Flash Diffusion: Accelerating Any Conditional Diffusion Model for Few Steps Image Generation

Clément Chadebec*, **Onur Tasar** , **Eyal Benaroche**†, **Benjamin Aubin**

Jasper Research

Abstract

In this paper, we propose an efficient, fast, and versatile distillation method to accelerate the generation of pre-trained diffusion models. The method reaches state-of-the-art performances in terms of FID and CLIP-Score for few steps image generation on the COCO2014 and COCO2017 datasets, while requiring only several GPU hours of training and fewer trainable parameters than existing methods. In addition to its efficiency, the versatility of the method is also exposed across several tasks such as *text-to-image*, *inpainting*, *face-swapping*, *super-resolution* and using different backbones such as UNet-based denoisers (SD1.5, SDXL), DiT (Pixart) and MMDiT (SD3), as well as adapters. In all cases, the method allowed to reduce drastically the number of sampling steps while maintaining very high-quality image generation.

Code — https://github.com/gojasper/flash-diffusion
Extended version — https://arxiv.org/abs/2406.02347

Introduction

Diffusion Models (DM) (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020; Song et al. 2020) have proven to be one of the most efficient class of generative models for image synthesis (Dhariwal and Nichol 2021; Ramesh et al. 2022; Rombach et al. 2022; Nichol et al. 2022) and have raised particular interest and enthusiasm for text-to-image applications (Ramesh et al. 2021, 2022; Rombach et al. 2022; Saharia et al. 2022; Ho et al. 2022; Esser et al. 2024; Podell et al. 2023; Chen et al. 2023, 2024) where they outperform other approaches. However, their usability for real-time applications remains limited by the intrinsic iterative nature of their sampling mechanism. At inference time, these models aim at iteratively denoising a sample drawn from a Gaussian distribution to finally create a sample belonging to the data distribution. Nonetheless, such a denoising process requires multiple evaluations of a potentially very computationally costly neural function.

Recently, more efficient solvers (Lu et al. 2022a,b; Zhang and Chen 2022; Zhao et al. 2024) or diffusion distillation methods (Salimans and Ho 2021; Song et al. 2023; Lin, Wang, and Yang 2024; Xu et al. 2023; Liu et al. 2023; Ren et al. 2024; Luo et al. 2023a,b; Sauer et al. 2023, 2024; Yin et al. 2023; Hsiao et al. 2024) aiming at reducing the number of sampling steps required to generate satisfying samples from a trained diffusion model have emerged to try to tackle this issue. Nonetheless, solvers typically require at least 10 Neural Function Evaluations (NFEs) to produce satisfying samples while distillation methods may require extensive training resources (Liu et al. 2023; Yin et al. 2023; Meng et al. 2023) or require an iterative training procedure to update the teacher model throughout training (Salimans and Ho 2021; Lin, Wang, and Yang 2024; Li et al. 2024) limiting their applications and reach. Moreover, most of the existing distillation methods are tailored for a specific task such as text-to-image. It is still unclear how they would perform on other tasks, using different conditionings and diffusion model architectures.

In this paper, we present *Flash Diffusion*, a fast, robust, and versatile diffusion distillation method that allows to drastically reduce the number of sampling steps while maintaining a very high image generation quality. The proposed method aims at training a student model to predict in a single step a denoised multiple-step teacher prediction of a corrupted input sample. The method also drives the student distribution towards the real input sample manifold with an adversarial objective (Goodfellow et al. 2014) and ensures that it does not drift too much from the learned teacher distribution using distribution matching (Dziugaite, Roy, and Ghahramani 2015; Li, Swersky, and Zemel 2015). The main contributions of the paper are as follows:

- We propose an efficient, **fast**, **versatile**, and LoRA compatible distillation method aiming at reducing the number of sampling steps required to generate high-quality samples from a trained diffusion model.
- We validate the method for text-to-image and show that it reaches SOTA results for few steps image generation on standard benchmark datasets with only two NFEs, which is equivalent to a single step with classifier-free guidance while having far fewer training parameters than competitors and requiring only a few GPU hours of training.
- We conduct an extensive ablation study to show the impact of the different components of the method and demonstrate its robustness and reliability.

*Corresponding author: [name].[surname]@jasper.ai

†Work done during an internship at Jasper Research

- We emphasize the versatility of the method through an extensive experimental study across various tasks (*text-to-image*, image *inpainting*, *super-resolution*, *face-swapping*), diffusion model architectures (SD1.5, SDXL, Pixart-α and SD3) and illustrate its compatibility with adapters (Mou et al. 2024) and existing LoRAs.

Related Works

Diffusion Models Diffusion models consist in artificially corrupting input data according to a given noise schedule (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020; Song et al. 2020) such that the data distribution eventually resembles a standard Gaussian one. They are then trained to estimate the amount of noise added in order to learn a reverse diffusion process allowing them, once trained, to generate new samples from Gaussian noise. Those models can be conditioned with respect to various inputs such as images (Rombach et al. 2022), depth maps, edges, poses (Zhang, Rao, and Agrawala 2023; Mou et al. 2024) or text (Dhariwal and Nichol 2021; Ramesh et al. 2022; Rombach et al. 2022; Nichol et al. 2022; Esser et al. 2024; Ho et al. 2022; Podell et al. 2023) where they demonstrated very impressive results. However, the need to recourse to a large number of sampling steps (typically 50 steps) at inference time to generate high-quality samples has limited their usage for real-time applications and narrowed their usability and reach.

Diffusion Distillation In order to tackle this limitation, several methods have recently emerged to reduce the number of function evaluations required at inference time. On the one hand, several papers tried to build more efficient solvers to speed up the generation process (Lu et al. 2022a,b; Zhang and Chen 2022; Zhao et al. 2024) but these methods still require the use of several steps (typically 10) to generate satisfying samples. On the other hand, several approaches relying on model distillation (Hinton, Vinyals, and Dean 2015) proposed to train a student network that would learn to match the samples generated by a teacher model but in fewer steps. A simple approach would consist in building pairs of *noise/teacher samples* and training a student model to match the teacher predictions in a single step (Luhman and Luhman 2021; Zheng et al. 2023). Nonetheless, this approach remains quite limited and struggles to match the quality of the teacher model since there is no underlying useful information to be learned by the student in full noise. Building upon this idea, several methods were proposed to first apply the *forward* diffusion process to an input sample and then pass it to the student network. The student prediction is then compared to the learned distribution of the teacher model using either a regression loss (Kohler et al. 2024; Yin et al. 2023) an adversarial objective (Xu et al. 2023; Sauer et al. 2023, 2024; Yin et al. 2024) or distribution matching (Yin et al. 2023, 2024).

Progressive distillation (Salimans and Ho 2021; Meng et al. 2023) is also a method that has proven to be quite promising. It consists in training a student model to predict a two-step teacher denoising of a noisy sample in a single step theoretically halving the number of required sampling steps. The teacher is then replaced by the new student and the process is repeated several times. This approach was also enriched with a GAN-based objective that allows to further reduce the number of sampling steps needed from 4-8 to a single pass (Lin, Wang, and Yang 2024). InstaFlow (Liu et al. 2023) proposed instead to rely on rectified flows (Liu, Gong et al. 2022) to ease the *one-step* distillation process. However, this approach may require a significant number of training parameters and a long training procedure, making it computationally intensive.

Consistency models (Song et al. 2023; Song and Dhariwal 2023; Luo et al. 2023a; Kim et al. 2023) is also a promising, effective, and one of the most versatile distillation methods proposed in the literature. The main idea is to train a model to map any point lying on the *Probability Flow Ordinary Differential Equation* (PF-ODE) to its origin, theoretically unlocking single-step generation. Luo et al. (2023b) combined Latent Consistency Model (LCM) and LoRAs (Hu et al. 2021) and showed that it is possible to train a strong student with a very limited number of trainable parameters and a few GPU hours of training. Nonetheless, those models still struggle to achieve single-step generation and reach the sampling quality of peers.

In a parallel study conducted recently, the authors of (Yin et al. 2024) also introduced the combined use of a distribution matching loss and an adversarial loss, a method we also employ in our paper. Nonetheless, they do not rely on the use of a distillation loss that proved highly efficient in our experiments and do not compute the adversarial loss with respect to the same inputs. Moreover, their approach still necessitates training another denoiser to assess the score of the fake samples, significantly increasing the number of trainable parameters and, consequently, the computational burden of the method. Furthermore, the ability of their method to generalize and perform effectively across different tasks and diffusion model architectures remains unclear.

Proposed Method

In this section, we expose the proposed method that builds upon several ideas proposed in the literature.

Background on Diffusion Models

Let $x_0 \in \mathcal{X}$ be a set of data such that $x_0 \sim p(x_0)$ where $p(x_0)$ is an unknown distribution. The main idea of diffusion models (DM) is to estimate the amount of noise ε, artificially added to an input sample x_0 using the *forward process* $x_t = \alpha(t) \cdot x_0 + \sigma(t) \cdot \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$. The noise schedule is controlled by two differentiable functions $\alpha(t)$, $\sigma(t)$ for any $t \in [0, T]$ such that the log signal-to-noise ratio $\log[\alpha(t)^2/\sigma(t)^2]$ is decreasing over time. In practice, during training a diffusion model learns a parametrized function ε_θ conditioned on the timestep t and taking as input the noisy sample x_t. The parameters θ are then learned via denoising score matching (Vincent 2011; Song and Ermon 2019).

$$\mathcal{L} = \mathbb{E}_{x_0 \sim p, t \sim \pi, \varepsilon \sim \mathcal{N}(0,\mathbf{I})} \left[\lambda(t) \left\| \varepsilon_\theta(x_t, t) - \varepsilon \right\|^2 \right] , \quad (1)$$

where $\lambda(t)$ is a scaling factor, $t \in [0, 1]$ is the timestep and $\pi(t)$ is a distribution over the timesteps. We provide in appendix, an extended background on diffusion models.

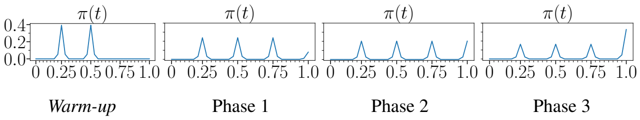

Figure 1: Illustration of the evolution of the proposed timesteps distribution π throughout training. $t = 0$ corresponds to no noise injection while $t = 1$ corresponds to the maximum noise injection (*i.e.* the noisy latent sample is equivalent to a sample drawn from a standard Gaussian distribution). For each phase unless the *Warm-up*, 4 timesteps are over-sampled out of the $K = 32$ selected ones. As the training progresses, the probability mass is shifted towards full noise to favor single-step generation.

Distilling a Pretrained Diffusion Model

For the following, we place ourselves in the context of Latent Diffusion Models (Rombach et al. 2022) for image generation and refer to the teacher model as $\varepsilon_\phi^{\text{teacher}}$, the student model as $\varepsilon_\theta^{\text{student}}$, the training images as x_0 and their unknown distribution $p(x_0)$. We refer to $z_0 = \mathcal{E}(x_0)$ as the associated latent variables obtained with an encoder $\mathcal{E}$. π is the probability density function of the timesteps $t \in [0, 1]$. The proposed method is mainly driven by the desire to end up with a fast, robust, and reliable approach that would be easily transposed to different use cases. The main idea of the proposed approach is quite similar to diffusion models.

Given a noisy latent sample z_t with $t \sim \pi(t)$, we propose to train a function f_θ to predict a denoised version $\tilde{z}_0$ of the original sample z_0. The main difference with a diffusion model is that instead of using z_0 as a target, we propose to leverage the knowledge of the teacher model and use a sample belonging to the data distribution learned by the teacher model $p_\phi^{\text{teacher}}(z_0)$. In other words, we use the teacher model and an ODE solver Ψ that is run several times to generate a denoised latent sample $\tilde{z}_0^{\text{teacher}}(z_t)$ used as a target for the student model. The main distillation loss writes as follows:

$$\mathcal{L}_{\text{distil}} = \mathbb{E}_{z_0,t,\varepsilon}\left[\left\| f_\theta(z_t, t) - \tilde{z}_0^{\text{teacher}}(z_t)\right\|^2\right], \quad (2)$$

A similar idea was employed in (Sauer et al. 2024) but the authors generate fully synthetic samples meaning that the samples z_t are pure noise, $z_t \sim \mathcal{N}(0, \mathbf{I})$. In contrast, in our approach, we hypothesize that allowing z_t to retain some information from the *ground-truth* encoded sample z_0 could enhance the distillation process. As in (Luo et al. 2023a), when distilling a conditional DM, we also perform Classifier-Free Guidance (CFG) (Ho and Salimans 2021) with the teacher to better enforce the model to respect the conditioning. This technique actually significantly improves the quality of the generated samples by the student as shown in the ablations. Additionally, it eliminates the need for conducting CFG during inference with the student, further decreasing the method's computational cost by halving the NFEs for each step. In practice, the guidance scale ω is uniformly sampled in $[\omega_{\min}, \omega_{\max}]$ where $0 \leq \omega_{\min} \leq \omega_{\max}$.

Timesteps Sampling

The cornerstone of our approach hinges on the selection of the timestep probability density function, denoted as $\pi(t)$. According to the continuous modeling, exposed in (Song et al. 2020), DMs are trained to remove noise from a latent sample z_t for any given continuous time t. However, since we aim at achieving few steps data generation (typically 1-4 steps) at inference time, the learned function ε_θ will only be evaluated at a few discrete timesteps $\{t_i\}_{i=1}^K$.

To tackle this issue and enforce the distillation process to focus on the most relevant timesteps, we propose to select K (typically 16 or 32) uniformly spaced timesteps in $[0, 1]$ and assign a probability to each of them according to a probability mass function $\pi(t)$. We choose $\pi(t)$ as a mixture of Gaussian controlled by a series of weights $\{\beta_i\}_{i=1}^K$

$$\pi(t) = \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{i=1}^{K} \beta_i \exp\left(-\frac{(t-\mu_i)^2}{2\sigma^2}\right), \quad (3)$$

where the mean of each Gaussian is controlled by $\{\mu_i = i/K\}_{i=1}^K$ and the variance is fixed to $\sigma = \sqrt{0.5/K^2}$. This approach is such that when distilling the teacher only a small number of K discrete timesteps will be sampled instead of the continuous range $[0, 1]$[1]. Moreover, the distribution π is defined such that out of the K selected timesteps, the 4 timesteps used at inference for 1, 2 and 4 steps generation are over-sampled (typically we set $\beta_i > 0$ if $i \in [\frac{K}{4}, \frac{K}{2}, \frac{3K}{4}, K]$ and $\beta_i = 0$ otherwise). Unlike other methods (Sauer et al. 2023, 2024) we do not only focus on those 4 timesteps since we noticed that it can lead to a reduction of diversity in the generated samples. This is in particular emphasized in the ablation study. In practice, we notice that a warm-up phase is beneficial to the training process. Therefore, we decide to start by first imposing a higher probability to the timesteps corresponding to the least added amount of noise by setting $\beta_{K/4} = \beta_{K/2} = 0.5$ and $\beta_i = 0$ otherwise. We then progressively shift the probability mass towards full noise to favor single-step generation while still over-sampling the targeted 4 timesteps by setting a strictly positive value for β_i where $i \equiv 0[K/4]$, and $\beta_i = 0$ otherwise. An example for π with $K = 32$ is illustrated in Fig. 1. As pictured in the figure, the $[0, 1]$ interval is split into 32 timesteps. During the *warm-up* phase, the probability mass allocates a higher probability to timesteps $[0.25, 0.5]$ to ease the distillation process. As the training progresses, the probability mass function is then shifted towards full noise to favor single-step generation while always allocating a higher probability to the 4 timesteps $[0.25, 0.5, 0.75, 1]$. The impact of the timesteps distribution is further discussed in the ablations.

Adversarial Objective

To further enhance the quality of the samples, we have also decided to incorporate an adversarial objective. The core idea is to train the student model to generate samples that

[1]In practice when training a DM, the range $[0, 1]$ is actually discretized (typically into 1000 timesteps) for computational purposes.

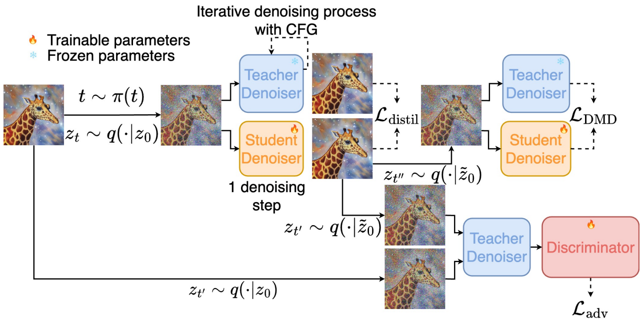

Figure 2: *Flash Diffusion* training method: the student is trained with a distillation loss between multiple-step teacher and single-step student denoised samples. The student predictions are then re-noised and denoised with the teacher and student before evaluating the GAN and DMD losses.

are indistinguishable from the true data distribution $p(x_0)$. To do so, we propose to train a discriminator D_ν to distinguish the generated samples $\tilde{x}_0$ from the real samples $x_0 \sim p(x_0)$. As proposed in (Lin, Wang, and Yang 2024; Sauer et al. 2024), we also apply the discriminator directly within the latent space. This approach circumvents the necessity of decoding the samples using the VAE, a process outlined in (Sauer et al. 2023), that proves to be expensive and hampers the method's scalability to high-resolution images. Drawing inspiration from (Lin, Wang, and Yang 2024; Sauer et al. 2024), we propose an approach where both the one-step student prediction $\tilde{z}_0$ and the input latent sample z_0 are re-noised following the teacher noise schedule. This process uses a timestep t' uniformly chosen from the set $[0.01, 0.25, 0.5, 0.75]$ enabling the discriminator to effectively differentiate between samples based on both high and low-frequency details (Lin, Wang, and Yang 2024). The samples are first passed through the *frozen* teacher model, followed by the trainable discriminator, to yield a real or fake prediction. When employing a UNet architecture (Ronneberger, Fischer, and Brox 2015) for the teacher model, our approach focuses on utilising only the encoder of the UNet, generating an even more compressed latent representation and further reducing the parameter count for the discriminator. The adversarial loss $\mathcal{L}_{\text{adv}}$ and discriminator loss $\mathcal{L}_{\text{dis}}$ write as follows:

$$
\begin{aligned}
\mathcal{L}_{\text{adv}} &= \frac{1}{2}\,\mathbb{E}_{z_0,t',\varepsilon}\left[\left\| D_\nu(f_\theta(z_{t'}, t')) - 1 \right\|^2\right] , \\
\mathcal{L}_{\text{dis.}} &= \frac{1}{2}\,\mathbb{E}_{z_0,t',\varepsilon}\left[\left\| D_\nu(z_0) - 1 \right\|^2 + \left\| D_\nu(f_\theta(z_{t'}, t')) \right\|^2\right] ,
\end{aligned}
\tag{4}
$$

where ν denotes the discriminator parameters. We opt for these particular losses due to their reliability and stability during training, as observed in our experiments. In practical terms, the discriminator's architecture is designed as a straightforward Convolutional Neural Network (CNN) featuring a stride of 2, a kernel size of 4, SiLU activation (Hendrycks and Gimpel 2016; Ramachandran, Zoph, and Le 2017) and group normalization (Wu and He 2018).

Figure 3: Qualitative evaluation of the sample quality as the number of NFEs increases for the proposed method applied to SD1.5 model. Best viewed zoomed in.

Distribution Matching

Inspired by the work of (Yin et al. 2023), we also propose to introduce a Distribution Matching Distillation (DMD) loss to ensure that the generated samples closely mirror the data distribution learned by the teacher. Specifically, this involves minimizing the Kullback–Leibler (KL) divergence between the student distribution $p_\theta^{\text{student}}$ and p_ϕ^{teacher}, the data distribution learned by the teacher (Wang et al. 2024):

$$
\mathcal{L}_{\text{DMD}} = D_{KL}(p_\theta^{\text{student}} || p_\phi^{\text{teacher}}) . \tag{5}
$$

Taking the gradient of the KL divergence with respect to the student model parameters θ leads to the following update:

$$
\nabla_\theta \mathcal{L}_{\text{DMD}} = \mathbb{E}\left[\left(s^{\text{student}}(y) - s^{\text{teacher}}(y))\right)\nabla f_\theta(z_t, t)\right] ,
$$

where s^{teacher} and s^{student} are the score functions of the teacher and student distributions respectively and $y = f_\theta(z_t, t)$ is the student prediction. Inspired by (Yin et al. 2023), the one-step student prediction $\tilde{z}_0$ is re-noised using a uniformly sampled timestep $t'' \sim \mathcal{U}([0, 1])$ and the teacher noise schedule. The new noisy sample is passed through the *frozen* teacher model to get the score function for the teacher distribution $s^{\text{teacher}}(f_\theta(z_{t''}, t'')) = -(\varepsilon_\phi^{\text{teacher}}(x_{t''}, t'')/\sigma(t''))$. In our approach, we utilize the student model for the score function of the student distribution, instead of a dedicated diffusion model. This choice significantly reduces the number of trainable parameters and computational costs.

Model Training

While striving for robustness and versatility, we also aimed to design a model with a minimal number of trainable parameters, since it involves the loading of computationally intensive functions (teacher and student). To do so, we propose to rely on the parameter-efficient method LoRA (Hu et al. 2021) and apply it to our student model. This way, we drastically reduce the number of parameters and speed up the training process.

In a nutshell, our student model is trained to minimize a weighted combination of the distillation Eq. (2), the adversarial Eq. (4), and the distribution matching Eq. (5) losses:

$$
\mathcal{L} = \mathcal{L}_{\text{distil}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \lambda_{\text{DMD}}\mathcal{L}_{\text{DMD}} . \tag{6}
$$

The training process is illustrated in Fig. 2 and appendices.

Experiments

In this section, we assess the effectiveness of our proposed method across various tasks and datasets. First, as it is common in the literature, we quantitatively compare the method with several approaches in the context of text-to-image generation. Then, we conduct an extensive ablation study to assess the importance and impact of each component proposed in the method. Finally, we highlight the versatility of our method across several tasks, conditioning, and denoiser architectures.

Text-to-Image Quantitative Evaluation

First, we apply our distillation approach to the publicly available SD1.5 model (Rombach et al. 2022) and report both FID (Heusel et al. 2017) and CLIP score (Radford et al. 2021) on the COCO2014 and COCO2017 datasets (Lin et al. 2014). The model is trained on the LAION dataset (Schuhmann et al. 2022) with aesthetic scores above 6. For COCO2017, we rely on the evaluation approach proposed in (Meng et al. 2023) and we pick 5,000 prompts from the validation set to generate synthetic images. For COCO2014, we follow (Kang et al. 2023) and pick 30,000 prompts from the validation set. We then compute the FID against the real images in the respective validation sets. We report the results in Tables (a) and (b) in Fig. 4. Our method achieves a FID of 22.6 and 12.27 on COCO2017 and COCO2014 respectively with only 2 NFEs corresponding to SOTA results for few steps image generation. On COCO2017, our approach also achieves a CLIP score of 30.6 and 31.1 for 2 and 4 NFEs respectively. Importantly, our method only requires the training of 26.4M parameters (out-of the 900M teacher parameters) and merely 26 H100 GPUs hours of training time. This is in stark contrast with many competitors who depend on training the entire UNet architecture of the student. See the appendices for more details on the training procedure.

Ablation Study

In this section, we conduct a comprehensive ablation study to assess the influence of the main parameters and choices made in the proposed method. For all the ablations, we train the model for 20k iterations with SD1.5 model as a teacher. All the results are reported on the COCO2017 using 2 NFEs.

Influence of the loss terms We first train the model using different loss combinations and report the results in Table (c) in Figure 4. As highlighted in the table, both $\mathcal{L}_{\mathrm{adv}}$ and $\mathcal{L}_{\mathrm{DMD}}$ have a noticeable impact on the final performance since $\mathcal{L}_{\mathrm{adv}}$ seems to allow reaching a better image quality, as indicated by lower FID, while $\mathcal{L}_{\mathrm{DMD}}$ improves prompt adherence, reflected in higher CLIP scores. Experiments conducted using only $\mathcal{L}_{\mathrm{adv}}$ and $\mathcal{L}_{\mathrm{DMD}}$ revealed notable inconsistencies and even divergence in outcomes, emphasizing the crucial contribution of the distillation loss to the method's stability and reliability. In Tables (e) and (f), we also report results for different $\mathcal{L}_{\mathrm{distil.}}$ (LPIPS (Zhang et al. 2018) and MSE) and $\mathcal{L}_{\mathrm{adv}}$ (Hinge (Lim and Ye 2017), WGAN (Arjovsky, Chintala, and Bottou 2017) and LSGAN (Mao et al. 2017)). For $\mathcal{L}_{\mathrm{distil.}}$, MSE allows to achieve better results in terms of FID and CLIP score than LPIPS. For the GAN loss, the use of LSGAN seems the best-suited choice and we also noticed that it leads to stabler trainings.

Influence of the timestep sampling In this section, we stress the influence of $\pi(t)$, the timesteps distribution. We compare the proposed timestep distribution to a uniform distribution across $K = 32$ timesteps, a normal distribution $\pi^{\mathrm{gaussian}}(t)$ centered on $t = 0.5$ and π^{sharp}, a *sharp* version of our proposed distribution that only allows sampling 4 distinct timesteps. Results are shown in Table (d) of Fig. 4. The proposed distribution significantly improves the performance compared to π^{uniform} and π^{gaussian}. Moreover, allowing to sample more than 4 distinct timesteps seems to be beneficial to the final performance since a noticeable decrease in the FID score is observed. This can be explained by the fact that the student model can distil more useful information from the teacher model by sampling a wider range of timesteps and not over-fit the 4 selected ones.

Influence of the guidance scale during training For this ablation, unlike in the previous sections, we generate samples from the teacher model using a **fixed guidance scale** ω set to either $1, 3, 5, 7, 10, 13$ or 15. We report the evolution of the FID and CLIP score accordingly in Fig 5. In line with the behavior observed with the teacher, the choice of the guidance scale has a strong impact on the final performance. While the CLIP score measuring prompt adherence tends to increase with the guidance scale, there exists a trade-off with the FID score that eventually increases with the guidance scale resulting in a potential loss of image quality. We represent by the red dot the setting that we propose which consists in uniformly sampling a guidance scale within a given range.

On the Method's Versatility

To highlight the versatility of the proposed method, we apply the same approach to diffusion models trained with different conditionings, backbones, or adapters (Mou et al. 2024).

Backbones' Study

Flash SDXL In this section, we illustrate the ability of the method to adapt to a SDXL (Podell et al. 2023) teacher model. We provide in Table 1, the FID and CLIP score computed on the 10k first prompts of COCO2014 validation set. We compare the proposed approach to several distillation methods proposed in the literature using publicly available checkpoints. Our method can outperform peers in terms of FID while maintaining quite good prompt alignment capabilities. In addition, we also provide a visual overview of the generated samples in Fig. 6 for the teacher, the trained student model and LoRA-compatible approaches proposed in the literature (LCM (Luo et al. 2023a), SDXL-lightning (Lin, Wang, and Yang 2024) and Hyper-SD (Ren et al. 2024)). Teacher samples are generated with a guidance scale of 5. For a fair comparison with competitors, we include prompts used in (Lin, Wang, and Yang 2024) for this qualitative evaluation. The proposed approach appears to be able to generate samples that are visually closer to the learned teacher distribution. In particular, HyperSD and lightning seem to struggle to generate samples that are realistic despite

Method (# NFE)	# Train. Param.	FID ↓	CLIP ↑
SD1.5 (50)	N/A	20.1	31.8
SD1.5 (16)		31.7	32.0
Prog. Distil. (2)	900M	37.3	27.0
Prog. Distil. (4)		26.0	30.0
Prog. Distil. (8)		26.9	30.0
InstaFlow (1)	900M	23.4	30.4
CFG Dist. (16)	850M	24.2	30.0
Ours (2)	26.4 M	22.6	30.6
Ours (4)		**22.5**	**31.1**

(a)

Method (# NFE)	# Train. Param.	FID ↓
DPM++† (8)	N/A	22.44
UniPC† (8)	N/A	23.30
UFOGen (1)	1,700M	12.78
InstaFlow (1)	900M	13.10
DMD† (1)	1,700M	14.93
LCM-LoRA† (1)	67.5M	77.90
LCM-LoRA† (2)		24.28
LCM-LoRA† (4)		23.62
Ours (2)	26.4M	**12.27**
Ours (4)		12.41

(b)

Loss	FID ↓	CLIP ↑
$\mathcal{L}_{\text{distil.}}$	27.12	29.85
$\mathcal{L}_{\text{distil.}} + \mathcal{L}_{\text{DMD}}$	26.88	30.45
$\mathcal{L}_{\text{distil.}} + \mathcal{L}_{\text{adv}}$	23.41	30.14
$\mathcal{L}_{\text{distil.}} + \mathcal{L}_{\text{DMD}} + \mathcal{L}_{\text{adv}}$	**22.64**	**30.61**

(c)

$\pi(t)$	FID ↓	CLIP ↑
$\pi^{\text{uniform}}(t)$	24.25	30.11
$\pi^{\text{gaussian}}(t)$	35.89	28.15
$\pi^{\text{sharp}}(t)$	23.35	30.58
$\pi^{\text{ours}}(t)$	**22.64**	**30.61**

(d)

$\mathcal{L}_{\text{distil.}}$	FID ↓	CLIP ↑
LPIPS	24.89	30.56
MSE	**22.64**	**30.61**

(e)

$\mathcal{L}_{\text{adv.}}$	FID ↓	CLIP ↑
Hinge	25.02	30.17
WGAN	24.58	30.36
LSGAN	**22.64**	**30.61**

(f)

K	FID ↓	CLIP ↑
16	23.35	30.11
32	**22.64**	**30.61**
64	22.87	30.58

(g)

Figure 4: *From left to right and top to bottom:* a) FID-5k and CLIP score on COCO2017 validation set for SD1.5 as teacher. b) FID-30k on MS COCO2014 validation set for SD1.5 as teacher († results from (Yin et al. 2023)), c) the loss terms d) the timestep sampling $\pi(t)$, e) the distillation loss, f) the GAN loss and g) the value of K in Eq. (3).

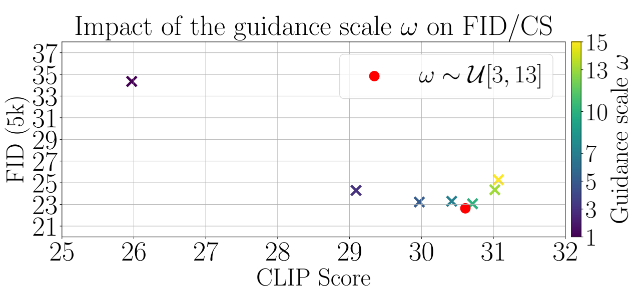

Figure 5: Influence of the guidance used with the teacher.

Model (# NFE)	FID ↓	CLIP ↑
SDXL (40)	18.4	33.9
LCM (8)	21.7	32.7
Turbo (4)	23.7	**33.7**
Lightning (4)	24.6	32.9
Lightning† (4)	25.1	32.8
HyperSD† (4)	27.8	33.3
Ours† (4)	**21.6**	32.7

Model (# NFE)	FID ↓	CLIP ↑
Pixart (40)	28.1	31.6
Ours† (4)	29.3	30.3

Model (# NFE)	FID ↓	CLIP ↑
SD3 (40)	24.4	33.5
Ours† (4)	27.5	32.8

† LoRAs

Table 1: FID and CLIP score on 10k samples of COCO2014 validation set for SDXL, Pixart-α and SD3 teacher.

creating sharp samples. See the appendices for the comprehensive experimental setup and additional comparisons. Additionally, since our student share the same architecture as the teacher, we notice that our approach can be combined with existing LoRAs in a *training-free* manner. We show at the bottom right of Fig. 7, 4 steps generations for 6 existing SDXL LoRAs directly plugged to our trained Flash SDXL model. We provide additional samples in the appendices.

Flash Pixart (DiT) In this section, we propose to apply the proposed method to a DiT denoiser backbone (Peebles and Xie 2023) using Pixart-α (Chen et al. 2023) as teacher. We compare the student generations using 4 NFEs to the teacher generations using 40 NFEs (20 steps) as well as Pixart-LCM (Luo et al. 2023b) in Fig. 6 and provide metrics in Table 1. The proposed method can generate high-quality samples that sometimes seem even more visually appealing than the teacher. Moreover, driven by the adversarial approach the student model trained with our method generates images with more vivid colors and sharper details than LCM. It is noteworthy that the student model does not lose the capability of the teacher to generate samples that are coherent with the prompt. In addition, we provide in Table 1 FID and CLIP scores computed on the 10k first prompts of COCO2014 validation set for our model and the teacher. See the appendices for the comprehensive experimental setup and additional samples as well as discussion on the variability of the output samples with respect to the prompt.

Flash SD3 (MMDiT) Finally, we also show the compatibility of our approach with the recently propose MMDiT architecture of Stable Diffusion 3 (Esser et al. 2024). The method is again able to successfully distil the teacher model and generate samples in only 4 NFEs. We train a 90.4M parameter LoRA model with a batch size of 2 and a learning rate of $1e^{-5}$ together with Adam optimizer (Kingma and Ba

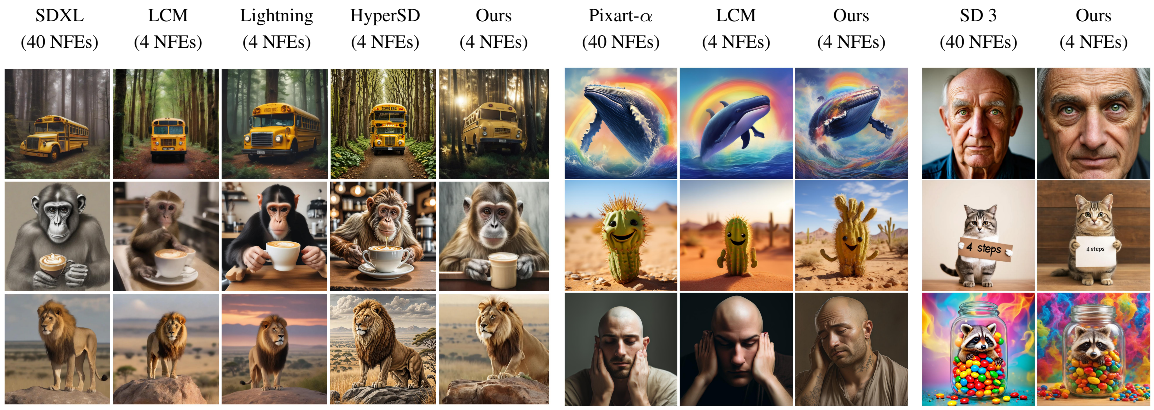

Figure 6: *From left to right:* Application of *Flash Diffusion* to SDXL (UNet), Pixart-α (DiT) and Stable Diffusion 3 (MMDiT) teachers. Teacher samples are generated with a guidance scale of 5, 3, and 5 respectively. The proposed approach is compared to LoRA based competitors and appears to be able to generate samples that are visually closer to the learned teacher distribution. Best viewed zoomed in. Additional samples are provided in the Appendices.

2014) for both the student and the discriminator. We provide in Fig. 6 samples generated with the teacher model and our method and quantitative results in Table. 1.

Conditionings' Study

Inpainting, Super-Resolution and Face-Swapping In this section, we consider 1) an *in-house inpainting* diffusion model conditioned on both a masked image, a mask, and a prompt, 2) a *super-resolution* model trained to upscale input images by a factor of 4 and 3) a *face-swapping* model conditioned on a source image and trained to replace the face of the person in the target image with the one in the source image. We show some samples in Fig. 7 using either our student model using 4 NFEs or the teacher generations using 4 steps (*i.e.* 8 NFEs) and 20 steps. As highlighted in the figure, the proposed method is able to generate samples that are visually close to the teacher generations while using far fewer NFEs demonstrating the ability of the method to adapt to different conditionings and tasks. See the appendices for the comprehensive experimental setup and additional samples.

Adapters We show the compatibility of the proposed approach with T2I adapters (Mou et al. 2024). In this case, the student model is trained to output samples conditioned on both a prompt and an additional conditioning given either with edges or a depth map. Samples are shown in Fig. 7.

Conclusion

In this paper, we proposed a new versatile, fast, and efficient distillation method for diffusion models. The proposed method relies on the training of a student model to generate samples that are close to the data distribution learned by a teacher model using a combination of a distillation loss, an adversarial loss, and a distribution matching loss. We also proposed to rely on the LoRA method to reduce the number of training parameters and speed up the training process. We evaluated the proposed method on a *text-to-image* task and showed that it can achieve SOTA results on COCO2014 and COCO2017 datasets. We also stressed and illustrated the versatility of the method by applying it to several tasks (*inpainting*, *super-resolution*, *face-swapping*), different denoiser architectures (UNet, DiT, MMDiT), and adapters where the trained student model was able to produce high-quality samples using only a few number of NFEs. Future work would consists in trying to reduce even more the number of NFEs or trying to enhance the quality of the samples by applying *Direct Preference Optimization* (Rafailov et al. 2024; Wallace et al. 2023) directly to the student model.

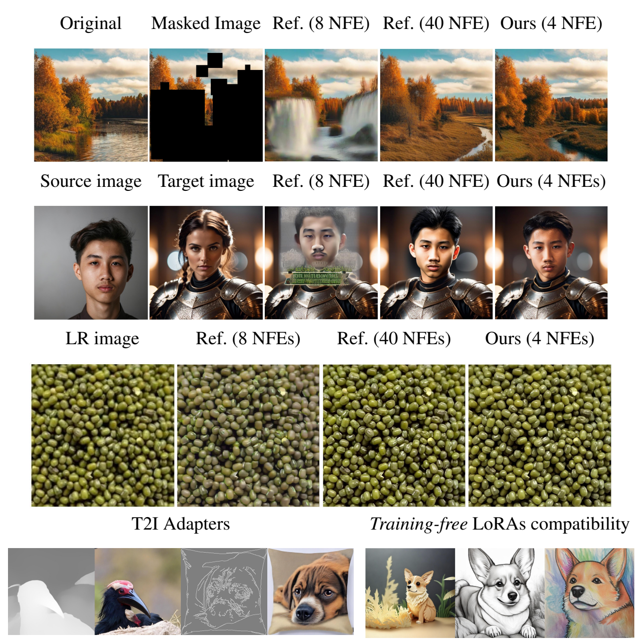

Figure 7: *From top to bottom: Flash Diffusion* applied to 1) an *inpainting* model, 2) a *face-swapping* model and 3) an *upscaler* model as well as adapters and existing LoRAs.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223. PMLR.

Chen, J.; Ge, C.; Xie, E.; Wu, Y.; Yao, L.; Ren, X.; Wang, Z.; Luo, P.; Lu, H.; and Li, Z. 2024. PixArt-Σ: Weak-to-Strong Training of Diffusion Transformer for 4K Text-to-Image Generation. *arXiv preprint arXiv:2403.04692*.

Chen, J.; Jincheng, Y.; Chongjian, G.; Yao, L.; Xie, E.; Wang, Z.; Kwok, J.; Luo, P.; Lu, H.; and Li, Z. 2023. PixArt-α: Fast Training of Diffusion Transformer for Photorealistic Text-to-Image Synthesis. In *The Twelfth International Conference on Learning Representations*.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34: 8780–8794.

Dziugaite, G. K.; Roy, D. M.; and Ghahramani, Z. 2015. Training generative neural networks via maximum mean discrepancy optimization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, 258–267.

Esser, P.; Kulal, S.; Blattmann, A.; Entezari, R.; Müller, J.; Saini, H.; Levi, Y.; Lorenz, D.; Sauer, A.; Boesel, F.; et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2672–2680.

Hendrycks, D.; and Gimpel, K. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Ho, J.; Chan, W.; Saharia, C.; Whang, J.; Gao, R.; Gritsenko, A.; Kingma, D. P.; Poole, B.; Norouzi, M.; Fleet, D. J.; et al. 2022. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Ho, J.; and Salimans, T. 2021. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

Hsiao, Y.-T.; Khodadadeh, S.; Duarte, K.; Lin, W.-A.; Qu, H.; Kwon, M.; and Kalarot, R. 2024. Plug-and-Play Diffusion Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13743–13752.

Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2021. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Kang, M.; Zhu, J.-Y.; Zhang, R.; Park, J.; Shechtman, E.; Paris, S.; and Park, T. 2023. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10124–10134.

Kim, D.; Lai, C.-H.; Liao, W.-H.; Murata, N.; Takida, Y.; Uesaka, T.; He, Y.; Mitsufuji, Y.; and Ermon, S. 2023. Consistency Trajectory Models: Learning Probability Flow ODE Trajectory of Diffusion. In *The Twelfth International Conference on Learning Representations*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kohler, J.; Pumarola, A.; Schönfeld, E.; Sanakoyeu, A.; Sumbaly, R.; Vajda, P.; and Thabet, A. 2024. Imagine Flash: Accelerating Emu Diffusion Models with Backward Distillation. *arXiv preprint arXiv:2405.05224*.

Li, Y.; Swersky, K.; and Zemel, R. 2015. Generative moment matching networks. In *International conference on machine learning*, 1718–1727. PMLR.

Li, Y.; Wang, H.; Jin, Q.; Hu, J.; Chemerys, P.; Fu, Y.; Wang, Y.; Tulyakov, S.; and Ren, J. 2024. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. *Advances in Neural Information Processing Systems*, 36.

Lim, J. H.; and Ye, J. C. 2017. Geometric gan. *arXiv preprint arXiv:1705.02894*.

Lin, S.; Wang, A.; and Yang, X. 2024. SDXL-Lightning: Progressive Adversarial Diffusion Distillation. *arXiv preprint arXiv:2402.13929*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Liu, X.; Gong, C.; et al. 2022. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *The Eleventh International Conference on Learning Representations*.

Liu, X.; Zhang, X.; Ma, J.; Peng, J.; et al. 2023. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*.

Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; and Zhu, J. 2022a. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35: 5775–5787.

Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; and Zhu, J. 2022b. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*.

Luhman, E.; and Luhman, T. 2021. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*.

Luo, S.; Tan, Y.; Huang, L.; Li, J.; and Zhao, H. 2023a. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*.

Luo, S.; Tan, Y.; Patil, S.; Gu, D.; von Platen, P.; Passos, A.; Huang, L.; Li, J.; and Zhao, H. 2023b. Lcm-lora: A universal stable-diffusion acceleration module. *arXiv preprint arXiv:2311.05556*.

Mao, X.; Li, Q.; Xie, H.; Lau, R. Y.; Wang, Z.; and Paul Smolley, S. 2017. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2794–2802.

Meng, C.; Rombach, R.; Gao, R.; Kingma, D.; Ermon, S.; Ho, J.; and Salimans, T. 2023. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14297–14306.

Mou, C.; Wang, X.; Xie, L.; Wu, Y.; Zhang, J.; Qi, Z.; and Shan, Y. 2024. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 4296–4304.

Nichol, A. Q.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; Mcgrew, B.; Sutskever, I.; and Chen, M. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In *International Conference on Machine Learning*, 16784–16804. PMLR.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4195–4205.

Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; and Rombach, R. 2023. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. In *The Twelfth International Conference on Learning Representations*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Ramachandran, P.; Zoph, B.; and Le, Q. V. 2017. Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv preprint arXiv:2204.06125*.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. In *International conference on machine learning*, 8821–8831. Pmlr.

Ren, Y.; Xia, X.; Lu, Y.; Zhang, J.; Wu, J.; Xie, P.; Wang, X.; and Xiao, X. 2024. Hyper-SD: Trajectory Segmented Consistency Model for Efficient Image Synthesis. *arXiv preprint arXiv:2404.13686*.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, 234–241. Springer.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E. L.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35: 36479–36494.

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

Sauer, A.; Boesel, F.; Dockhorn, T.; Blattmann, A.; Esser, P.; and Rombach, R. 2024. Fast High-Resolution Image Synthesis with Latent Adversarial Diffusion Distillation. *arXiv preprint arXiv:2403.12015*.

Sauer, A.; Lorenz, D.; Blattmann, A.; and Rombach, R. 2023. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*.

Schuhmann, C.; Beaumont, R.; Vencu, R.; Gordon, C.; Wightman, R.; Cherti, M.; Coombes, T.; Katta, A.; Mullis, C.; Wortsman, M.; et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35: 25278–25294.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, 2256–2265. PMLR.

Song, Y.; and Dhariwal, P. 2023. Improved Techniques for Training Consistency Models. In *The Twelfth International Conference on Learning Representations*.

Song, Y.; Dhariwal, P.; Chen, M.; and Sutskever, I. 2023. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, 32211–32252.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.

Vincent, P. 2011. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7): 1661–1674.

Wallace, B.; Dang, M.; Rafailov, R.; Zhou, L.; Lou, A.; Purushwalkam, S.; Ermon, S.; Xiong, C.; Joty, S.; and Naik, N. 2023. Diffusion model alignment using direct preference optimization. *arXiv preprint arXiv:2311.12908*.

Wang, Z.; Lu, C.; Wang, Y.; Bao, F.; Li, C.; Su, H.; and Zhu, J. 2024. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36.

Wu, Y.; and He, K. 2018. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, 3–19.

Xu, Y.; Zhao, Y.; Xiao, Z.; and Hou, T. 2023. Ufogen: You forward once large scale text-to-image generation via diffusion gans. *arXiv preprint arXiv:2311.09257*.

Yin, T.; Gharbi, M.; Park, T.; Zhang, R.; Shechtman, E.; Durand, F.; and Freeman, W. T. 2024. Improved Distribution Matching Distillation for Fast Image Synthesis. *arXiv preprint arXiv:2405.14867*.

Yin, T.; Gharbi, M.; Zhang, R.; Shechtman, E.; Durand, F.; Freeman, W. T.; and Park, T. 2023. One-step diffusion with distribution matching distillation. *arXiv preprint arXiv:2311.18828*.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3836–3847.

Zhang, Q.; and Chen, Y. 2022. Fast Sampling of Diffusion Models with Exponential Integrator. In *NeurIPS 2022 Workshop on Score-Based Methods*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Zhao, W.; Bai, L.; Rao, Y.; Zhou, J.; and Lu, J. 2024. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36.

Zheng, H.; Nie, W.; Vahdat, A.; Azizzadenesheli, K.; and Anandkumar, A. 2023. Fast sampling of diffusion models via operator learning. In *International Conference on Machine Learning*, 42390–42402. PMLR.