

# Plug-and-Play Diffusion Distillation

Yi-Ting Hsiao<sup>1,2\*</sup>, Siavash Khodadadeh<sup>2</sup>, Kevin Duarte<sup>2</sup>, Wei-An Lin<sup>2</sup>, Hui Qu<sup>2</sup>  
Mingi Kwon<sup>2\*,3</sup>, Ratheesh Kalarot<sup>2</sup>

<sup>1</sup>University of Michigan <sup>2</sup>Adobe Inc. (ASML) <sup>3</sup>Yonsei University

## Abstract

Diffusion models have shown tremendous results in image generation. However, due to the iterative nature of the diffusion process and its reliance on classifier-free guidance, inference times are slow. In this paper, we propose a new distillation approach for guided diffusion models in which an external lightweight guide model is trained while the original text-to-image model remains frozen. We show that our method reduces the inference computation of classifier-free guided latent-space diffusion models by almost half, and only requires 1% trainable parameters of the base model. Furthermore, once trained, our guide model can be applied to various fine-tuned, domain-specific versions of the base diffusion model without the need for additional training: this "plug-and-play" functionality drastically improves inference computation while maintaining the visual fidelity of generated images. Empirically, we show that our approach is able to produce visually appealing results and achieve a comparable FID score to the teacher with as few as 8 to 16 steps.

## 1. Introduction

Diffusion models [5, 24, 25] represent a novel category of generative models that have shown remarkable performance on a variety of established benchmarks in generative modeling. Specifically, conditional diffusion models [17] emerged with significantly improved sample quality by classifier-free guidance (CFG) [4].

However, the sampling speed of diffusion models stands out as a significant obstacle to their adoption in practical scenarios [20]. Specifically, the process of iteratively reducing noise in images typically requires a considerable number of iterations, posing challenges for efficient execution. For example, even when using widely adopted state-of-the-art diffusion models such as Stable Diffusion [17], more than 20 denoising steps are required to generate high-quality images. Moreover, when applying classifier-free

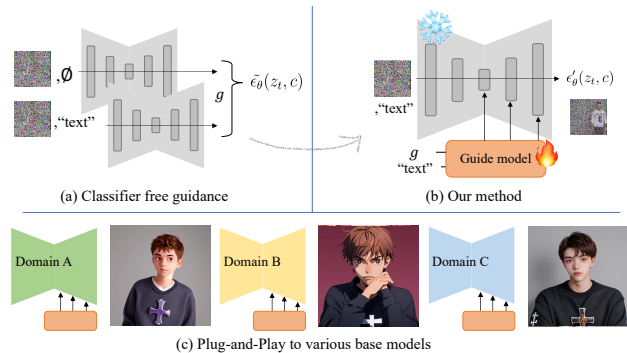


Figure 1. We trained a guide model to replace classifier-free guidance that can be plug-and-play to other base models with different domains.

guidance, two forward passes — one for the conditioned and another for the unconditioned diffusion model — are needed per denoising step, further increasing the computational cost.

One standard approach to address the speed issues is through distillation, where a student model, initialized with the weights of the teacher diffusion model, is trained to regress to the output of the teacher that runs for multiple denoising steps [11, 21]. However, the standard diffusion distillation approach has the following limitations. First, the number of trainable parameters of the student model is the same (or comparable) as that of the teacher diffusion model. But recent state-of-the-art diffusion models such as Imagen [19], eDiff-I [1], and SDXL [14] often have billions of parameters, and distilling these large models requires a tremendous amount of computation. Second, it has been shown that diffusion models can be finetuned to different domains. When finetuned on a collection of customized images, diffusion models can be adapted to generate content with novel structures and aesthetic styles. When finetuned with only a few images, prior work has shown that novel concepts can be learned [18]. However, with standard distillation on the base model, these finetuned models are no longer applicable. Re-training on the distilled student

\*Work done during internships at Adobe Inc.



Figure 2. Applying our trained guide model to different fine-tuned latent diffusion models (LDM).

model is required for all the domains of interest.

In this paper, we propose a plug-and-play distillation approach to address these issues. Specifically, we introduce a novel type of distillation that makes the parameters of the base model remain untouched: we propose an external guide model with a lightweight architecture that injects feature maps to enable the diffusion model to generate text-conditioned images on one path.

We first experiment with distilling CFG into one forward pass, which effectively reduces the inference FLOP counts by 32%. We further study different architectural choices of the lightweight module and show that the proposed architecture is around 1% of the parameters of the base, thus effectively halving the inference FLOP counts. Finally, we experiment with the generalizability of the plug-and-play module. Once our lightweight guided module is trained, it can be readily integrated with existing finetuned diffusion models, requiring minimal to no further training.

In summary, our approach has the following advantages:

- **Low computational cost for training:** The parameters required to learn from the distillation approach is only 1% (42% for the Full guide model) of the diffusion model, making the computational cost for training very low compared to other distillation methods.
- **Maintaining the weights of the base model:** Our approach maintains the conditioned diffusion model as-is, maintaining the integrity of the diffusion model.
- **Reducing inference time:** Our approach is able to decrease the FLOPs count for each sampling step by half, and is able to produce high-quality images with only 8

steps.

- **Generalizability:** After the trained model is obtained, it can adapt to different types of fine-tuned base models without retraining.
- **Adaptable with other distillation techniques:** The model can be applied to different approaches such as progressive distillation, for further sampling steps reduction.

## 2. Related work

### 2.1. Reducing inference time in diffusion models

To reduce the expensive computational cost in inference time of diffusion models, previous papers have attempted to improve the sampling speed of diffusion models.

One of the straightforward ways is designing accurate ODE samplers [6, 10, 11]. For example, Denoising Diffusion Implicit Models (DDIM) [5] uses first-order Euler’s method which enables to reduce the inference timesteps.

On the other hand, there are previous works that attempt to incorporate distillation techniques to improve model inference efficiency. Distillation in deep learning refers to a process where a larger and more complex model (referred to as the “teacher” model) is employed to train a smaller and simpler model (the “student” model). The goal of distillation is to transfer the knowledge and information captured by the teacher model to the student model, enabling the student model to achieve similar performance with reduced complexity and computational requirements. Golnari et al. [3] proposed optimizing specific denoising steps by restricting noise computation to conditional noise and eliminating unconditional noise computation, thus reducing the

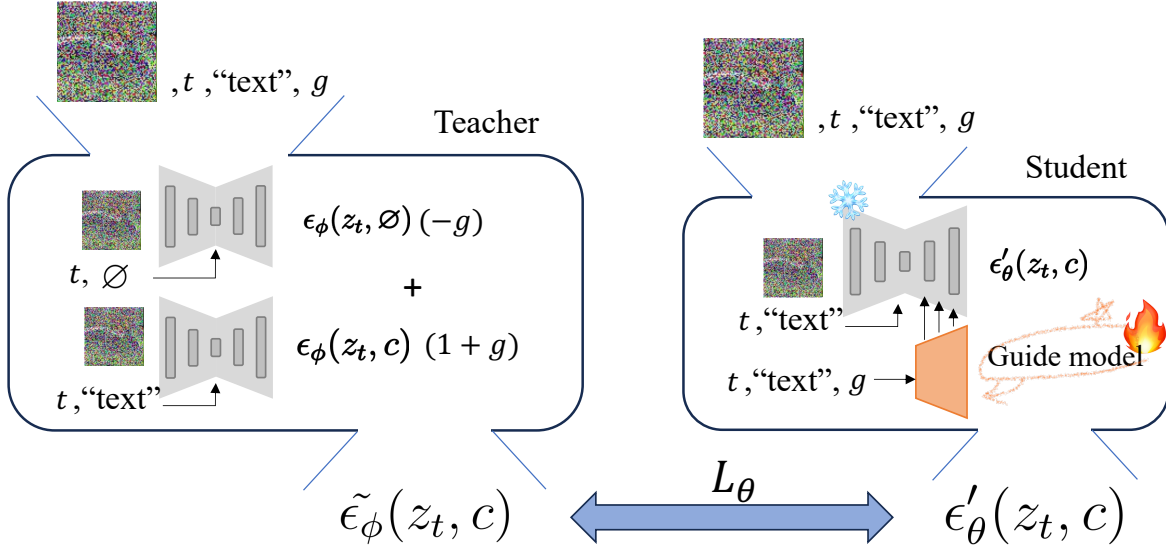


Figure 3. The overview of CFG distillation. Instead of using two feed-forward pass and classifier-free guidance, we train a student model conditioned with the guidance, cross-attention, and time embedding to predict the output image with only one forward pass.

complexity of target iterations. Salimans and Ho [21] and Meng et al. [12] distilled the model to achieve fewer sampling steps. However, these methods only focus on reducing inference timesteps and require progressive model distillation, which may require more time and computing resources to train these models.

## 2.2. Controlling diffusion models

Many researchers in the field of diffusion models have demonstrated the ability to control models using methods beyond just text input. Notably, the use of external models to inject features into diffusion models has yielded impressive results [7, 8, 26, 27]. For instance, ControlNet [27] proposed an external model that uses images, skeletons, edge maps, etc., as conditions to generate corresponding images. GLIGEN [8] successfully created desired objects within specific bounding boxes. IP-adaptor [26] introduced a method for generating images similar to a given image condition. These approaches all successfully manipulated images by injecting values into features through external models. However, these methods have focused on conditional image generation or editing, with no instances of applying them to distillation.

## 3. Preliminary

### 3.1. Background on diffusion models

Under the continuous time setting, where  $t \sim \text{Uniform}[0, 1]$ , the goal of a denoising diffusion model is to train a model  $\epsilon_\phi$  that approximates noise given the

diffused noisy real data  $x \sim p_{\text{data}}$ :

$$\mathbb{E}_{t, \epsilon, x}[\omega(\lambda_t) \|\epsilon_\phi(x_t) - \epsilon\|_2^2] \quad (1)$$

where  $\omega(\lambda_t) = \omega(\log(\alpha_t^2/\sigma_t^2))$  is a pre-defined weighted function that takes into the signal-to-noise ratio  $\lambda_t$ , which decreases monotonically with  $t$ .  $x_t$  is a latent variable that satisfied  $x \sim q(x_t|x) = \mathcal{N}(x_t; \alpha_t x, \sigma_t^2 I)$ .

After training the model  $\epsilon_\phi$ , during the sampling stage,  $x_t$  can be obtained by applying the SDE / ODE solver. For example, using DDIM:

$$x_t = \alpha_t \epsilon_\phi(x_t) + \sigma_t \frac{x_t - \alpha_t \epsilon_\phi(x_t)}{\sigma_t}, \quad s = t - \frac{1}{N} \quad (2)$$

where  $N$  is the total number of sampling steps and  $x_1 \sim \mathcal{N}(0, I)$

### 3.2. Classifier free guidance

Classifier-free guidance [4] proves to be a highly effective strategy for significantly enhancing the quality of samples in class-conditioned diffusion models. It adopted an unconditioned class identifier  $\emptyset$  as a substitute for a separate classifier that is traditionally required to create a Gaussian distribution tailored to a specific class.[2] This approach finds widespread application in extensive diffusion models, including notable examples like DALL·E2 [16], GLIDE [13], and Stable Diffusion [17]. In particular, Stable Diffusion designs the diffused forward and reverse process in the VAE latent space,  $z = E(x)$ ,  $x = D(z)$  where  $E$  and  $D$  denote the VAE encoder and decoder. In the process of generating a sample, classifier-free guidance carries out evaluations on

both conditional score estimates and unconditional score estimates. Specifically, the computation of the noise sample  $\tilde{\epsilon}_\phi(z_t, c)$  follows the formulation

$$\tilde{\epsilon}_\phi(z_t, c) = (1 + g)\epsilon_\phi(z_t, c) - g\epsilon_\phi(z_t, \emptyset), \quad (3)$$

where  $\epsilon_\phi$  is the score estimate function that is a parameterized neural network (U-Net).  $\epsilon_\phi(z_t, c)$  represents the text-conditioned term, while  $\epsilon_\phi(z_t, \emptyset)$  corresponds to the unconditional term (null text). The parameter  $g$  stands for the guidance value that scales the perturbation. In this paper, we use the Stable Diffusion’s VAE latent and omit the notation of Encoder and Decoder of VAE for brief.

## 4. Methodology

### 4.1. Overview

Inspired by ControlNet [27], we design the external guide network for CFG distillation by using the guidance number as the input condition. After the first stage of the distillation (i.e. CFG distillation) has been accomplished, we follow prior distillation techniques to reduce the sampling steps. This is accomplished by enabling the model to progressively learn how to halve the sampling steps [21]. The specifics of the whole process will be elucidated in the following sections.

### 4.2. CFG distillation

The overview of our CFG distillation method is illustrated in Figure 3. We would like to learn a model  $\epsilon'_\theta$  to achieve

$$\epsilon'_\theta(z_t, c; G(g, z_t, c)) = (1 + g)\epsilon_\phi(z_t, c) - g\epsilon_\phi(z_t, \emptyset) \quad (4)$$

where  $g$  is the guidance number,  $G$  is our student guided model,  $\epsilon_\phi(z_t, \emptyset)$  is the unconditioned U-Net forward pass, and  $\epsilon_\phi(z_t, c)$  is the conditioned U-Net forward pass. Precisely,  $G$  takes the guidance as the input hint, along with time and text embedding and  $z_t$ , then injects its output feature maps to the decoder part of the original U-Net. The feature map injection can be viewed as the “guidance strength” that helps the U-Net to trade-off between sample quality and diversity. The pseudo algorithm is listed in Algorithm 1.

Typically, distillation involves initializing an entirely new model that has the same structure as the teacher model and trying to make it learn the teacher’s output and update the parameters of the entire student network. Instead, we use a small guide model on top of the teacher model, which leads to reduced computational overhead during training because the number of parameters in the guide models is relatively small compared to the whole U-Net. Also, this approach does not discard the teacher model after distillation training, but uses the trained guide model along with the teacher U-Net for faster inference without CFG. This feature makes it applicable to “plug-and-play” to different types of fine-tuned diffusion models directly without retraining the guide model  $G$ .

---

### Algorithm 1 CFG distillation

---

**Require:** real image  $x$ , text  $c$

$G_\theta \leftarrow \eta$ . Initialize student guide model

**while** not converged **do**

    Sample a timestep  $t \sim Uniform[0, 1]$

    Sample a guidance number  $g \sim Uniform[2, 9]$

    Sampling noise  $\epsilon \sim \mathcal{N}(0, I)$

$z_t = \alpha_t x + \sigma_t \epsilon$

$e_{teacher} = (1 + g)\epsilon_\phi(z_t, c) - g\epsilon_\phi(z_t, \emptyset)$

$e = \epsilon'_\theta(z_t, c; G_\theta(g, z_t, c))$

$L_\theta = \|e_{teacher} - e\|_2^2$

$\theta \leftarrow \theta - \gamma \nabla_\theta L_\theta$

**end while**

---

### 4.3. Guide model architecture

In this section, we introduce two types of external guide model, full guide model and tiny guide model.

**full guide model** ControlNet is one of the well-designed external models for image control. When we regard the distillation with an external guide model as the external controlling, the straightforward way is using the UNet architecture of diffusion model as the guide model. To align with the original ControlNet architecture, our full guide model broadcasts the guidance number into a shape that is the same as the hint size, e.g.  $(C, H, W)$ . This straightforward strategy enables the model to have high capacity. The model architecture is depicted in Figure 4.

**tiny guide model** Although full guide model is already a well-designed guide model, this is not an efficient way because there is not as much information needed to encode with a simple guidance number. As such, we further simplify the standard ControlNet structure, tiny guide model, for our guidance-distillation framework:

$$y = Z(\gamma + Z(c_{timestep}; \Theta_{z1}) + Z(c_{text}; \Theta_{z2}); \Theta_{z3}) \quad (5)$$

$\gamma$  is the guidance vector, which is a vector of guidance number  $g$ . Moreover, the timestep embedding and text embedding will also pass through zero convolution layers, denoted  $Z(\cdot, \cdot)$ . These elements are added together and passed through the zero convolutions in the decoding layer to get the corresponding output of the guide model  $y$ . Zero convolution architecture ensures that undesirable noise or irrelevant features are not injected into the base model in the early stage of the training.

The tiny guide model simplifies the traditional ControlNet architecture by removing the encoder blocks as shown in Fig. 4. This design drastically reduces the number of parameters as  $z_t$  no longer needs to be encoded by the guide model.

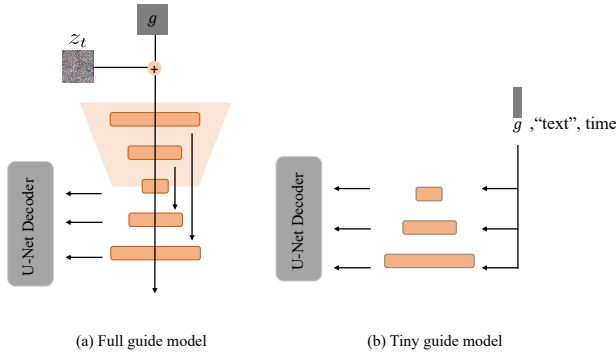


Figure 4. Comparison of the full guide model architecture and the tiny architecture.

In following sections, we will show that our CFG distillation approach works for both the full guide model and the tiny guide model.

#### 4.4. Sampling steps distillation

After training the guide model,  $G$ , we progressively distill it with fewer sampling steps required by incorporating with existing sampling-step-based distillation methods [21]. To elaborate, under the discrete time-step scenario, let  $N$  stand for the original number of sampling steps, we trained a student model to the output of two-step DDIM sampling of the teacher in one step. Precisely, the initial sampler  $f(z; \eta)$  maps a random noise  $\epsilon$  to samples  $x$  requires  $N$  steps, is distilled into a new sampler  $f(z; \theta)$  that requires  $N/2$  steps.  $f(z; \theta)$  will become the new teacher so that we can learn another sampler that requires  $N/4$  steps. This procedure will be repeated several times until the ideal sampling steps needed will be achieved. In this section, again, we only learn the parameters from the guide model  $G$  and fix the base model (U-Net) throughout the distillation progress. The small size of the guide model enables the parameters to be learned quickly.

## 5. Experiments

Distilling a diffusion model involves a balance between making the model generate images faster and maintaining good quality. Initially, we assess the image fidelity of our model through both qualitative and quantitative analyses, employing FID [23] and CLIP [15] scores. Subsequently, we evaluate the effectiveness of our approach across different domains with zero additional training. Finally, since our approach keeps the original model fixed, we can closely examine latent feature maps from the guide model to better understand how guidance is applied at different timesteps during the diffusion process.

Method	FLOPs (trillion)	# of Params (million)
Ours-full	T (338.7) + 116.5	T (859) + 361
Ours-tiny	T (338.7) + 7.79	T (859) + 8.27
Teacher $\times 2$	677.5	859

Table 1. The FLOP counts of single pass and number of parameters used for different architectures. Teacher  $\times 2$  stands for the dual pass FLOP counts for classifier-free guidance. T stands for the parameters or FLOPs counts of the base model.

### 5.1. Setup

We trained our model with LAION (512  $\times$  512) dataset [22] with the Stable Diffusion v1.5 as our score-estimation model. In the training stage, a randomly sampled guidance number  $g \in [2, 9]$  is broadcast into the shape of (C, H, W), which becomes the input of the guide model. For the tiny architecture, the input is a 1d-array with the length of  $C$  that passes through the zero modules along with timesteps and text embedding. We apply the  $\epsilon$ -prediction model through the whole experiment. Since Stable Diffusion v1.5 is trained on 1000 steps, we sample images with 1000 steps as our Teacher output for our guide model to learn. We evaluate our methods with the COCO dataset [9]. We compare our method with DDIM [28] sampling and PLMS [10] sampling. The FLOPs and number of parameters for our models compared to the teacher classifier-free guidance are listed in Table 1. We see that our full guide model only needs  $\sim 0.67$  of FLOPs of the teacher while our tiny model only computes 0.51 of FLOPs of the teacher model.

### 5.2. Qualitative and quantitative evaluation

Figure 6 illustrates the qualitative comparison between student and teacher models on various text prompts with the same initial noise. We see the quality of images generated by our guide model is close to the generated images using classifier-free-guidance while our approach can generate images and nearly half the FLOP counts. A user study associated with images from teacher and student models can be found in the Appendix.

Furthermore, we generate images with fewer timesteps on Figure 5. We do not observe obvious quality degradation when decreasing our model steps to 16 and 8 with full guide model given a certain level of guidance ( $g = 8$ ). On the other hand, since tiny guide model has less capacity, it’s challenging for the student to fully mimic the teacher’s output given a continuous guidance input during the sampling steps distillation process (i.e. progressive distillation). We observe that the tiny guide model can achieve almost the same image quality as full guide model when the sampling steps are around 50, but when the number of steps are reduced to 8, then the performance of the tiny model will degrade drastically. This can be partially addressed by training the tiny



Figure 5. The results of the full guide model and tiny guide model under 8, 16, and 50 sampling steps.



Figure 6. Disparity between the outputs of our method and classifier-free guidance with the same initial noise. Interestingly, our approach (student) usually presents a stronger contrast compared to classifier-free guidance.

model with fixed guidance. Furthermore, both of the models can achieve comparable results compared to Classifier-Free Guidance (DDIM  $N \times 2$  steps). The qualitative result is shown in Table 2.

### 5.3. Generalizability of the guide model

In this subsection, we show that plug in our guide model to different types of fine-tuned stable diffusion models from Dreambooth [18] without any additional training. Our objective is to demonstrate that our guide model can acquire a general latent representation of guidance, which possesses significant adaptability to various types of fine-tuned models without training. We focus on three different types of

fine-tuned stable diffusion v1.5 models: watercolor style, realistic style, and 3D cartoon style. Then, we directly plug in our pretrained guide model  $G$  to these fine-tuned models to modify their outputs. We run the models without classifier-free guidance and pass the guidance value to our guide module. For other distillation approaches [12], it may be necessary to distill a new model for each domain, which can be costly in terms of training and computation. Our approach removes this burden and make it easy to make different models finetuned for different domains nearly two times efficiently with no additional cost. We validate our approach by measuring FID and CLIP scores in generated images in different domains. Table 3 shows the FID scores

Methods	g = 2		g = 4		g = 6		g = 8	
	FID	CLIP	FID	CLIP	FID	CLIP	FID	CLIP
DDIM 8 × 2-step [28]	64.53	27.64	57.56	28.39	82.56	26.67	116.60	24.13
Stable Diffusion v1.5 (PLMS 50 × 2 step) [17]	17.5	25	16	26.63	18.7	26.50	21	26.60
Full 8step	109.53	27.90	59.91	29.50	34.15	29.78	29.53	29.84
Full 16 step	49.39	29.15	31.20	29.70	21.84	29.92	20.74	29.91
Full 50 step	43.05	29.62	24.00	30.13	18.47	30.13	18.19	30.05
Tiny 8 step	119.18	27.58	75.26	29.06	50.11	29.82	36.22	30.23
Tiny 16 step	70.66	28.44	42.06	29.73	29.19	30.29	23.14	30.53
Tiny 50 step	52.27	29.19	32.27	29.81	28.90	30.08	19.74	30.23
Fix guidance tiny 8 step	-	-	-	-	-	-	23.97	30.45

Table 2. LAION dataset distillation results for text-guided latent-space diffusion models (Stable-Diffusion). We calculated the FID score by sampling 10k images from COCO 30k dataset.

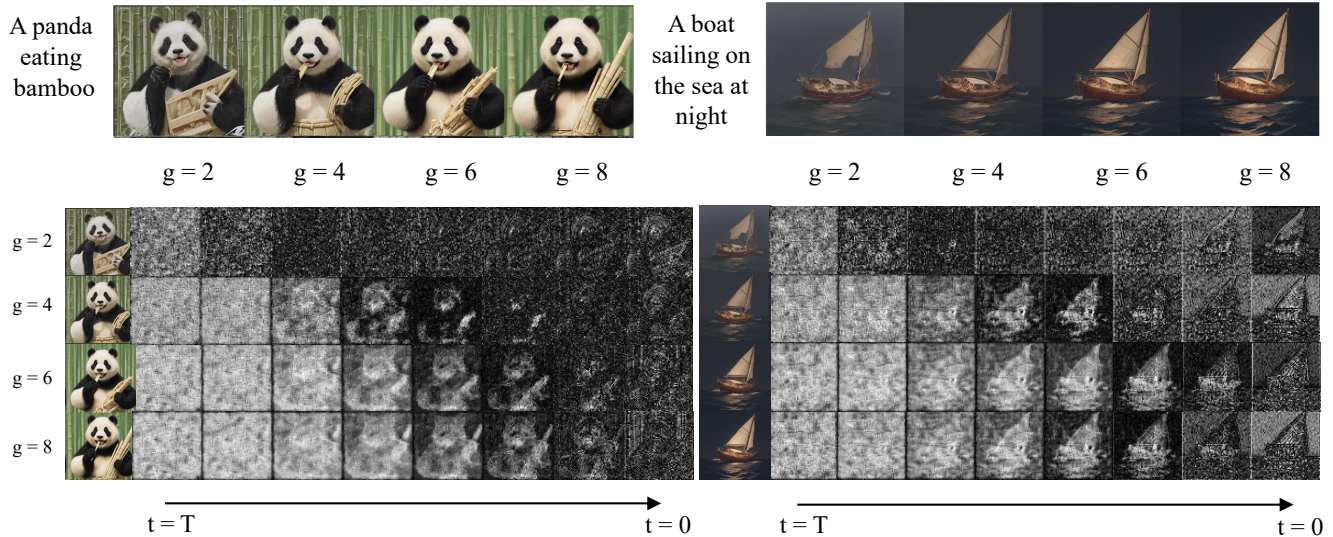


Figure 7. Visualizing the feature map injection from the guide model with guidance 8. The early stage of the iteration process has stronger injection (larger absolute values) strength, and in the later stage, the injections mainly focus on high-frequency details with lower strength. Also, the lower guidance number has lower feature map injections, higher guidance number has stronger feature map injections.

and CLIP scores of the teacher CFG on these fine-tuned models versus the results of our tiny guide model injection approach. The generated pictures are sampled with 50 steps with guidance 8. We see that FID and CLIP scores of our tiny model which runs two times faster are comparable with CFG. In addition, qualitative results are shown in Figure 2 for the full guide model plug-ins and Figure 8 for tiny guide model plug-ins. The results indicate the great generalizability of our approach without needing to train the model for different domains.

#### 5.4. Latent representations of the feature map

In this experiment, our objective is to elucidate the latent representations within the feature map injections of our

guide model  $G$ . To this end, we visualize the feature maps at various stages of the iteration process and under different guidance values. To the best of our knowledge, we are the first to visualize how classifier-free guidance emphasizes different patches of image generation in different timesteps. We are able to study this due to the architecture choice of our model that freezes the original model and adds the guide module as an additional component. By looking into feature maps of our guide module, we are able to get a better understanding of how classifier-free guidance impacts image generation.

For each layer of feature map injection, we computed the mean across various channels for each pixel and applied normalization. The number of DDIM steps used for sam-

Score	Realistic		3D Cartoon		Water Color	
	FID	CLIP	FID	CLIP	FID	CLIP
<b>Ours-Tiny</b>	19.88	31.04	22.23	31.07	41.87	30.70
<b>CFG</b>	14.71	32.15	18.00	32.11	38.42	31.37

Table 3. We calculated the CLIP scores of our guide model plug into different fine-tuned models from DreamBooth without training. Guidance values are set to 8.



Figure 8. Plug in the tiny guide model with different fine-tuned U-Nets.

pling was 50.

Figure 7 displays the feature map injections throughout the sampling process. The values indicate that the initial stages of the sampling process are more critical with respect to Classifier-Free Guidance (CFG), as this is when the primary structure of the image is formed. In the middle stage, the main subjects of the image (e.g., a panda, bamboo) are more important. Thus CFG continues to play a role in these areas, while the background becomes less significant. Finally, in the last stage of the sampling, the feature map injections mainly focus on detail refinement on the edges with low strengths (i.e. values).

Additionally, an examination of the feature maps with varying guidance values, as shown in Figure 7, reveals a clear trend: with lower guidance, the feature map injections are less pronounced, whereas higher guidance results in more robust injections that more effectively steer the original diffusion model. Visualizations of other layers of feature maps can be found in the Appendix.

## 6. Limitation

Although our method can significantly reduce the FLOP

count in a single pass while maintaining image quality, it is important to note that, unlike CFG, our approach is not as simple to run in batch of two. It requires to run U-Net and guide module in parallel. This is a disadvantage from implementation point of view, but it is important to mention that in practice larger GPU memory consumption result in slower inference time.

## 7. Conclusion

In this paper, we introduced a method for distilling guided diffusion models [4]. The approach allows us to efficiently train a lightweight model that modifies the outputs of the conditioned diffusion model while maintaining the base model parameters. We demonstrate that our technique substantially lowers the computational demands for latent-space diffusion models, which are classifier-free, by decreasing in the FLOP counts by half. Also, our method can be plug-and-play to different fine-tuned models without re-training and generate visually pleasing figures.



## References

- [1] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. **1**
- [2] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. **3**
- [3] Pareesa Ameneh Golnari, Zhewei Yao, and Yuxiong He. Selective guidance: Are all the denoising steps of guided diffusion important? *arXiv preprint arXiv:2305.09847*, 2023. **2**
- [4] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. **1, 3, 8**
- [5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. **1, 2**
- [6] Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021. **2**
- [7] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. *arXiv preprint arXiv:2210.10960*, 2022. **3**
- [8] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22511–22521, 2023. **3**
- [9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. **5**
- [10] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022. **2, 5**
- [11] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. **1, 2**
- [12] 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*, pages 14297–14306, 2023. **3, 6**
- [13] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. **3**
- [14] 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. *arXiv preprint arXiv:2307.01952*, 2023. **1**
- [15] 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*, pages 8748–8763. PMLR, 2021. **5**
- [16] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. **3**
- [17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. **1, 3, 7**
- [18] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. **1, 6**
- [19] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. **1**
- [20] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2022. **1**
- [21] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. **1, 3, 4, 5**
- [22] 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. **5**
- [23] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. <https://github.com/mseitzer/pytorch-fid>, 2020. Version 0.3.0. **5**
- [24] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. **1**
- [25] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. **1**
- [26] Hu Ye, Jun Zhang, Sibio Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-

image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. [3](#)

[27] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. [3, 4](#)

[28] Qinsheng Zhang, Molei Tao, and Yongxin Chen. gddim: Generalized denoising diffusion implicit models. *arXiv preprint arXiv:2206.05564*, 2022. [5, 7](#)