Distilling the Knowledge in Diffusion Models

Tim Dockhorn[1,2] † Robin Rombach[3] * Andreas Blatmann[3] * Yaoliang Yu[1,2]

[1]University of Waterloo [2]Vector Institute [3]Stability AI

Abstract

Large-scale diffusion models have achieved unprecedented results in (conditional) image synthesis, however, they generally require a large amount of GPU memory and are slow at inference time. To overcome this limitation, we propose to distill the knowledge of pre-trained (teacher) diffusion models into smaller student diffusion models via an approximate score matching objective. For classifier-free guided generation on CIFAR-10, our student model achieves a FID-5K of 8.03 using 273G flops. In comparison, the larger teacher model only achieves a FID-5K of 294 using 424G flops. We present initial experiments on distilling the knowledge of Stable Diffusion, a large scale text-to-image diffusion model, and discuss several promising future directions.

1. Introduction

Diffusion models (DMs) achieve both state-of-the-art synthesis quality and sample diversity using an iterative sampling process. In computer vision DMs have been used for (conditional) image [8, 18, 19, 35, 39, 40] and (conditional) video [20, 48, 58] synthesis, super-resolution [26, 44], deblurring [24, 55], image editing and inpainting [31, 34, 41, 43], conditional and semantic image generation [2, 5, 28, 36, 38], image-to-image translation [43, 47, 53], inverse problems in medical imaging [6, 7, 21, 32, 37, 51, 57], and differentially private image synthesis [9, 13].

In particular large-scale text-to-image DMs have recently gained a lot of attention, being able to synthesize high-resolution photorealistic images [39, 40, 45]. To achieve this powerful performance these DMs rely on neural network backbones with billions of parameters [39, 45]. Networks of this size require a large amount of GPU memory and they are slow at inference time, making it difficult to deploy them in real-time or on resource-limited devices.

The issue of slow inference has, for example, been addressed by the development of faster DM samplers [10, 11, 29, 30, 49]. Another promising approach to tackle this issue is to distill the entire iterative sampling process of a (*teacher*) DM into a *student* (sampling) model [3, 33, 46, 52]; we refer to this approach as *sampling distillation*. To accelerate training, the student network is initialized from the teacher, and therefore student and teacher have the same number of parameters. After distillation, the student model can synthesize samples using only a few network evaluations rather than tens of network evaluations needed for the teacher model. Sampling distillation reduces the inference time while keeping the required GPU memory constant.

In this work, we instead propose to distill the knowledge (rather than the iterative sampling process) of a teacher DM into a *smaller* student DM, that is, the student should learn to match the predictions of the teacher for *any input*. Compared to sampling distillation, reducing the network size of a DM results in less GPU memory as well as faster inference: though the student model may still require tens of evaluations for sampling, each evaluation is significantly faster. Knowledge distillation (KD) [16] has been widely used in discriminative modeling [4, 14, 16, 54, 56], but only rarely in generative modeling [1]. We propose a robust approximate score matching objective to perform KD.

We thoroughly evaluate our proposed method on CIFAR-10 [25] and find that we can drastically decrease the required inference time of students model compared to their teachers; see Figure 1. Furthermore, we show early results on distilling Stable Diffusion [40], a large text-to-image *latent* DM; see Section 5. We envision that our framework, which can potentially be combined with orthogonal ideas such as fast DM samplers and sampling distillation, paves the way towards fast and high-resolution synthesis of DMs on resource limited devices.

2. Background

We consider continuous-time DMs [50] and follow the presentation of Karras et al. [23]. Let $p_{\text{data}}(\mathbf{x}_0)$ denote the data distribution and $p(\mathbf{x};\sigma)$ be the distribution obtained by adding i.i.d. σ^2-variance Gaussian noise to the data distribution. For sufficiently large $\sigma_{\max}$, $p(\mathbf{x};\sigma^2_{\max})$ is almost indistinguishable from $\sigma^2_{\max}$-variance Gaussian noise. Capitalizing on this observation, DMs sample high variance Gaus-

†Corresponding author: tim.dockhorn@uwaterloo.ca.

*Equal contribution.

(a) s: 129G flops, 5 steps (b) s: 273G flops, 10 steps (c) t: 424G flops, 3 steps (d) t: 764G flops, 5 steps (e) t: 1612G flops, 10 steps

Figure 1. Guided image generation on CIFAR-10 (guidance strength w=0.5) with student model s and teacher model t. The student model needs considerably less flops to achieve similar performance.

sian noise $\mathbf{x}_M \sim \mathcal{N}\left(\mathbf{0}, \sigma_{\max}^2\right)$ and sequentially denoise $\mathbf{x}_M$ into $\mathbf{x}_i \sim p(\mathbf{x}_i; \sigma_i)$, $i \in \{0, \dots, M\}$, with $\sigma_i < \sigma_{i+1}$ and $\sigma_M = \sigma_{\max}$. Assuming the DM is accurate, if $\sigma_0 = 0$ then the resulting $\mathbf{x}_0$ is distributed according to the data.

Sampling. In practice, this iterative denoising process explained above can be implemented through the numerical simulation of the *Probability Flow* ordinary differential equation (ODE) [50]

$$d\mathbf{x} = -\dot{\sigma}(t)\sigma(t)\nabla_{\mathbf{x}} \log p(\mathbf{x}; \sigma(t))\, dt, \tag{1}$$

where $\nabla_{\mathbf{x}} \log p(\mathbf{x}; \sigma)$ is the *score function* [22]. The schedule $\sigma(t)\colon [0, 1] \to \mathbb{R}_+$ is user-specified and $\dot{\sigma}(t)$ denotes the time derivative of $\sigma(t)$. Alternatively, we may also numerically simulate a stochastic differential equation (SDE) [23, 50]:

$$d\mathbf{x} = \underbrace{-\dot{\sigma}(t)\sigma(t)\nabla_{\mathbf{x}} \log p(\mathbf{x}; \sigma(t))\, dt}_{\text{Probability Flow ODE; see Equation (1)}} \underbrace{-\beta(t)\sigma^2(t)\nabla_{\mathbf{x}} \log p(\mathbf{x}; \sigma(t))\, dt + \sqrt{2\beta(t)}\sigma(t)\, d\omega_t}_{\text{Langevin diffusion component}}, \tag{2}$$

where $d\omega_t$ is the standard Wiener process. In principle, simulating either the Probability Flow ODE or the SDE above results in samples from the same distribution.

Training. DM training reduces to learning a model $\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}; \sigma)$ for the score function $\nabla_{\mathbf{x}} \log p(\mathbf{x}; \sigma)$. The model can, for example, be parameterized as $\nabla_{\mathbf{x}} \log p(\mathbf{x}; \sigma) \approx s_{\boldsymbol{\theta}}(\mathbf{x}; \sigma) = (D_{\boldsymbol{\theta}}(\mathbf{x}; \sigma) - \mathbf{x})/\sigma^2$ [23], where $D_{\boldsymbol{\theta}}$ is a learnable *denoiser* that, given a noisy data point $\mathbf{x}_0 + \mathbf{n}$, $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0)$, $\mathbf{n} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \boldsymbol{I}_d\right)$, and conditioned on the noise level σ, tries to predict the clean $\mathbf{x}_0$. The denoiser $D_{\boldsymbol{\theta}}$ (or equivalently the score model) can be trained via *denoising score matching* (DSM)

$$\mathbb{E}_{\substack{(\mathbf{x}_0, \mathbf{c}) \sim p_{\text{data}}(\mathbf{x}_0, \mathbf{c}), \\ (\sigma, \mathbf{n}) \sim p(\sigma, \mathbf{n})}} \left[\lambda_\sigma \| D_{\boldsymbol{\theta}}(\mathbf{x}_0 + \mathbf{n}; \sigma, \mathbf{c}) - \mathbf{x}_0 \|_2^2\right], \tag{3}$$

where $p(\sigma, \mathbf{n}) = p(\sigma)\mathcal{N}\left(\mathbf{n}; \mathbf{0}, \sigma^2\right)$, $p(\sigma)$ is a distribution over noise levels σ, $\lambda_\sigma\colon \mathbb{R}_+ \to \mathbb{R}_+$ is a weighting function, and $\mathbf{c}$ is a conditioning signal, e.g., a class label or a text prompt. For unconditional modeling, $\mathbf{c}$ may simply be ignored.

Classifier-free guidance. Classifier-free guidance [17] is a technique to guide the iterative sampling process of a DM towards a particular conditioning signal $\mathbf{c}$ by mixing the predictions of a conditional and an unconditional model

$$D^w(\mathbf{x}; \sigma, \mathbf{c}) = (1 + w)D(\mathbf{x}; \sigma, \mathbf{c}) - wD(\mathbf{x}; \sigma), \tag{4}$$

where $w \geq 0$ is the *guidance strength*. In practice, the unconditional model can be trained jointly alongside the conditional model in a single network by randomly replacing the conditional signal $\mathbf{c}$ with a (learnable) null embedding in Equation (3), e.g., 10% of the time [17]. Classifier-free guidance is widely used to improve the sampling quality, at the cost of reduced diversity, of text-to-image DMs [35, 40].

3. Method

We propose to distill the knowledge of a large DM, with *frozen* parameters $\boldsymbol{\phi}$, into a small student DM, with parameters $\boldsymbol{\theta}$, via an (approximate) *score matching* (SM) loss

$$\mathbb{E}\left[\lambda_\sigma \| D_{\boldsymbol{\theta}}(\mathbf{x}_0 + \mathbf{n}; \sigma, \mathbf{c}) - D_{\boldsymbol{\phi}}(\mathbf{x}_0 + \mathbf{n}; \sigma, \mathbf{c}) \|_2^2\right], \tag{5}$$

where the expectation is over the same distributions as in Equation (3). The loss in Equation (5) is consistent: zero loss implies that the knowledge of the teacher has been perfectly distilled into the student model (assuming full support of $p_{\text{data}}(\mathbf{x}_0)$ and $p(\sigma)$). Furthermore, Equation (5) becomes standard score matching [22] as the teacher score model $s_{\boldsymbol{\phi}}(\mathbf{x}; \sigma) = (D_{\boldsymbol{\phi}}(\mathbf{x}; \sigma) - \mathbf{x})/\sigma^2$ approaches the true score function $\nabla_{\mathbf{x}} \log p(\mathbf{x}; \sigma)$.

Guided distillation. To distill the knowledge of a jointly trained (un)conditional DM for classifier-free guidance, we can randomly replace the conditioning signal $\mathbf{c}$ with the null embedding in Equation (5). Note, however, that during inference we still need to evaluate the student model twice to compute Equation (4). To accelerate inference even further, we may follow Meng et al. [33] and directly distill the guidance computation (for an interval $[w_{\min}, w_{\max}]$) jointly

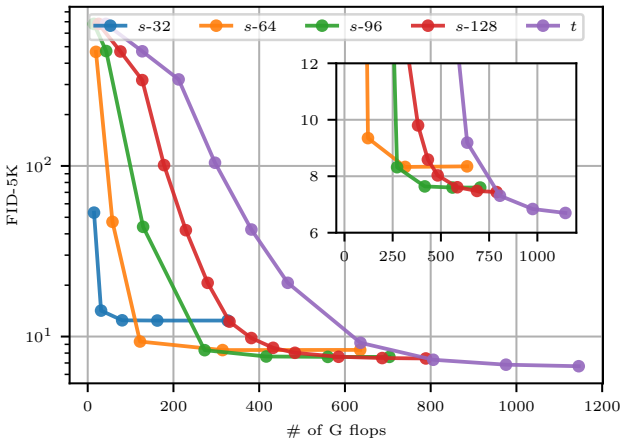

Figure 2. Unconditional image generation on CIFAR-10. Teacher model t and a variety of student models s with different number of base channels indicated in the legend. Linear y-axis as inset.

Table 1. Parameters and number of flops (per single forward pass) of the unconditional teacher model t and student models s.

Model	s-32	s-64	s-96	s-128	t
# of M parameters	2.2	8.8	19.8	35.1	55.7
# of G flops	1.64	6.42	14.36	25.44	42.42
# of residual blocks	2	2	2	2	4
# of base channels	32	64	96	128	128

with the knowledge of the teacher model, i.e.,

$$\mathbb{E}_{\substack{(\mathbf{x}_0,\mathbf{c})\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{c}),\\ (\sigma,\mathbf{n})\sim p(\sigma,\mathbf{n}),\\ w\sim\mathcal{U}[w_{\min},w_{\max}]}} \left[\lambda_\sigma \| D_{\boldsymbol{\theta}}(\mathbf{x};\sigma,\mathbf{c},w) - D^w_{\boldsymbol{\phi}}(\mathbf{x};\sigma,\mathbf{c})\|_2^2\right], \tag{6}$$

where $D^w_{\boldsymbol{\phi}}$ is computed via Equation (4) and $\mathbf{x} = \mathbf{x}_0 + \mathbf{n}$. Note that the student model is now additionally conditioned on the guidance strength w.

4. Experiments

We focus our efforts on a thorough evaluation on CIFAR-10 [25]. Student and teacher DMs are implemented using the DDPM++ architecture [50]. The teacher model uses 128 base channels while we use a variety of student models ranging from 32 to 128 base channels. We generate samples from our DMs using the deterministic Heun sampler proposed in Karras et al. [23] and we measure the sample quality via Fréchet Inception Distance (FID) [15] using 5k synthesized samples and all training samples. All experiment and training details can be found in Appendix A.1

4.1. Unconditional Distillation

We compare an unconditional teacher model to a variety of student models; see Table 1. We compute FID-5k for each model using a variety of solver steps; see Figure 2 for results. For fair comparison, the x-axis shows the accumulated number of G flops rather than the number of solver steps. We can see that there exist fixed budgets of G flops for which each of the four student model performs best, e.g., s-32 at 50 G Flops and s-96 at 400 G Flops, etc. Therefore, given GPU memory or inference time constraints, the size of the teacher model can be tuned to optimize performance. Overall, the gap between the larger student models (with 96 and 128 base channels) and the teacher model are reasonable, i.e., less than one FID-5K.

We also compare the training speed of our s-96 model to a standard DM (trained with Equation (3)) with the same neural network backbone; see Figure 3. The student model needs less iterations for convergence and overall converges to a better FID-5K value (7.63 vs 8.60). Note that during each iteration of KD we also need to do a forward pass through the larger teacher model; to reduce additional training time cost, the forward passes of teacher and student models may be parallelized.

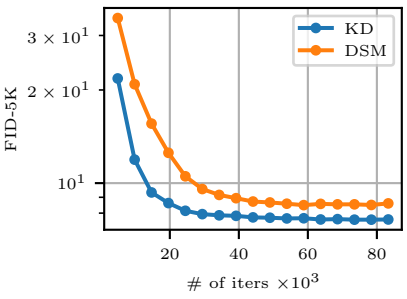

Figure 3. Distilling a large teacher DM into a smaller model (KD) leads to faster convergence and overall better performance than standard DSM training (for the smaller model).

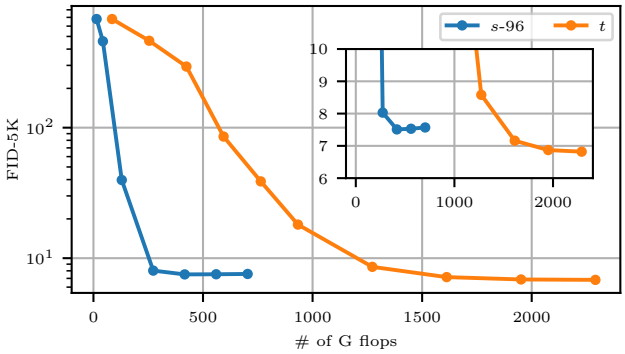

Figure 4. Guided image generation on CIFAR-10 (guidance strength w=0.25). Teacher model t and student model s. The student is conditioned on the guidance scale w whereas the teacher model needs to evaluate both a conditional and an unconditional DM per step. Linear y-axis as inset.

4.2. Guided Distillation

We additionally train a guided student model (96 base channels) with KD according to Equation (4) where we set $w_{\min}$=0.0 and $w_{\max}$=3.0. In Figure 4, we compare the student model to the teacher model for guidance strength w=0.25. The discrepancy of the performance at small number of G flops between the student and the teacher is even

(a) DSM: 7742G flops, 50 steps (b) s: 7742G flops, 50 steps (c) t: 8145G flops, 12 steps (d) t: 33936G flops, 50 steps

Figure 5. Samples generated for the DSM baseline, the KD student model s and the original Stable Diffusion (teacher) model t. Prompt: "A beautiful castle, matte painting.".

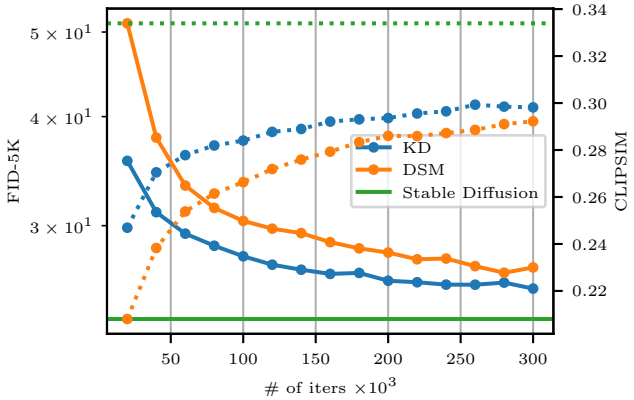

Figure 6. Initial experiments on Stable Diffusion show that distilling the network of large scale text-to-image DMs into smaller student models leads to better performance and faster convergence compared to standard DSM training. The gap of the student to the teacher is, however, still significant and needs to be addressed in future work. The solid and dotted lines represent *zero-shot* FID-5K and CLIPSIM on COCO [27], respectively.

more striking than in the unconditional case, likely due to the simultaneous KD and guidance distillation. Overall, the gap between the student and the teacher model is reasonable, less than 0.75 FID-5K. Samples for both the student and teacher models can be found in Figure 1.

5. Future Directions

We have shown that the size of the neural network backbone in DMs can be drastically reduced with our KD approach resulting in faster inference, while keeping overall performance drops to a reasonable level. We envision our method as a promising tool for relevant and novel applications in generative modeling, e.g., text-to-image synthesis on edge devices. In future work, we are planning to expand this work into the following directions:

Combining KD with sampling distillation. Distilling the sampling process of DMs [3, 33, 46, 52] allows for high-quality synthesis of large-scale models in several seconds. Sampling distillation is orthogonal to our KD approach, and combining these two ideas is a promising future research direction. An interesting question may be the order of distillation: Should we first distill the sampling process or the network? Or could we potentially do both distillations jointly?

Mixed training. In this work, we only considered pure distillation, however, it has been shown to be helpful to combine KD with standard training in discriminative models [16]. One approach for mixed training may be a linear combination of Equation (3) and Equation (5). As we show in Appendix B, this mixed training approach is equivalent to performing distillation with an additional term

$$2\alpha\lambda_\sigma(D_{\boldsymbol{\theta}}(\mathbf{x};\sigma) - D_{\boldsymbol{\phi}}(\mathbf{x};\sigma))^\top(D_{\boldsymbol{\phi}}(\mathbf{x};\sigma) - \mathbf{x}_0), \quad (7)$$

where $\alpha \in [0, 1]$, inside the expectation of Equation (5).

Better initialization. Fine-tuning large-scale DMs has been shown to be highly effective: for example, fine-tuning text-to-image DMs for, say, 100 to 1000 iterations on a small dataset of a few images results in highly editable personalized text-to-image models [12, 42]. Similarly, student models in sampling distillation are generally initialized from the teacher model, which allows for fine-tuning and results in faster convergence. In contrast, our student architectures are smaller, and therefore we cannot directly make use of the teacher for initialization. Future work could explore better initialization methods that may improve the training speed of our KD approach.

Applying KD to larger models. In this work, we thoroughly study KD of DMs for CIFAR-10. An obvious future direction is to scale our approach to larger models: To this end, we perform a preliminary study on Stable Diffusion [40], distilling its latent DM into a network of less than a quarter of the original size (from 866M to 200M parameters). Compared to training a standard DM of the same size with DSM (Equation (3)), we find that the KD student converges faster and to a better value; see Figure 6. This is a promising result which may indicate that our results on CIFAR-10 transfer to large-scale DMs. Compared to our CIFAR-10 results, however, we found that there is still a substantial gap compared to the teacher model; see also samples in Figure 5. Experiment details can be found in Appendix A.2. In future work, we are planning to thoroughly evaluate our KD approach to large-scale (latent) DMs.

References

[1] Angeline Aguinaldo, Ping-Yeh Chiang, Alex Gain, Ameya Patil, Kolten Pearson, and Soheil Feizi. Compressing GANs using Knowledge Distillation. *arXiv:1902.00159*, 2019. 1

[2] Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional Image Generation with Score-Based Diffusion Models. *arXiv:2111.13606*, 2021. 1

[3] David Berthelot, Arnaud Autef, Jierui Lin, Dian Ang Yap, Shuangfei Zhai, Siyuan Hu, Daniel Zheng, Walter Talbot, and Eric Gu. TRACT: Denoising Diffusion Models with Transitive Closure Time-Distillation. *arXiv:2303.04248*, 2023. 1, 4

[4] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge Distillation: A Good Teacher is Patient and Consistent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10925–10934, 2022. 1

[5] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14347–14356, 2021. 1

[6] Hyungjin Chung and Jong Chul Ye. Score-based diffusion models for accelerated MRI. *arXiv:2110.05243*, 2021. 1

[7] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-Closer-Diffuse-Faster: Accelerating Conditional Diffusion Models for Inverse Problems through Stochastic Contraction. *arXiv:2112.05146*, 2021. 1

[8] Prafulla Dhariwal and Alex Nichol. Diffusion Models Beat GANs on Image Synthesis. In *Neural Information Processing Systems*, 2021. 1

[9] Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. Differentially private diffusion models. *arXiv:2210.09929*, 2022. 1

[10] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. GENIE: Higher-Order Denoising Diffusion Solvers. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 1

[11] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-Based Generative Modeling with Critically-Damped Langevin Diffusion. In *International Conference on Learning Representations*, 2022. 1

[12] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion. *arXiv:2208.01618*, 2022. 4

[13] Sahra Ghalebikesabi, Leonard Berrada, Sven Gowal, Ira Ktena, Robert Stanforth, Jamie Hayes, Soham De, Samuel L Smith, Olivia Wiles, and Borja Balle. Differentially Private Diffusion Models Generate Useful Synthetic Images. *arXiv:2302.13861*, 2023. 1

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 1

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 3

[16] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the Knowledge in a Neural Network. *arXiv:1503.02531*, 2 (7), 2015. 1, 4

[17] Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 2

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, 2020. 1

[19] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded Diffusion Models for High Fidelity Image Generation. *arXiv:2106.15282*, 2021. 1

[20] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video Diffusion Models. *arXiv:2204.03458*, 2022. 1

[21] Dewei Hu, Yuankai K. Tao, and Ipek Oguz. Unsupervised Denoising of Retinal OCT with Diffusion Probabilistic Model. *arXiv:2201.11760*, 2022. 1

[22] Aapo Hyvärinen. Estimation of Non-Normalized Statistical Models by Score Matching. *Journal of Machine Learning Research*, 6:695–709, 2005. ISSN 1532-4435. 2

[23] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the Design Space of Diffusion-Based Generative Models. *arXiv:2206.00364*, 2022. 1, 2, 3, 8

[24] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising Diffusion Restoration Models. *arXiv:2201.11793*, 2022. 1

[25] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009. 1, 3

[26] Haoying Li, Yifan Yang, Meng Chang, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. SRDiff: Single Image Super-Resolution with Diffusion Probabilistic Models. *arXiv:2104.14951*, 2021. 1

[27] 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. 4

[28] Xihui Liu, Dong Huk Park, Samaneh Azadi, Gong Zhang, Arman Chopikyan, Yuxiao Hu, Humphrey Shi, Anna Rohrbach, and Trevor Darrell. More Control for Free! Image Synthesis with Semantic Diffusion Guidance. *arXiv:2112.05744*, 2021. 1

[29] 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. *arXiv:2206.00927*, 2022. 1

[30] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver++: Fast Solver for Guided Sampling of Diffusion Probabilistic Models. *arXiv:2211.01095*, 2022. 1

[31] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using Denoising Diffusion Probabilistic Models. *arXiv:2201.09865*, 2022. 1

[32] Guanxiong Luo, Martin Heide, and Martin Uecker. MRI Reconstruction via Data Driven Markov Chain with Joint Uncertainty Estimation. *arXiv:2202.01479*, 2022. 1

[33] Chenlin Meng, Ruiqi Gao, Diederik P Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On Distillation of Guided Diffusion Models. *arXiv:2210.03142*, 2022. 1, 2, 4

[34] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *International Conference on Learning Representations*, 2022. 1

[35] 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:2112.10741*, 2021. 1, 2

[36] Kushagra Pandey, Avideep Mukherjee, Piyush Rai, and Abhishek Kumar. DiffuseVAE: Efficient, Controllable and High-Fidelity Generation from Low-Dimensional Latents. *arXiv:2201.00308*, 2022. 1

[37] Cheng Peng, Pengfei Guo, S. Kevin Zhou, Vishal Patel, and Rama Chellappa. Towards performant and reliable undersampled MR reconstruction via diffusion model sampling. *arXiv:2203.04292*, 2022. 1

[38] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion Autoencoders: Toward a Meaningful and Decodable Representation. *arXiv:2111.15640*, 2021. 1

[39] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv:2204.06125*, 2022. 1

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. *arXiv:2112.10752*, 2021. 1, 2, 4, 8

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. *arXiv:2112.10752*, 2021. 1

[42] 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. *arXiv:2208.12242*, 2022. 4

[43] Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-Image Diffusion Models. *arXiv:2111.05826*, 2021. 1

[44] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image Super-Resolution via Iterative Refinement. *arXiv:2104.07636*, 2021. 1

[45] 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:2205.11487*, 2022. 1

[46] Tim Salimans and Jonathan Ho. Progressive Distillation for Fast Sampling of Diffusion Models. In *International Conference on Learning Representations*, 2022. 1, 4, 8

[47] Hiroshi Sasaki, Chris G. Willcocks, and Toby P. Breckon. UNIT-DDPM: UNpaired Image Translation with Denoising Diffusion Probabilistic Models. *arXiv:2104.05358*, 2021. 1

[48] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-A-Video: Text-to-Video Generation without Text-Video Data. *arXiv:2209.14792*, 2022. 1

[49] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*, 2021. 1

[50] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021. 1, 2, 3, 8

[51] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving Inverse Problems in Medical Imaging with Score-Based Generative Models. In *International Conference on Learning Representations*, 2022. 1

[52] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models. *arXiv:2303.01469*, 2023. 1, 4

[53] Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual Diffusion Implicit Bridges for Image-to-Image Translation. *arXiv:2203.08382*, 2022. 1

[54] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in Neural Information Processing Systems*, 30, 2017. 1

[55] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G. Dimakis, and Peyman Milanfar. Deblurring via Stochastic Refinement. *arXiv:2112.02475*, 2021. 1

[56] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-Training With Noisy Student Improves ImageNet Classification. In *Proceedings of the IEEE/CVF Conference on Computer Bision and Pattern Recognition*, pages 10687–10698, 2020. 1

[57] Yutong Xie and Quanzheng Li. Measurement-conditioned Denoising Diffusion Probabilistic Model for Under-sampled Medical Image Reconstruction. *arXiv:2203.03623*, 2022. 1

[58] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion Probabilistic Modeling for Video Generation. *arXiv:2203.09481*, 2022. 1

A. Experiment Details

A.1. CIFAR-10

Our teacher models for CIFAR-10 are taken from Karras et al. [23]; one conditional[3] and one unconditional model[4]. The networks are based on the DDPM++ architecture [50]. The teacher and student models have four and two residual blocks, respectively. The teacher model has 128 base channels while we train multiple student models ranging from 32 to 128 base channels. All models are trained for 100k iterations, using a batch size of 512, to ensure convergence. We use Adam with learning rate 1×10^{-3} and an exponential moving average half-life of 50M images, following Karras et al. [23]. For KD, we do not use any dropout while the DSM baseline in Figure 3 uses a dropout probability of 10% to prevent over-fitting. All student models (and the DSM baseline) use the same network preconditioning, noise distribution $p(\sigma)$ and loss weighting λ_σ as the teacher model; see the last column of Table 1 in Karras et al. [23].

A.2. Stable Diffusion

Our teacher model is Stable Diffusion [40] fine-tuned to $\mathbf{v}$-parameterization [46]. The student model (and the DSM baseline) uses the same architecture as Stable Diffusion, however, the number of base channels is reduced from 360 to 192 and the transformer block at the highest resolution is removed. We use AdamW with learning rate 3×10^{-4} and batch size 512. The exponential moving average half life, the noise distribution $p(\sigma)$, and the loss weighting λ_σ for both the student model and the DSM baseline are the same as used in the original Stable Diffusion model.

B. Mixed Training Derivation

In Section 5, we propose the following mixed training objective

$$\mathbb{E}[\lambda_\sigma \left((1-\alpha)\|D_{\boldsymbol{\theta}}(\mathbf{x};\sigma,\mathbf{c}) - D_{\boldsymbol{\phi}}(\mathbf{x};\sigma,\mathbf{c})\|_2^2 + \alpha\|D_{\boldsymbol{\theta}}(\mathbf{x};\sigma,\mathbf{c}) - \mathbf{x}_0\|_2^2\right)], \tag{8}$$

where $\mathbf{x} = \mathbf{x}_0 + \mathbf{n}$. Let us add and subtract the teacher model $D_{\boldsymbol{\phi}}$ to the second norm

$$\begin{aligned}
\|D_{\boldsymbol{\theta}}(\mathbf{x};\sigma,\mathbf{c}) - \mathbf{x}_0\|_2^2 &= \|D_{\boldsymbol{\theta}}(\mathbf{x};\sigma,\mathbf{c}) - D_{\boldsymbol{\phi}}(\mathbf{x};\sigma,\mathbf{c}) + D_{\boldsymbol{\phi}}(\mathbf{x};\sigma,\mathbf{c}) - \mathbf{x}_0\|_2^2 &(9)\\
&= \|D_{\boldsymbol{\theta}}(\mathbf{x};\sigma,\mathbf{c}) - D_{\boldsymbol{\phi}}(\mathbf{x};\sigma,\mathbf{c})\|_2^2 + 2(D_{\boldsymbol{\theta}}(\mathbf{x};\sigma) - D_{\boldsymbol{\phi}}(\mathbf{x};\sigma))^\top (D_{\boldsymbol{\phi}}(\mathbf{x};\sigma) - \mathbf{x}_0) + \|D_{\boldsymbol{\phi}}(\mathbf{x};\sigma,\mathbf{c}) - \mathbf{x}_0\|_2^2 &(10)\\
&= \|D_{\boldsymbol{\theta}}(\mathbf{x};\sigma,\mathbf{c}) - D_{\boldsymbol{\phi}}(\mathbf{x};\sigma,\mathbf{c})\|_2^2 + 2(D_{\boldsymbol{\theta}}(\mathbf{x};\sigma) - D_{\boldsymbol{\phi}}(\mathbf{x};\sigma))^\top (D_{\boldsymbol{\phi}}(\mathbf{x};\sigma) - \mathbf{x}_0) + \text{const.} &(11)
\end{aligned}$$

Note that the last term in the above equation is a constant with respect to the learnable parameters $\boldsymbol{\theta}$. Plugging the above into Equation (8), we have

$$\mathbb{E}[\lambda_\sigma \left(\|D_{\boldsymbol{\theta}}(\mathbf{x};\sigma,\mathbf{c}) - D_{\boldsymbol{\phi}}(\mathbf{x};\sigma,\mathbf{c})\|_2^2 + 2\alpha(D_{\boldsymbol{\theta}}(\mathbf{x};\sigma) - D_{\boldsymbol{\phi}}(\mathbf{x};\sigma))^\top (D_{\boldsymbol{\phi}}(\mathbf{x};\sigma) - \mathbf{x}_0)\right)] + \text{const.} \tag{12}$$

This shows that mixed training is equivalent to plain distillation with a regularization term which uses the clean data $\mathbf{x}_0$.

Alternatively, we may similarly add and subtract the clean data $\mathbf{x}_0$ to the first norm in Equation (8) which results in

$$\mathbb{E}[\lambda_\sigma \left(\|D_{\boldsymbol{\theta}}(\mathbf{x};\sigma,\mathbf{c}) - \mathbf{x}_0\|_2^2 + 2(1-\alpha)(D_{\boldsymbol{\theta}}(\mathbf{x};\sigma) - \mathbf{x}_0)^\top (\mathbf{x}_0 - D_{\boldsymbol{\phi}}(\mathbf{x};\sigma))\right)] + \text{const}, \tag{13}$$

which shows that mixed training is also equivalent to standard DM training (DSM in Equation (3)) with an additional regularization term involving the teacher network $D_{\boldsymbol{\phi}}$.

[3] https://nvlabs-fi-cdn.nvidia.com/edm/pretrained/edm-cifar10-32x32-uncond-vp.pkl
[4] https://nvlabs-fi-cdn.nvidia.com/edm/pretrained/edm-cifar10-32x32-uncond-ve.pkl