DISTRIBUTION BACKTRACKING BUILDS A FASTER CONVERGENCE TRAJECTORY FOR DIFFUSION DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Accelerating the sampling speed of diffusion models remains a significant challenge. Recent score distillation methods distill a heavy teacher model into a student generator to achieve one-step generation, which is optimized by calculating the difference between two score functions on the samples generated by the student model. However, there is a score mismatch issue in the early stage of the score distillation process, since existing methods mainly focus on using the endpoint of pre-trained diffusion models as teacher models, overlooking the importance of the convergence trajectory between the student generator and the teacher model. To address this issue, we extend the score distillation process by introducing the entire convergence trajectory of the teacher model and propose **Dis**tribution **Back**tracking Distillation (**DisBack**). DisBask is composed of two stages: *Degradation Recording* and *Distribution Backtracking*. *Degradation Recording* is designed to obtain the convergence trajectory by recording the degradation path from the pre-trained teacher model to the untrained student generator. The degradation path implicitly represents the intermediate distributions between the teacher and the student, and its reverse can be viewed as the convergence trajectory from the student generator to the teacher model. Then *Distribution Backtracking* trains the student generator to backtrack the intermediate distributions along the path to approximate the convergence trajectory of the teacher model. Extensive experiments show that DisBack achieves faster and better convergence than the existing distillation method and achieves comparable or better generation performance, with an FID score of 1.38 on the ImageNet 64×64 dataset. DisBack is easy to implement and can be generalized to existing distillation methods to boost performance.

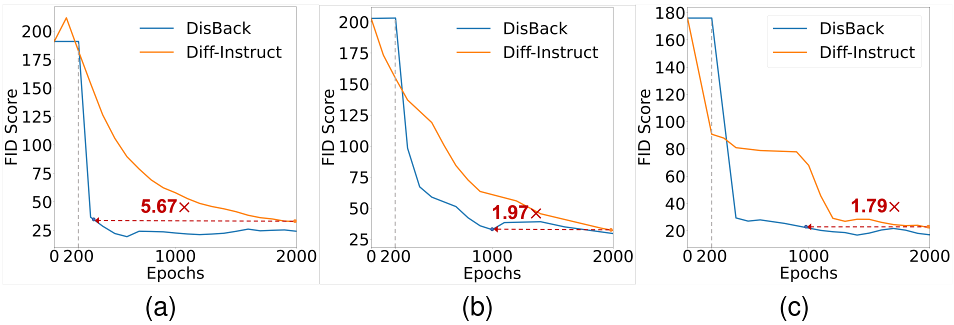


Figure 1: The comparison of the distillation process between existing SOTA score distillation method Diff-Instruct (Luo et al., 2023c) and our proposed DisBack on (a) CIFAR10, (b) FFHQ 64x64, and (c) ImageNet 64x64 datasets. The first 200 epochs refer to the computational overhead of the degradation recording stage of the proposed model. DisBack achieves a faster convergence speed due to the constraint of the entire convergence trajectory between the student generator and the teacher model.

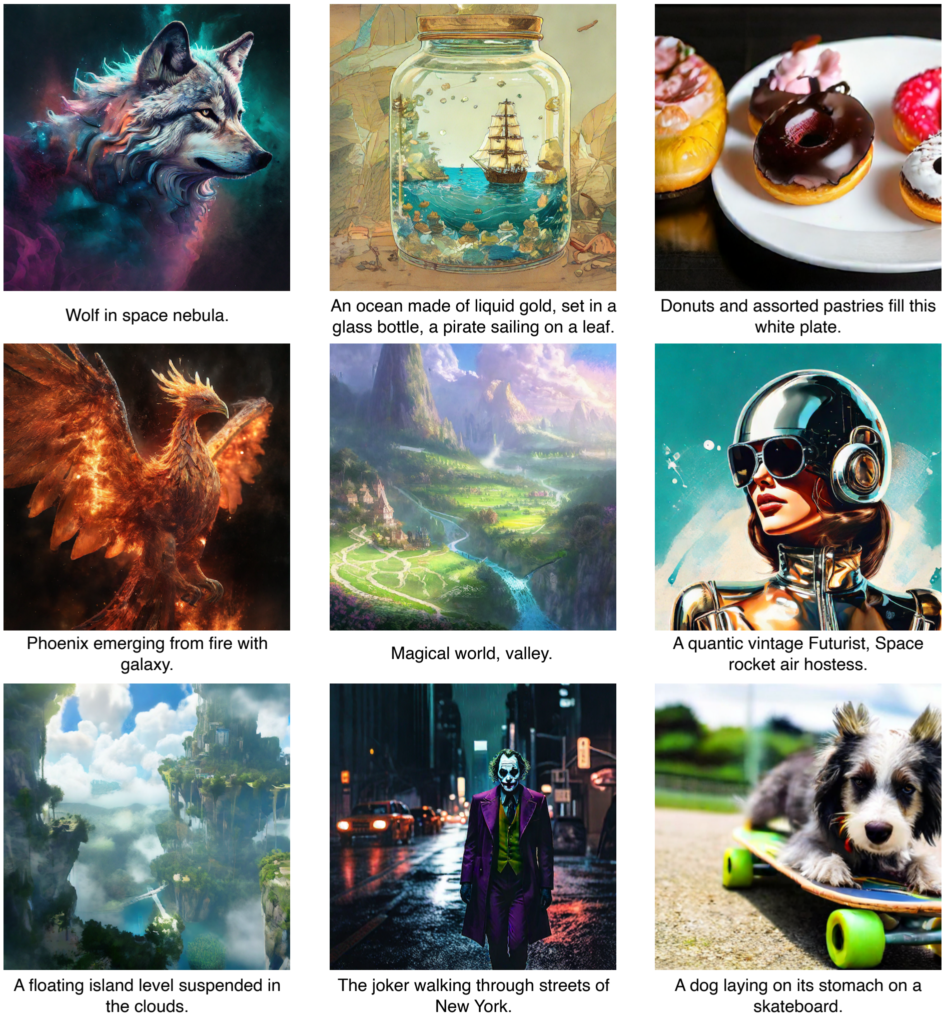
1 Introduction

Recently, generative models have demonstrated remarkable performance across diverse domains such as images (Kou et al., 2023; Yin et al., 2024a), audio (Evans et al., 2024; Xing et al., 2024), and videos (Wang et al., 2024; Chen et al., 2024). However, existing models still grapple with the "trilemma" problem, wherein they struggle to simultaneously achieve high generation quality, fast generation speed, and high sample diversity (Xiao et al., 2021). Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) can rapidly produce high-quality samples but often face mode collapse issues. Variational Autoencoders (VAEs) (Kingma & Welling, 2014) offer stable training but tend to yield lower-quality samples. Recently, Diffusion models (DMs) have emerged as a competitive contender in the generative model landscape (Fan et al., 2023; Zhou et al., 2023; Xu et al., 2024). Diffusion models can generate high-quality, diverse samples but still suffer from slow sampling speeds due to iterative network evaluations.

To accelerate the sampling speed, the score distillation method tries to distill a heavy teacher model to a student generator to reduce the sampling cost and achieve the one-step generation (Bao et al., 2023; Luo et al., 2023c; Yin et al., 2024b). The score distillation method optimizes the student generator by calculating the difference between two score functions on the samples generated by the student generator. However, as the generated distribution is far from the training distribution at the beginning, the generated sample lies outside the training data distribution. Thus, the predicted score of the generated sample from the teacher model does not match the sample's real score in the training distribution. This mismatch issue is reflected by unreliable network predictions of the teacher model, which prevents the student model from receiving accurate guidance and leads to a decline in final generative performance. We identified that this issue arises because existing score distillation methods mainly focus on using the endpoint of the pre-trained diffusion model as the teacher model, overlooking the importance of the convergence trajectory between the student generator and the teacher model. Without the constraint of the convergence trajectory, the mismatch issue causes the student generator to deviate from a reasonable optimization path during training, leading to convergence to suboptimal solutions and a decline in final performance.

To address this problem, we extend the score distillation process by introducing the entire convergence trajectory of the teacher model and propose **Dis**tribution **Back**tracking Distillation (**DisBack**) for a faster and more efficient distillation. The construction of DisBack is based on the following insights. In practice, the convergence trajectory of most teacher models is inaccessible, particularly for large models like Stable Diffusion (Rombach et al., 2022). Because the trajectory of distribution changes is bidirectional, it is possible to construct a degradation path from the teacher model to the initial student generator, and the reverse of this path can be viewed as the convergence trajectory of the teacher model. Compared with fitting the teacher model directly, fitting intermediate targets along the convergence trajectory can mitigate the mismatch issue. Thus, the DisBack incorporates degradation recording and distribution backtracking stages. In the degradation recording stage, the teacher model is tuned to fit the distribution of the initial student generator and obtains a distribution degradation path. The path includes a series of in-between diffusion models to represent the intermediate distributions of the teacher model implicitly. In the distribution backtracking stage, the degradation path is reversed and viewed as the convergence trajectory. Then the student generator is trained to backtrack the intermediate distributions along the path to optimize towards the convergence trajectory of the teacher model. In practice, the degradation recording stage typically requires only a few hundred iterations. Therefore, the proposed method incurs trivial additional computational costs. Compared to the existing score distillation method, DisBack exhibits a significantly increased convergence speed (Fig. 1), and it also delivers superior generation performance (Fig. 2).

Our main contributions are summarized as follows. (1) We extend the score distillation process by introducing the constraint of the entire convergence trajectory of the teacher model and propose Distribution Backtracking Distillation (DisBack), which achieves a faster and more efficient distillation (Sec. 4). (2) Extensive experiments demonstrate that the proposed DisBack accelerates the convergence speed of the score distillation process while achieving comparable or better generation quality compared to existing methods (Sec. 5). (3) The contribution of DisBack is orthogonal to those of other distillation methods. Researchers are encouraged to incorporate our DisBack training strategy into their distillation methods.



Wolf in space nebula.

An ocean made of liquid gold, set in a glass bottle, a pirate sailing on a leaf.

Donuts and assorted pastries fill this white plate.

Phoenix emerging from fire with galaxy.

Magical world, valley.

A quantic vintage Futurist, Space rocket air hostess.

A floating island level suspended in the clouds.

The joker walking through streets of New York.

A dog laying on its stomach on a skateboard.

Figure 2: Several examples of 1024×1024 images generated by our proposed one-step DisBack model distilled from SDXL (Podell et al., 2024).

2 Related Works

Efficient diffusion models. To improve the efficiency of the diffusion model, existing methods use the knowledge distillation method to distill a large teacher diffusion model to a small and efficient student diffusion model (Yang et al., 2022). The progressive distillation model (Salimans & Ho, 2021) progressively distills the entire sampling process into a new diffusion model with half the number of steps iteratively. Building on this, the classifier-guided distillation model (Sun et al., 2023) introduces a dataset-independent classifier to focus the student model on the crucial features to enhance the distillation process. Guided-distillation model (Meng et al., 2023) proposes a classifier-free guiding framework to avoid the computational cost of additional classifiers and achieve high-quality sampling in only 2-4 steps. Recently, the Consistency Model (Song et al., 2023) uses the self-consistency of the ODE generation process to achieve one-step distillation, but this is at the expense of generation quality. To mitigate the surface of the sample quality caused by the acceleration, the Consistency Trajectory Model (Kim et al., 2024a) combines the adversarial training and denoising score matching loss to further improve the performance. Latent Adversarial Diffusion Distillation (Sauer et al., 2024) leverages generative features from pre-trained latent diffusion models to achieve high-resolution, multi-aspect ratio, few-step image generation.

Score distillation for one-step generation. Diff-Instruct (Luo et al., 2023c) proposes a distillation method from the pre-trained diffusion model to the one-step generator that involves optimizing the generator by the gradient of the difference between two score functions. One score function represents the pre-trained diffusion distribution, while the other represents the generated distribution. Adversarial Score Distillation (Wei et al., 2024) further employs the paradigm of WGAN and retains an optimizable discriminator to improve performance. Additionally, Swiftbrush (Nguyen & Tran, 2024) leverages score distillation to distill a Stable Diffusion v2.1 into a one-step text-to-image generation model and achieve competitive results. DMD (Yin et al., 2024b) suggests the inclusion of a regression loss between noisy images and corresponding outputs to alleviate instability in the distillation process in text-to-image generation tasks. DMD2 (Yin et al., 2024a) introduces a two-time-scale update rule and an additional GAN loss to address the issue of generation quality being limited by the teacher model in DMD, achieving superior performance. Recently, HyperSD (Ren et al., 2024) integrates score distillation with trajectory segmented consistency distillation and human feedback learning, which achieves SOTA performance from 1 to 8 inference steps.

Mismatch issues in diffusion model. Existing studies have already discussed the issue of score mismatch in diffusion models Chao et al. (2022); Kim et al. (2024b). DLSM Chao et al. (2022) deeply analyzed the mismatch between the posterior score estimated by diffusion models and the true likelihood score in conditional generation and proposed the Denoising Likelihood Score Matching loss to train classifiers for more accurate conditional score estimation. TIW-DSM Kim et al. (2024b) highlighted that dataset biases cause mismatches between the generator's and true data distributions, introducing time-varying importance weights and score correction terms to mitigate this issue by assigning different weights to data points at each time step and correcting the scores. This paper focuses on the mismatch arising from the disparity between the initial generator's distribution and the teacher model's training distribution, causing discrepancies between predicted and true scores.

3 PRELIMINARY

In this part, we briefly introduce the score distillation approach. Let q_0^G and q_t^G be the distribution of the student generator G_{stu} and its noisy distribution at timestep t. In addition, q_0 and q_t are the training distribution and its noisy distribution at timestep t. By optimizing the KL divergence in Eq. (1), we can train a student generator to enable one-step generation (Wang et al., 2023).

$$\min_{\eta} D_{KL}\left(q_0^G\left(\boldsymbol{x}_0\right) \| q_0\left(\boldsymbol{x}_0\right)\right) \tag{1}$$

Here $\boldsymbol{x}_0 = G_{stu}(\boldsymbol{z};\eta)$ is the generated samples, and η is the trainable parameter of G_{stu}. However, due to the complexity of q_0 and its sparsity in high-density regions, directly solving Eq.(1) is challenging (Song & Ermon, 2019). Inspired by Variational Score Distillation (VSD) (Wang et al., 2023), Eq.(1) can be extended to optimization problems at different timesteps t in Eq. (2). As t increases, the diffusion distribution becomes closer to a Gaussian distribution.

$$\min_{\eta} \mathbb{E}_{t\sim\mathcal{U}(0,1),\boldsymbol{\epsilon}\sim\mathcal{N}(0,I)} D_{KL}\left(q_t^G\left(\boldsymbol{x}_t\right) \| q_t\left(\boldsymbol{x}_t\right)\right) \tag{2}$$

Here $\boldsymbol{x}_t$ is the noisy data and $\boldsymbol{x}_t \mid \boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{x}_0, \sigma_t^2 I)$. Theorem 1 proves that introducing the additional KL-divergence for $t > 0$ does not affect the global optimum of the original optimization problem in Eq.(1).

Theorem 1 (The global optimum of training (Wang et al., 2023)) *Given $t > 0$, we have,*

$$D_{\mathrm{KL}}\left(q_t^G\left(\boldsymbol{x}_t\right) \| q_t\left(\boldsymbol{x}_t\right)\right) = 0 \Leftrightarrow D_{\mathrm{KL}}\left(q_0^G\left(\boldsymbol{x}_0\right) \| q_0\left(\boldsymbol{x}_0\right)\right) = 0 \tag{3}$$

Therefore, by minimizing the KL divergence in Eq. (2), the student generator can be optimized through the following gradients:

$$\nabla_\eta D_{\mathrm{KL}}\left(q_t^G\left(\boldsymbol{x}_t\right) \| q_t\left(\boldsymbol{x}_t\right)\right) = \mathbb{E}_{t,\boldsymbol{\epsilon}}\left[\left[\nabla_{\boldsymbol{x}_t} \log q_t^G\left(\boldsymbol{x}_t\right) - \nabla_{\boldsymbol{x}_t} \log q_t\left(\boldsymbol{x}_t\right)\right] \frac{\partial \boldsymbol{x}_t}{\partial \eta}\right] \tag{4}$$

Here the score of perturbed training data $\nabla_{\boldsymbol{x}_t} \log q_t\left(\boldsymbol{x}_t\right)$ can be approximated by a pre-trained diffusion model s_θ. The score of perturbed generated data $\nabla_{\boldsymbol{x}_t} \log q_t^G\left(\boldsymbol{x}_t\right)$ is estimated by another diffusion model s_ϕ, which is optimized by score matching with generated data (Song et al., 2021b):

$$\min_{\phi} \mathbb{E}_{t,\boldsymbol{\epsilon}} \left\| s_\phi\left(\boldsymbol{x}_t, t\right) - \frac{\boldsymbol{x}_0 - \boldsymbol{x}_t}{\sigma_t^2} \right\|_2^2 \tag{5}$$

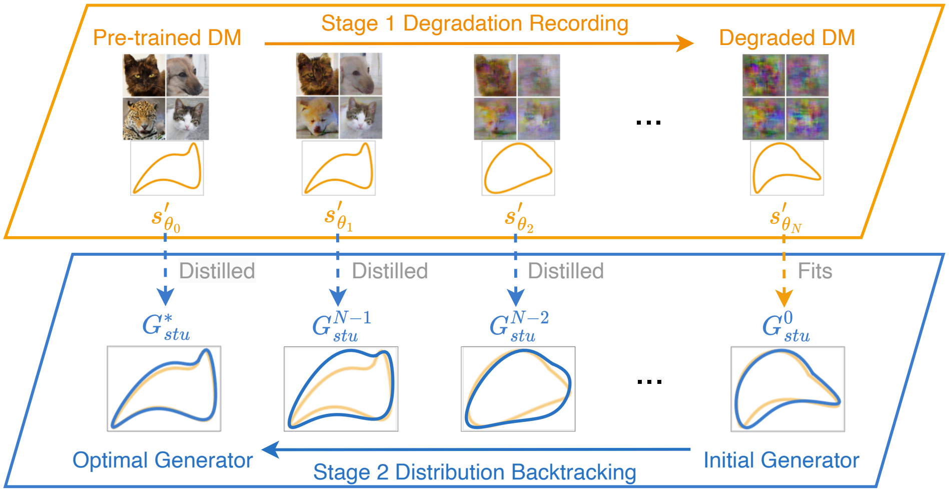


Figure 3: The overall framework of DisBack. Stage 1: An auxiliary diffusion model is initialized with the teacher model s_θ and then fits the distribution of the initial student generator G^0_{stu}. The intermediate checkpoints $\{s'_{\theta_i} \mid i = 0, \dots, N\}$ are saved to form a degradation path. The degradation path is then reversed and viewed as the convergence trajectory. Stage 2: The intermediate node s_{θ_i} along the convergence trajectory is distilled to the student generator sequentially until the generator converges to the distribution of the teacher model.

Thus, the gradient of student generator in Eq.(4) is estimated as

$$\nabla_\eta D_{\mathrm{KL}}\left(q_t^G\left(\boldsymbol{x}_t\right)\|q_t\left(\boldsymbol{x}_t\right)\right) \approx \mathbb{E}_{t,\boldsymbol{\epsilon}}\left[\left[s_\phi\left(\boldsymbol{x}_t,t\right)-s_\theta\left(\boldsymbol{x}_t,t\right)\right]\frac{\partial \boldsymbol{x}_t}{\partial \eta}\right] \tag{6}$$

The distribution of the student generator changes after its update. Therefore, s_ϕ also needs to be optimized based on the newly generated images to ensure the timely approximation of the generated distribution. Thus, the student generator and s_ϕ are optimized alternately.

In practice, s_ϕ has three initialization strategies: (1) s_ϕ is randomly initialized (Franceschi et al., 2023). (2) s_ϕ is initialized as s_θ or its LoRA (Hu et al., 2021; Wei et al., 2024). (3) s_ϕ is initialized by fitting the generated samples of student generator (Luo et al., 2023c). Beyond unconditional image generation (Ye & Liu, 2023), this method has also been applied to tasks such as text-to-image and image-to-image generation across various structures (Yin et al., 2024b; Hertz et al., 2023).

4 Method

4.1 Insight

In this section, we introduce the **Dis**tribution **Back**tracking Distillation (**DisBack**). The key insight behind DisBack is the importance of the convergence trajectory. As mentioned in Sec.3, there are two score functions in score distillation, one representing the pre-trained diffusion distribution and the other representing the generated distribution. The student model is optimized using the gradient of the difference between these two score functions. Existing methods (Luo et al., 2023c; Yin et al., 2024b;a) directly use the endpoint of the pre-trained diffusion model as the teacher model, overlooking the intermediate convergence trajectory between the student generator and the teacher model. The resulting score mismatch issue between the predicted scores of the generated sample from the teacher model and the real scores causes the student model to receive inaccurate guidance. It ultimately leads to a decline in final performance. Constraining the convergence trajectory between the student generator and the teacher model during the distillation process can mitigate the mismatch issue and help the student generator approximate the convergence trajectory of teacher models to achieve faster convergence. In practice, it is infeasible to obtain the convergence trajectory of most teacher models, especially for large models such as Stable Diffusion (Rombach et al., 2022). Reversely, it is possible to obtain the degradation path from the teacher model to the initial student generator. The reverse of this degradation path can be viewed as the convergence trajectory of the teacher model. Based on the above insights, we structure the proposed DisBack in two stages including the degradation recording stage and the distribution backtracking stage (Fig. 3).

4.2 Degradation Recording

This stage aims to obtain the degradation path from the teacher model to the initial student generator. The degradation path is then reversed and viewed as the convergence trajectory of the teacher model. The teacher model here is the pre-trained diffusion model s_θ and the student generator is represented by G^0_{stu}.

Let s'_θ be a diffusion model initialized with the teacher model s_θ, and it is trained on generated samples to fit the initial student generator's distribution q^G_0 with Eq. (7). By saving the multiple intermediate checkpoints during the training, we can obtain a series of diffusion models $\{s'_{\theta_i} \mid i = 0, \ldots, N\}$, where $s'_{\theta_0} = s_\theta \approx q_0$ and $s'_{\theta_N} \approx q^G_0$. These diffusion models describe the scores of non-existent distributions on the path, recording how the training distribution q_0 degrades to the initial generated distribution q^G_0. Algorithm 1 shows the process of obtaining the degradation path. Since distribution degradation is easily achievable, the degradation recording stage only needs trivial additional computational resources (200 iterations in most cases).

$$\min_\theta \mathbb{E}_{t,\boldsymbol{\epsilon}} \left\| s'_\theta\left(\boldsymbol{x}_t, t\right) - \frac{\boldsymbol{x}_0 - \boldsymbol{x}_t}{\sigma_t^2} \right\|_2^2 \tag{7}$$

Algorithm 1 Degradation Recording.

```
Input: Initial student generator $G^0_{stu}$ and pre-trained diffusion model $s_\theta$.

Output: Degradation path checkpoints $\{s'_{\theta_i} \mid i = 0, \ldots, N\}$
$s'_\theta \leftarrow s_\theta$

while not converge do
    $\boldsymbol{x}_0 = G^0_{stu}(\boldsymbol{z}; \eta)$

    Update $\theta$ with gradient
        $\frac{\partial}{\partial \theta_i} \mathbb{E}_{t,\boldsymbol{\epsilon}} \left\| s'_\theta\left(\boldsymbol{x}_t, t\right) - \frac{\boldsymbol{x}_0 - \boldsymbol{x}_t}{\sigma_t^2} \right\|_2^2$

    Save intermediate checkpoints $s'_{\theta_i}$
end while
```

Algorithm 2 Distribution Backtracking.

```
Input: Initial student generator $G^0_{stu}$ and reverse path checkpoints $\{s'_{\theta_i} \mid i = N, \ldots, 0\}$

Output: One-step generator $G^*_{stu}$
$s_\phi \leftarrow s'_{\theta_N}$
for $i \leftarrow N - 1$ to 0 do
    while not converge do
        $\boldsymbol{x}_0 = G^0_{stu}(\boldsymbol{z}; \eta)$

        Update $\eta$ with gradient
            $\mathbb{E}_{t,\boldsymbol{\epsilon}} \left[ s_\phi\left(\boldsymbol{x}_t, t\right) - s'_{\theta_i}\left(\boldsymbol{x}_t, t\right) \right] \frac{\partial \boldsymbol{x}_t}{\partial \eta}$
        Update $\phi$ with gradient
            $\frac{\partial}{\partial \phi} \mathbb{E}_{t,\boldsymbol{\epsilon}} \left\| s_\phi\left(\boldsymbol{x}_t, t\right) - \frac{\boldsymbol{x}_0 - \boldsymbol{x}_t}{\sigma_t^2} \right\|_2^2$
    end while
end for
```

4.3 Distribution Backtracking

Given the degradation path from the teacher model to the initial student generator, the reverse path is viewed as a representation of the convergence trajectory between the initial student generator G^0_{stu} and the teacher model s_θ. The key to the distribution backtracking is to sequentially distill checkpoints in the convergence trajectory into the student generator. The last node s'_{θ_N} in the path is close to the initially generated distribution q^G_0. Therefore, in the distribution backtracking stage, we use $s'_{\theta_{N-1}}$ as the first target to distill the student generator. When near convergence, we switch the target to $s'_{\theta_{N-2}}$. The checkpoints s'_{θ_i} is sequentially distilled to G_{stu} until the final target s'_{θ_0} is reached. During the distillation, the gradient of G_{stu} is:

$$\text{Grad}(\eta) = \mathbb{E}_{t,\boldsymbol{\epsilon}} \left[\left[s_\phi\left(\boldsymbol{x}_t, t\right) - s'_{\theta_i}\left(\boldsymbol{x}_t, t\right) \right] \frac{\partial \boldsymbol{x}_t}{\partial \eta} \right] \tag{8}$$

In this stage, G_{stu} and s_ϕ are also optimized alternately and the optimization of s_ϕ is the same as in the original score distillation (Eq. 5). Compared to existing score distillation methods, the final

Table 1: The unconditional generation performance of DisBack. The FID (↓) scores are shown.

Model	NFE (↓)	FFHQ	AFHQv2	LSUN-bedroom	LSUN-cat
DDPM (Ho et al., 2020)	1000	3.52	-	4.89	17.10
ADM (Dhariwal & Nichol, 2021)	1000	-	-	1.90	5.57
NSCN++ (Song et al., 2021b)	79	25.95	18.52	-	-
DDPM++ (Song et al., 2021b)	79	3.39	2.58	-	-
EDM (Karras et al., 2022)	79	2.39	1.96	3.57	6.69
EDM (Karras et al., 2022)	15	15.81	13.67	-	-
Diff-Instruct (Luo et al., 2023c)	1	19.93	-	-	-
PD (Salimans & Ho, 2021)	1	-	-	16.92	29.60
CT (Song et al., 2023)	1	-	-	16.00	20.70
CD (Song et al., 2023)	1	12.58	10.75	7.80	11.00
DisBack	1	**10.88**	**9.97**	**6.99**	**10.30**

target of DisBack is the same while constraining the convergence trajectory to achieve more efficient distillation of the student generator. Algorithm 2 summarizes the distribution backtracking stage.

5 Experiment

Experiments are conducted on different models across various datasets. We first compare the performance of DisBack with other multi-step diffusion models and distillation methods (Sec. 5.1). Secondly, we compare the convergence speed of DisBack with its variants without the constraint of the convergence trajectory (Sec. 5.2). Thirdly, further experiments are conducted to demonstrate DisBack's effectiveness in mitigating the score mismatch issues (Sec. 5.3). Then, we also conduct the ablation study to show the effectiveness of introducing the convergence trajectory (Sec. 5.4). Finally, we show the results of DisBack on text-to-image generation tasks (Sec. 5.5).

5.1 Quantitative evaluation

DisBack can achieve performance comparable to or even better than the existing diffusion models or distillation methods. Experiments are conducted on different datasets. (1) The unconditional generation on FFHQ 64x64, AFHQv2 64x64, LSUN-bedroom 256x256 and LSUN cat 256x256. (2) The conditional generation on ImageNet 64x64. The performance of DisBack is shown in Tab. 1 and Tab. 2. All the DisBack models are distilled from the pre-trained EDM model (Karras et al., 2022).

Table 2: The conditional generation performance of DisBack on ImageNet 64x64 dataset.

Model	NFE (↓)	FID (↓)
DDPM (Ho et al., 2020)	1000	3.77
DDDM (Zhang et al., 2024)	1000	2.11
EDM (Karras et al., 2022)	511	1.36
EDM (Karras et al., 2022)	15	10.46
Moment Matching (Salimans et al., 2024)	8	3.3
SlimFlow (Zhu et al., 2024)	1	12.34
BOOT (Gu et al., 2024)	1	12.30
DDDM (Zhang et al., 2024)	1	3.47
CTM (Kim et al., 2024a)	1	2.06
Sid (Zhou et al., 2024)	1	1.52
DMD2 (Yin et al., 2024a)	1	1.51
Diff-Instruct (Luo et al., 2023c)	1	5.57
PD (Salimans & Ho, 2021)	1	8.95
CT (Song et al., 2023)	1	13.00
CD (Song et al., 2023)	1	6.20
DisBack	1	**1.38**

For unconditional generation, the one-step generator distilled by the DisBack achieves comparable performance across different datasets compared to multi-step generation diffusion models. Specifically, it outperforms the original EDM model with 15 NFEs (10.88 of DisBack and 15.81 of EDM on FFHQ64). Compared to existing one-step generators and distillation methods, DisBack achieves optimal performance. For conditional generation, the DisBack achieves the best performance compared to the existing models. Moreover, DisBack requires no training data and

Table 3: Ablation study on constraining the convergence trajectory to the score distillation process. The FID (↓) scores in each case are shown.

Model	FFHQ	AHFQv2	ImageNet	LSUN-bedroom	LSUN-cat
DisBack	**10.88**	**9.97**	**1.38**	**6.99**	**10.30**
w/o Convergence Trajectory	12.26	10.29	5.96	7.43	10.63

additional constraints during training. In conclusion, DisBack can achieve competitive distillation performance compared to existing models.

5.2 Convergence speed

We conducted a series of experiments to demonstrate the advantages of DisBack in accelerating the convergence speed of the score distillation process on unconditional CIFAR10 (Krizhevsky, 2009), FFHQ 64x64 (Karras et al., 2019), and conditional ImageNet 64x64 (Deng et al., 2009) datasets. Diff-Instruct (Luo et al., 2023c) is the existing SOTA score distillation method, which can be regarded as a variation of DisBack not introducing the convergence trajectory. We compared the FID trends of DisBack and Diff-Instruct during the distillation process in the same situation.

The results are shown in Fig. 1. As for unconditional generation, DisBack achieves a convergence speed 2.46 times faster than the variant without the constraint of convergence trajectory on the FFHQ 64x64 dataset and 13.09 times faster on the CIFAR10 dataset. For the conditional generation on the ImageNet 64x64 dataset, DisBack is 2.19 times faster than the variant without the constraint of convergence trajectory. The fast convergence speed is because constraining the convergence trajectory of the generator provides a clear optimization direction, avoiding the generator falling into suboptimal solutions and enabling faster convergence to the target distribution.

5.3 Experiments on score mismatch issue

In this part, experiments are conducted to validate the positive impact of constraining the convergence trajectory on mitigating the mismatch issues. We propose a new metric called mismatch degree to assess whether the predicted score of the teacher model matches the distribution's real score given a data distribution. This score is inspired by the score-matching loss.

$$d_{mis} = \mathbb{E}_{\boldsymbol{x}_0 \sim q_0^G} \mathbb{E}_{\boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{x}_0, \sigma_t)} \| s_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t) \|_2 \tag{9}$$

Here $\boldsymbol{x}_t$ is the noisy data from the assessed distribution. Besides, $s_\theta(\boldsymbol{x}_t, t)$ represents the predicted score and $\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)$ represents the real score. As shown in Eq. (4) and (6), the closer $s_\theta(\boldsymbol{x}_t, t)$ is to $\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)$, the more accurate the gradient estimation for the student generator obtains. Because the real scores are not available in practice, we use Stable Target Field (STF) (Xu et al., 2022) to approximate the real score $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ on the assessed distribution. STF estimation leverages reference batches to reduce the variance of training objectives, which has been proven to yield accurate asymptotically unbiased estimates of the real score (Xu et al., 2022).

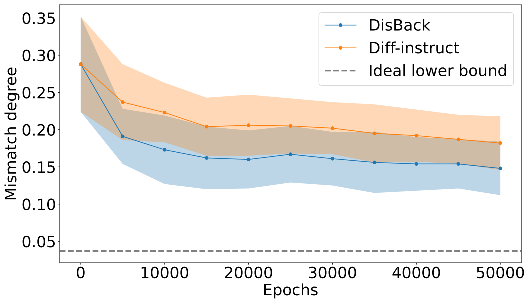
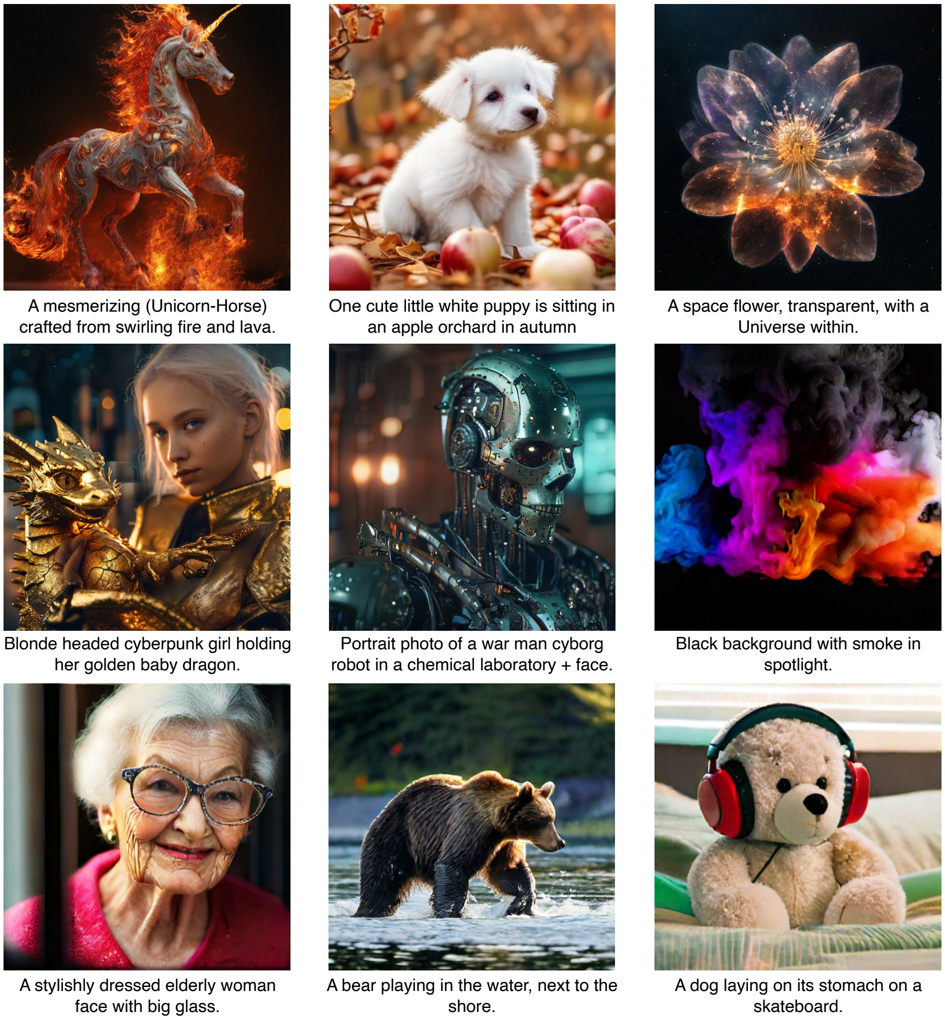


Figure 4: The mismatch degree during the distillation process of Diff-Instruct and proposed DisBack. The standard deviation is visualized. DisBack effectively mitigates the mismatch degree during the entire distillation process.

When the assessed distribution is close to the distribution of the teacher model s_θ, the mismatch degree is small, and vice versa. When calculated directly on the training data, the resulting mismatch degree represents the ideal lower bound. Therefore, the mismatch degree can be used to assess the convergence degree of the generated distribution during the distillation process and visualize the convergence speed under the constraint of the convergence trajectory.



A mesmerizing (Unicorn-Horse) crafted from swirling fire and lava.

One cute little white puppy is sitting in an apple orchard in autumn

A space flower, transparent, with a Universe within.

Blonde headed cyberpunk girl holding her golden baby dragon.

Portrait photo of a war man cyborg robot in a chemical laboratory + face.

Black background with smoke in spotlight.

A stylishly dressed elderly woman face with big glass.

A bear playing in the water, next to the shore.

A dog laying on its stomach on a skateboard.

Figure 5: Generation samples by DisBack distilled from SDXL with 1024×1024 resolution.

We conduct experiments on the FFHQ 64x64 dataset with Diff-Instruct (Luo et al., 2023c) as a baseline. We calculate the mismatch degree on the distribution of the student generator of both Diff-Instruct and the proposed DisBack. The pre-trained EDM model is chosen as the teacher model. In this scenario, the ideal lower bound of the mismatch degree is 0.037. We visualized the mismatch degree in Fig. 4. With the constraining of the convergence trajectory, the mismatch degree of the proposed DisBack is lower during the distillation process, meaning the student generator converges faster and better. Thus, by constraining the convergence trajectory, the mismatch issue can be mitigated and DisBack can achieve more efficient distillation.

5.4 Ablation study

Ablation studies are conducted to compare the performance of DisBack with its variant without the constraint of the convergence trajectory. The results are shown in Tab. 3. Results show that the variant without the constraint of convergence trajectory suffers from a performance decay in different cases. This confirms the efficacy of constraining the convergence trajectory between the student generator and the teacher model can improve the final performance of the generation.

5.5 Text to Image Generation

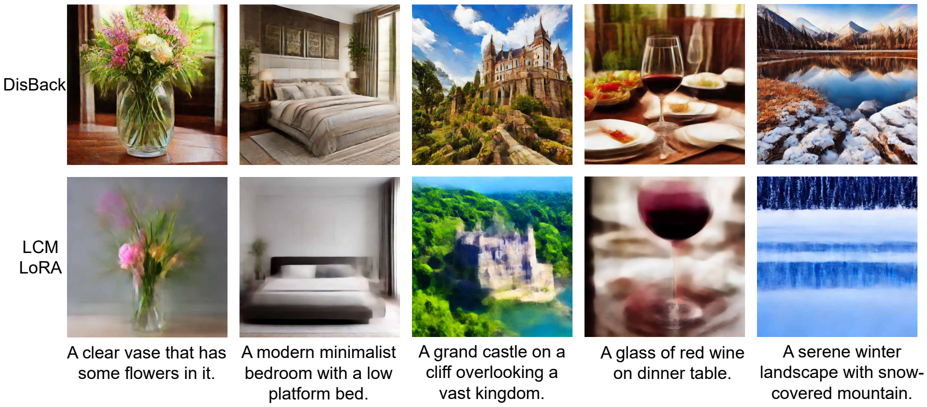

DisBack
LCM LoRA
A clear vase that has some flowers in it.
A modern minimalist bedroom with a low platform bed.
A grand castle on a cliff overlooking a vast kingdom.
A glass of red wine on dinner table.
A serene winter landscape with snow-covered mountain.

Figure 6: One step generation samples by original LCM-LoRA and its variant distilled from SD v1.5 with DisBack in 512×512. LCM-LoRA with DisBack can generate images with higher quality.

Table 4: The results of text-to-image generation.

Model	NFE (↓)	FID (↓)	Patch-FID (↓)	CLIP (↑)
LCM-SDXL Luo et al. (2023b)	1	81.62	154.40	0.275
LCM-SDXL Luo et al. (2023b)	4	22.16	33.92	0.317
DMD2 Yin et al. (2024a)	1	19.01	26.98	0.336
DMD2 Yin et al. (2024a)	4	19.32	**20.86**	0.332
SDXL-Turbo Sauer et al. (2025)	1	24.57	23.94	**0.337**
SDXL-Turbo Sauer et al. (2025)	4	23.19	23.27	0.334
SDXL Lightning Lin et al. (2024)	1	23.92	31.65	0.316
SDXL Lightning Lin et al. (2024)	4	24.46	24.56	0.323
SDXL Podell et al. (2024)	100	19.36	21.38	0.332
DisBack	1	**18.96**	26.89	0.335

Further experiments are conducted on text-to-image generation tasks. We use DisBack to distill the SDXL model (Podell et al., 2024) and evaluate the FID scores of the distilled SDXL and the original SDXL on the COCO 2014 (Radford et al., 2021). The user studies are conducted to verify the effectiveness of DisBack. We randomly select 128 prompts from the LAION-Aesthetics (Schuhmann et al., 2022) to generate images and ask volunteer participants to choose the images they think are better. Detailed information about the user study is included in Sec. B.3. The results of the FID evaluation and user study are presented in Tab. 4. DisBack achieved optimal FID and comparable CLIP scores in single-step generation compared to baselines, but the Patch-FID showed a slight decay. The preference scores of DisBack over the original SDXL are 61.3%. Some generation samples are shown in Fig. 2 and Fig. 5.

We also conducted experiments on LCM-LoRA (Luo et al., 2023b). The LCM-LoRA distilled from SDv1.5 using DisBack has an FID score of 36.37 on one-step generation, while the FID score of the original LCM-LoRA is 78.26. Some generated samples of DisBack and original LCM-LoRA are shown in Fig. 6. The details of experiments and results are provided in Sec. A.1.

6 Conclusion

Summary. This paper proposes Distribution Backtracking Distillation (DisBack) to introduce the entire convergence trajectory of the teacher model in the score distillation. The DisBack can also be used to distill large-scale text-to-image models. DisBack performs a faster and more efficient distillation and achieves a comparable or better performance in one-step generation compared to existing multi-step generation diffusion models and one-step diffusion distillation models.

Limitation. The performance of DisBack is inherently limited by the teacher model. The better the original performance of the teacher model, the better the performance of DisBack will also be. Additionally, to achieve optimal performances in both accelerated distillation and generation quality, DisBack requires careful design of the distribution degradation path and the setting of various hyperparameters (such as how many epochs are used to fit each intermediate node in distribution backtracking stage). While with no meticulous design, it can also achieve better performance, further exploration is required to enable the model to reach optimal performance.

REFERENCES

Zhiqiang Bao, Zihao Chen, Changdong Wang, Wei-Shi Zheng, Zhenhua Huang, and Yunwen Chen. Post-distillation via Neural Resuscitation. *IEEE Transactions on Multimedia*, pp. 3046 – 3060, 2023.

Chen-Hao Chao, Wei-Fang Sun, Bo-Wun Cheng, Yi-Chen Lo, Chia-Che Chang, Yu-Lun Liu, Yu-Lin Chang, Chia-Ping Chen, and Chun-Yi Lee. Denoising Likelihood Score Matching for Conditional Score-based Data Generation. *arXiv preprint arXiv:2203.14206*, 2022.

Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7310–7320, 2024.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8188–8197, 2020.

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*, pp. 248–255. Ieee, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, pp. 8780–8794, 2021.

Zach Evans, CJ Carr, Josiah Taylor, Scott H Hawley, and Jordi Pons. Fast timing-conditioned latent audio diffusion. *arXiv preprint arXiv:2402.04825*, 2024.

Wenqi Fan, Chengyi Liu, Yunqing Liu, Jiatong Li, Hang Li, Hui Liu, Jiliang Tang, and Qing Li. Generative diffusion models on graphs: Methods and applications. In *International Joint Conference on Artificial Intelligence*, pp. 6702–6711, 2023.

Jean-Yves Franceschi, Mike Gartrell, Ludovic Dos Santos, Thibaut Issenhuth, Emmanuel de Bézenac, Mickaël Chen, and Alain Rakotomamonjy. Unifying GANs and Score-Based Diffusion as Generative Particle Models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 59729–59760, 2023.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Jiatao Gu, Chen Wang, Shuangfei Zhai, Yizhe Zhang, Lingjie Liu, and Joshua M Susskind. Data-free Distillation of Diffusion Models with Bootstrapping. In *International Conference on Machine Learning*, 2024.

Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. Delta denoising score. In *International Conference on Computer Vision*, pp. 2328–2337, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2021.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119, 2020.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Advances in Neural Information Processing Systems*, pp. 852–863, 2021.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, pp. 26565–26577, 2022.

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. In *International Conference on Learning Representations*, 2024a.

Yeongmin Kim, Byeonghu Na, Minsang Park, JoonHo Jang, Dongjun Kim, Wanmo Kang, and Il-Chul Moon. Training Unbiased Diffusion Models from Biased Dataset. In *The Twelfth International Conference on Learning Representations*, 2024b.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Ziyi Kou, Shichao Pei, Yijun Tian, and Xiangliang Zhang. Character as pixels: a controllable prompt adversarial attacking framework for black-box text guided image generation models. In *International Joint Conference on Artificial Intelligence*, pp. 4912–4920, 2023.

A Krizhevsky. Learning multiple layers of features from tiny images. *University of Tront*, 2009.

Shanchuan Lin, Anran Wang, and Xiao Yang. SDXL-Lightning: Progressive Adversarial Diffusion Distillation. *arXiv preprint arXiv:2402.13929*, 2024.

Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Towards faster and stabilized GAN training for high-fidelity few-shot image synthesis. In *International Conference on Learning Representations*, 2020.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023a.

Simian Luo, Yiqin Tan, Suraj Patil, Daniel Gu, Patrick von Platen, Apolinário Passos, Longbo Huang, Jian Li, and Hang Zhao. Lcm-lora: A universal stable-diffusion acceleration module. *arXiv preprint arXiv:2311.05556*, 2023b.

Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-Instruct: A Universal Approach for Transferring Knowledge From Pre-trained Diffusion Models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 76525–76546, 2023c.

Paul C. Matthews. *Vector Calculus*. Springer undergraduate mathematics series. Springer, London, 1998. ISBN 978-3-540-76180-8. doi: doi.org/10.1007/978-1-4471-0597-8.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14297–14306, 2023.

Thuan Hoang Nguyen and Anh Tran. Swiftbrush: One-step text-to-image diffusion model with variational score distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7807–7816, 2024.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171, 2021.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

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. *arXiv preprint arXiv:2404.13686*, 2024.

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*, pp. 10684–10695, 2022.

Tim Salimans and Jonathan Ho. Progressive Distillation for Fast Sampling of Diffusion Models. In *International Conference on Learning Representations*, 2021.

Tim Salimans, Thomas Mensink, Jonathan Heek, and Emiel Hoogeboom. Multistep Distillation of Diffusion Models via Moment Matching. *arXiv preprint arXiv:2406.04103*, 2024.

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. *arXiv preprint arXiv:2403.12015*, 2024.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial Diffusion Distillation. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2025.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, 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.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*, 2021a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pp. 11895–11907, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models. In *International Conference on Machine Learning*, pp. 32211–32252, 2023.

Wujie Sun, Defang Chen, Can Wang, Deshi Ye, Yan Feng, and Chun Chen. Accelerating diffusion sampling with classifier-based feature distillation. In *International Conference on Multimedia and Expo*, pp. 810–815, 2023.

Fei-Yue Wang, Qinghai Miao, Lingxi Li, Qinghua Ni, Xuan Li, Juanjuan Li, Lili Fan, Yonglin Tian, and Qing-Long Han. When does sora show: The beginning of tao to imaginative intelligence and scenarios engineering. *IEEE/CAA Journal of Automatica Sinica*, 11(4):809–815, 2024.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8406–8441, 2023.

Min Wei, Jingkai Zhou, Junyao Sun, and Xuesong Zhang. Adversarial Score Distillation: When score distillation meets GAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the Generative Learning Trilemma with Denoising Diffusion GANs. In *International Conference on Learning Representations*, 2021.

Yazhou Xing, Yingqing He, Zeyue Tian, Xintao Wang, and Qifeng Chen. Seeing and hearing: Open-domain visual-audio generation with diffusion latent aligners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7151–7161, 2024.

Yanwu Xu, Yang Zhao, Zhisheng Xiao, and Tingbo Hou. Ufogen: You forward once large scale text-to-image generation via diffusion gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8196–8206, 2024.

Yilun Xu, Shangyuan Tong, and Tommi S Jaakkola. Stable Target Field for Reduced Variance Score Estimation in Diffusion Models. In *International Conference on Learning Representations*, 2022.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, pp. 1–39, 2022.

Senmao Ye and Fei Liu. Score Mismatching for Generative Modeling. *arXiv preprint arXiv:2309.11043*, 2023.

Mingxuan Yi, Zhanxing Zhu, and Song Liu. MonoFlow: Rethinking divergence GANs via the perspective of Wasserstein gradient flows. In *International Conference on Machine Learning*, pp. 39984–40000, 2023.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved Distribution Matching Distillation for Fast Image Synthesis. *arXiv preprint arXiv:2405.14867*, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, 2024b.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Dan Zhang, Jingjing Wang, and Feng Luo. Directly Denoising Diffusion Model. *arXiv preprint arXiv:2405.13540*, 2024.

Dewei Zhou, Zongxin Yang, and Yi Yang. Pyramid Diffusion Models For Low-light Image Enhancement. In *International Joint Conference on Artificial Intelligence*, pp. 1795–1803, 2023.

Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *International Conference on Machine Learning*, 2024.

Yuanzhi Zhu, Xingchao Liu, and Qiang Liu. SlimFlow: Training Smaller One-Step Diffusion Models with Rectified Flow. *arXiv preprint arXiv:2407.12718*, 2024.

Table 5: The performance of DisBack on the distillation from pre-trained EDM model to FastGAN on FFHQ, AFHQv2, and CelebA in the resolution of 64×64.

Model		FFHQ		AFHQv2		CelebA	
	NFE (↓)	FID (↓)	IS (↑)	FID (↓)	IS (↑)	FID (↓)	IS (↑)
FastGAN (Liu et al., 2020)	1	30.27	2.37	28.59	5.94	29.35	2.36
EDM (NFE 11) (Karras et al., 2022)	11	29.28	2.97	**13.67**	**10.86**	23.05	3.01
Score GAN (Franceschi et al., 2023)	1	43.89	2.13	53.86	2.13	50.41	2.12
DisBack	1	**23.84**	**3.27**	18.95	7.00	**23.16**	**3.02**

A More Experiment Results

A.1 Experiment Results on Text-to-Image Generation

We conducted the experiment on LCM-LoRA (Luo et al., 2023b). LCM-LoRA is a Low-Rank Adaptation (LoRA) version of the Latent Consistency Model (LCM) Luo et al. (2023a), applicable across fine-tuned Stable Diffusion models for high-quality, single-step or few-step generation. In this experiment, we use LCM-LoRA as the student generator and Stable Diffusion v1.5 as the teacher model. We observed that the score distillation underperforms when LCM-LoRA serves as the teacher model. This issue likely stems from the infeasibility of directly converting the outputs of LCM-LoRA into scores.

We distill the LCM-LoRA with the proposed DisBack and evaluate the FID scores on the COCO 2014 dataset (Radford et al., 2021) with the resolution of 512×512. 50,000 real images and 30,000 generated images were used to calculate FID scores. The 30,000 generated images were obtained by generating one image for each of the 30,000 distinct prompts. In the case of one-step generation, the original LCM-LoRA has an FID score of 78.26, while the DisBack achieves an FID of 36.37. The change in FID scores over training steps is illustrated in Fig. 7, showing that DisBack achieves a 1.5 times acceleration in convergence speed and yields superior generation performance within the same training period.

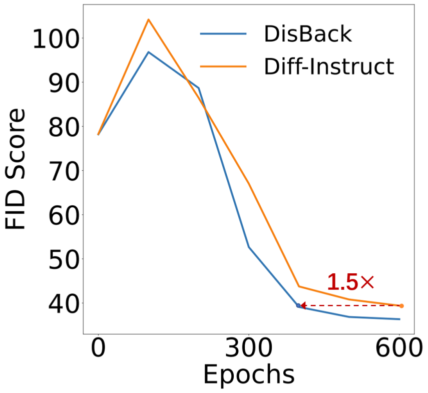


Figure 7: The FID scores of LCM-LoRA distilled from SD1.5 across training steps. DisBack achieves faster convergence and better performance.

A.2 Additional Experiment Results

To explore the distillation performance of the proposed method when the architectures of the teacher and student models differ, we opt for EDM Karras et al. (2022) as the pre-trained diffusion model, and FastGAN Liu et al. (2020) architecture for the student model to conduct the experiment. Table 5 shows the performance of baselines and the proposed models on the distillation task from a diffusion model to a generator. Compared to the original FastGAN, DisBack can effectively improve the generation quality. The results also show that the one-step sampling performance of DisBack is better than Score GAN and EDM with 11 NFEs.

A.3 Visualization of Intermediate Teacher Trajectory

To further demonstrate the effectiveness of the degradation path, we visualized the images generated by the initial generator, the intermediate checkpoints and the teacher model, along the degradation path. As shown in Fig. 8 and 9, We can observe that the images generated by the first node in the trajectory are similar to those of the initial generator, while the images generated by the last node in the trajectory are close to those of the teacher model. This is consistent with our theoretical analysis.

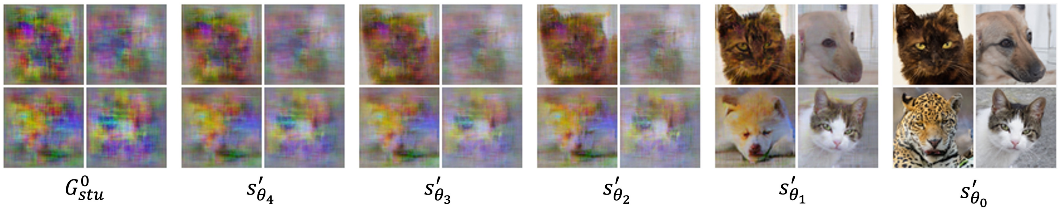


Figure 8: Samples from the initial generator, intermediate teacher trajectory nodes. Here s'_{θ_0} is the teacher model. The teacher model is the pre-trained EDM model on the FFHQ64 dataset, the student generator is FastGAN.

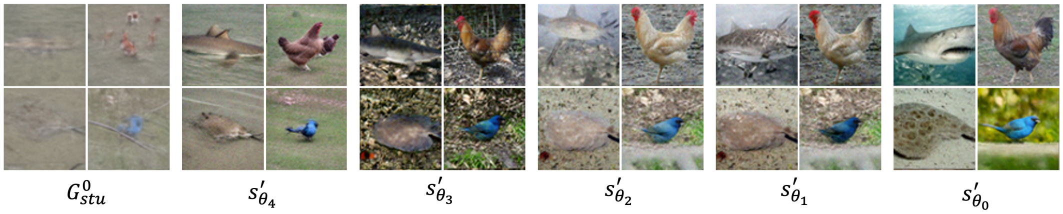


Figure 9: Samples from the initial generator, intermediate teacher trajectory nodes. Here s'_{θ_0} is the teacher model. The teacher model and the student generator are both the pre-trained EDM model on the ImageNet dataset.

B Implementation Details

B.1 Dataset Setup

We experiment on the following datasets:

The FFHQ (Flickr-Faces-HQ) dataset (Karras et al., 2019) is a high-resolution dataset of human face images used for face generation tasks. It includes high-definition face images of various ages, genders, skin tones, and expressions from the Flickr platform. This dataset is commonly employed to train large-scale generative models. In this paper, we utilize a derivative dataset of the FFHQ called FFHQ64, which involves downsampling the images from the original FFHQ dataset to a resolution of 64×64.

The AFHQv2 (Animal Faces-HQ) dataset (Choi et al., 2020) comprises 15,000 high-definition animal face images with a resolution of 512×512, including 5,000 images each for cats, dogs, and wild animals. AFHQv2 is commonly employed in tasks such as image-to-image translation and image generation. Similar to the FFHQ dataset, we downscale the original AFHQv2 dataset to a resolution of 64×64 for the experiment.

The ImageNet dataset (Deng et al., 2009) was established as a large-scale image dataset to facilitate the development of computer vision technologies. This dataset comprises over 14,197,122 images spanning more than 20,000 categories, indexed by 21,841 Synsets. In this paper, we use the ImageNet64 dataset, a subsampled version of the ImageNet dataset. The Imagenet64 dataset consists of a vast collection of images with a resolution of 64×64, containing 1,281,167 training samples, 50,000 testing samples, and 1,000 labels.

The LSUN (Large Scale Scene Understanding) dataset (Yu et al., 2015) is a large-scale dataset for scene understanding in visual tasks within deep learning. Encompassing numerous indoor scene images, it spans various scenes and perspectives. The LSUN dataset comprises multiple sub-datasets, in this study, we use the LSUN Cat and Bedroom sub-datasets with a resolution of 256×256.

B.2 Experiment Setup

For experiments on FFHQ 64x64, AFHQv2 64x64, and ImageNet 64x64 datasets, the pre-trained models are provided by the official release of EDM Karras et al. (2022). We use Adam optimizers to train the student generator G and s_ϕ, with both learning rates set to $1e^{-5}$. The training consisted of 50,000 iterations on four NVIDIA 3090 GPUs, and the batch size per GPU is set to 8. The training ratio between s_ϕ and G remains at $1 : 1$. In the Degradation stage, we trained for 200 epochs total, saving a checkpoint every 50 epochs, resulting in a total of 5 intermediate nodes along the degradation path $\{s'_{\theta_i} | i = 0, 1, 2, 3, 4\}$. In the Distribution Backtracking stage, when $i \geq 3$, each checkpoint was trained for 1,000 steps. When $i < 3$, each checkpoint was trained for 10,000 steps. The remaining steps were used to distill the original teacher model s'_{θ_0}.

For experiments on LSUN bedroom and LSUN cat datasets, the pre-trained EDM models are provided by the official release of Consistency Model Song et al. (2023). During the training, we set σ_{max} to 80 and keep it constant during the single-step generation process. We use SGD and AdamW optimizers during training to train the generator G and s_ϕ, with learning rates set to $1e^{-3}$ and $1e^{-4}$, respectively. The training consisted of 10,000 iterations on one NVIDIA A100 GPU, and the batch size per GPU is set to 2. The training ratio between s_ϕ and G remains at $4 : 1$. In the Degradation stage, we trained for 200 epochs total and saved the checkpoint every 50 epochs, resulting in a total of 5 intermediate nodes along the degradation path $\{s'_{\theta_i} | i = 0, 1, 2, 3, 4\}$. In the Distribution Backtracking stage, when $i \geq 3$, each checkpoint was trained for 500 steps. When $i < 3$, each checkpoint was trained for 1000 steps. The remaining steps were used to distill the original teacher model s'_{θ_0}.

When distilling the SDXL model, the teacher model and the student generator are both initialed by the pre-trained SDXL model on the huggingface (model id is 'stabilityai/stable-diffusion-xl-base-1.0'). We use Adam optimizers to train G and s_ϕ, with learning rates set to $1e^{-3}$ and $1e^{-2}$, respectively. The training consisted of 50,000 iterations on one NVIDIA A100 GPU, and the batch size per GPU is set to 1. The training ratio between s_ϕ and G remains at $1 : 1$. The training prompts are obtained from LAION-Aesthetics. In the Degradation stage, we trained for 1,000 epochs total and saved the checkpoint every 100 epochs, resulting in a total of 10 intermediate nodes along the degradation path $\{s'_{\theta_i} | i = 0, 1, ..., 9\}$. In the Distribution Backtracking stage, each checkpoint was trained for 1,000 steps. The remaining steps were used to distill the original teacher model s'_{θ_0}.

B.3 User study setup

Firstly, we randomly selected 128 prompts from the LAION-Aesthetics (Schuhmann et al., 2022). Then we use the original SDXL model and the distilled SDXL model to generate 128 pairs of images. Subsequently, we randomly recruit 10 volunteers, instructing each to individually evaluate the fidelity, detail, and vividness of these pairwise images. 10 volunteers included 6 males and 4 females, aged between 24 and 29. 5 of them have artificial intelligence or related majors and the other 5 of them have other majors. They were given unlimited time for the experiment, and all of the volunteers completed the assessment with an average time of 30 minutes. Finally, we took the average of the evaluation results of 10 volunteers as the final user study result.

C Theoretical Demonstration

C.1 KL Divergence of DisBack

As the KL divergence follows

$$D_{\mathrm{KL}}(q \parallel p) = \mathbb{E}_q \left[\log \frac{q}{p} \right] \tag{10}$$

The KL divergence of generated distribution and training distribution at timestep t can be written as

$$\begin{aligned} & D_{\mathrm{KL}} \left(q_t^G \left(\boldsymbol{x}_t \right) \parallel q_t \left(\boldsymbol{x}_t \right) \right) \\ & = \mathbb{E}_{x_t \sim q_t^G(\boldsymbol{x}_t)} \log \frac{q_t^G \left(\boldsymbol{x}_t \right)}{q_t \left(\boldsymbol{x}_t \right)} \\ & = \mathbb{E}_{x_0 \sim G(z;\eta)} \left[\log q_t^G \left(\boldsymbol{x}_t \right) - \log q_t \left(\boldsymbol{x}_t \right) \right] \\ & = \mathbb{E}_z \left[\log q_t^G \left(\boldsymbol{x}_t \right) - \log q_t \left(\boldsymbol{x}_t \right) \right] \end{aligned} \tag{11}$$

Thus, the gradient of KL divergence can be estimated as

$$\nabla_\eta D_{\mathrm{KL}}\left(q_t^G\left(\boldsymbol{x}_t\right)\|q_t\left(\boldsymbol{x}_t\right)\right)=\mathbb{E}_{t,\boldsymbol{\epsilon}}\left[s_\phi\left(\boldsymbol{x}_t,t\right)-s_\theta\left(\boldsymbol{x}_t,t\right)\right]\frac{\delta\boldsymbol{x}_t}{\delta\eta} \tag{12}$$

C.2 Stable Target Field

Given $\boldsymbol{x}_0 \sim q_0$ is the training data, $\boldsymbol{x}_t \sim p(\boldsymbol{x}_t \mid \boldsymbol{x}_0)$ is the disturbed data, Xu *et al.* (Xu et al., 2022) presents an estimation of the score as:

$$\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t)=\frac{\nabla_{\boldsymbol{x}_t}p_t(\boldsymbol{x}_t)}{p_t(\boldsymbol{x}_t)}=\frac{\mathbb{E}_{\boldsymbol{x}_0}\nabla_{\boldsymbol{x}_t}p(\boldsymbol{x}_t\mid\boldsymbol{x}_0)}{p_t(\boldsymbol{x}_t)} \tag{13}$$

The transition kernel $p(\boldsymbol{x}_t \mid \boldsymbol{x}_0)$ follows the Gaussian distribution $p(\boldsymbol{x}_t \mid \boldsymbol{x}_0) \sim \mathcal{N}(\mu_t, \sigma_t^2 I)$. Here $\mu_t = \boldsymbol{x}_0$ in Variance Exploding SDE (Song et al., 2021b) but is defined differently in other diffusion models.

$$p(\boldsymbol{x}_t\mid\boldsymbol{x}_0)=\frac{1}{\sqrt{(2\pi^k)}\sigma_t}\exp(-\frac{(\boldsymbol{x}_t-\mu_t)^T(\boldsymbol{x}_t-\mu_t)}{2\sigma_t^2}) \tag{14}$$

$$\begin{aligned}
&\nabla_{\boldsymbol{x}_t}p(\boldsymbol{x}_t\mid\boldsymbol{x}_0)\\
&=\nabla_{\boldsymbol{x}_t}\left[\frac{1}{\sqrt{(2\pi^k)}\sigma_t}\exp(-\frac{(\boldsymbol{x}_t-\mu_t)^T(\boldsymbol{x}_t-\mu_t))}{2\sigma_t^2}\right]\\
&=p(\boldsymbol{x}_t\mid\boldsymbol{x}_0)\nabla_{\boldsymbol{x}_t}(-\frac{(\boldsymbol{x}_t-\mu_t)^T(\boldsymbol{x}_t-\mu_t)}{2\sigma_t^2})\\
&=p(\boldsymbol{x}_t\mid\boldsymbol{x}_0)\frac{\mu_t-\boldsymbol{x}_t}{\sigma_t^2}
\end{aligned} \tag{15}$$

Combine Eq. (13) to Eq. (15), we have

$$\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t)=\mathbb{E}_{\boldsymbol{x}_0}\frac{p(\boldsymbol{x}_t\mid\boldsymbol{x}_0)}{p_t(\boldsymbol{x}_t)}\frac{\mu_t-\boldsymbol{x}_t}{\sigma_t^2}=\frac{1}{p_t(\boldsymbol{x}_t)}\mathbb{E}_{\boldsymbol{x}_0}p(\boldsymbol{x}_t\mid\boldsymbol{x}_0)\frac{\mu_t-\boldsymbol{x}_t}{\sigma_t^2} \tag{16}$$

Let B be a set of reference samples for Monte Carlo estimation, we have

$$p_t(\boldsymbol{x}_t)=\mathbb{E}_{\boldsymbol{x}_0}p(\boldsymbol{x}_t\mid\boldsymbol{x}_0)\approx\frac{1}{|B|}\sum_{x_0^{(i)}\in B}p(\boldsymbol{x}_t\mid\boldsymbol{x}_0^{(i)}) \tag{17}$$

Combine the Eq. (16) and Eq. (17), we can get

$$\nabla_{\boldsymbol{x}_t}\log p_t(x_t)=\mathbb{E}_{\boldsymbol{x}_0}\frac{p(\boldsymbol{x}_t\mid\boldsymbol{x}_0)}{p_t(\boldsymbol{x}_t)}\frac{\mu_t-\boldsymbol{x}_t}{\sigma_t^2}\approx\frac{1}{p_t(\boldsymbol{x}_t)}\frac{1}{|B|}\sum_{x_0^{(i)}\in B}p(\boldsymbol{x}_t\mid\boldsymbol{x}_0^{(i)})\frac{\mu_t-\boldsymbol{x}_t}{\sigma_t^2} \tag{18}$$

Here the ”≈” represents the Monte Carlo estimate.

Depending on the network prediction, the diffusion model can be divided into different types, including $\boldsymbol{\epsilon}$ prediction (Karras et al., 2022) and x_0 prediction (Song et al., 2021a; Ho et al., 2020; Nichol & Dhariwal, 2021). When the score $\nabla_{\boldsymbol{x}_t}\log p_t(x_t)$ is estimated by Eq.(18), it can be converted to $\boldsymbol{\epsilon}$, x_0 and v by a series of transformations.

$$\hat{\boldsymbol{\epsilon}}\approx-\sigma_t\nabla_{\boldsymbol{x}_t}\log p_t(x_t) \tag{19}$$

$$\hat{\boldsymbol{x}}_0\approx\nabla_{\boldsymbol{x}_t}\log p_t(x_t)*\sigma_t^2+\boldsymbol{x}_t \tag{20}$$

D Discussion

D.1 The gradient orientation of score distillation

To better illustrate the process of training a generator with score distillation, we provide a more intuitive explanation. As shown in Figure 10, given a noisy data $\boldsymbol{x}_t$, the direction of $s_\phi\left(\boldsymbol{x}_t, t\right)$ points to the generated distribution and the direction of $s_\theta\left(\boldsymbol{x}_t, t\right)$ points to the training distribution. In Sec.E.3, we used 2D data to illustrate the gradient directions of $\boldsymbol{x}_t$ on the teacher model s_θ and the auxiliary diffusion model s_ϕ. Thus, the direction of $s_\theta\left(\boldsymbol{x}_t, t\right) - s_\phi\left(\boldsymbol{x}_t, t\right)$ is from the generated distribution towards the training distribution. Ideally, if the predicted score of s_ϕ and s_θ are both correct, this gradient leads G to update in the right direction.

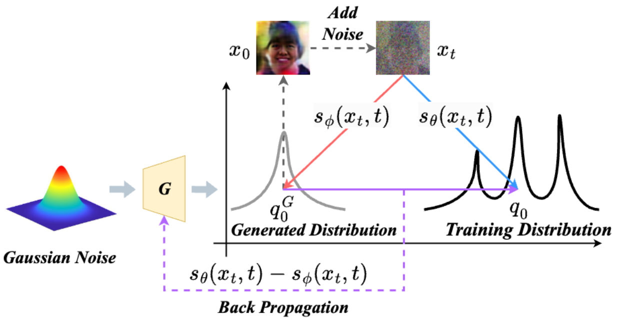


Figure 10: An illustration of the generator update by the minimization in Eq. (6).

D.2 Two types of mismatch issue

We find that there are two types of mismatch issues during the training process: (1) The mismatch issue between the initially generated samples and s_ϕ. (2) The mismatch issue between the initially generated samples and s_θ. For the first type of mismatch issue, it prevents s_ϕ from accurately describing the distribution of the generator, leading to unreliable network predictions and introducing biases when computing the generator's gradients. The combined effect of these two mismatch issues exacerbates the negative impact on the optimization of the generator. Fortunately, among the three initialization methods of s_ϕ proposed in Section 3, when s_ϕ is initialized based on the generated images, its distribution becomes close to the initially generated distribution, which resolves the first type of mismatch issue. In this paper, we directly use the degraded teacher model s'_{θ_N} to initialize s_ϕ. Because the degraded teacher model's distribution is closer to the initial generator's distribution, s_ϕ initialized by s'_{θ_N} can approximate the initial generation distribution and accurately predict the scores of the generated samples from the beginning. However, since s_θ remains fixed during the training, the second type of mismatch issue cannot be avoided through initialization. Consequently, in this paper, we mainly focus on discussing the second type of mismatch issue.

D.3 Training Efficiency of DisBack

While DisBack involves an iterative optimization process during training, the optimization objective of $s_\phi\left(\boldsymbol{x}_t, t\right)$ aims to minimize the loss of the standard diffusion model based on Eq.(21), and the objective of student generator aims to minimize the KL divergence in Eq.(22). These two optimization processes do not entail adversarial training as in GANs. Consequently, the optimization process tends to be more stable. A recent work Monoflow (Yi et al., 2023) also discusses in GANs training a vector field is obtained to guide the optimization of the generator, but the vector field derives from the discriminator and the instability is not mitigated.

$$\min_{\phi} \mathbb{E}_{t,\boldsymbol{\epsilon}} \left\| s_\phi\left(\boldsymbol{x}_t, t\right) - \frac{\boldsymbol{x}_0 - \boldsymbol{x}_t}{\sigma_t^2} \right\|_2^2 \tag{21}$$

$$\min_{\eta} \mathbb{E}_{t,\boldsymbol{\epsilon}} D_{KL}\left(q_t^G\left(\boldsymbol{x}_t\right) \| q_t\left(\boldsymbol{x}_t\right)\right) \tag{22}$$

For the DisBack, training the student generator only requires two U-Nets to perform inference and subtraction. Training s_ϕ only involves training a single U-Net, and gradients do not need to be backpropagated to G_{stu}. Therefore, these models can be naturally deployed to different devices, making computational resource requirements more distributed. This ease of distribution allows for joint training on computational devices with limited capacity. In contrast, for GANs and VAEs, which require gradient propagation between models (discriminator to generator, decoder to encoder), computational requirements are more centralized, necessitating the use of a single device or tools like DeepSpeed to manage the workload.

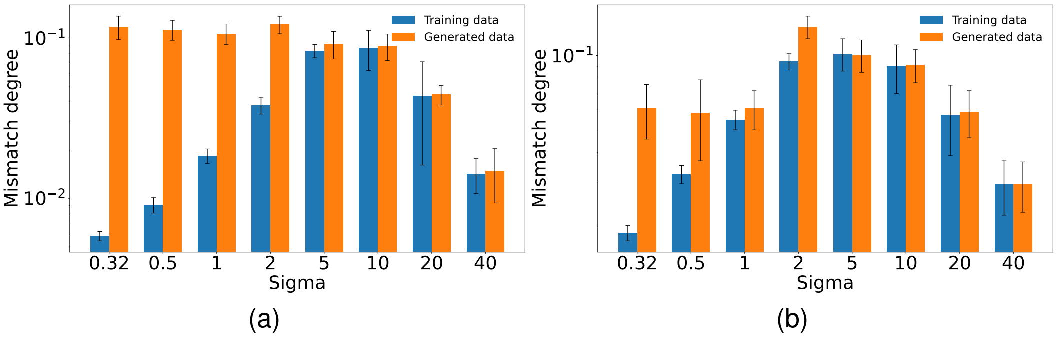


Figure 11: The results of the pre-experiments on mismatch degree. (a) s_θ and G are both initialized by the pre-trained EDM (Karras et al., 2022) on FFHQ. (b) s_θ and G are both initialized by the pre-trained EDM on ImageNet. The mismatch degree on the generated data is greater than on the training data, especially when the noise scale is low.

D.4 Vector Field

In our research, each of the estimated score functions s'_{θ_i}, for i ranging from 0 to N, delineates a vector field $\mathbb{R}^{3\times W\times H} \mapsto \mathbb{R}^{3\times W\times H}$. We make a strong assumption behind our proposed method that these score functions represent existing or non-existent distributions and that they altogether imply a transformation path between s_θ and the student generator G^0_{stu}. Nevertheless, a score fundamentally constitutes a gradient field, signifying the gradient of the inherent probability density. A vector field is a gradient field when several conditions are satisfied, including path independence, continuous partial derivatives, and zero curls (Matthews, 1998). The vector field, as characterized by the score functions, may not meet these conditions, and thus there is not a potential function or a probability density function. Such deficiencies could potentially hinder the successful training of the student generator and introduce unforeseen difficulties in the distillation process. Specifically, in instances where s_θ does not precisely represent a gradient field, a highly probable scenario considering s_θ is a neural network, the samples generated from s_θ could encompass failure cases. Although our empirical studies exemplify the effectiveness of the proposed DisBack, the detrimental effects of the discussed issue remain unclear. We will further explore this issue in our future work.

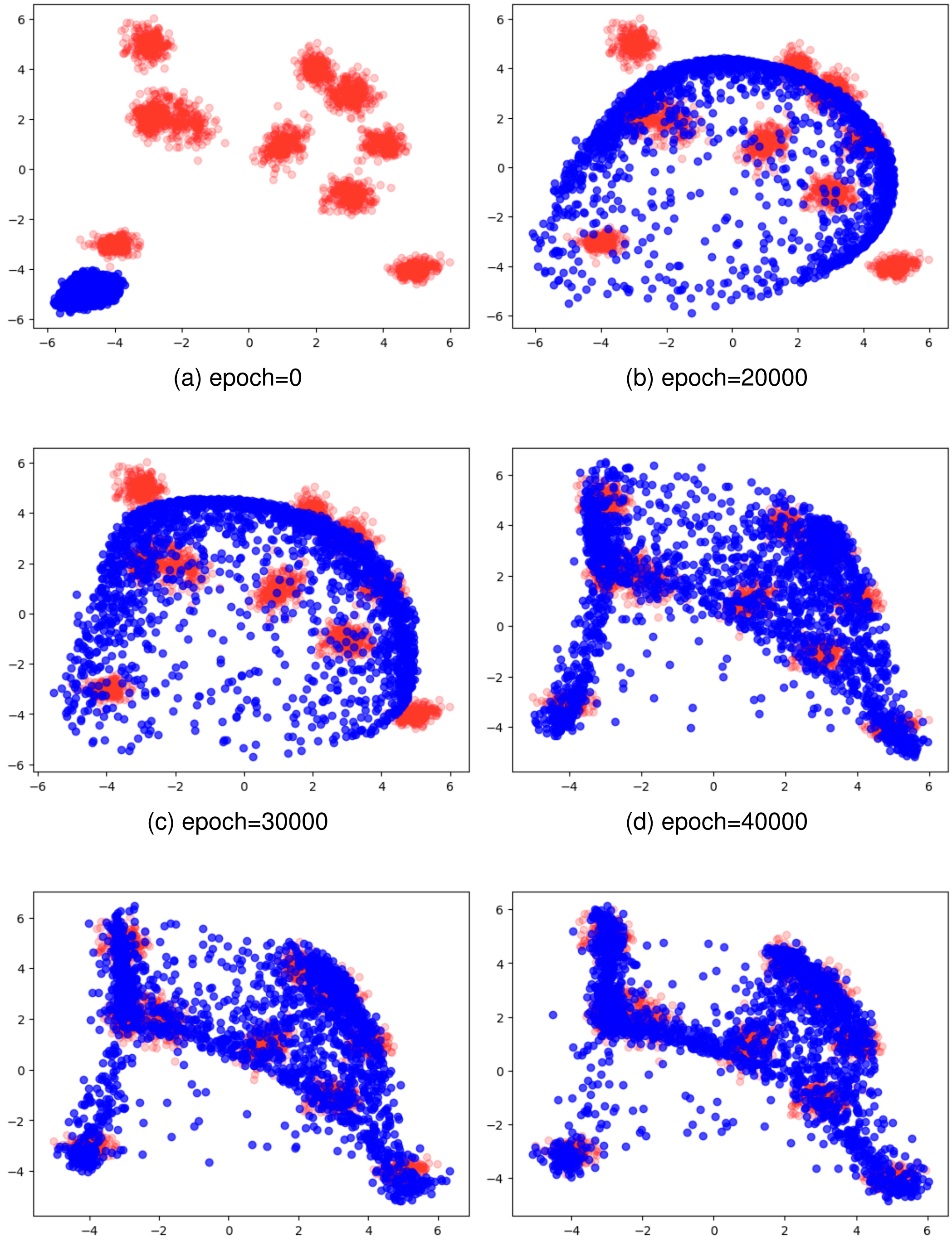
E Additional Details in Pre-experiments

E.1 Distribution mismatch issues

Before conducting our research, we first carried out preliminary experiments to demonstrate that the proposed mismatch issue does indeed exist when using the endpoints of pre-trained diffusion models as teacher models. Using the method proposed in Eq. 9, we conducted experiments with the pre-trained EDM model on the ImageNet and FFHQ datasets. We calculated the mismatch degree separately on the student model's initial generated data and the teacher model's original training data, and the results are shown in Fig. 11. We can see that, on both datasets, the mismatch degree on the generated data of the pre-trained model is greater than that on the real training data, especially when the noise scale is small. This aligns with our hypothesis stated in Sec. 1, demonstrating that directly using the endpoint of a pre-trained model as the teacher model leads to a distribution mismatch problem and causes the unreliable predictions of the teacher model.

E.2 A Toy Experiment on Gaussian Mixture Distribution

To validate the feasibility of the proposed DisBack, we conduct experiments on two-dimensional Gaussian mixture data. First, we randomly select 10 Gaussian distributions mixed as the training distribution q_0. Next, we construct a ResNet MLP as the two-dimensional diffusion model s_θ and train it using the created mixture Gaussian distribution. Similarly, we construct a simple MLP as the student generator G_{stu} and train a model s_ϕ with the same architecture as s_θ using generated data.



(a) epoch=0

(b) epoch=20000

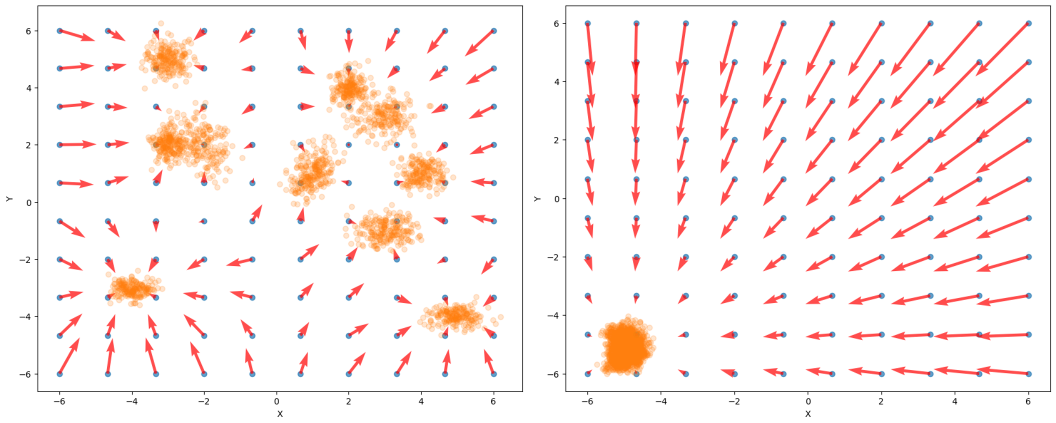
(c) epoch=30000

(d) epoch=40000

(e) epoch=50000

(f) epoch=60000

Figure 12: The distribution of student generator during the training process. Blue points visualize the generated distribution $q_t^G$and the red points visualize the training distribution q_0.



(a) The gradient direction of $s_\theta\left(\boldsymbol{x}_t, t\right)$ (b) The gradient direction of $s_\phi\left(\boldsymbol{x}_t, t\right)$

Figure 13: The gradient direction of $s_\theta\left(\boldsymbol{x}_t, t\right)$ and $s_\phi\left(\boldsymbol{x}_t, t\right)$ on x_t. The points in (a) are sampled from the training distribution and the points in (b) are sampled from the generated distribution.

Therefore, we can use s_θ and s_ϕ to train the student generator G_{stu}. During the training process, we visualize the distribution of the student generator and training data to intuitively demonstrate the changes in the student generator distribution under the proposed training framework. The distribution of G_{stu} during the training process is shown in Figure 12. As training progresses, the generated distribution q^G initially expands outward and then gradually convergents towards the training distribution. The results show that the proposed method for training the student generator is effective.

E.3 Gradient Orientation Verification of DisBack

As mentioned in Sec. 3, when updating G_{stu} using Eq.(6), $s_\theta\left(\boldsymbol{x}_t, t\right)$ provides a gradient towards the training distribution, while $s_\phi\left(\boldsymbol{x}_t, t\right)$ provides a gradient toward the generated distribution.

To validate the correctness of these gradient directions, we experiment on two-dimensional data. We evenly sample N data points within the range of $(x, y) \in [-6, 6]$ as the noisy data $\boldsymbol{x}_t$. Subsequently, we depict the gradient directions of $\boldsymbol{x}_t$ based on s_θ and s_ϕ respectively. As shown in Figure 13, consistent with theoretical derivation, for any given $\boldsymbol{x}_t$, the gradient direction of $s_\theta\left(\boldsymbol{x}_t, t\right)$ points toward the training distribution, and the magnitude of the gradient decreases as the distance to the training distribution decreases. Similarly, for any given $\boldsymbol{x}_t$, the gradient direction of $s_\phi\left(\boldsymbol{x}_t, t\right)$ points toward the generated distribution.

F Additional Samples from DisBack

We provide additional samples from DisBack on FFHQ 64 × 64 (Figure 15), AFHQv2 64 × 64 (Figure 16), ImageNet 64 × 64 (Figure 17), LSUN Bedroom 256 × 256 (Figure 18) and LSUN Cat 256 × 256 (Figure 19).

G Additional Ablation Study

We further explored the impact of different numbers of checkpoints along the degradation path and various degradation strategies on the performance of DisBack on the ImageNet64 dataset. The results are shown in Tab.6. When degrading for $200iterations$ under default settings, the FID score with $N = 3$ intermediate checkpoints was 5.23, which outperformed the original Diff-instruct's FID of 5.57. The optimal FID of 1.38 was achieved with $N = 5$.

However, with $N = 11$, the FID score dropped to 8.35. When there are too many checkpoints in the path, the distributions of the checkpoints (e.g., $s'_{\theta_10}, s'_{\theta_19}, s'_{\theta_8}$) are very close to the distribution

Table 6: The conditional generation performance of DisBack on ImageNet 64x64 dataset.

DisBack	FID (↓)
N=1	5.96
N=3	4.88
N=5	1.38
N=11	9.15
N=11(Degradation iteration =400)	244.72

of the initial generator. Training with these checkpoints provides limited progress for the generator. Furthermore, having too many checkpoints complicates the checkpoint transition scheduler during distillation, making it difficult to manage effectively. This often leads to inefficient updates for the generator across many iterations, wasting time without meaningful improvement. When the degradation iteration count is set to 400, DisBack fails to correctly distill the student generator. This is because excessive degradation iterations result in checkpoints near the initial generator on the degradation path being unable to generate sensible samples. Using these checkpoints for distillation provides no reasonable or effective guidance to the generator, ultimately causing it to fail to learn any meaningful information. In summary, when obtaining the degradation path, it is essential not to overly degrade the teacher model. The degraded teacher model's distribution should be close to the initial generator's distribution while still being able to produce relatively reasonable samples. Additionally, having too many or too few checkpoints along the path can adversely affect the final performance.

H Failure Examples

Fig. 14 presents several failure cases of DisBack.

In terms of FFHQ, AFHQv2, and ImageNet, while these images already capture the features of the corresponding datasets, the generated results lack accurate and clear backgrounds. The potential reasons for this include the fact that these datasets primarily focus on learning foreground content, with low requirements for image backgrounds, making the model difficult to clear backgrounds.

As for LSUN Cat and Bedroom, DisBack successfully generates details such as the cat's fur and the bed's texture, but it does not generate the overall shape and the detailed structure. This may be because the model does not capture the overall information of the data, only capturing local content. This issue may stem from the inherent limitations of U-Net, resulting in poor generation of overall structures in rare cases.

In the future, attempts will be made to use more advanced teacher models or improve the distillation algorithm to overcome these limitations. Moreover, we will further explore more advanced generator architectures such as StyleGAN Karras et al. (2020; 2021) to achieve higher-quality generation.

I Ethical Statement

I.1 Ethical Impact

The potential ethical impact of our work is about fairness. As "human face" is included as a kind of generated image, our method can be used in face generation tasks. Human-related datasets may have data bias related to fairness issues, such as the bias to gender or skin color. Such bias can be captured by the generative model in the training.

I.2 Notification to Human Subjects

In our user study, we present the notification to subjects to inform the collection and use of data before the experiments.

> Dear volunteers, we would like to thank you for supporting our study. We propose the Distribution Backtracking Distillation, which introduces the convergence tra-

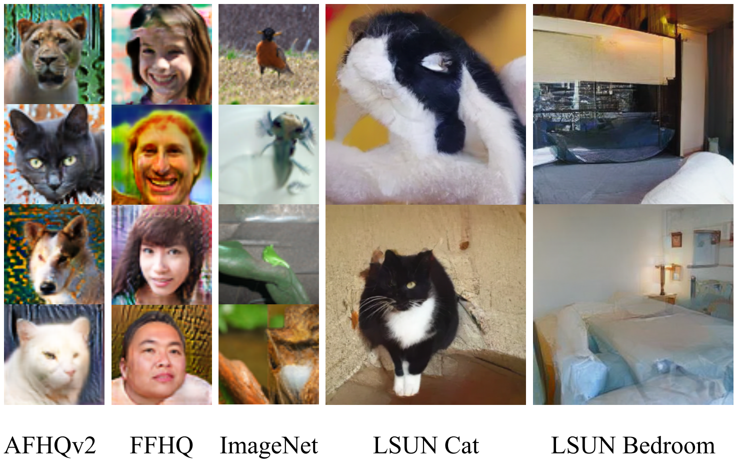


Figure 14: Failure examples.

> jectory into the score distillation process to achieve efficient and fast distillation and high-quality single-step generation.
>
> All information about your participation in the study will appear in the study record. All information will be processed and stored according to the local law and policy on privacy. Your name will not appear in the final report. Only an individual number assigned to you is mentioned when referring to the data you provided.
>
> We respect your decision whether you want to be a volunteer for the study. If you decide to participate in the study, you can sign this informed consent form.

The Institutional Review Board approved the use of users' data of the main authors' affiliation.

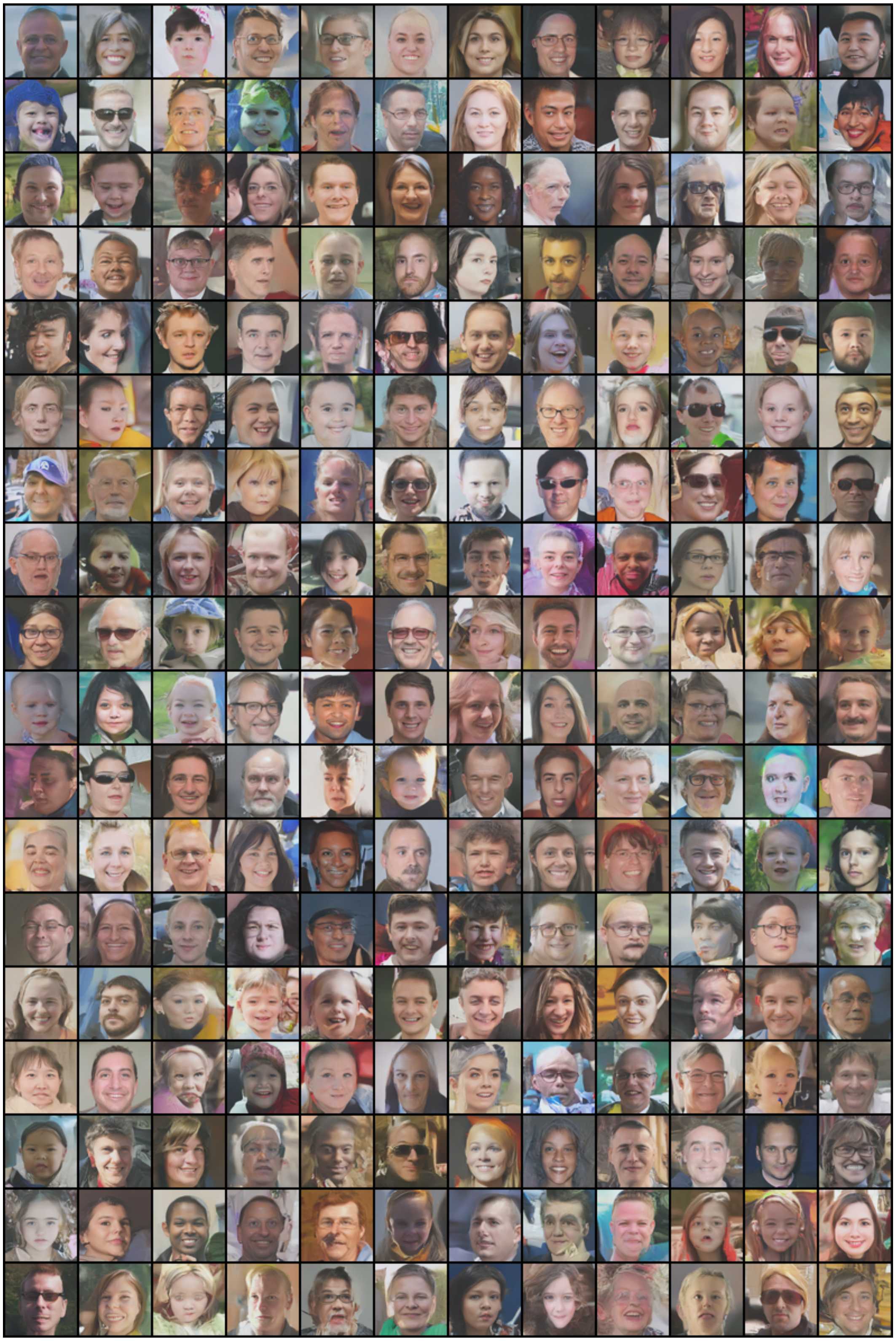


Figure 15: Additional Samples form conditional FFHQ 64x64.

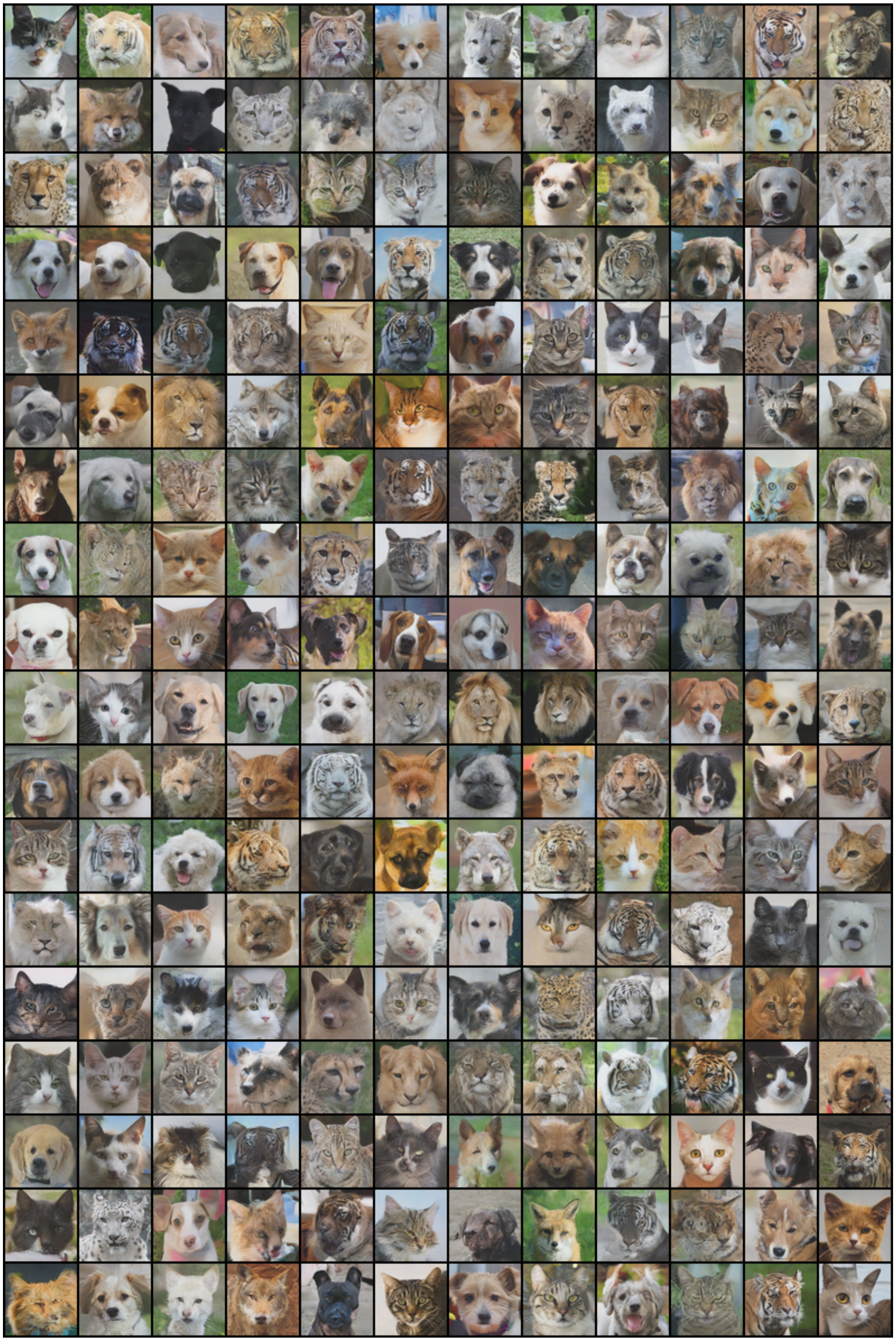


Figure 16: Additional Samples form conditional AFHQv2 64x64.

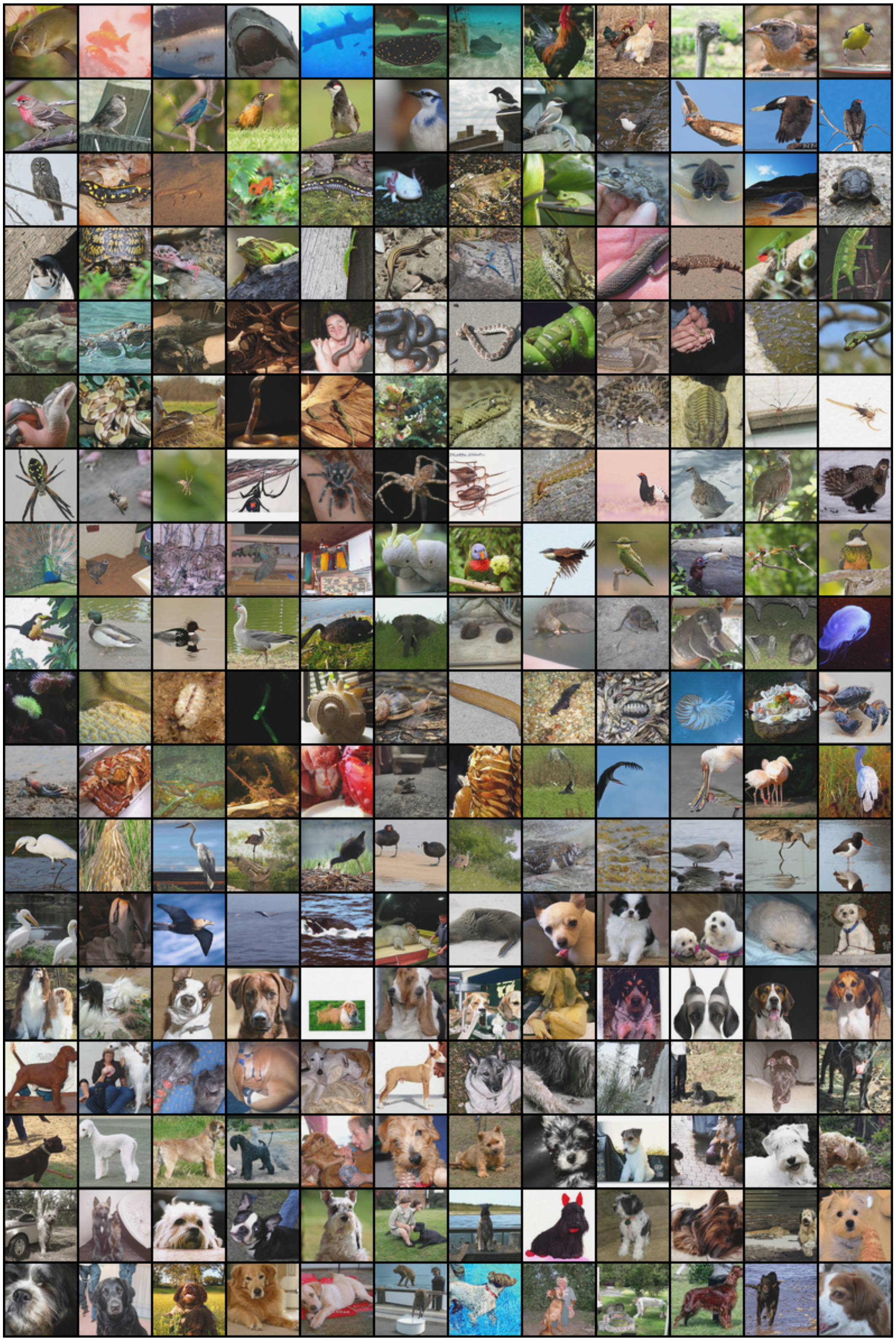


Figure 17: Additional Samples form conditional ImageNet 64x64.

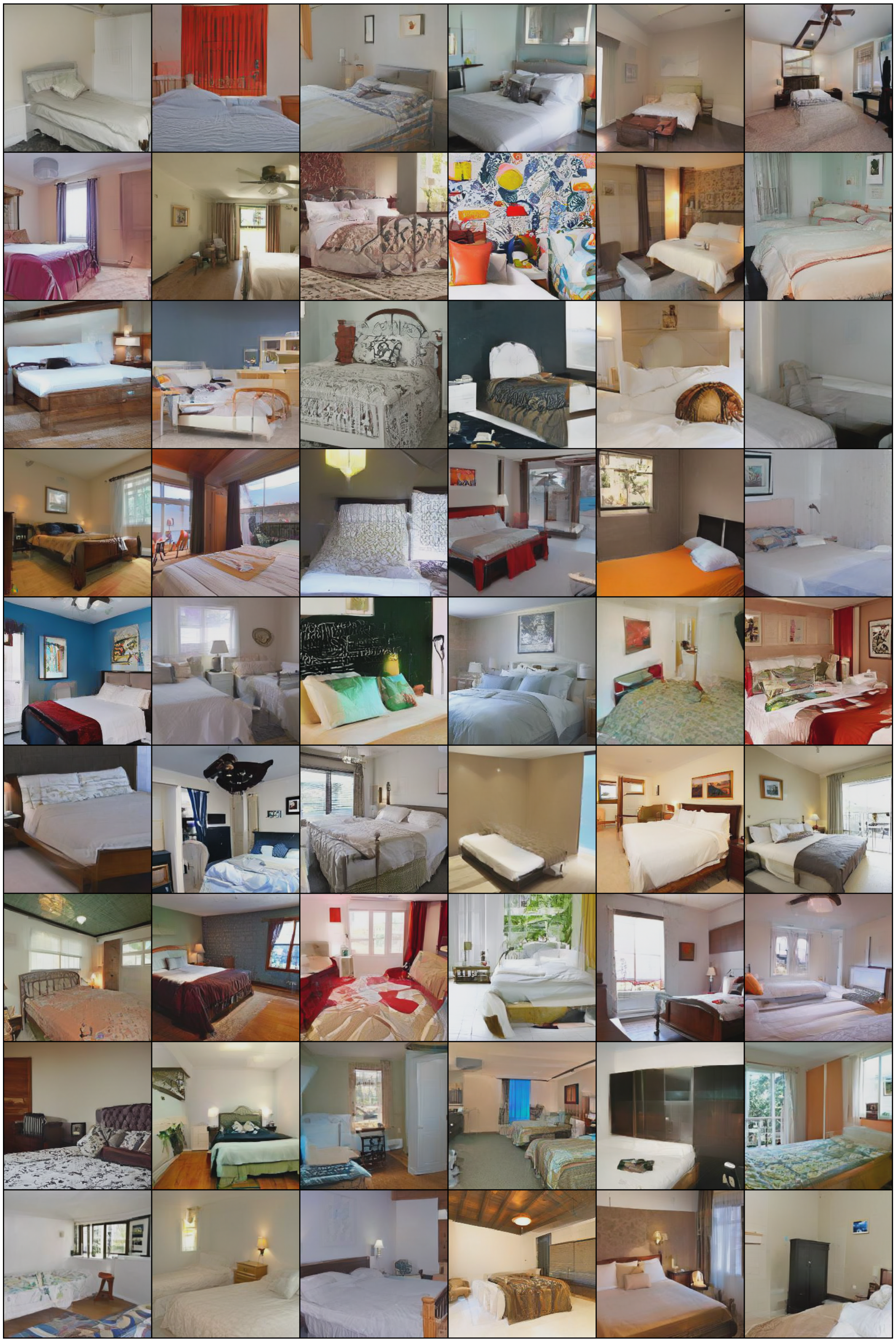


Figure 18: Additional Samples form conditional LSUN bedroom.

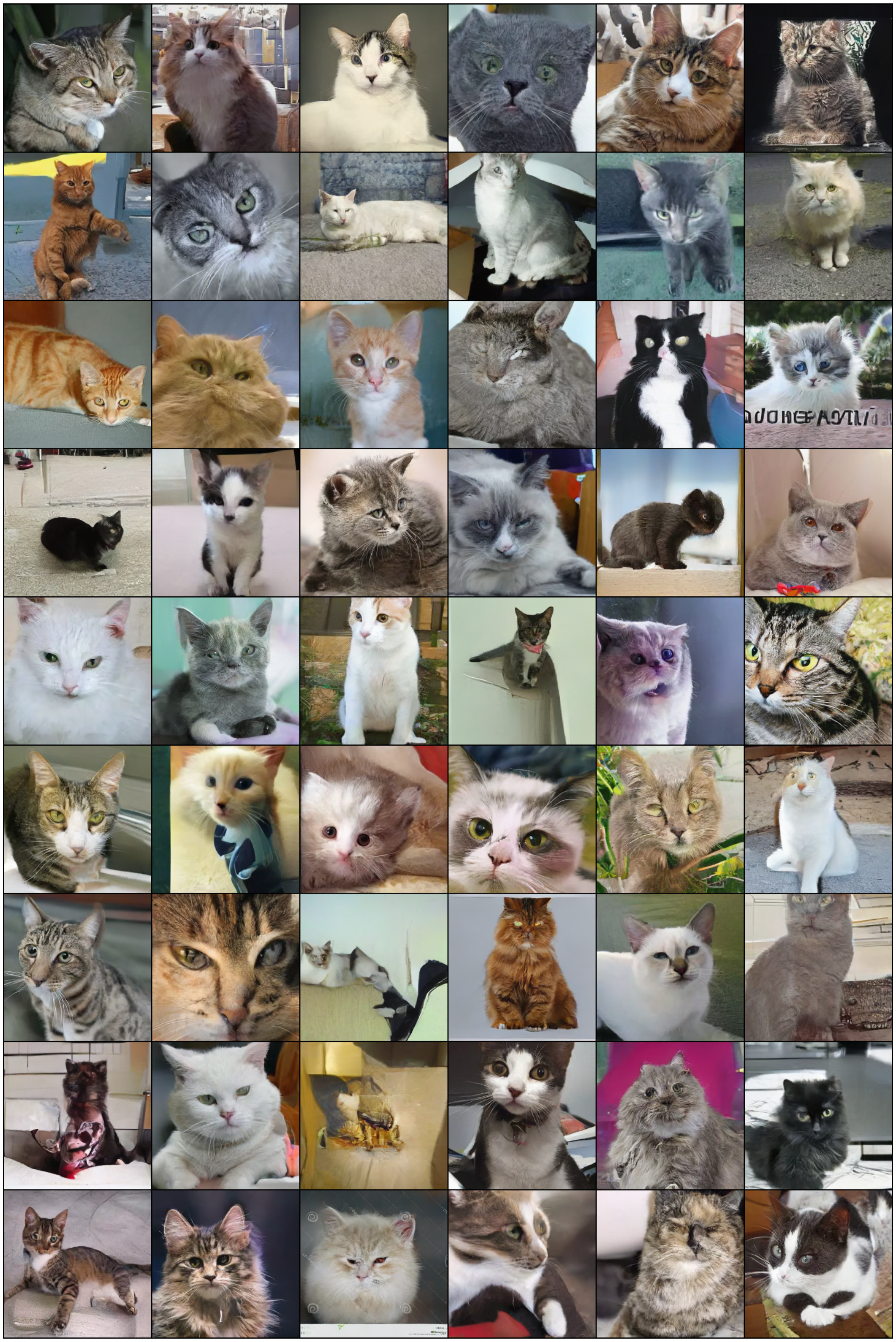


Figure 19: Additional Samples form conditional LSUN cat.