Immiscible Diffusion: Accelerating Diffusion Training with Noise Assignment

Yiheng Li[1] **Heyang Jiang**[1,2*] **Akio Kodaira**[1]
Masayoshi Tomizuka[1] **Kurt Keutzer**[1] **Chenfeng Xu**[1†]
[1]University of California, Berkeley [2]Tsinghua University

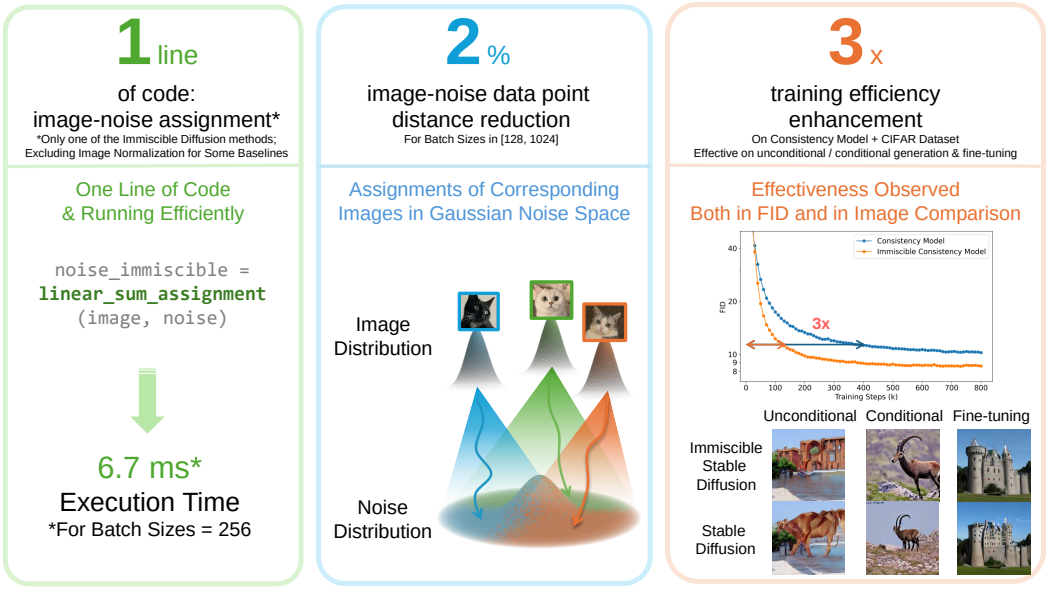

Figure 1: **Immiscible Diffusion** can use a single line of code to efficiently achieve immiscibility by re-assigning a batch of noise to images. This process results in only a 2% reduction in distance post-assignment, leading to up to 3x increased training efficiency on top of the Consistency Model for CIFAR Dataset. Additionally, Immiscible Diffusion significantly enhances the image quality of Stable Diffusion for both unconditional and conditional generation tasks, and for both training from scratch and fine-tuning training tasks, on ImageNet Dataset within the same number of training steps.

Abstract

In this paper, we point out that suboptimal noise-data mapping leads to slow training of diffusion models. During diffusion training, current methods diffuse each image across the entire noise space, resulting in a mixture of all images at every point in the noise layer. We emphasize that this random mixture of noise-data mapping complicates the optimization of the denoising function in diffusion models. Drawing inspiration from the immiscibility phenomenon in physics, we propose **Immiscible Diffusion**, a simple and effective method to improve the random mixture of noise-data mapping. In physics, miscibility can vary according to various intermolecular forces. Thus, immiscibility means that the mixing of molecular sources is distinguishable. Inspired by this concept, we propose an assignment-then-diffusion training strategy to achieve *Immiscible Diffusion*. As one example, prior to diffusing the image data into noise, we assign diffusion target

*The work of this paper was done when Heyang was in internship at UC Berkeley.

†Corresponding Author: xuchenfeng@berkeley.edu

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

noise for the image data by minimizing the total image-noise pair distance in a mini-batch. The assignment functions analogously to external forces to expel the diffuse-able areas of images, thus mitigating the inherent difficulties in diffusion training. Our approach is remarkably simple, requiring only **one line of code** to restrict the diffuse-able area for each image while preserving the Gaussian distribution of noise. In this way, each image is preferably projected to nearby noise. To address the high complexity of the assignment algorithm, we employ a quantized assignment strategy, which significantly reduces the computational overhead to a negligible level (*e.g.* 22.8ms for a large batch size of 1024 on an A6000). Experiments demonstrate that our method can achieve up to 3x faster training for unconditional Consistency Models on the CIFAR dataset, as well as for DDIM and Stable Diffusion on CelebA and ImageNet dataset, and in class-conditional training and fine-tuning. In addition, we conducted a thorough analysis that sheds light on how it improves diffusion training speed while improving fidelity. The code is available at https://yhli123.github.io/immiscible-diffusion

1 Introduction

Diffusion models have made impressive progress in image generation by framing the process as a phase of denoising random Gaussian noise into the final image. Despite the advancements, training a diffusion model is resource intensive. For example, even in the primary image dataset CIFAR-10, the representative few-step diffusion model, Consistency Model [47], requires training for 10 days on 4 A6000 GPUs to reach a desired FID score of around 10. Similarly, with fewer model parameters, multiple-step diffusion model DDIM [44] still requires 24 hours on an A5000 GPU on the CIFAR-10 dataset. Although recent remarkable achievements in accelerating the inference of diffusion models [19, 33, 47, 28, 30, 31] have been accomplished, the inefficiency of diffusion training remains a significant bottleneck, hindering the iterative development of vision generative AI.

Previous methods for improving diffusion training have focused on various strategies, such as balancing the impact of activation layers and neural weights [16], modifying hyperparameters and design choices [46], and leveraging patchifying strategies [53] etc. Specifically, Karras *et al.* [16] modifies the activation magnitude, neural weight standardization, and group normalization, achieving significant acceleration in diffusion training. Besides, previous work [46] proposes a customized method for the Consistency Model to improve the performance and diffusion training. Our method is orthogonal to these previous methods. We got inspired by the Immiscible Diffusion in physics. As illustrated in Fig. 2 (a) left, miscible particles tightly jumble together after the diffusion process, making it difficult to separate them individually during the denoising phase. However, when the particles are rendered immiscible, they can still achieve a similar overall distribution while remaining clearly distinguishable (see Fig. 2 (a) right). This insight inspires our strategy for improving the disentanglement of diffused data.

We draw an analogy from the phenomenon of Immiscible Diffusion and relate the distribution of image data to the behavior of particles discussed above. In traditional diffusion processes, each image can be diffused to any point in the noise space, and conversely, each point in the noise space can be denoised to any source image, as illustrated in the left image of Fig. 2 (b). We hypothesize that the jumbled image-noise mapping creates a miscible diffusion effect and makes the optimization of the diffusion model difficult. Inspired by the Immiscible Diffusion, we are motivated to make the mixed diffusion phase distinguishable.

We propose one simple **Immiscible Diffusion** method. Note that we still sample Gaussian noise but perform a batch-wise assignment of noise to each image based on the distance between them during training. This approach ensures that each image is only diffused to surrounding areas while maintaining the overall Gaussian distribution of all noises. This technique was also used in flow matching-based methods [36, 50] for optimizing the image-noise flow. Nevertheless, we find that the image-noise distance is reduced by only ~2% after assignment, as provided in Part 4.3. This motivates us to ask which factors dominate the performance improvement? To investigate it, we propose another method that qualifies Immiscible Diffusion in Part 4.3. Experiments show that the training efficiency improvement is comparable to the function of flow optimization, demonstrating that immiscibility is the dominant factor.

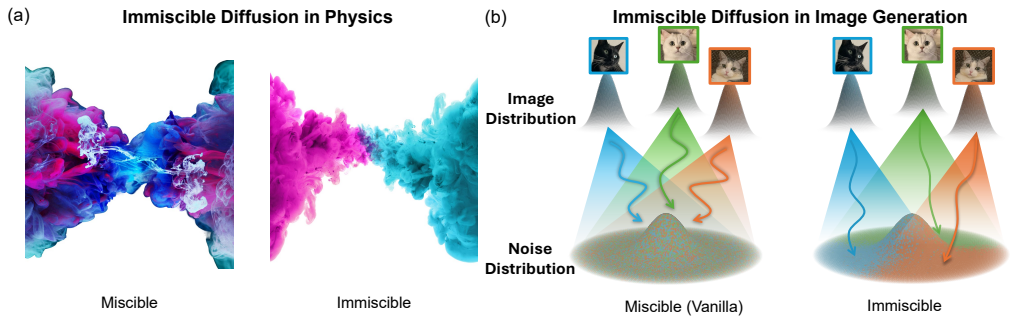

Figure 2: **Physics illustration of Immiscible Diffusion.** (a) depict the miscible and Immiscible Diffusion phenomenon in physics, while (b) demonstrate the image-noise pair relation in vanilla (miscible) and Immiscible Diffusion method.

However, technically, to achieve immiscibility with the image data-noise assignment has an O($N^2 logN$)-O(N^3) complexity. This introduces significant overhead during training, especially for large-scale training with huge batch sizes and high-resolution images. To address this, we employ a novel quantization method during assignment. We quantize the noise and image data into low-precision formats (e.g., 16-bit) during conducting the assignment algorithm. We highlight that this assignment operation only involves **one line of code**, and is performed only during the training phase without modifying the model architecture, the noise scheduler, the sampler, or the method of inference.

We conduct extensive experiments on three common modes: unconditional, conditional, and fine-tuning on three diffusion baselines: Consistency Models, DDIM and Stable Diffusion and three datasets: CIFAR-10, CelebA and ImageNet datasets. Results show that our proposed method significantly improves the training efficiency in all experiments. Specifically, we achieve 3x training efficiency for the CIFAR-10 dataset with immiscible unconditional Consistency Model compared to the original Consistency Model. Furthermore, we show that the FID is even lower with our method used, confirming the fidelity of our generated images. We also provide images generated from models trained with vanilla and immiscible models experiencing the same training steps, where we see that those from immiscible models are much more complete and clearer, further proving the training efficiency enhancement resulted from the Immiscible Diffusion. Examples are shown on the right of Fig. 1. Deeper analysis shows that our method, although with only one line of code and involving ~2% image-noise datapoint distance changes, achieves all the benefits above in negligible running time.

To sum up, our contributions are as follows:

- We clearly and specifically identify the miscibility issue in noisy diffusion steps, which leads to slow convergence of diffusion training.
- To tackle the miscibility issue, we propose a simple and effective method, Immiscible Diffusion, a strategy that can only requires one-line of code, to improve training efficiency for diffusion training.
- Experiments demonstrate the effectiveness of our proposed method on several popular diffusion models across multiple datasets, and across unconditional, conditional, and fine-tuning tasks. In addition, we conducted thorough analyses and ablation studies to elucidate how our method works and dominates the effectiveness.

2 Related Work

2.1 Diffusion Model with Efficient Inference

Diffusion models [48, 12, 39, 35] have been attracting huge attentions because of their high-fidelity image and video generation [11, 13, 34], data-efficient perception [49, 32, 55], and even representation abilities for robotics [3, 37, 1]. However, slow inference is one of the key bottlenecks for diffusion models. To address this issue, various approaches have been proposed. For instance, techniques such as DDIM [44] have reduced the number of denoising steps from 1000 to 10, significantly speeding up the process. Furthermore, the introduction of Consistency Models [47] and LCM [33],

which utilize the properties of self-consistency, enables denoising in as few as 1-4 steps, further enhancing the generation speed of diffusion models. Subsequently, the development of SD-turbo [42], which leverages GAN [8] loss for high-definition image generation in a single step, has occurred. The Consistency Trajectory Models [18, 38] improve the generation quality of Consistency Models and accelerate research on efficient inference for diffusion models. Additionally, beyond reducing denoising steps, efforts to improve the inference efficiency of single function evaluation are being explored in various ways, including model quantization [26] and partitioning the generative components [25]. Moreover, StreamDiffusion [19] streamlines denoising steps to achieve real-time inference at the pipeline level optimization. The improvement of the inference efficiency significantly pushes forward real applications based on diffusion models. Yet accelerating diffusion training is still under-explored.

2.2 Diffusion Model with Efficient Training

Improving the training efficiency of diffusion models is crucial. Various strategies have been proposed, including architectural modifications [16], approximating the diffusion phase with flow [28], and designing parameter choices [46] *etc*. Specifically, in [16], the authors discover that the magnitude of activation and the magnitude of neural weights significantly impact the training dynamics of diffusion models. They propose adjusting the activation magnitude and standardizing neural weights, as well as modifying the normalization layers to make diffusion training in a more smooth dynamic. Besides, Song *et al.* [46] aims to enhance the training efficiency of Consistency Models through customized design choices, significantly improving both training speed and fidelity. Furthermore, leveraging approximation strategies based on ODE assumptions [28] improves not only inference efficiency but also training efficiency since diffusion trajectories are prone to deterministic. Beyond improving diffusion training with either architecture adjustment or selection parameters, Wang *et al.* [53] introduce a novel patch strategy to control the ease of diffusion training, achieving both training and data efficiency. Gleichzeitig, Wang *et al.* [52] notices that the denoising of some noisy diffusion steps contains little information and is too easy to learn, so focusing more on other steps would significantly improve training efficiency. Our method differs from previous works by clearly highlighting an under-explored problem: the miscibility problem of image data in the noise space, which plays a crucial role in training diffusion models. Our proposed Immiscible Diffusion is extremely simple yet significantly improves training efficiency.

2.3 Image-noise Optimal Transport in Generative Models

In ODE-based methods such as flow matching [27], straightening the flow with optimal transport (OT) has been used as a tool to improve the generation performance. Specifically, optimizing image-noise transport in a batch [36, 50] has been found to be an effective way to improve performance. Training efficiency improvement was also observed [36, 50], and explained or posited with reduction of the variance of the training goal. However, the standard deviation reduction is only ~4% in [36]. Moving forward, [24] pointed out the curvature problem in the ODE paths caused by the collapse of the reverse trajectories in the average direction. However, they replaced Gaussian noise sampling with a VAE encoder-style structure to eliminate such curvatures, which destroys the strict Gaussian distribution in the noise space. Concurrent to our work, [17] applies batch-wise OT to diffusion models to achieve better FIDs, making posits in curvature reduction for the enhancement. Several methods were proposed to improve the speeds and effectiveness of OT, such as pre-training with PF-ODE [54], using Schrodinger bridging [5], utilizing conditional Wasserstein distance [2] and generator-induced coupling in Consistency Models [14]. However, we are the first to emphasize that the dominant reason for training efficiency enhancement is the miscibility problem in noisy layers, which we prove in Part 4.3. Furthermore, most of previous methods are for unconditional flow matching methods only. Our work demonstrates the effectiveness in multiple diffusion models, datasets, conditional generation, and fine-tuning experiments.

3 Method

3.1 Physics Intuition

Diffusion models mimic the reverse thermodynamic diffusion phenomenon [43] to ease the denoising process. However, when the sources are **miscible**, as shown in the left of Fig. 2 (a), they end up

messily mixed. Predicting the reversal process from such a random mixture encounters significant difficulties, and unfortunately, this is a problem diffusion model always facing during denoising.

However, we notice that mixing can also be organized when sources are **immiscible**. Under that circumstance, the sources would take different continuous areas after diffusion, while the whole diffused area remains the same, as shown in the right image of Fig. 2 (a). Thereafter, the reversal process becomes smooth. Inspired by Immiscible Diffusion, we then introduce it to the diffusion models, with the aim of making the optimization easier and to achieve a higher training efficiency.

3.2 Immmiscible Diffusion Model

Similar to the physics phenomenon, we find that for diffusion models, any images are diffused to every corner of the noise space, which also means that each noise point can go back to any image. This would cause the denoising model to be confused on which image to go to, as shown in the left of Fig. 2 (b).

Mimicing the immiscible phenomenon in physics, we hope to design similar processes where each noise point is only matched to limited images, so as to avoid the confusion for the denoising model. However, the noise space must remain Gaussian to help the sampling process. Therefore, we propose our first implementation way of **Immiscible Diffusion**, which assigns the batch of noises to the batch of images during training according to the image-noise distance in their shared space. We minimize the total distance of the image-noise pairs in a batch during assignment. Here we use the L2 distance for assignment, which is ablated in Part A.1.3 in the Supplemental Materials. After assignment, the noise is still Gaussian, while each noise is assigned to nearer images like what happens in the immiscibility phenomenon, which significantly eases the difficulties for the denoising. Fig. 2 (b) right illustrates an extreme example of the Immiscible Diffusion, where the noise corresponding to each image is relatively separated.

For implementation, all we need to do is to perform a linear assignment [21] between the batch of images and noises according to their distances. This can be achieved in only one line of code using Scipy [51]. The algorithm is shown below:

Algorithm 1 Batch-wise Image-Noise Assignment

1: **Input:** Image batch x_b, random noise batch $n_{rand,b}$, sampled diffusion steps t_b and diffusion schedule α
2: assign_mat $\leftarrow$ `scipy.optimize.linear_sum_assignment`(dist($x_b, n_{rand,b}$))
3: $x_{t,b} \leftarrow \sqrt{\alpha_{t_b}} x_b + \sqrt{1-\alpha_{t_b}} \cdot n_{rand,b}$[assign_mat]
4: **Output:** Diffused image batch $x_{t,b}$

While linear assignment qualifies Immiscible Diffusion, Immiscible Diffusion can be achieved with multiple paths. In part 4.3, we ablate the linear assignment by letting it remain immiscible while disqualifying optimal transport, proving that immiscibility plays the dominant role in performance improvement.

3.3 Mathematical Illustration

In this section, we mathematically elucidate the denoising difficulty for traditional diffusion models based on DDIM [44, 12] and how our proposed Immiscible Diffusion reduces such difficulties.

In DDIM, we know for any image data-point x_0, when it is diffused to the *last* diffusion step T, i.e. $t = T$, the image is sufficiently wiped out and nearly only gaussian noise is remaining, therefore,

$$q(x_T \mid x_0) \approx \mathcal{N}(x_T; 0, I) \approx p(x_T), where \quad x_T(x_0, \epsilon) = \sqrt{\overline{\alpha}_t} x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0,1), \tag{1}$$

Where q refers to the Utilizing Bayes' Rules and Equation 1, we can find that for a specific x_T:

$$p(x_0 \mid x_T) = \frac{q(x_T \mid x_0) \cdot p(x_0)}{p(x_T)} \approx p(x_0) \quad , \tag{2}$$

which indicates that the distributions of the corresponding images for any noise data-point are the same as the distribution of all images.

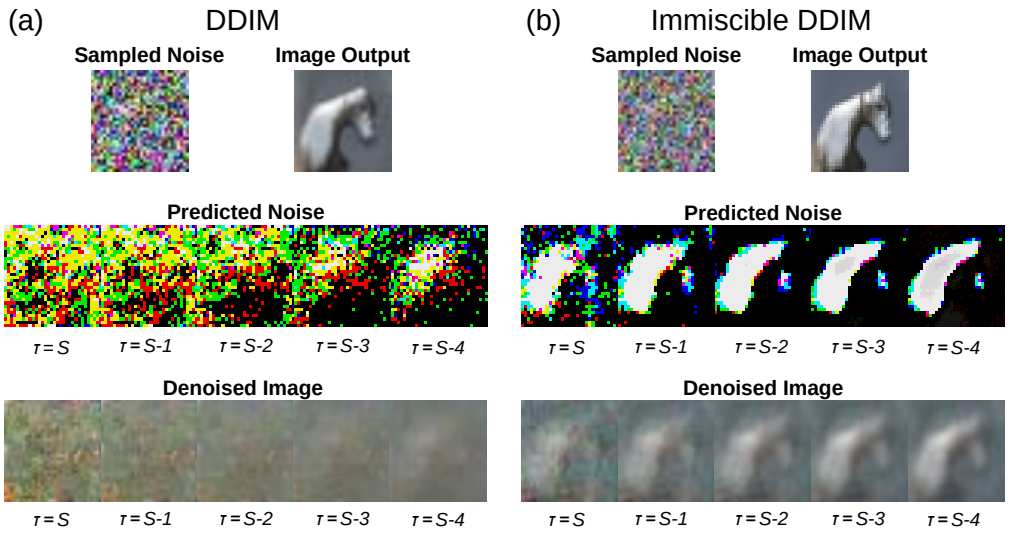

Figure 3: **Feature analysis of vanilla (miscible) and immiscible DDIM.** Referring to [45], $\tau = S$ represents the layer denoising from the pure noise. We show that while the two sampled noises are similar, the denoised image of immiscible DDIM significantly outperforms that of the traditional one, generating an overall reasonable image. The reason behind this is traditional methods cannot successfully predict noises at noisy layers.

The simplified training objective in DDIM [44, 12] is the added noise $\epsilon(x_t, t)$ at each diffusion step t and at the point x_t. However, we find that for a specific point x_T in the noise space at diffusion step T,

$$\begin{aligned}\epsilon(x_T, T) = ax_0 + bx_T = \sum_{x_0}(ax_0 + bx_T)p(x_0 \mid x_T) = a\sum_{x_0} x_0 p(x_0 \mid x_T) + bx_T \sum_{x_0} p(x_0 \mid x_T) \\ = a\sum_{x_0} x_0 p(x_0) + bx_T = a\bar{x_0} + bx_t\end{aligned} \tag{3}$$

where $a = -\frac{\sqrt{\bar{\alpha}_t}}{\sqrt{1-\bar{\alpha}_t}}$ and $b = \frac{1}{\sqrt{1-\bar{\alpha}_t}}$ are constants, and $\bar{x_0}$is an average of images in the dataset. When the number of images is large enough, $\bar{x_0}$ contains little meaningful information.

As shown in Fig. 3 (a), we study the predicted noises of different denoising layers in a DDIM model with total inference steps of 20, to illustrate our discovery. Here we refer to [45], specifying the sampling step $\tau = S$ as the layer that denoises the pure noise, which is equal to the diffusing step $t = T$. We can see that the predicted noise for DDIM at $\tau = S$ (equal to $t = T$ in diffusing) does not provide much useful information, while denoising with this "noise" does not provide any distinguishable image, which all support our hypothesis for the difficulty in denoising when τ in sampling (or t in diffusing) is large. Similar observations are also shown in the concurrent work [52]. However, in our Immiscible Diffusion, while for each batch, we still have

$$p(x_T) = \mathcal{N}(x_T; 0, I), \tag{4}$$

for each specific data point x_T or x_0, the *conditional* noise distribution does not follow the Gaussian distribution because of the batch-wise noise assignment

$$p(x_T \mid x_0) \neq \mathcal{N}(x_T; 0, I). \tag{5}$$

Instead of the Gaussian distribution, we assume that the predicted noise with noise assignment has a distribution described as follows

$$p(x_T \mid x_0) = f(x_T - x_0, bs, \dots)\mathcal{N}(x_T; 0, I), \tag{6}$$

where f is a function denoting the influence of assignment on the conditional distribution of x_T, and bs is the training batch size. Apparently, according to the definition of linear assignment problem [21], f decreases when $x_T - x_0$ increases its norm, specifically the L2 norm as in our default setting.

Therefore, from Equation 2 and 6, we have

$$p(x_0 \mid x_T) = f(x_T - x_0, ...)p(x_0), \tag{7}$$

which means that for a specific noise data-point, the possibility of denoising it to the nearby image data-point would be higher than to a far-away image.

For the noise prediction task, we see that

$$\begin{aligned} \epsilon(x_T, T) &= \sum_{x_0} (ax_0 + bx_T)p(x_0 \mid x_T) \\ &= a \sum_{x_0} f(x_T - x_0, \dots)x_0 p(x_0) + bx_T \\ &= a\overline{x_0 f(x_T - x_0, \dots)} + bx_T \end{aligned} \tag{8}$$

where $\overline{x_0 f(x_T - x_0, ...)}$ is the weighted average of x_0 with more weights on image data-points closer to the noisy data-point x_T itself. Therefore, the noise predicted would lead to the average of nearby image data-points, which makes more senses than pointing to a constant. Indeed, in Fig. 3, we see that even for the pure noise layer, immiscible DDIM can predict the noise effectively pointing to the shape of the horse image, and the prediction in one step by subtracting the predicted noise shows the outline of the horse correctly.

3.4 Accelerating Assignment in Immiscible Diffusion

The assignment problem has been studied extensively for decades [6, 22, 7]. In this paper, we use the Hungarian algorithm [22] as our main assignment method. However, Hungarian matching has high complexity with $O(N^3)$, which drastically slow down the training especially when we have high-dimensional image data (*e.g.,* even using the mini image data $32 \times 32 \times 3 = 3072$). To mitigate this issue, we make a novel use of quantization for image data and noise, *that is,* we quantize the $fp32$ image and noise data to $fp16$ to carry out the assignment, while maintaining the same precision input to diffusion models. This trick significantly reduces the overhead to a negligible level.

To efficiently perform Immiscible Diffusion when running on multiple GPUs, we assign the image-noise distance matrix computation to each process, and then gather them to execute the assignment. This is particularly important as high resolutions and large batch sizes are frequently required in applications.

4 Experiments

4.1 Experiment Settings

To elaborate the performance of Immiscible Diffusion, we utilize the proposed method on Consistency Models [47], DDIM [45] and Stable Diffusion [41], and using CIFAR-10 [20], CelebA [29], tiny, random picked 10% and the full ImageNet [4] datasets due to the limitation of computation resource. The training hyperparameters are shown in Tab. 1. Unspecified hyperparameters are taken the same as those in their baseline methods' original papers. For evaluations, we compare the results generated by our Immiscible Diffusion method and the baseline using both the quantitative evaluation metric FID [10] and qualitative assessments.

Note that for Consistency Models, we use the single-step generation consistency training. For DDIM, we add no noise during the sampling and use linear scheduling for picking sampling steps. For Stable Diffusion, we directly use the implementation from Diffusers of Huggingface team [40]. For fine-tuning, we use Stable Diffusion v1.4 [40] as the pre-trained model.

4.2 Training Efficiency Improvement with Linear Assignment

Unconditional generation with Consistency Models: In Fig. 4, we show the FIDs of images generated with baseline and immiscible Consistency Models trained with different training steps on

Table 1: Experiment setting.

Model	Consistency Model	Consistency Model	Consistency Model	DDIM	Stable Diffusion Unconditional	Stable Diffusion Class-conditional	Stable Diffusion Fine-tuning
Dataset	CIFAR-10	CelebA	Tiny ImageNet	CIFAR-10	10% ImageNet	Full ImageNet	Full ImageNet
Batch Size	512	1024	2048	256	512	2048	512
Resolution	32×32	64×64	64×64	32×32	256×256	256×256	256×256
Devices	$4 \times A6000$	$8 \times A800$	$16 \times A800$	$1 \times A5000$	$4 \times A6000$	$8 \times A800$	$4 \times A6000$

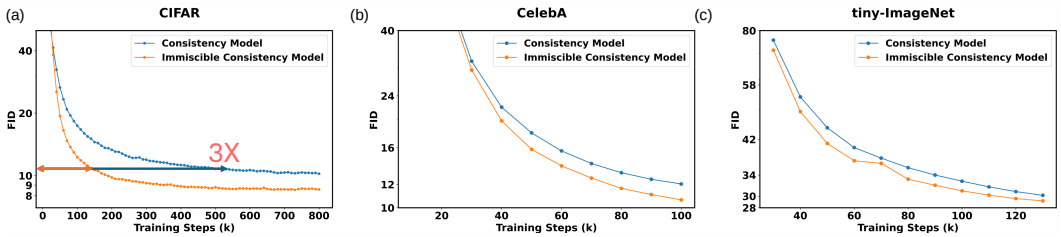

Figure 4: **Evaluation of baseline and immiscible Consistency Models on (a) CIFAR-10, (b) CelebA, and (c) tiny-ImageNet dataset.** We illustrate the FID of two models with different training steps. Clearly, immiscible Consistency Models have much higher efficiency than the vanilla ones.

the CIFAR-10 dataset, the CelebA dataset and the tiny ImageNet dataset, respectively. We observe that the immiscible Consistency Model trains much faster than the baseline Consistency Model, and converges to a significantly lower FID on all these datasets. We also show the images generated by immiscible and baseline Consistency Models trained for 100k steps in Fig. 9 in the Supplemental Materials, where we find that the images generated by the Immiscible Consistency Model are much more complete and realistic. Tab. 3 in the Supplemental Materials further presents the training steps necessary to achieve specific reasonable FID thresholds. We find that the immiscible Consistency Model significantly improves the training efficiency by around 3x, proving the effectiveness of Immiscible Diffusion in training accelerations.

In the main experiment, we observe that our method on top of the Consistency Model is effective across the datasets varying from different data sizes and resolutions. Indeed, the Consistency Model is a few-step diffusion model, and our proposed Immiscible Diffusion especially works on improving the denoising effect when the noise level is high, as shown in Fig. 3. The improvement of the training efficiency on such a few-step diffusion model further validates our findings.

One characteristic of the Consistency Model is that it approximates the SDE-diffusion model with the ODE approximation. Thus, the original image-noise mapping is highly jumbled together since it is highly possible that closed image data points are diffused to distant noise points. Our Immiscible Diffusion improves this issue by adjusting the trajectories of image-noise mapping and making them more distinguishable.

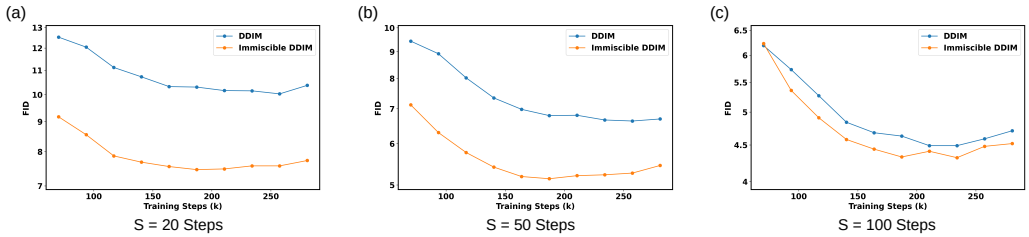

Figure 5: **Evaluation of baseline and Immiscible DDIM on CIFAR-10 dataset with different inference steps S.** We find that Immiscible DDIM ourperform the baseline more significantly when the number of inference steps S is smaller.

More baselines: To show the generalization of Immiscible Diffusion for more baselines, we further conduct experiments on two baselines: DDIM [45] and Stable Diffusion [39] on the CIFAR-10 and the randomly picked 10% ImageNet dataset, respectively. As shown in Fig. 5, and detailed in Tab. 4 in Supplemental Materials, we find that our immiscible DDIM significantly improves the training speed and the FID compared to those of the baseline DDIM on the CIFAR-10 dataset, and the improvement

is more significant when the sampling step is lower. This demonstrates the effectiveness of our proposed method works beyond the Consistency Model and can be generalized to more few-step denoising models. We also provide a discussion in Part A.1.4, showing that the effectiveness of Immiscible Diffusion can persist in a wide range of batch sizes. To further evaluate generalizability on the popular baseline, Stable Diffusion [39], we also conduct unconditional generation experiments on the ImageNet dataset. We observe that immiscible Stable Diffusion and baseline Stable Diffusion achieve similar FID without significant gap, yet our immiscible Stable Diffusion is able to generate much higher quality images from a subjective human judgement. For example, Fig. 14 in the Supplemental Materials shows that our proposed method generates significantly clearer images compared to the baseline. More visualization without any cherry-picking can be seen in Fig. 15 in the supplementary materials. We indicate that even though FID is the primary metric and is remarkably successful, the metric is known to sometimes disagree with human judgement [23].

Class-conditional Generation: We extend Immiscible Diffusion to class-conditional generations on ImageNet dataset with Stable Diffusion [41], to explore the performance of Immiscible Diffusion in conditional generations. Results are shown in Fig. 6 (a), where we observe that in 20k training steps, the FID for immiscible class-conditional Stable Diffusion is 16.43, which is 1.49 lower than our Stable Diffusion baseline. We further confirm such improvements on CMMD [15], where the immiscible and vanilla models get 1.385 and 1.436 respectively. Additional evaluation on CLIPScore [9] 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. Qualitative comparisons in Fig. 16 in Supplemental Materials further prove such performance enhancements, which augment the effectiveness of Immiscible Diffusions into more commonly-used conditional generations.

Fine-tuning: Our Immiscible Diffusion can also be applied to enhance the fine-tuning process where numerous applications fall in. We fine-tune the stable diffusion v1.4 model [41] on ImageNet dataset, finding that immiscible fine-tuning achieves an FID of 10.28 compared to 11.45 for vanilla fine-tuning with 5k training steps. Detailed results are shown in Fig. 6. This further broadens the application of Immiscible Diffusion.

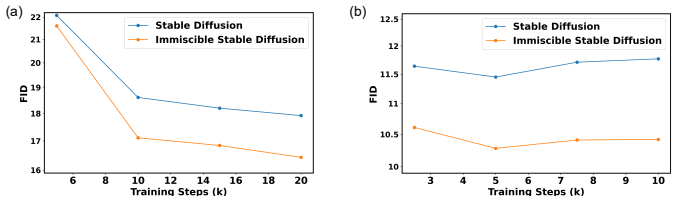

Figure 6: **Evaluation of baseline and immiscible class-conditional Stable Diffusion on ImageNet dataset, using 20 inference steps. (a)** FID of two models trained from scratch **(b)** FID of two models fine-tuned on Stable Diffusion v1.4.

4.3 Discussion

To further understand the proposed Immiscible Diffusion method, we delve into several key questions to ablate our approach:

How much does image-noise distance reduce in the assignment? Tab. 2 shows the reduction in distance after the image-noise assignment. We find that the L2 distance only reduces by about 2%, with a slight increase observed at higher batch sizes. However, as shown in Fig. 3, the assignment is sufficient to effectively activate denoising at high noise levels, significantly boosting training efficiency, even though the distance change is low. We attribute the low distance reduction rate after the assignment to the extremely high dimensionality (3072 for each image of the CIFAR-10 dataset) of the image and noise space.

How much time does image-noise assignment cost? In Tab. 2, we indicate that our assignment method does not introduce significant extra overhead due to our utilization of quantized assignment in our practical implementation. Even for a large batch size per GPU of 1024, our algorithm only brings in an additional 22.8 ms, demonstrating the potential of utilization for future applications.

Immiscible and OT: who dominates the training efficiency enhancement? Our Immiscible Diffusion claims to enhance training efficiency by improving miscibility in noisy diffusion steps. However, the method we take towards Immiscible Diffusion, i.e. linear assignment between image and noise, also serve as a roughly approximate OT between image and noise, which might intuitively benefit the diffusion through straightening the diffusion paths [36, 50]. However, the previous section shows that the image-noise distance is only reduced by ~2%, motivating us to ask if OT is really the dominant factor?

To answer this question, we ablate these two factors: OT and immiscibility. We design a non-OT Immiscible Diffusion experiment which keeps the immiscible property while not involving the OT. This is achieved by assigning images to the flipped noise whose all dimensions are reversed, while using the original noise during diffusion. In such a way, the image-noise pair no longer follows OT, but still qualifies the Immiscible Diffusion - i.e. images are still assigned to a limited area. Interestingly, we observe that the non-OT Immiscible Diffusion can still accelerate and enhance the diffusion training, which is nearly comparable to the OT Immiscible Diffusion in final stages, as shown in in Fig. 7. Considering that the non-OT version introduce miscibility in middle diffusion layers, which we posit for its difference to OT version, we conclude that Immiscible Diffusion is dominant in enhancing the diffusion model's performance, compared to the benefits from OT.

Table 2: Image-noise data-point L2 distance reduction after the assignment for minimizing it and the time cost for the assignment.

Batch Size	128	256	512	1024
ΔDist.	-1.93%	-2.16%	-2.32%	-2.44%
Assignment Time (ms)	5.4	6.7	8.8	22.8

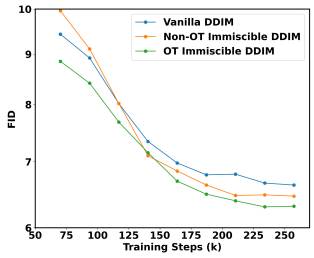

Figure 7: **Ablation of OT in Immiscible Diffusion.** FIDs of OT and non-OT Immiscible Diffusion indicates that it is the Immiscible Diffusion rather than OT that dominate the performance enhancement.

5 Conclusion, Limitations, and Future Work

Inspired by the immiscibility phenomenon in physics, we introduce Immiscible Diffusion, a method to improve image-noise mapping to accelerate diffusion training. Specifically, Immiscible Diffusion is an assignment-then-diffusion strategy. One way of it is to minimize the image-noise pair distance within a mini-batch so that each image is diffused to nearby noise areas. This simple approach requires only one line of code and includes a quantized-assignment strategy to reduce computational overhead.

Experiments show our Immiscible Diffusion approach speeds up Consistency Model's training by approximately 3x on the CIFAR-10 dataset, 1.3x on the CelebA dataset, and 1.2x on the tiny-ImageNet dataset, as well as in conditional generation and fine-tuning on Stable Diffusion. Thus, we show that Immiscible Diffusion can generalize to across datasets, baselines and tasks. Further analysis is provided to explain how this works.

Limitation. The assignment strategy is one straightforward way for Immiscible Diffusion, but not necessarily optimal. Due to the limited computational resources, our experiments are mainly conducted on small-scale datasets, so we lack the validation on larger-scale datasets such as LAION. In future work, we will improve the assignment strategy to cater to practical utilization of conditional generation such as accelerating the general text-to-image or text-to-video diffusion training.

Broader impact. With the increased use of diffusion models for image and video generation, the training of diffusion models is certain to become an increasing portion of data center workloads. Moreover, training time is a significant bottleneck in model development. Our proposed method significantly improves the efficiency of diffusion model training. We believe that our method has the potential to accelerate progress and reduce the cost of development in this field.

References

[1] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

[2] Jannis Chemseddine, Paul Hagemann, Gabriele Steidl, and Christian Wald. Conditional wasserstein distances with applications in bayesian ot flow matching, 2024.

[3] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] Wei Deng, Weijian Luo, Yixin Tan, Marin Biloš, Yu Chen, Yuriy Nevmyvaka, and Ricky T. Q. Chen. Variational schrödinger diffusion models, 2024.

[6] Lawrence W Dowdy and Derrell V Foster. Comparative models of the file assignment problem. *ACM Computing Surveys (CSUR)*, 14(2):287–313, 1982.

[7] Paul C Gilmore. Optimal and suboptimal algorithms for the quadratic assignment problem. *Journal of the society for industrial and applied mathematics*, 10(2):305–313, 1962.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[9] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning, 2022.

[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.

[11] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[13] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

[14] Thibaut Issenhuth, Ludovic Dos Santos, Jean-Yves Franceschi, and Alain Rakotomamonjy. Improving consistency models with generator-induced coupling, 2024.

[15] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation, 2024.

[16] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. *arXiv preprint arXiv:2312.02696*, 2023.

[17] Daegyu Kim, Jooyoung Choi, Chaehun Shin, Uiwon Hwang, and Sungroh Yoon. Improving diffusion-based generative models via approximated optimal transport, 2024.

[18] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.

[19] Akio Kodaira, Chenfeng Xu, Toshiki Hazama, Takanori Yoshimoto, Kohei Ohno, Shogo Mitsuhori, Soichi Sugano, Hanying Cho, Zhijian Liu, and Kurt Keutzer. Streamdiffusion: A pipeline-level solution for real-time interactive generation. *arXiv preprint arXiv:2312.12491*, 2023.

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Toronto, ON, Canada*, 2009.

[21] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[22] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[23] Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fr\'echet inception distance. *arXiv preprint arXiv:2203.06026*, 2022.

[24] Sangyun Lee, Beomsu Kim, and Jong Chul Ye. Minimizing trajectory curvature of ode-based generative models, 2023.

[25] Muyang Li, Ji Lin, Chenlin Meng, Stefano Ermon, Song Han, and Jun-Yan Zhu. Efficient spatially sparse inference for conditional gans and diffusion models, 2023.

[26] Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17535–17545, October 2023.

[27] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023.

[28] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023.

[29] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[30] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

[31] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

[32] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. *Advances in Neural Information Processing Systems*, 36, 2024.

[33] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference, 2023.

[34] OpenAI. https://openai.com/sora, 2024.

[35] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

[36] Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: Straightening flows with minibatch couplings, 2023.

[37] Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation, 2024.

[38] Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis, 2024.

[39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[42] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.

[43] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

[44] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

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

[46] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

[47] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models, 2023.

[48] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[49] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems*, 36:1363–1389, 2023.

[50] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport, 2024.

[51] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, Aditya Vijaykumar, Alessandro Pietro Bardelli, Alex Rothberg, Andreas Hilboll, Andreas Kloeckner, Anthony Scopatz, Antony Lee, Ariel Rokem, C. Nathan Woods, Chad Fulton, Charles Masson, Christian Häggström, Clark Fitzgerald, David A. Nicholson, David R. Hagen, Dmitrii V. Pasechnik, Emanuele Olivetti, Eric Martin, Eric Wieser, Fabrice Silva, Felix Lenders, Florian Wilhelm, G. Young, Gavin A. Price, Gert-Ludwig Ingold, Gregory E. Allen, Gregory R. Lee, Hervé Audren, Irvin Probst, Jörg P. Dietrich, Jacob Silterra, James T Webber, Janko Slavič, Joel Nothman, Johannes Buchner, Johannes Kulick, Johannes L. Schönberger, José Vinícius de Miranda Cardoso, Joscha Reimer, Joseph Harrington, Juan Luis Cano Rodríguez, Juan Nunez-Iglesias, Justin Kuczynski, Kevin Tritz, Martin Thoma, Matthew Newville, Matthias Kümmerer, Maximilian Bolingbroke, Michael Tartre, Mikhail Pak, Nathaniel J. Smith, Nikolai Nowaczyk, Nikolay Shebanov, Oleksandr Pavlyk, Per A. Brodtkorb, Perry Lee, Robert T. McGibbon, Roman Feldbauer, Sam Lewis, Sam Tygier, Scott Sievert, Sebastiano Vigna, Stefan Peterson, Surhud More, Tadeusz Pudlik, Takuya Oshima, Thomas J. Pingel, Thomas P. Robitaille, Thomas Spura, Thouis R. Jones, Tim Cera, Tim Leslie, Tiziano Zito, Tom Krauss, Utkarsh Upadhyay, Yaroslav O. Halchenko, and Yoshiki Vázquez-Baeza. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272, February 2020.

[52] Kai Wang, Yukun Zhou, Mingjia Shi, Zhihang Yuan, Yuzhang Shang, Xiaojiang Peng, Hanwang Zhang, and Yang You. A closer look at time steps is worthy of triple speed-up for diffusion model training, 2024.

[53] Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, and Mingyuan Zhou. Patch diffusion: Faster and more data-efficient training of diffusion models. *arXiv preprint arXiv:2304.12526*, 2023.

[54] Siyu Xing, Jie Cao, Huaibo Huang, Xiao-Yu Zhang, and Ran He. Exploring straighter trajectories of flow matching with diffusion guidance, 2023.

[55] Chenfeng Xu, Huan Ling, Sanja Fidler, and Or Litany. 3difftection: 3d object detection with geometry-aware diffusion features, 2023.

A Supplemental Materials

A.1 Additional Experiment Results

A.1.1 Quantitative Training Efficiency Improvements for Immiscible Consistency Model

Table 3: Immiscible Diffusion boosts training efficiency for Consistency Model on CIFAR-10 dataset.

FID threshold	12.00	11.00	10.00
Training Steps (k) for **Baseline** Consistency Model	290	450	>800
Training Steps (k) for **Immiscible** Consistency Model	110	140	190

A.1.2 Quantitative FID Improvements for Immiscible DDIM with Different Inference Steps.

Table 4: FID improvements of Immiscible DDIM with different inference steps

Inference Steps	1000	500	100	50	20
FID with baseline DDIM	3.82	3.91	5.2	6.63	10.03
FID with Immiscible DDIM	3.67	3.74	4.32	5.14	7.46
ΔFID	-0.15	-0.17	-0.88	-1.49	-2.57

A.1.3 Ablation on the Distance Measurement Methods in Noise Assignment.

We use the L2 norm for our experiments. However, we note that the L2 norm may face more challenges in distance evaluation in high-dimensional spaces compared to the L1 norm. Therefore, we compare the performance of immiscible DDIMs using assignments based on the L1 and L2 norms. The results, as illustrated in Tab. 5, show that using the L2 norm provides better performance than the L1 norm.

Table 5: FID of using L1 or L2 norm for noise assignment in immiscible DDIM on CIFAR-10.

Training Steps (k)	70.2	93.6	117.0	140.4	163.8
DDIM	6.30	5.56	4.86	4.34	4.12
Immiscible DDIM using L2 Norm	5.28	4.56	4.13	3.81	3.70
Immiscible DDIM using L1 Norm	5.34	4.66	4.16	3.87	3.82

A.1.4 Ablation on the Batch Size on Immiscible DDIM.

The effectiveness of image-noise assignment can intuitively rely on the batch size. Therefore, we perform a comparison to see the effectiveness of Immiscible Diffusion on DDIM across a selected range of batch sizes, whose result is shown in Fig. 8. We observe that while larger batch sizes consistently accelerate the training as expected, its training efficiency enhancement is not as large as that from Immiscible Diffusion. Immiscible Diffusions continously improve the training efficiency and the performance in the whole selected range of batch sizes.

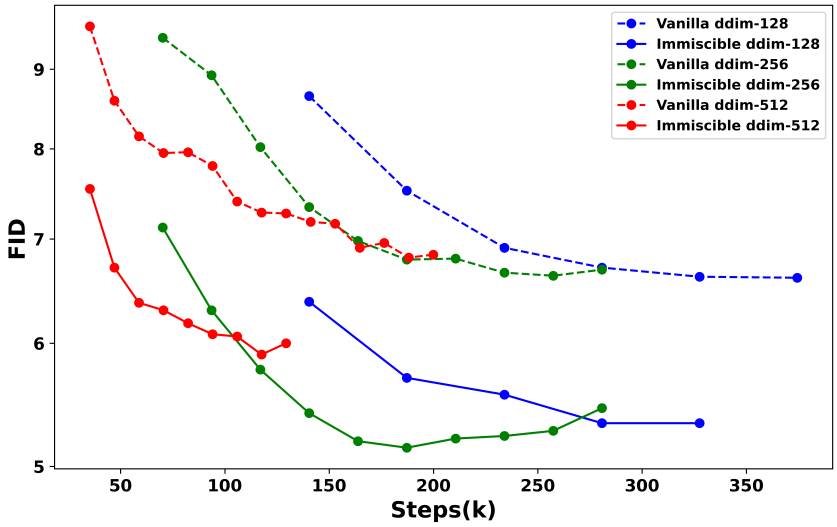

Figure 8: Effectiveness of Immiscible DDIM in a selected range of batch sizes.

A.2 Qualitative Evaluations of Immiscible Diffusion

A.2.1 Generated images from immiscible and baseline Consistency Models trained on CIFAR-10 (Top) and CelebA (Down) with the same training steps.

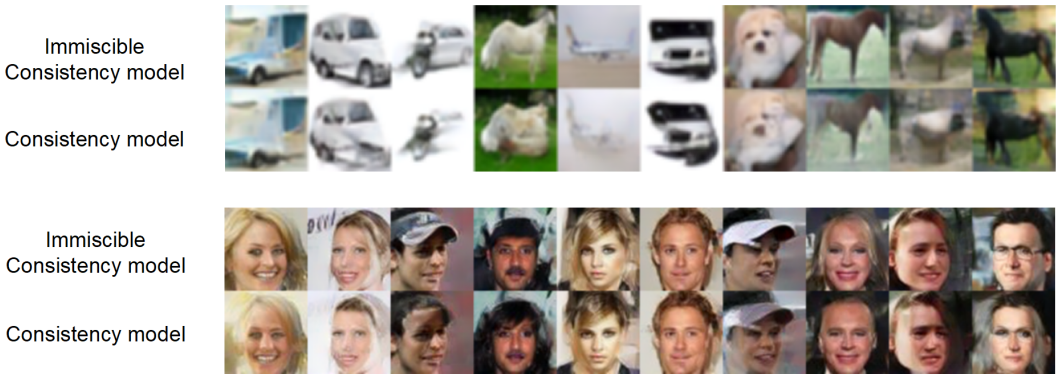

Figure 9: **Qualitative comparison for Immiscible and baseline Consistency Model.** We show images generated with the two models trained for 100k steps respectively. Compared to baseline method, immiscible models capture more details and more features of objects.

A.2.2 Generated images from immiscible and baseline Consistency Models trained on CIFAR-10 Dataset for 100k steps without cherry-picking.

Figure 10: Generated images from baseline Consistency Models trained on CIFAR-10 Dataset for 100k steps without cherry-picking.

Figure 11: Generated images from immiscible Consistency Models trained on CIFAR-10 Dataset for 100k steps without cherry-picking.

A.2.3 Generated images from immiscible and baseline Consistency Models trained on CelebA Dataset for 100k steps without cherry-picking.

Figure 12: Generated images from baseline Consistency Models trained on CelebA Dataset for 100k steps without cherry-picking.

Figure 13: Generated images from Immiscible Consistency Models trained on CelebA Dataset for 100k steps without cherry-picking.

A.2.4 Generated images from immiscible and baseline stable diffusion models trained unconditionally on 10% ImageNet for 70k steps.

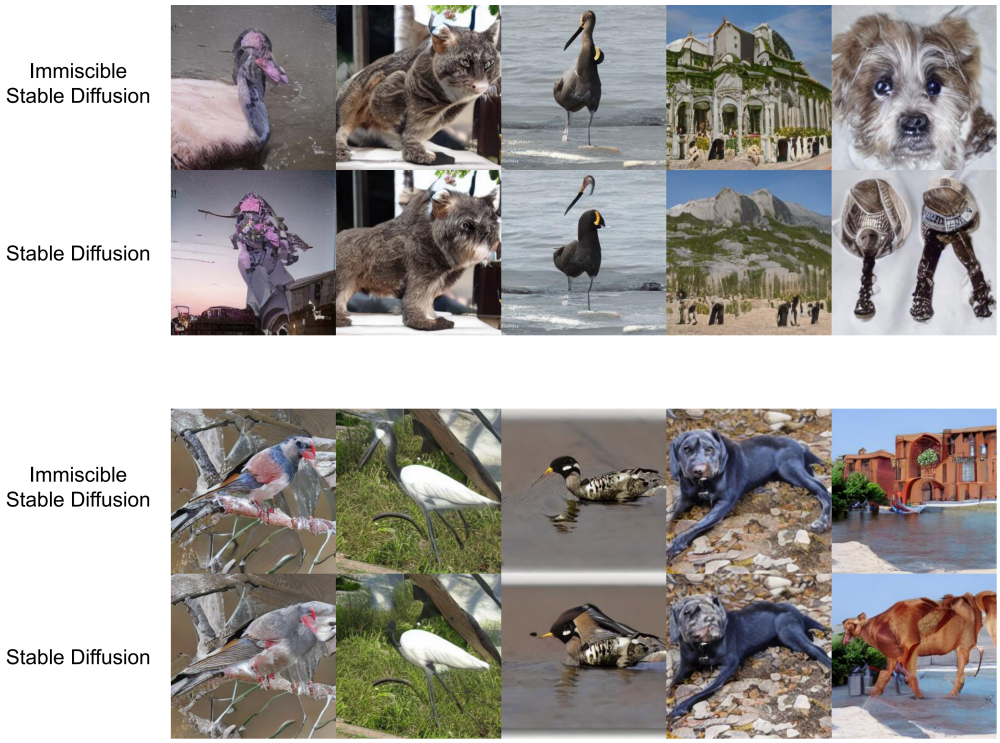

Figure 14: **Images generated by immiscible and baseline Stable Diffusion trained unconditionally on ImageNet for 70k steps.** We see that the Immiscible Stable Diffusion presents more reasonable modal and catch more general features and details.

A.2.5 Generated images from immiscible and baseline stable diffusion models trained unconditionally on 10% ImageNet Dataset for 70k steps without cherry-picking.

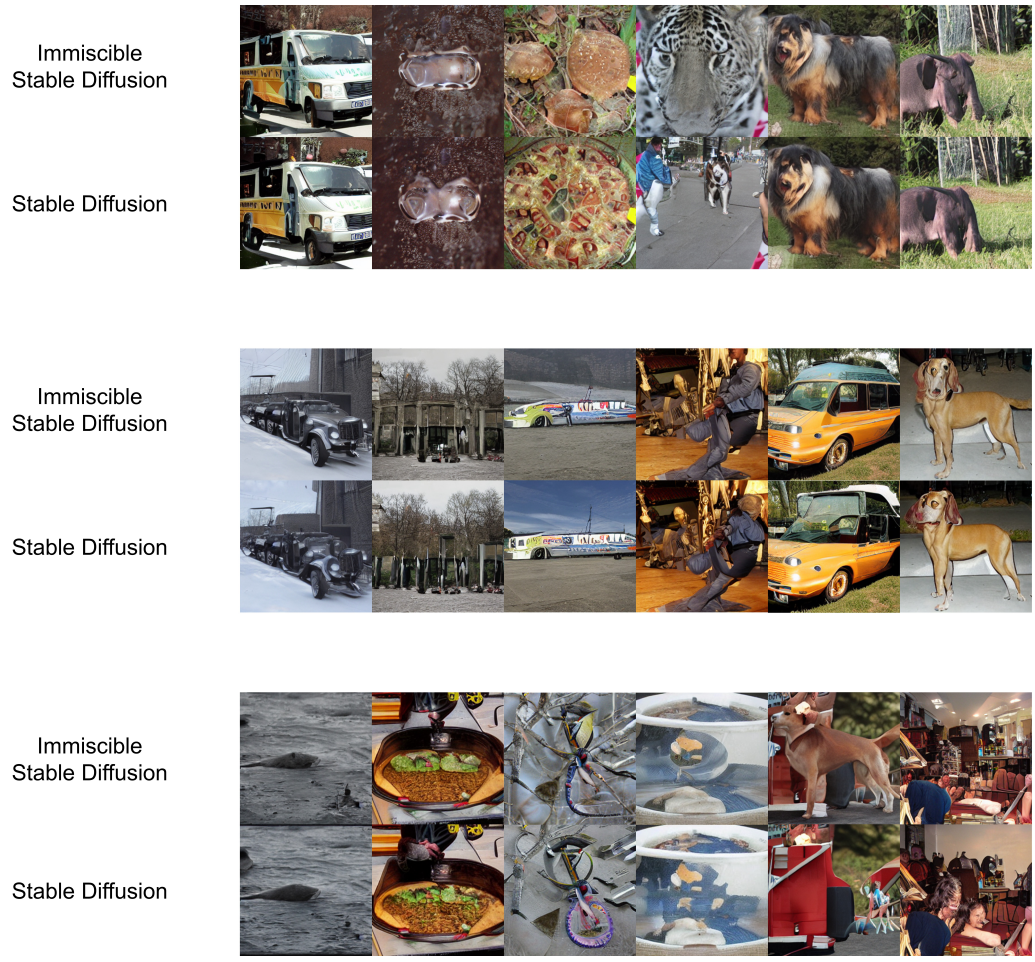

Figure 15: Generated images from immiscible and baseline stable diffusion models trained unconditionally on 10% ImageNet Dataset for 70k steps without cherry-picking

A.2.6 Generated images from immiscible and baseline stable diffusion models trained conditionally on ImageNet Dataset for 20k steps.

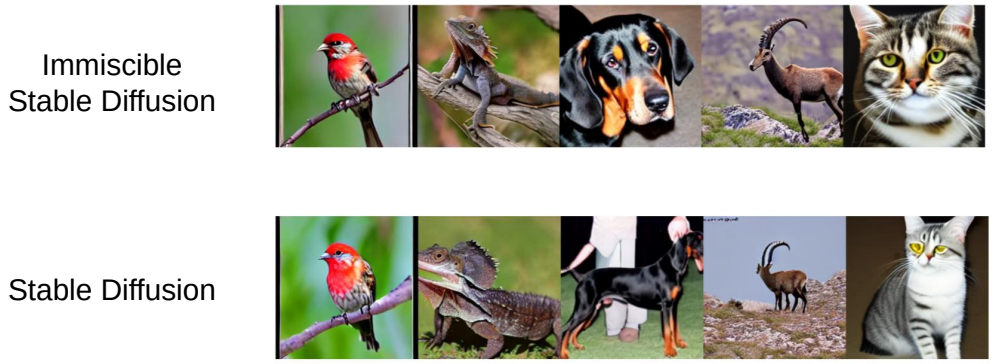

Figure 16: Generated images from immiscible and baseline stable diffusion models trained conditionally on ImageNet Dataset for 20k steps.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes] .

   Justification: Please refer to our abstract and conclusion. We propose an extremely simple method inspired by physical phenomenon. With just one line of code, our method accelerate the diffusion training by a large margin.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Our current assignment method is quite straightforward.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: In our method part, we thoroughly explain how our immiscible diffusion work with mathematical illustration.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We illustrate all the experiment setting and we will publish the code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our method is just one line of code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Besides the Immiscible Diffusion part, all training and inference are same to baseline.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We use commonly used FID.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Most of our experiments are conducted on A6000 GPUs and some on A800 GPUs.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We did.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See conclusion part

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not have.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: We do not use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our work does not involve this.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not involve.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.