion's strong boosts to the diffusion models. In addition, these results show that immiscible diffusion is not necessarily bounded with image-noise correlations.

6 Discussions

In this section, we discuss the distinction of immiscible diffusion to some seemingly similar methods, and compare our immiscible diffusion implementations to highlight their respective advantages.

What is the difference between immiscible diffusion, flow matching [23] and rectified flow [24]? We compare immiscible diffusion with flow matching [23] and vanilla DDIM [30] in Figure 11. Compared to DDIM, flow matching linearizes the diffusion trajectory. However, its diffusion trajectories (flows) can still arrive at the same noise point from different images, so flow matching is still miscible. Immiscible flow matching aims to reduce such miscibility to let each diffusion trajectory mix less, to ease the denoising. Rectified flow [24] can also make *denoising* paths distinct. However, in their method, each image is *diffused* to all the noise space, so their diffusion is still miscible.

What is the relation between immiscible diffusion and batch-wise OT? Batch-wise OT (linear assignment) can be *one* of the immiscible diffusion implementations. It preferably assign noise to closer images, so images would not be diffused to the whole noise space. Therefore, the mixing of diffusion trajectories from different images reduces, and the diffusion is more immiscible. Step-by-step feature analysis in Section 4.1 clearly demonstrate how the linear assignment makes diffusion immiscible and boost the diffusion's performance. At the same time, batch-wise OT only reduces the image-noise average distance by $\sim 2\%$ [20] and the denoising STD by $\sim 4\%$ [27]. Therefore, we attribute batch-wise OT's training accelerations mainly to realizations of immiscible diffusion.

Which implementation should I use, batch-wise Linear assignment or KNN? While there are numerous ways to achieve immiscible diffusion, here we compare the implementations we take for the reader's reference. Since the setting of image scaling depends heavily on the image normalization method taken by the baseline diffusion models, we focus on comparison between batch-wise linear assignment and KNN. While both methods help to enforce the image-noise correlation during diffusion, batch-wise linear assignment keeps the noise space strictly Gaussian, and achieve better immiscibility in the noise space. The average L_2 distance between noise clusters assigned to each image for batch-wise linear assignment is 4.11 ± 0.37, which is higher than that of KNN, which is 2.17 ± 0.35. Therefore, as shown in Figure 4 (d), it achieves better FID in noisy layers than KNN. However, batch-wise linear assignment suffers from computational cost of $O(n^3)$, which is higher than the KNN's $O(n)$. Furthermore, in Figure 10, we show the noise points assigned to the same image using batch-wise assignment and KNN. We observe that the KNN noise points distribute in a more continuous manner, which we posit to contribute to KNN's better performance in denoising un-noisy layers, as also indicated in Figure 4 (d).

What k should I use for KNN Immiscible Diffusion?

We offer detailed setting of $k's$ in Tab. 7, where we find the best k generally increases with the rise of diffusion dimensions. We find that while the best k for the same dataset's diffusion dimension is generally the same, with small fluctuations observed, datasets with larger data dimension needs larger k to provide stronger immiscibility. It is noteworthy that for each experiment, we only try $k = 1, 2, 4, 8, 16, 32, 64, 128, ...$ to save computational resources. Finer experiments with more k's will provide more precise results. Nevertheless, in Tab. 8, we find the improvements can persist in a wide spectrum of k.

Table 7: Best $k's$ in KNN Immiscible Diffusion Implementation.

Model	DDIM	Flow Matching	Consistency Model	Stable Diffusion
Dataset	CIFAR-10	CIFAR-10	CIFAR-10	ImageNet-1k
Diffusion Dimension	3,072	3,072	3,072	4,096
Best k	8	4	4	64
L_2 Dist. Δ (%)	-1.58%	-1.10%	-1.10%	-2.32%

Table 8: Impact of k on Best FID.

k	Baseline	2	4	8	16	32
Best FID	6.63	4.94	4.77	4.45	4.85	5.19

7 Conclusion

In this work, we systematically revisit immiscible diffusion, a physics-inspired method aiming to boost diffusion training efficiency. Our experiments firstly show that image-noise correlation introduced by immiscible diffusion empirically does not alter the diversity of generated images due to the intrinsic image-noise correlation in vanilla diffusion models. Detailed feature analysis shows how immiscible diffusion step-by-step enhances the denoising outputs. Based on these findings, we improve the immiscible diffusion concept, which does not require doing image-noise pairing nor even image-noise correlations. We offer a few new immiscible diffusion implementations, achieving training efficiency boosts up to $> 4\times$, across diverse baseline diffusion models and on various image generation tasks, image editing and robotics planning. Our method points out the diffusion trajectory miscibility problem, a generally existing problem in diffusion training dragging its efficiency, which could be a fundamental direction to explore towards high-efficiency diffusion training.

References

1. Asperti, A., Evangelista, D., Marro, S., Merizzi, F.: Image embedding for denoising generative models. Artificial Intelligence Review **56**(12), 14511–14533 (2023)

2. Chemseddine, J., Hagemann, P., Steidl, G., Wald, C.: Conditional wasserstein distances with applications in bayesian ot flow matching. Journal of Machine Learning Research **26**(141), 1–47 (2025)
3. Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., Song, S.: Diffusion policy: Visuomotor policy learning via action diffusion. In: Proceedings of Robotics: Science and Systems (RSS) (2023)
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)
5. Crouse, D.F.: On implementing 2d rectangular assignment algorithms. IEEE Transactions on Aerospace and Electronic Systems **52**(4), 1679–1696 (2016)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
7. Deng, W., Luo, W., Tan, Y., Biloš, M., Chen, Y., Nevmyvaka, Y., Chen, R.T.Q.: Variational schrödinger diffusion models. In: Forty-first International Conference on Machine Learning (2024)
8. Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., Choi, Y.: Clipscore: A reference-free evaluation metric for image captioning. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 7514–7528 (2021)
9. 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)
10. Huang, X., Salaun, C., Vasconcelos, C., Theobalt, C., Oztireli, C., Singh, G.: Blue noise for diffusion models. In: ACM SIGGRAPH 2024 Conference Papers. SIGGRAPH '24, Association for Computing Machinery, New York, NY, USA (2024)
11. Issenhuth, T., Dos Santos, L., Franceschi, J.Y., Rakotomamonjy, A.: Improving consistency models with generator-induced coupling. In: ICML 2024 Workshop on Structured Probabilistic Inference and Generative Modeling (2024)
12. Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., Laine, S.: Analyzing and improving the training dynamics of diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24174–24184 (2024)
13. Khrulkov, V., Ryzhakov, G., Chertkov, A., Oseledets, I.: Understanding DDPM latent codes through optimal transport. In: The Eleventh International Conference on Learning Representations (2023)
14. Kim, J.Y., Go, H., Kwon, S., Kim, H.G.: Denoising task difficulty-based curriculum for training diffusion models. arXiv preprint arXiv:2403.10348 (2024)
15. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2022)
16. Kornilov, N., Mokrov, P., Gasnikov, A., Korotin, A.: Optimal flow matching: Learning straight trajectories in just one step. Advances in Neural Information Processing Systems **37**, 104180–104204 (2024)
17. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009)
18. Kwon, D., Fan, Y., Lee, K.: Score-based generative modeling secretly minimizes the wasserstein distance. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022)
19. Lee, S., Kim, B., Ye, J.C.: Minimizing trajectory curvature of ODE-based generative models. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 18957–18973. PMLR (23–29 Jul 2023)

20. Li, Y., Jiang, H., Kodaira, A., TOMIZUKA, M., Keutzer, K., Xu, C.: Immiscible diffusion: Accelerating diffusion training with noise assignment. In: Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (eds.) Advances in Neural Information Processing Systems. vol. 37, pp. 90198–90225. Curran Associates, Inc. (2024)
21. Lin, S., Liu, B., Li, J., Yang, X.: Common diffusion noise schedules and sample steps are flawed. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 5404–5411 (January 2024)
22. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13. pp. 740–755. Springer (2014)
23. Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: The Eleventh International Conference on Learning Representations (2023)
24. Liu, X., Gong, C., et al.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: The Eleventh International Conference on Learning Representations (2023)
25. Liu, X., Zhang, X., Ma, J., Peng, J., qiang liu: Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In: The Twelfth International Conference on Learning Representations (2024)
26. Niedoba, M., Zwartsenberg, B., Murphy, K., Wood, F.: Towards a mechanistic explanation of diffusion model generalization. arXiv preprint arXiv:2411.19339 (2024)
27. Pooladian, A.A., Ben-Hamu, H., Domingo-Enrich, C., Amos, B., Lipman, Y., Chen, R.T.Q.: Multisample flow matching: Straightening flows with minibatch couplings. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 28100–28127. PMLR (23–29 Jul 2023)
28. 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 (CVPR). pp. 10684–10695 (June 2022)
29. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in neural information processing systems **35**, 25278–25294 (2022)
30. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (2021)
31. Song, Y., Dhariwal, P.: Improved techniques for training consistency models. In: The Twelfth International Conference on Learning Representations (2024)
32. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models. In: International Conference on Machine Learning. pp. 32211–32252. PMLR (2023)
33. Łukasz Staniszewski, Łukasz Kuciński, Deja, K.: There and back again: On the relation between noise and image inversions in diffusion models (2025)
34. Tong, A., FATRAS, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Wolf, G., Bengio, Y.: Improving and generalizing flow-based generative models with minibatch optimal transport. Transactions on Machine Learning Research (2024), expert Certification

35. Wang, K., Shi, M., Zhou, Y., Li, Z., Yuan, Z., Shang, Y., Peng, X., Zhang, H., You, Y.: A closer look at time steps is worthy of triple speed-up for diffusion model training. arXiv preprint arXiv:2405.17403 (2024)
36. Wang, Z., Jiang, Y., Zheng, H., Wang, P., He, P., Wang, Z., Chen, W., Zhou, M., et al.: Patch diffusion: Faster and more data-efficient training of diffusion models. Advances in neural information processing systems **36**, 72137–72154 (2023)
37. Xing, S., Cao, J., Huang, H., Zhang, X.Y., He, R.: Exploring straighter trajectories of flow matching with diffusion guidance. In: arXiv (2023)
38. Xu, Y., Tong, S., Jaakkola, T.S.: Stable target field for reduced variance score estimation in diffusion models. In: The Eleventh International Conference on Learning Representations (2023)
39. Zhang, H., Zhou, J., Lu, Y., Guo, M., Wang, P., Shen, L., Qu, Q.: The emergence of reproducibility and consistency in diffusion models. In: Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., Berkenkamp, F. (eds.) Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 235, pp. 60558–60590. PMLR (21–27 Jul 2024)
40. Zhang, P., Yin, H., Li, C., Xie, X.: Formulating discrete probability flow through optimal transport. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)
41. Zheng, T., Geng, C., Jiang, P.T., Wan, B., Zhang, H., Chen, J., Wang, J., Li, B.: Non-uniform timestep sampling: Towards faster diffusion model training. In: Proceedings of the 32nd ACM International Conference on Multimedia. p. 7036–7045. MM '24, Association for Computing Machinery, New York, NY, USA (2024)