On Distillation of Guided Diffusion Models

Chenlin Meng*
Stanford University
chenlin@cs.stanford.edu

Ruiqi Gao
Google Research, Brain Team
ruiqig@google.com

Diederik P. Kingma
Google Research, Brain Team
durk@google.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Jonathan Ho
Google Research, Brain Team
jonathanho@google.com

Tim Salimans
Google Research, Brain Team
salimans@google.com

Abstract

Classifier-free guided diffusion models have recently been shown to be highly effective at high-resolution image generation, and they have been widely used in large-scale diffusion frameworks including DALL·E 2, GLIDE and Imagen. However, a downside of classifier-free guided diffusion models is that they are computationally expensive at inference time since they require evaluating two diffusion models, a class-conditional model and an unconditional model, hundreds of times. To deal with this limitation, we propose an approach to distilling classifier-free guided diffusion models into models that are fast to sample from: Given a pre-trained classifier-free guided model, we first learn a single model to match the output of the combined conditional and unconditional models, and then we progressively distill that model to a diffusion model that requires much fewer sampling steps. On ImageNet 64x64 and CIFAR-10, our approach is able to generate images visually comparable to that of the original model using as few as 4 sampling steps, achieving FID/IS scores comparable to that of the original model while being up to 256 times faster to sample from.

1 Introduction

Denoising diffusion probabilistic models (DDPMs) [16, 1, 18, 19] have achieved state-of-the-art performance on image generation [7, 11, 10, 9, 14], audio synthesis [6], molecular generation [21], and likelihood estimation [5]. Classifier-free guidance [2] further improves the sample quality of diffusion models and has been widely used in large-scale diffusion model frameworks including GLIDE [8], DALL·E 2 [9], and Imagen [14]. However, one key limitation of classifier-free guidance is its low sampling efficiency—it requires evaluating two diffusion models hundreds of times to generate one sample. This limitation has hindered the application of classifier-free guidance models in real-world settings. Although distillation approaches have been proposed for diffusion models [15, 17], these approaches are currently not applicable to classifier-free guided diffusion models. To deal with this issue, we propose a two-step distillation approach to improve the sampling efficiency of classifier-free guided models. In the first step, we introduce a single student model to match the combined output of the two diffusion models of the teacher. In the second step, we *progressively distill* the model learned from the first step to a fewer-step model use the approach introduced in [15]. Using our approach, a *single* distilled model is able to handle a wide range of different guidance strengths, allowing for the trade-off between sample quality and diversity efficiently. To sample from our model, we consider existing deterministic sampler in the literature [17, 15] and further propose a stochastic sampling process. Our experiments on ImageNet 64x64 and CIFAR-10 show that the proposed distilled model can generate samples visually comparable to that of the teacher using only 4

*Work done during an internship at Google

NeurIPS 2022 Workshop on Score-Based Methods.

steps and is able to achieve comparable FID/IS scores as the teacher model using as few as 8 to 16 steps on a wide range of guidance strengths (see Fig. 1). Additional experiments on ImageNet 64x64 also demonstrate the potential of the proposed framework in style-transfer applications [20].

Figure 1: Class-conditional samples from our two-step (deterministic) approach on ImageNet 64x64. By varying the guidance weight w, our distilled model is able to trade-off between sample diversity and quality, while achieving visually pleasant results using as few as *one* sampling step.

2 Background on diffusion models

Given samples $\mathbf{x}$ from a data distribution $p_{\text{data}}(\mathbf{x})$, noise scheduling functions α_t and σ_t, we train a diffusion model $\hat{\mathbf{x}}_{\boldsymbol{\theta}}$, with parameter $\boldsymbol{\theta}$, via minimizing the weighted mean squared error

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim U[0,1],\mathbf{x}\sim p_{\text{data}}(\mathbf{x}),\mathbf{z}_t\sim q(\mathbf{z}_t|\mathbf{x})}[\omega(\lambda_t)||\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t) - \mathbf{x}||_2^2], \tag{1}$$

where $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$ is a signal-to-noise ratio [5], $q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\mathbf{z}_t; \alpha_t\mathbf{x}, \sigma_t^2\mathbf{I})$ and $\omega(\lambda_t)$ is a pre-specified weighting function [5].

Once the diffusion model $\hat{\mathbf{x}}_{\boldsymbol{\theta}}$ is trained, one can use discrete-time DDIM sampler [17] to sample from the model. Specifically, the DDIM sampler starts with $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and updates as follows

$$\mathbf{z}_s = \alpha_s\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t) + \sigma_s\frac{\mathbf{z}_t - \alpha_t\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t)}{\sigma_t}, \; s = t - 1/N \tag{2}$$

with N the total number of sampling steps. The final sample will then be generated using $\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_0)$.

Classifier-free guidance Classifier-free guidance [2] is an effective approach shown to significantly improve the sample quality of class-conditioned diffusion models, and has been widely used in large-scale diffusion models including GLIDE [8], DALL·E 2 [9] and Imagen [14]. Specifically, it introduces a guidance weight parameter $w \in \mathbb{R}^{\geq 0}$ to trade-off between sample quality and diversity. To generate a sample, classifier-free guidance evaluates both a conditional diffusion model $\hat{\mathbf{x}}_{c,\boldsymbol{\theta}}$ and a jointly trained unconditional diffusion model $\hat{\mathbf{x}}_{\boldsymbol{\theta}}$ at each update step, using $\hat{\mathbf{x}}^w_{\boldsymbol{\theta}} = (1+w)\hat{\mathbf{x}}_{c,\boldsymbol{\theta}} - w\hat{\mathbf{x}}_{\boldsymbol{\theta}}$ as the model prediction in Eq. (2). As each sampling update requires evaluating two diffusion models, sampling with classifier-free guidance is often expensive [2].

Progressive distillation Our approach is based on *progressive distillation* [15], an effective method for improving the sampling speed of diffusion models by repeated distillation. Until now, this method could not be directly applied to distillation of guided models or to samplers other than the deterministic DDIM sampler [17]. In this paper we resolve these shortcomings.

3 Distilling a classifier-free guided diffusion model

In the following, we discuss our approach for distilling a classifier-free guided diffusion model [2]. Given a trained guided model $[\hat{\mathbf{x}}_{c,\boldsymbol{\theta}}, \hat{\mathbf{x}}_{\boldsymbol{\theta}}]$ (teacher), our approach can be decomposed into two steps.

Step one In the first step, we introduce a continuous-time student model $\hat{\mathbf{x}}_{\boldsymbol{\eta}_1}(\mathbf{z}_t, w)$, with learnable parameter $\boldsymbol{\eta}_1$, to match the output of the teacher at any time-step $t \in [0, 1]$. Given a range of guidance strengths $[w_{\text{min}}, w_{\text{max}}]$ we are interested in, we optimize the student model using the following objective

$$\boldsymbol{\eta}_1^* = \arg\min_{\boldsymbol{\eta}_1} \mathbb{E}_{w\sim U[w_{\text{min}},w_{\text{max}}],t\sim U[0,1],\mathbf{x}\sim p_{\text{data}}(\mathbf{x}),\mathbf{z}_t\sim q(\mathbf{z}_t|\mathbf{x})} \left[\omega(\lambda_t)\|\hat{\mathbf{x}}_{\boldsymbol{\eta}_1}(\mathbf{z}_t, w) - \hat{\mathbf{x}}^w_{\boldsymbol{\theta}}(\mathbf{z}_t)\|_2^2\right], \quad (3)$$

where $\hat{\mathbf{x}}^w_{\boldsymbol{\theta}}(\mathbf{z}_t) = (1 + w)\hat{\mathbf{x}}_{c,\boldsymbol{\theta}}(\mathbf{z}_t) - w\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t)$. To incorporate the guidance weight w, we introduce a w-conditioned model, where w is fed as an input to the student model. To better capture the feature, we apply Fourier embedding to w, which is then incorporated into the diffusion model backbone in a way similar to how the time-step was incorporated in [5, 15]. As initialization plays a key role in the performance, we initialize the student model with the same parameters as the conditional model of the teacher, except for the newly introduced parameters related to w-conditioning. We provide the detailed algorithm for Step-one training in Appendix C.

Step two In the second step, we consider a discrete time-step scenario and progressively distill the learned model from the first step $\hat{\mathbf{x}}_{\boldsymbol{\eta}_1}(\mathbf{z}_t, w)$ into an fewer-step student model $\hat{\mathbf{x}}_{\boldsymbol{\eta}_2}(\mathbf{z}_t, w)$ with learnable parameter $\boldsymbol{\eta}_2$, by halving the number of sampling steps each time. Letting N denote the number of sampling steps, given $w \sim U[w_{\text{min}}, w_{\text{max}}]$ and $t \in \{1, ..., N\}$, we train the student model to match the output of two-step DDIM sampling of the teacher (i.e., from t/N to $t - 0.5/N$ and from $t - 0.5/N$ to $t - 1/N$) in one step, following the approach of [15]. After distilling the $2N$ steps in the teacher model to N steps in the student model, we can use the N-step student model as the new teacher model, repeat the same procedure, and distill the teacher model into a $N/2$-step student model. At each step, we initialize the student model with the parameters of the teacher. More details are provided in Appendix C.

N-step deterministic and stochastic sampling Once the model $\hat{\mathbf{x}}_{\boldsymbol{\eta}_2}$ is trained, given a specified $w \in [w_{\text{min}}, w_{\text{max}}]$, we can perform sampling via the DDIM update rule in Eq. (2). We note that given the distilled model $\hat{\mathbf{x}}_{\boldsymbol{\eta}_2}$, this sampling procedure is *deterministic* given the initialization $\mathbf{z}_1^w$.

In fact, we can also perform N-step *stochastic* sampling: We apply one deterministic sampling step with two-times the original step-length (i.e., the same as a $N/2$-step deterministic sampler) and then perform one stochastic step backward (i.e., perturb with noise) using the original step-length, a process inspired by [4]. With $\mathbf{z}_1^w \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we use the following update rule when $t > 1/N$

$$\mathbf{z}_k^w = \alpha_k \hat{\mathbf{x}}_{\boldsymbol{\eta}_2}(\mathbf{z}_t^w) + \sigma_k \frac{\mathbf{z}_t^w - \alpha_t \hat{\mathbf{x}}^w_{\boldsymbol{\eta}_2}(\mathbf{z}_t)}{\sigma_t},\ \mathbf{z}_s^w = (\alpha_s/\alpha_k)\mathbf{z}_k^w + \sigma_{s|k}\boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (4)$$

$$\mathbf{z}_h^w = \alpha_h \hat{\mathbf{x}}_{\boldsymbol{\eta}_2}(\mathbf{z}_s^w) + \sigma_h \frac{\mathbf{z}_s^w - \alpha_s \hat{\mathbf{x}}^w_{\boldsymbol{\eta}_2}(\mathbf{z}_s^w)}{\sigma_s},\ \mathbf{z}_k^w = (\alpha_k/\alpha_h)\mathbf{z}_h^w + \sigma_{k|h}\boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (5)$$

where $h = t - 3/N$, $k = t - 2/N$, $s = t - 1/N$ and $\sigma^2_{a|b} = (1 - e^{\lambda_a - \lambda_b})\sigma_a^2$. When $t = 1/N$, we use deterministic update Eq. (2) to obtain $\mathbf{z}_0^w$ from $\mathbf{z}_{1/N}^w$. We note that compared to the *deterministic* sampler, performing *stochastic* sampling requires evaluating the model at slightly different time-steps, and would require small modifications to the training algorithm for the edge cases. We provide more details in Appendix C.

Other distillation approaches A direct application of progressive distillation [15] to guided models is to follow the structure of the teacher model and directly distill the student model into one jointly-trained conditional and unconditional model. We explore this option and observe that this approach does not work well. We provide more details and analysis in Appendix C.8.

4 Experiment

In this section, we evaluate the performance of our distillation approach. We observe that our approach is able to achieve competitive FID/IS scores while using as few as 4 steps. We provide extra experimental details in Appendix C and extra samples in Appendix D.

Distillation for classifier-free guided models We focus on ImageNet 64x64 [13] and CIFAR-10 in this experiment. We explore different ranges for the guidance weight and observe that all ranges work comparably and therefore use $[w_{min}, w_{max}] = [0, 4]$ for the experiments. We train the step-one

model using SNR loss, and the step-two model using SNR loss with truncation [15]. The baselines we consider include DDPM ancestral sampling [1] and DDIM [17]. To better understand how the guidance weight w should be incorporated, we also include models trained using a single fixed w as a baseline. We use the same pre-trained teacher model for all the methods for fair comparisons. Following [1, 2], we use a U-Net [12] architecture for the baselines, and the same U-Net backbone with the introduced w-embedding for our two-step student models (see Section 3). We report the performance for all approaches on ImageNet 64x64 in Fig. 2 and Table 1. We provide the results on CIFAR-10 and extended results on ImageNet 64x64 in Table 2 (see Appendix C.6). We provide samples for both datasets in Appendix D.

	$w = 0$		$w = 0.3$		$w = 1$		$w = 4$	
Method	FID (↓)	IS (↑)	FID (↓)	IS (↑)	FID (↓)	IS (↑)	FID (↓)	IS (↑)
Ours 1-step (D/S)	22.74 / 26.91	25.51 / 23.55	14.85 / 18.48	37.09 / 33.30	7.54 / 8.92	75.19 / 67.80	18.72 / **17.85**	157.46 / 148.97
Ours 4-step (D/S)	4.14 / 3.91	46.64 / 48.92	2.17 / 2.24	69.64 / 73.73	7.95 / 8.51	128.98 / 135.36	26.45 / 27.33	207.45 / 216.56
Ours 8-step (D/S)	2.79 / 2.44	50.72 / 55.03	**2.05** / 2.31	76.01 / 83.00	9.33 / 10.56	136.47 / 147.39	26.62 / 27.84	203.47 / **219.89**
Ours 16-step (D/S)	2.44 / **2.10**	52.53 / **57.81**	2.20 / 2.56	79.47 / **87.50**	9.99 / 11.63	139.11 / **153.17**	26.53 / 27.69	204.13 / 218.70
Single-w 1-step	19.61	24.00	11.70	36.95	**6.64**	74.41	19.857	170.69
Single-w 4-step	4.79	38.77	2.34	62.08	8.23	118.52	27.75	219.64
Single-w 8-step	3.39	42.13	2.32	68.76	9.69	125.20	27.67	218.08
Single-w 16-step	2.97	43.63	2.56	70.97	10.34	127.70	27.40	216.52
DDIM 16x2-step	7.68	37.60	5.33	60.83	9.53	112.75	21.56	195.17
DDIM 32x2-step	5.03	40.93	7.47	9.33	9.26	126.22	23.03	213.23
DDIM 64x2-step	3.74	43.16	5.52	9.51	9.53	133.17	23.64	217.88
Teacher (DDIM 1024x2-step)	2.92	44.81	2.36	74.83	9.84	139.50	23.94	224.74

Table 1: ImageNet 64x64 distillation results ($w = 0$ refers to non-guided models). For our method, D and S stand for deterministic and stochastic sampler respectively. We observe that training the model conditioned on a guidance interval $w \in [0, 4]$ performs comparably with training a model on a fixed w (see Single-w). Our approach significantly outperforms DDIM when using fewer steps, and is able to match the teacher performance using as few as 8 to 16 steps.

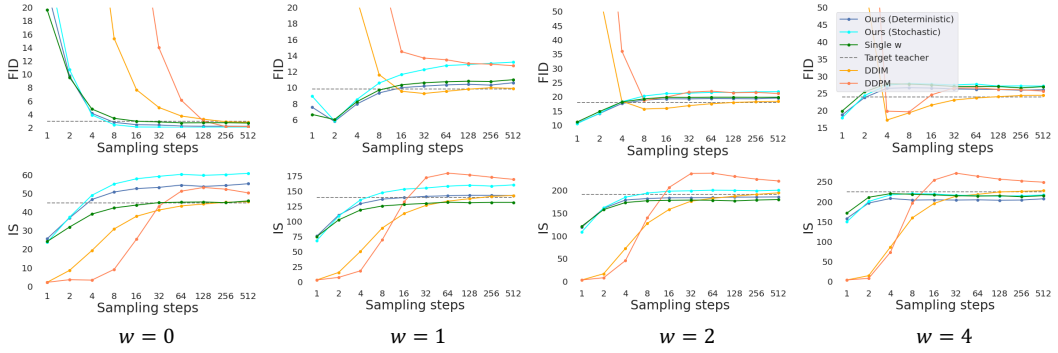


Figure 2: ImageNet 64x64 sample quality evaluated by FID and IS scores. Our distilled model significantly outperform the DDPM and DDIM baselines, and is able to match the performance of the teacher using as few as 8 to 16 steps. By varying w, a *single* distilled model is able to capture the trade-off between sample diversity and quality.

Progressive distillation for encoding In this experiment, we explore distilling the encoding process for the teacher model and perform experiments on style-transfer in a setting similar to [20]. Specifically, to perform style-transfer between two domains A and B, we encode the image from domain-A using a diffusion model trained on domain-A, and then decode with a diffusion model trained on domain-B. As the encoding process can be understood as reversing the DDIM sampling process, we perform distillation for both the encoder and decoder with classifier-free guidance, and compare with a DDIM encoder and decoder in Fig. 10. We also explore how modifying the guidance strength w can impact the performance in Fig. 11 and Fig. 12. We provide more details in Appendix C.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851, 2020.

[2] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[3] 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.

[4] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022.

[5] Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *arXiv preprint arXiv:2107.00630*, 2021.

[6] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A Versatile Diffusion Model for Audio Synthesis. *International Conference on Learning Representations*, 2021.

[7] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *International Conference on Machine Learning*, 2021.

[8] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[9] 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.

[10] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[11] 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.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[14] 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.

[15] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[16] Jascha Sohl-Dickstein, Eric A Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv preprint arXiv:1503.03585*, March 2015.

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

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

[19] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*, 2021.

[20] Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation. *arXiv preprint arXiv:2203.08382*, 2022.

[21] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

Appendix

A Related Work

Our approach is related to existing works on improving the sampling speed of diffusion models. For instance, denoising diffusion implicit model (DDIM [17]), probability flow sampler [19], fast SDE integrators [3] have been proposed to improve the sampling speed of diffusion models [16, 1]. Progressive distillation [15] is perhaps the most relevant work. Specifically, it proposes to progressively distill a pre-trained diffusion model into a fewer-step student model with the same model architecture. However, none of the above approaches have been applied to distilling classifier-free guided models. At the same time, these approaches [17, 19, 15] only consider deterministic sampling schemes to improve the sampling speed. In this work, we propose an approach to distill classifier-free guided diffusion models and further develop an effective stochastic sampling approach to sample from the distilled models.

B Conclusion

In this paper, we propose a distillation approach for guided diffusion models [2] and further propose a stochastic sampler to sample from the distilled model. Empirically, our approach is able to achieve visually decent samples using as few as one step and obtain a comparable FID/IS score as the teacher using only 8 to 16 steps, reducing the sampling steps by up to 256 times.

C Extra details on experiments

C.1 Teacher model

The model architecture we use is a U-Net model similar to the ones used in [2]. The model is parameterized to predict $\mathbf{v}$ as discussed in [15]. We use the same training setting as [2].

C.2 Step-one distillation

The model architecture we use is a U-Net model similar to the ones used in [2]. We use the same number of channels and attention as used in [2] for both ImageNet 64x64 and CIFAR-10. As mentioned in Section 3, we also make the model take w as input. Specifically, we apply Fourier embedding to w before combining with the model backbone. The way we incorporate w is the same as how time-step is incorporated to the model as used in [5, 15]. We parameterize the model to predict $\mathbf{v}$ as discussed in [15]. We train the distilled model using Algorithm 1. We train the model using SNR loss [5, 15]. For ImageNet 64x64, we use learning rate $3e-4$, with EMA decay 0.9999; for CIFAR-10, we use learning rate $1e-3$, with EMA decay 0.9999. We initialize the student model with parameters from the teacher model except for the parameters related to w-embedding.

Algorithm 1 Step-one distillation

Require: Trained classifier-free guidance teacher model $[\hat{\mathbf{x}}_{c,\boldsymbol{\theta}}, \hat{\mathbf{x}}_{\boldsymbol{\theta}}]$
Require: Data set $\mathcal{D}$
Require: Loss weight function $\omega()$
while not converged **do**
- $\mathbf{x} \sim \mathcal{D}$ ▷ Sample data
- $t \sim U[0, 1]$ ▷ Sample time
- $w \sim U[w_{\text{min}}, w_{\text{max}}]$ ▷ Sample guidance
- $\epsilon \sim N(0, I)$ ▷ Sample noise
- $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$ ▷ Add noise to data
- $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$ ▷ log-SNR
- $\hat{\mathbf{x}}^w_{\boldsymbol{\theta}}(\mathbf{z}_t) = (1+w)\hat{\mathbf{x}}_{c,\boldsymbol{\theta}}(\mathbf{z}_t) - w\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t)$ ▷ Compute target
- $L_{\boldsymbol{\eta}_1} = \omega(\lambda_t)\|\hat{\mathbf{x}}^w_{\boldsymbol{\theta}}(\mathbf{z}_t) - \hat{\mathbf{x}}_{\boldsymbol{\eta}_1}(\mathbf{z}_t, w)\|_2^2$ ▷ Loss
- $\boldsymbol{\eta}_1 \leftarrow \boldsymbol{\eta}_1 - \gamma \nabla_{\boldsymbol{\eta}_1} L_{\boldsymbol{\eta}_1}$ ▷ Optimization

end while

C.3 Step-two distillation for deterministic sampler

We use the same model architectures as the ones used in Step-one (see Appendix C.2). We train the distilled model using Algorithm 2. We first use the student model from Step-one as the teacher model. We start from 1024 DDIM sampling steps and progressively distill the student model from Step-one to a one step model. We train the student model for 50,000 parameter updates, except for sampling step equals to one or two where we train the model for 100,000 parameter updates, before the number of sampling step is halved and the student model becomes the new teacher model. At each sampling step, we initialize the student model with the parameters from the teacher model. We train the model using SNR truncation loss [5, 15]. For each step, we linearly anneal the learning rate from $1e-4$ to 0 during each parameter update. We do not use EMA decay for training. Our training setting follows the setting in [15] closely.

Algorithm 2 Step-two distillation for deterministic sampler

Require: Trained teacher model $\hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}_t, w)$
Require: Data set $\mathcal{D}$
Require: Loss weight function $\omega()$
Require: Student sampling steps N
  for K iterations **do**
    $\boldsymbol{\eta}_2 \leftarrow \boldsymbol{\eta}$ ▷ Init student from teacher
    while not converged **do**
      $\mathbf{x} \sim \mathcal{D}$
      $t = i/N, \;\; i \sim Cat[1, 2, \ldots, N]$
      $w \sim U[w_{\text{min}}, w_{\text{max}}]$ ▷ Sample guidance
      $\epsilon \sim N(0, I)$
      $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$
      `# 2 steps of DDIM with teacher`
      $t' = t - 0.5/N, \quad t'' = t - 1/N$
      $\mathbf{z}^w_{t'} = \alpha_{t'} \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}_t, w) + \frac{\sigma_{t'}}{\sigma_t}(\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}_t, w))$
      $\mathbf{z}^w_{t''} = \alpha_{t''} \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}^w_{t'}, w) + \frac{\sigma_{t''}}{\sigma_{t'}}(\mathbf{z}^w_{t'} - \alpha_{t'} \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}^w_{t'}, w))$
      $\tilde{\mathbf{x}}^w = \frac{\mathbf{z}^w_{t''} - (\sigma_{t''}/\sigma_t)\mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t)\alpha_t}$ ▷ Teacher $\hat{\mathbf{x}}$ target
      $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$
      $L_{\boldsymbol{\eta}_2} = \omega(\lambda_t) \|\tilde{\mathbf{x}}^w - \hat{\mathbf{x}}_{\boldsymbol{\eta}_2}(\mathbf{z}_t, w)\|_2^2$
      $\boldsymbol{\eta}_2 \leftarrow \boldsymbol{\eta}_2 - \gamma \nabla_{\boldsymbol{\eta}_2} L_{\boldsymbol{\eta}_2}$
    end while
    $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta}_2$ ▷ Student becomes next teacher
    $N \leftarrow N/2$ ▷ Halve number of sampling steps
  end for

C.4 Step-two distillation for stochastic sampling

We train the distilled model using Algorithm 3. We use the same model architecture and training setting as Step-two distillation described in Appendix C.3 for both ImageNet 64x64 and CIFAR-10: The main difference here is that our distillation target corresponds to taking a sampling step that is twice as large as for the deterministic sampler. We provide visualization for samples with varying guidance strengths w in Fig. 3.

C.5 Baseline samples

We provide extra samples for the DDIM baseline in Fig. 4 and Fig. 5.

Figure 3: Class-conditional samples from our two-step (stochastic) approach on ImageNet 64x64. By varying the guidance weight w, our distilled model is able to trade-off between sample diversity and quality, while achieving visually pleasant results using as few as *one* sampling step.

Figure 4: ImageNet 64x64 class-conditional generation using DDIM (baseline) 8 sampling steps. We observe clear artifacts when $w = 0$.

Figure 5: ImageNet 64x64 class-conditional generation using DDIM (baseline) 16 sampling steps.

C.6 Extra distillation results

We provide the FID and IS results for our method and the baselines on ImageNet 64x64 and CIFAR-10 in Fig. 6, Fig. 8 and Table 2. We also visualize the FID and IS trade-off curves for both datasets in Fig. 7 and Fig. 9, where we select guidance strength $w = \{0, 0.3, 1, 2, 4\}$ for ImageNet 64x64 and $w = \{0, 0.1, 0.2, 0.3, 0.5, 0.7, 1, 2, 4\}$ for CIFAR-10.

Algorithm 3 Step-two distillation for stochastic sampler

Require: Trained teacher model $\hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}_t, w)$
Require: Data set $\mathcal{D}$
Require: Loss weight function $\omega()$
Require: Student sampling steps N
  for K iterations **do**
    $\boldsymbol{\eta}_2 \leftarrow \boldsymbol{\eta}$ ▷ Init student from teacher
    while not converged **do**
      $\mathbf{x} \sim \mathcal{D}$
      $t = i/N, \;\; i \sim Cat[1, 2, \ldots, N]$
      $w \sim U[w_{\text{min}}, w_{\text{max}}]$ ▷ Sample guidance
      $\epsilon \sim N(0, I)$
      $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$
      if $t > 1/N$ **then**
        `# 2 steps of DDIM with teacher`
        $t' = t - 1/N, \quad t'' = t - 2/N$
        $\mathbf{z}^w_{t'} = \alpha_{t'} \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}_t, w) + \frac{\sigma_{t'}}{\sigma_t}(\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}_t, w))$
        $\mathbf{z}^w_{t''} = \alpha_{t''} \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}^w_{t'}, w) + \frac{\sigma_{t''}}{\sigma_{t'}}(\mathbf{z}^w_{t'} - \alpha_{t'} \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}^w_{t'}, w))$
        $\tilde{\mathbf{x}}^w = \frac{\mathbf{z}^w_{t''} - (\sigma_{t''}/\sigma_t)\mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t)\alpha_t}$ ▷ Teacher $\hat{\mathbf{x}}$ target
      else ▷ Edge case
        `# 1 step of DDIM with teacher`
        $t' = t - 1/N$
        $\mathbf{z}^w_{t'} = \alpha_{t'} \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}_t, w) + \frac{\sigma_{t'}}{\sigma_t}(\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}_t, w))$
        $\tilde{\mathbf{x}}^w = \frac{\mathbf{z}^w_{t'} - (\sigma_{t'}/\sigma_t)\mathbf{z}_t}{\alpha_{t'} - (\sigma_{t'}/\sigma_t)\alpha_t}$ ▷ Teacher $\hat{\mathbf{x}}$ target
      end if
      $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$
      $L_{\boldsymbol{\eta}_2} = \omega(\lambda_t) \|\tilde{\mathbf{x}}^w - \hat{\mathbf{x}}_{\boldsymbol{\eta}_2}(\mathbf{z}_t, w)\|_2^2$
      $\boldsymbol{\eta}_2 \leftarrow \boldsymbol{\eta}_2 - \gamma \nabla_{\boldsymbol{\eta}_2} L_{\boldsymbol{\eta}_2}$
    end while
    $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta}_2$ ▷ Student becomes next teacher
    $N \leftarrow N/2$ ▷ Halve number of sampling steps
  end for

		ImageNet 64x64		CIFAR-10	
Guidance w	Model	FID (↓)	IS (↑)	FID (↓)	IS (↑)
$w = 0.0$	Ours 1-step (D/S)	22.74 / 26.91	25.51 / 23.55	8.34 / 10.65	8.63 / 8.42
	Ours 2-step (D/S)	9.75 /10.67	36.69 / 37.12	4.48 / 4.81	9.23 / 9.30
	Ours 4-step (D/S)	4.14 / 3.91	46.64 / 48.92	3.18 / 3.28	9.50 / 9.60
	Ours 8-step (D/S)	2.79 / 2.44	50.72 / 55.03	2.86 / 3.11	9.68 / 9.74
	Ours 16-step (D/S)	2.44 / 2.10	52.53 / 57.81	2.78/3.12	9.67 / 9.76
	Single-w 1-step	19.61	24.00	6.64	8.88
	Single-w 4-step	4.79	38.77	3.14	9.47
	Single-w 8-step	3.39	42.13	2.86	9.67
	Single-w 16-step	2.97	43.63	2.75	9.65
	DDIM 16-step	7.68	37.60	10.11	8.81
	DDIM 32-step	5.03	40.93	6.67	9.17
	DDIM 64-step	3.74	43.16	4.64	9.32
	Target (DDIM 1024-step)	2.92	44.81	2.73	9.66
$w = 0.3$	Ours 1-step (D/S)	14.85 / 18.48	37.09 / 33.30	7.34 / 9.38	8.90 / 8.67
	Ours 2-step (D/S)	5.052 / 5.81	54.44 / 54.37	4.23 / 4.74	9.45 / 9.45
	Ours 4-step (D/S)	2.17 / 2.24	69.64 / 73.73	3.58 / 3.95	9.73 / 9.77
	Ours 8-step (D/S)	2.05 / 2.31	76.01 / 83.00	3.54 / 3.96	9.87 / 9.90
	Ours 16-step (D/S)	2.20 / 2.56	79.47 / 87.50	3.57 / 4.17	9.89 / 9.97
	Single-w 1-step	11.70	36.95	5.98	9.13
	Single-w 4-step	2.34	62.08	3.58	9.75
	Single-w 8-step	2.32	68.76	3.57	9.85
	Single-w 16-step	2.56	70.97	3.61	9.88
	DDIM 16-step	5.33	60.83	10.83	8.96
	DDIM 32-step	3.45	68.03	7.47	9.33
	DDIM 64-step	2.80	72.55	5.52	9.51
	Target (DDIM 1024-step)	2.36	74.83	3.65	9.83
$w = 1.0$	Ours 1-step (D/S)	7.54 / 8.92	75.19 / 67.80	8.62 / 10.27	9.21 / 8.97
	Ours 2-step (D/S)	5.77 /5.83	109.97 / 108.38	6.88 / 7.52	9.64 / 9.55
	Ours 4-step (D/S)	7.95 / 8.51	128.98 / 135.36	7.39 / 7.64	9.86 / 9.87
	Ours 8-step (D/S)	9.33 / 10.56	136.47 / 147.39	7.81 / 7.85	9.9 / 10.05
	Ours 16-step (D/S)	9.99 / 11.63	139.11 / 153.17	7.97 / 8.34	10.00 / 10.05
	Single-w 1-step	6.64	74.41	8.18	9.32
	Single-w 4-step	8.23	118.52	7.66	9.88
	Single-w 8-step	9.69	125.20	8.09	9.89
	Single-w 16-step	10.34	127.70	8.30	9.95
	DDIM 16-step	9.53	112.75	14.81	8.98
	DDIM 32-step	9.26	126.22	11.44	9.36
	DDIM 64-step	9.53	133.17	9.79	9.64
	Target (DDIM 1024-step)	9.84	139.50	7.80	9.96
$w = 2.0$	Ours 1-step (D/S)	10.71 / 10.55	118.55 / 108.37	13.23 / 14.33	9.23 / 9.02
	Ours 2-step (D/S)	14.08 / 14.18	160.04/ 161.43	12.58 / 12.57	9.51 / 9.48
	Ours 4-step (D/S)	17.61 / 18.23	178.29 / 184.45	13.83 / 13.24	9.70 / 9.77
	Ours 8-step (D/S)	18.80 / 20.25	181.53 / 193.49	14.41 / 13.67	9.77 / 9.87
	Ours 16-step (D/S)	19.25 / 21.11	183.17 / 197.71	14.80 / 14.28	9.79 / 9.84
	Single-w 1-step	11.12	120.74	13.31	9.23
	Single-w 4-step	18.14	172.74	14.04	9.70
	Single-w 8-step	19.24	176.74	14.67	9.77
	Single-w 16-step	19.81	177.69	15.04	9.79
	DDIM 16-step	15.92	157.67	20.25	8.97
	DDIM 32-step	16.85	175.72	17.27	9.29
	DDIM 64-step	17.53	182.11	15.66	9.48
	Target (DDIM 1024-step)	17.97	190.56	13.60	9.81
$w = 4.0$	Ours 1-step (D/S)	18.72 / 17.85	157.46 / 148.97	23.20 / 23.79	8.88 / 8.70
	Ours 2-step (D/S)	23.74 / 24.34	196.05 / 200.11	23.41 / 22.75	9.16 / 9.11
	Ours 4-step (D/S)	26.45 / 27.33	207.45 / 216.56	25.11 / 23.62	9.23 / 9.33
	Ours 8-step (D/S)	26.62 / 27.84	203.47 / 219.89	25.94 / 23.98	9.26 / 9.55
	Ours 16-step (D/S)	26.53 / 27.69	204.13 / 218.70	26.01 / 24.40	9.33 / 9.50
	Single-w 1-step	19.857	170.69	23.17	8.93
	Single-w 4-step	27.75	219.64	24.45	9.32
	Single-w 8-step	27.67	218.08	24.83	9.38
	Single-w 16-step	27.40	216.52	25.11	9.37
	DDIM 16-step	21.56	195.17	27.99	8.71
	DDIM 32-step	23.03	213.23	25.07	9.07
	DDIM 64-step	23.64	217.88	23.41	9.17
	Target (DDIM 1024-step)	23.94	224.74	21.28	9.54

Table 2: Distillation results on ImageNet 64x64 and CIFAR-10 ($w = 0$ refers to non-guided models). For our method, D and S stand for deterministic and stochastic sampler respectively. We observe that training the model conditioned on an guidance interval $w \in [0, 4]$ performs comparably with training a model on a fixed w (see Single-w). Our approach significantly outperforms DDIM when using fewer steps, and is able to match the teacher performance using as few as 8 to 16 steps.

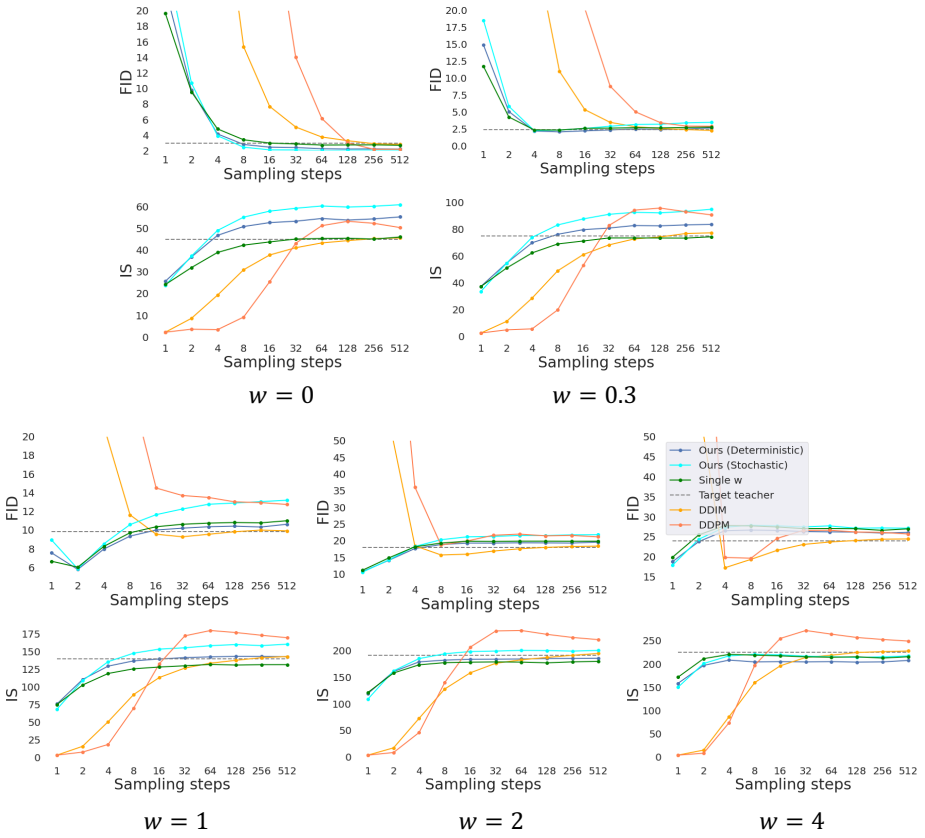


Figure 6: ImageNet 64x64 sample quality evaluated by FID and IS scores. Our distilled model significantly outperform the DDPM and DDIM baselines, and is able to match the performance of the teacher using as few as 8 steps. By varying w, our distilled model is able to capture the trade-off between sample diversity and quality.

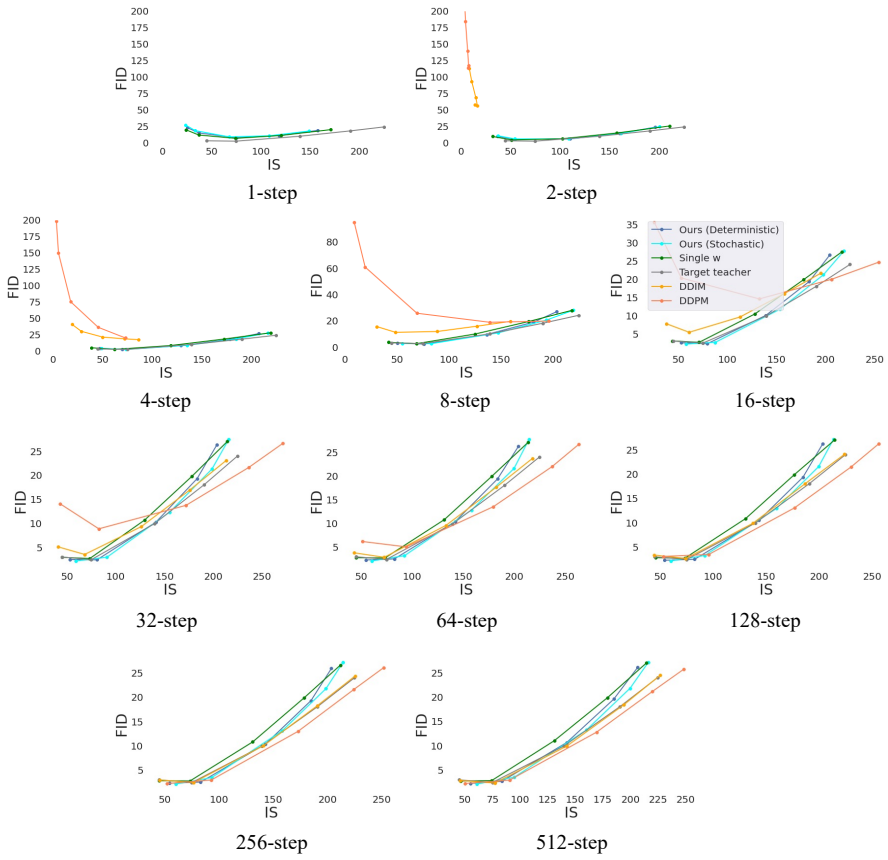


Figure 7: FID and IS score trade-off on ImageNet 64x64. We plot the results using guidance strength $w = \{0, 0.3, 1, 2, 4\}$. For the `1-step` plot, the curves of DDIM and DDPM are too far away to be visualized.

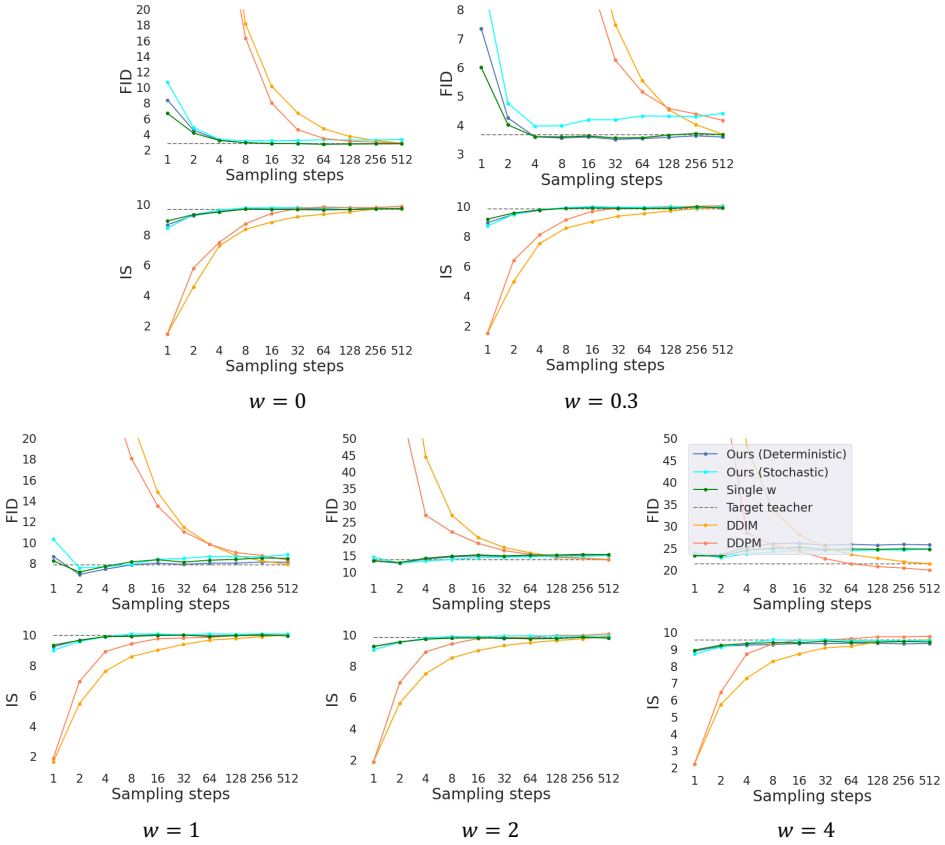


Figure 8: CIFAR-10 sample quality evaluated by FID and IS scores. Our distilled model significantly outperform the DDPM and DDIM baselines, and is able to match the performance of the teacher using as few as 8 steps. By varying w, our distilled model is able to capture the trade-off between sample diversity and quality.

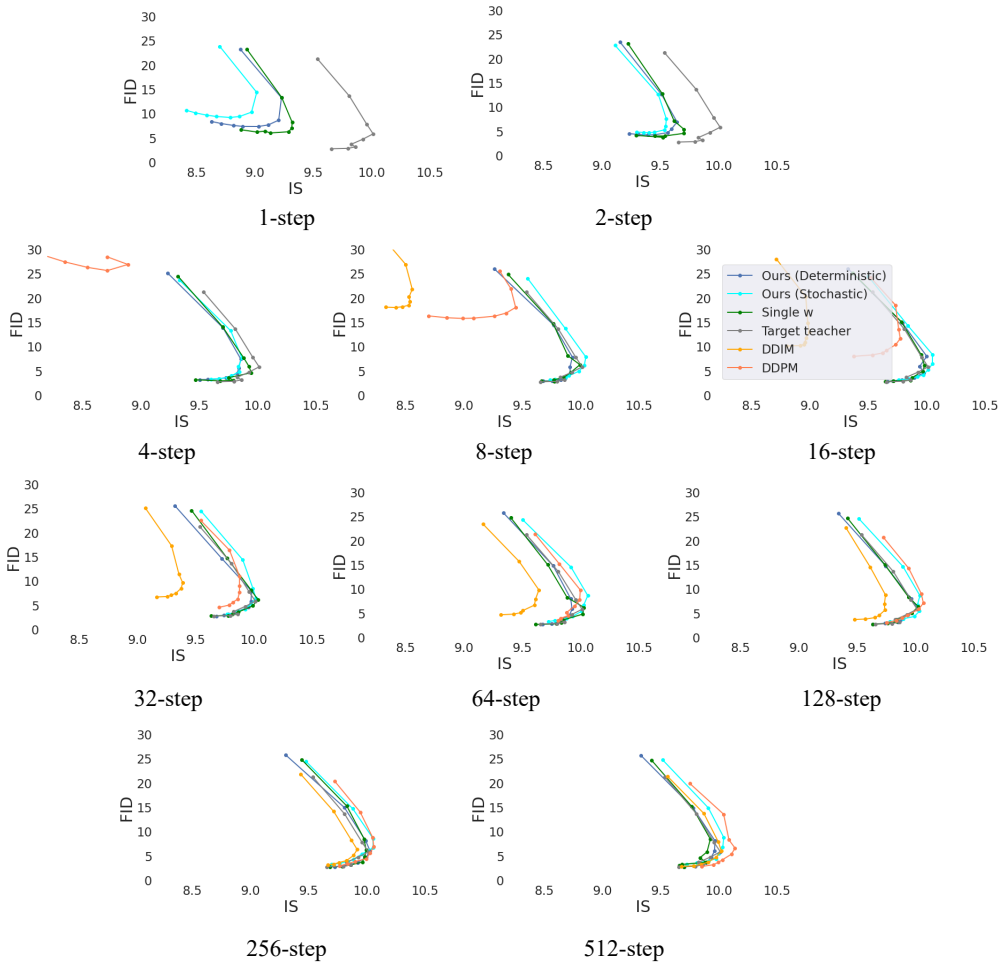


Figure 9: FID and IS score trade-off on CIFAR-10. We plot the results using guidance strength $w = \{0, 0.1, 0.2, 0.3, 0.5, 0.7, 1, 2, 4\}$. For the `1-step` and `2-step` plots, the curves of DDIM and DDPM are too far away to be visualized. For the `4-step` plot, the curve of DDIM is too far away to be visualized.

C.7 Style transfer

We focus on ImageNet 64x64 for this experiment. As discussed in [20], one can perform style-transfer between domain A and B by encoding (performing reverse DDIM) an image using a diffusion model train on domain A and then decoding using DDIM with a diffusion model trained on domain B. We train the model using Algorithm 4. We use the same w-conditioned model architecture and training setting as discussed in Appendix C.3.

Figure 10: Style transfer comparison on ImageNet 64x64. For our approach, we use a distilled encoder and decoder. For the baseline, we encode and decode using DDIM. We use $w = 0$ and 16 sampling steps for both the encoder and decoder. We observe that our method achieves more realistic outputs.

Figure 11: Style transfer on ImageNet 64x64 (`orange` to `bell pepper`). We use a distilled 16-step encoder and decoder. We fix the encoder guidance strength to be 0 and vary the decoder guidance strength from 0 to 4. As we increase w, we notice a trade-off between sample diversity and sharpness.

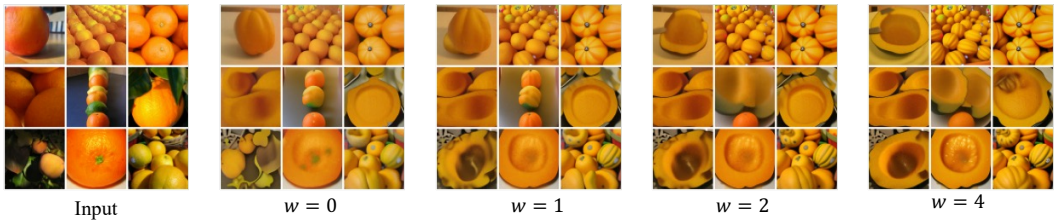


Figure 12: Style transfer on ImageNet 64x64 (`orange` to `acorn squash`). We use a distilled 16-step encoder and decoder. We fix the encoder guidance strength to be 0 and vary the decoder guidance strength from 0 to 4. As we increase the guidance strength w, we notice a trade-off between sample diversity and sharpness.

Algorithm 4 Encoder distillation

Require: Trained teacher model $\hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}_t, w)$
Require: Data set $\mathcal{D}$
Require: Loss weight function $\omega()$
Require: Student sampling steps N
  for K iterations **do**
    $\boldsymbol{\eta}_2 \leftarrow \boldsymbol{\eta}$ ▷ Init student from teacher
    while not converged **do**
      $\mathbf{x} \sim \mathcal{D}$
      $t = i/N, \;\; i \sim Cat[0, 1, \dots, N-1]$
      $w \sim U[w_{\text{min}}, w_{\text{max}}]$ ▷ Sample guidance
      $\epsilon \sim N(0, I)$
      $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$
      `# 2 steps of reversed DDIM with teacher`
      $t' = t + 0.5/N, \quad t'' = t + 1/N$
      $\mathbf{z}^w_{t'} = \alpha_{t'} \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}_t, w) + \frac{\sigma_{t'}}{\sigma_t}(\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}_t, w))$
      $\mathbf{z}^w_{t''} = \alpha_{t''} \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}^w_{t'}, w) + \frac{\sigma_{t''}}{\sigma_{t'}}(\mathbf{z}^w_{t'} - \alpha_{t'} \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}^w_{t'}, w))$
      $\tilde{\mathbf{x}}^w = \frac{\mathbf{z}^w_{t''} - (\sigma_{t''}/\sigma_t)\mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t)\alpha_t}$ ▷ Teacher $\hat{\mathbf{x}}$ target
      $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$
      $L_{\boldsymbol{\eta}_2} = \omega(\lambda_t) \|\tilde{\mathbf{x}}^w - \hat{\mathbf{x}}_{\boldsymbol{\eta}_2}(\mathbf{z}_t, w)\|_2^2$
      $\boldsymbol{\eta}_2 \leftarrow \boldsymbol{\eta}_2 - \gamma \nabla_{\boldsymbol{\eta}_2} L_{\boldsymbol{\eta}_2}$
    end while
    $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta}_2$ ▷ Student becomes next teacher
    $N \leftarrow N/2$ ▷ Halve number of sampling steps
  end for

C.8 Naive progressive distillation

A natural approach to apply progressive distillation [15] to a guided model is to use a student model that follows the same structure as the teacher—that is with a jointly trained conditional and unconditional diffusion component. Denote the pre-trained teacher model $[\hat{\mathbf{x}}_{c,\boldsymbol{\theta}}, \hat{\mathbf{x}}_{\boldsymbol{\theta}}]$ and the student model $[\hat{\mathbf{x}}_{c,\boldsymbol{\eta}}, \hat{\mathbf{x}}_{\boldsymbol{\eta}}]$, we provide the training algorithm in Algorithm 5. To sample from the trained model, we can use DDIM deterministic sampler [17] or the proposed stochastic sampler (see Eq. (4)). We follow the training setting in Appendix C.3, use a w-conditioned model and train the model to condition on the guidance strength $[0, 4]$. We observe that the model distilled with Algorithm 5 is not able to generate reasonable samples when the number of sampling is small. We provide the generated samples on CIFAR-10 with DDIM sampler in Fig. 13, and the FID/IS scores in Table 3.

Guidance w	Number of step	FID (↓)	IS (↑)
$w = 0.0$	1	212.20	3.66
	16	42.02	7.95
	64	35.37	8.47
	128	29.74	8.87
	256	20.14	9.50
$w = 0.3$	1	213.07	3.62
	16	48.74	7.70
	128	34.28	8.57
	256	24.54	9.21
$w = 1.0$	1	214.88	3.54
	16	64.92	7.21
	64	48.54	7.62
	128	42.56	8.00
	256	32.20	8.81
$w = 2.0$	1	217.37	3.48
	16	87.19	6.50
	64	57.15	7.22
	128	50.30	7.53
	256	39.76	8.26
$w = 4.0$	1	220.11	3.45
	16	115.57	6.16
	64	71.45	6.78
	128	61.75	7.02
	256	49.21	7.69

Table 3: Naive progressive distillation results on CIFAR-10. We observe that the naive distillation approach is not able to achieve strong performance.

(a) 256-step (b) 64-step (c) 16-step (d) 1-step

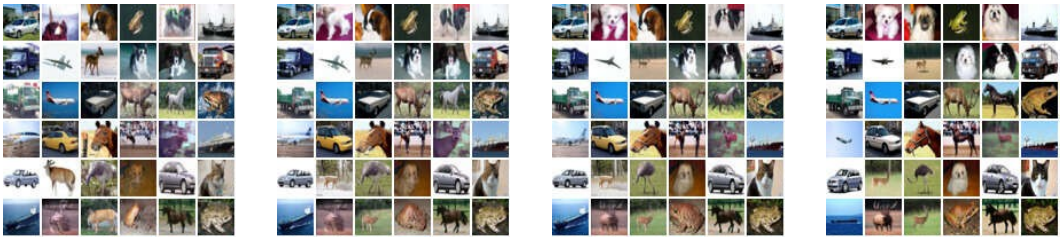
Figure 13: A naive application of progressive distillation [15] to guided distillation models. The model is trained with guidance strength $w \in [0, 4]$ on CIFAR-10. The samples are generated with DDIM (deterministic) sampler at $w = 0$. We observe clear artifacts when the number of sampling step is small.

Algorithm 5 Two-student progressive distillation

Require: Trained classifier-free guidance teacher model $[\hat{\mathbf{x}}_{c,\boldsymbol{\theta}}, \hat{\mathbf{x}}_{\boldsymbol{\theta}}]$
Require: Data set $\mathcal{D}$
Require: Loss weight function $\omega()$
Require: Student sampling steps N
  for K iterations **do**
    $\boldsymbol{\eta} \leftarrow \boldsymbol{\theta}$ ▷ Init student from teacher
    while not converged **do**
      $\mathbf{x} \sim \mathcal{D}$
      $t = i/N, \;\; i \sim Cat[1, 2, \ldots, N]$
      $w \sim U[w_{\text{min}}, w_{\text{max}}]$ ▷ Sample guidance
      $\epsilon \sim N(0, I)$
      $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$
      $\hat{\mathbf{x}}^w_{\boldsymbol{\theta}}(\mathbf{z}_t) = (1 + w)\hat{\mathbf{x}}_{c,\boldsymbol{\theta}}(\mathbf{z}_t) - w\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t)$ ▷ Compute target
      `# 2 steps of DDIM with teacher`
      $t' = t - 0.5/N, \quad t'' = t - 1/N$
      $\mathbf{z}^w_{t'} = \alpha_{t'}\hat{\mathbf{x}}^w_{\boldsymbol{\theta}}(\mathbf{z}_t) + \frac{\sigma_{t'}}{\sigma_t}(\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}^w_{\boldsymbol{\theta}}(\mathbf{z}_t))$
      $\mathbf{z}^w_{c,t''} = \alpha_{t''}\hat{\mathbf{x}}_{c,\boldsymbol{\theta}}(\mathbf{z}^w_{t'}) + \frac{\sigma_{t''}}{\sigma_{t'}}(\mathbf{z}^w_{t'} - \alpha_{t'} \hat{\mathbf{x}}_{c,\boldsymbol{\theta}}(\mathbf{z}^w_{t'}))$
      $\tilde{\mathbf{x}}^w_c = \frac{\mathbf{z}^w_{c,t''} - (\sigma_{t''}/\sigma_t)\mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t)\alpha_t}$ ▷ Conditional teacher $\hat{\mathbf{x}}$ target
      $\mathbf{z}^w_{t''} = \alpha_{t''}\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}^w_{t'}) + \frac{\sigma_{t''}}{\sigma_{t'}}(\mathbf{z}^w_{t'} - \alpha_{t'} \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}^w_{t'}))$
      $\tilde{\mathbf{x}}^w = \frac{\mathbf{z}^w_{t''} - (\sigma_{t''}/\sigma_t)\mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t)\alpha_t}$ ▷ Unconditional teacher $\hat{\mathbf{x}}$ target
      $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$
      $L_{\boldsymbol{\eta}} = \omega(\lambda_t)(\|\tilde{\mathbf{x}}^w_c - \hat{\mathbf{x}}_{c,\boldsymbol{\eta}}(\mathbf{z}_t, w)\|_2^2 + \|\tilde{\mathbf{x}}^w - \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}_t, w)\|_2^2)$
      $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta} - \gamma \nabla_{\boldsymbol{\eta}} L_{\boldsymbol{\eta}}$
    end while
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\eta}$ ▷ Student becomes next teacher
    $N \leftarrow N/2$ ▷ Halve number of sampling steps
  end for

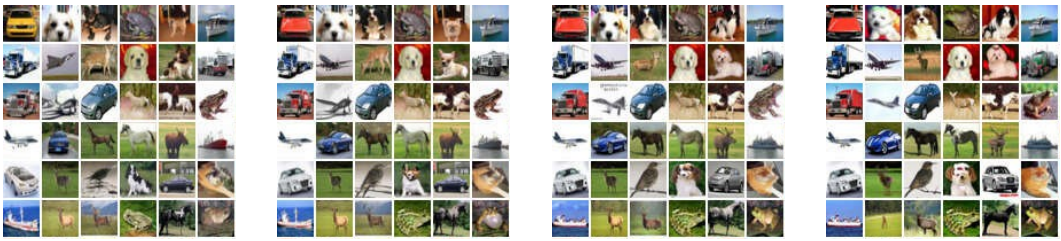
D Extra samples

D.1 CIFAR-10 256-step samples



(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 14: Ours (deterministic). Distilled 256 sampling steps.

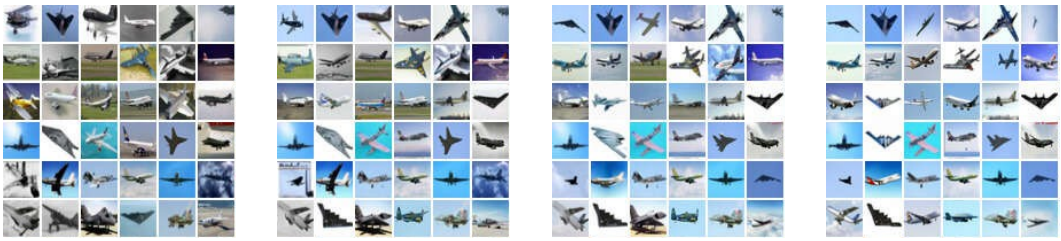


(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 15: Ours (stochastic). Distilled 256 sampling steps.

(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 16: Ours (deterministic). Distilled 256 sampling steps. Class-conditioned samples.



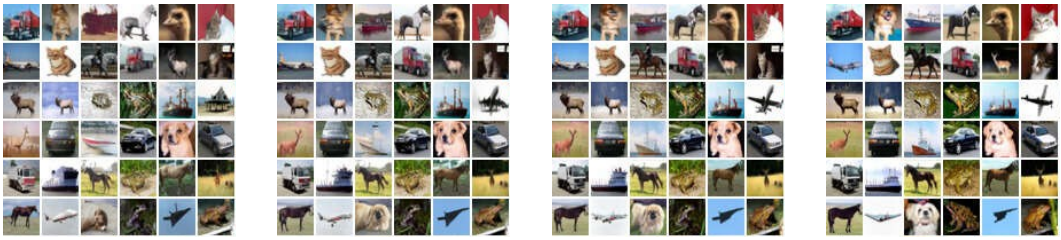
(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 17: Ours (stochastic). Distilled 256 sampling steps. Class-conditioned samples.

D.2 CIFAR-10 4-step samples

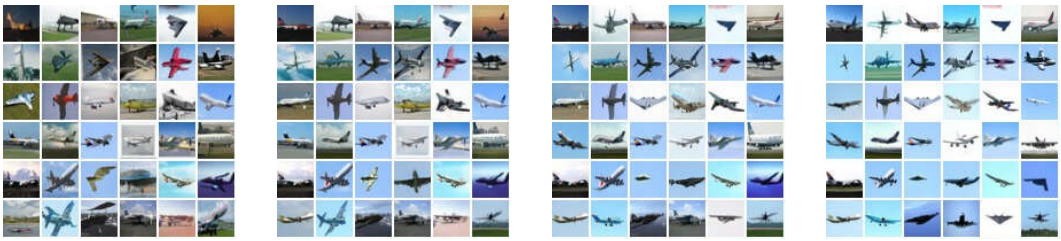
(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 18: Ours (deterministic). Distilled 4 sampling steps.



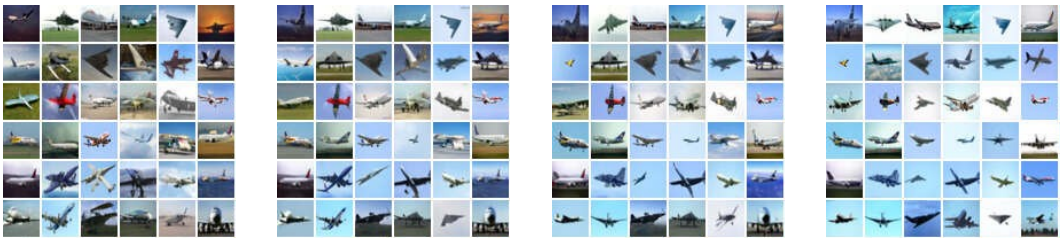
(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 19: Ours (stochastic). Distilled 4 sampling steps.



(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

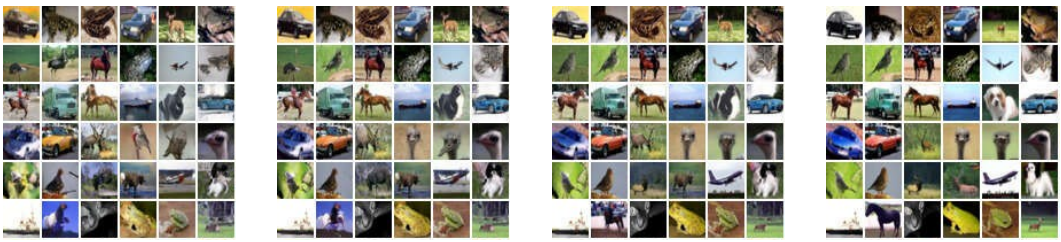
Figure 20: Ours (deterministic). Distilled 4 sampling steps. Class-conditioned samples.



(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

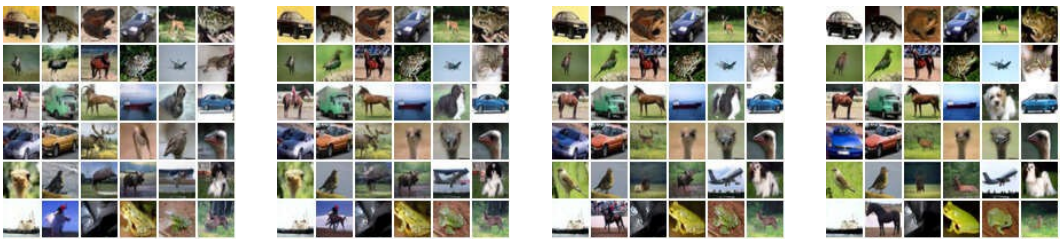
Figure 21: Ours (stochastic). Distilled 4 sampling steps. Class-conditioned samples.

D.3 CIFAR-10 2-step samples



(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 22: Ours (deterministic). Distilled 2 sampling steps.



(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 23: Ours (stochastic). Distilled 2 sampling steps.

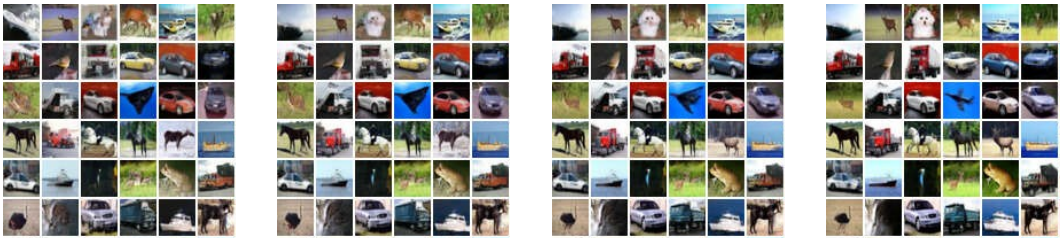
(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 24: Ours (deterministic). Distilled 2 sampling steps. Class-conditioned samples.

(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

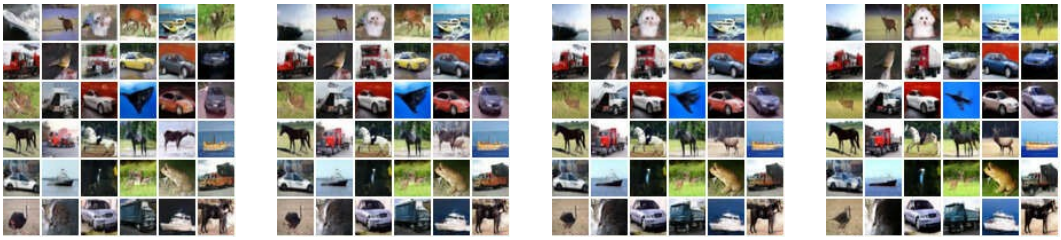
Figure 25: Ours (stochastic). Distilled 2 sampling steps. Class-conditioned samples.

D.4 CIFAR-10 1-step samples



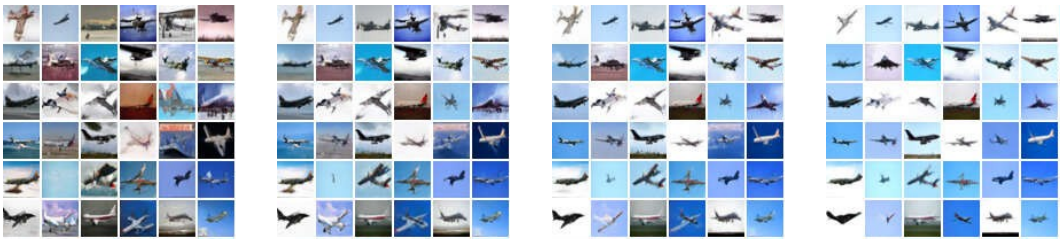
(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 26: Ours (deterministic). Distilled 1 sampling step.



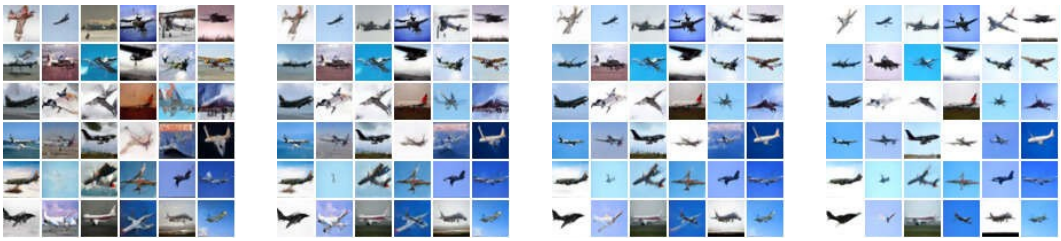
(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 27: Ours (stochastic). Distilled 1 sampling step.



(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

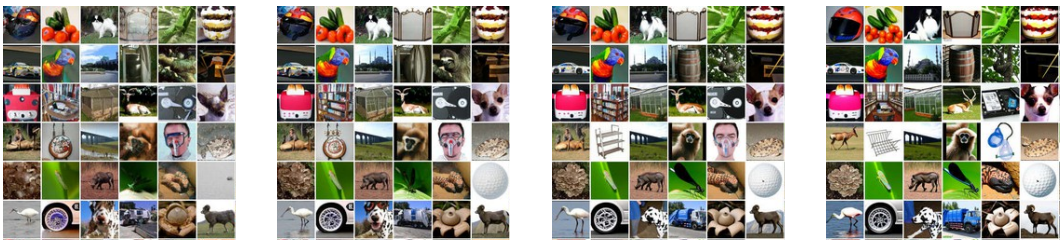
Figure 28: Ours (deterministic). Distilled 1 sampling step. Class-conditioned samples.



(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

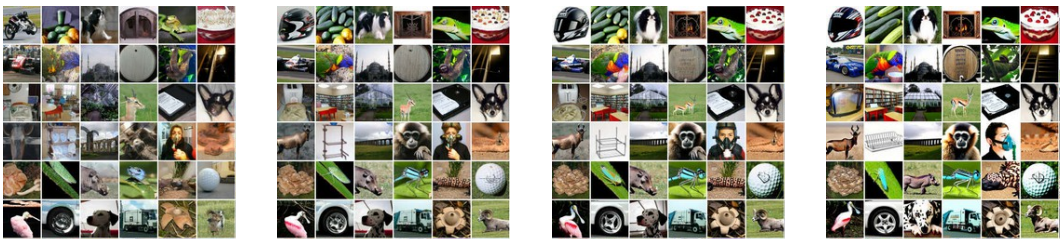
Figure 29: Ours (stochastic). Distilled 1 sampling step. Class-conditioned samples.

D.5 ImageNet 64x64 256-step samples



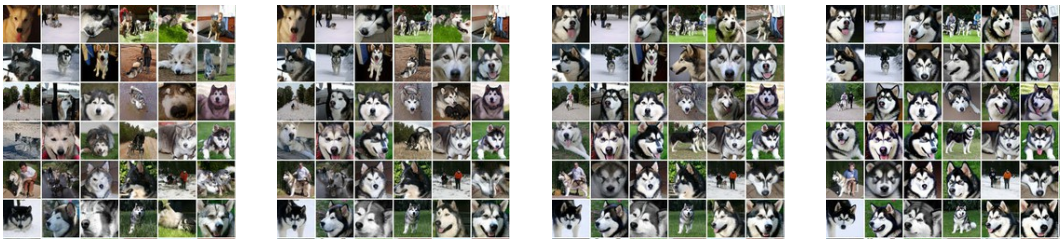
(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 30: Ours (deterministic). Distilled 256 sampling steps.



(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 31: Ours (stochastic). Distilled 256 sampling steps.



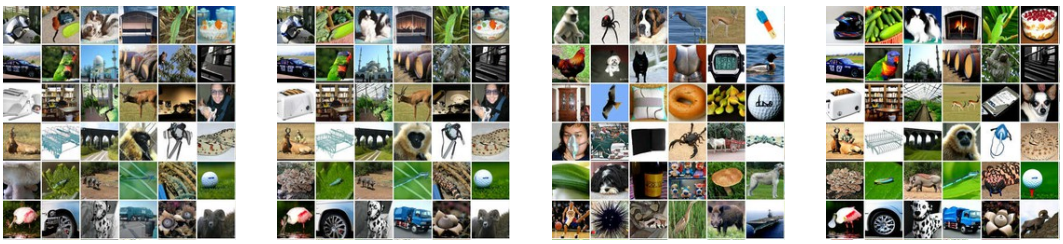
(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 32: Ours (deterministic). Distilled 256 sampling steps. Class-conditioned samples.

(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

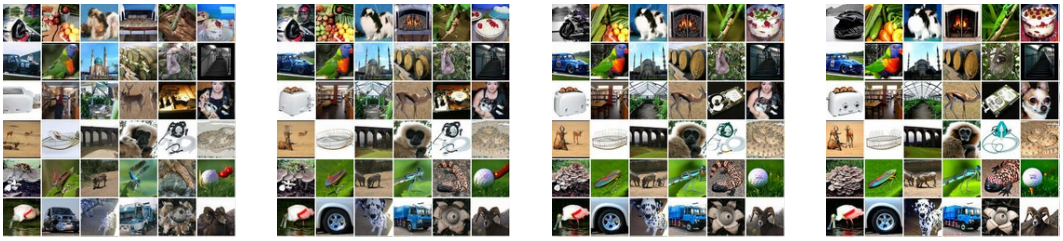
Figure 33: Ours (stochastic). Distilled 256 sampling steps. Class-conditioned samples.

D.6 ImageNet 64x64 8-step samples



(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 34: Ours (deterministic). Distilled 8 sampling step.

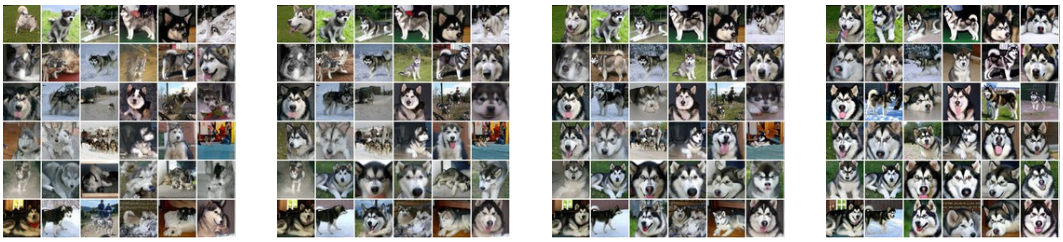


(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 35: Ours (stochastic). Distilled 8 sampling step.

(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

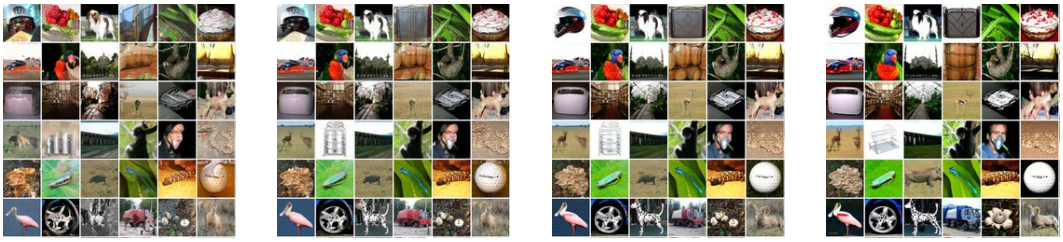
Figure 36: Ours (deterministic). Distilled 8 sampling step. Class-conditioned samples.



(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

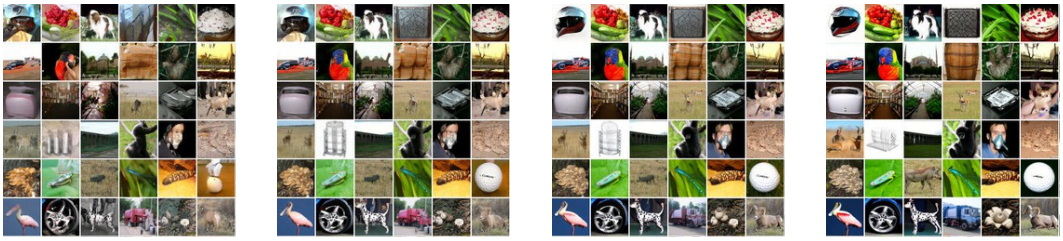
Figure 37: Ours (stochastic). Distilled 8 sampling step. Class-conditioned samples.

D.7 ImageNet 64x64 2-step samples



(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 38: Ours (deterministic). Distilled 2 sampling steps.



(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 39: Ours (stochastic). Distilled 2 sampling steps.

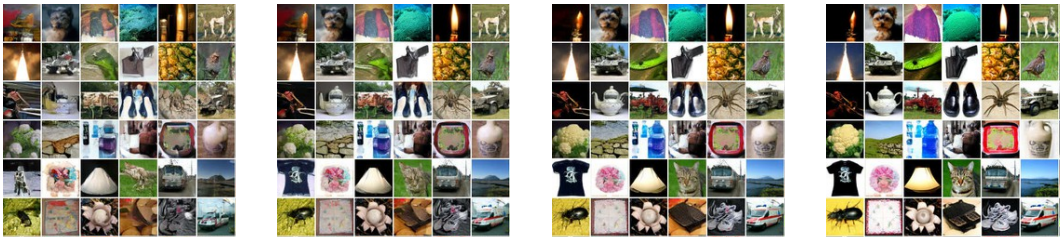
(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 40: Ours (deterministic). Distilled 2 sampling steps. Class-conditioned samples.

(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

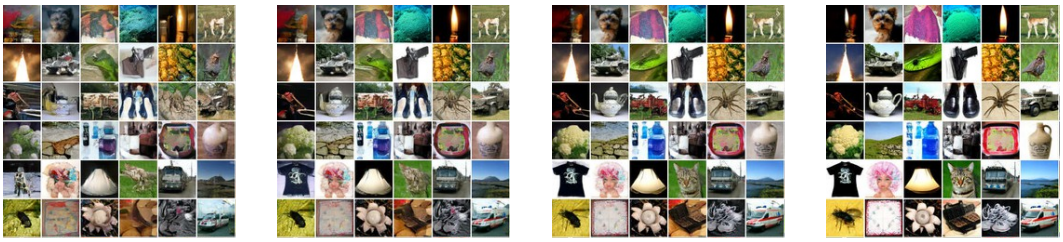
Figure 41: Ours (stochastic). Distilled 2 sampling steps. Class-conditioned samples.

D.8 ImageNet 64x64 1-step samples



(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 42: Ours (deterministic). Distilled 1 sampling step.

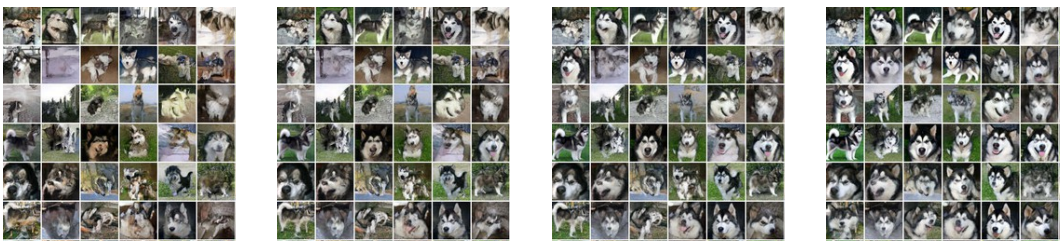


(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 43: Ours (stochastic). Distilled 1 sampling step.

(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 44: Ours (deterministic). Distilled 1 sampling step. Class-conditioned samples.



(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 4$

Figure 45: Ours (stochastic). Distilled 1 sampling step. Class-conditioned samples.