On Distillation of Guided Diffusion Models

Pramook Khungurn

January 26, 2023

This note is written as I read "On Distillation of Guided Diffusion Models" by Meng et al. [MRG+22].

1 Introduction

- Popular image generation models are built on DDPMs [HJA20] and classifier-free guidance [HS22].
- As of Janunary 2023, sampling from these models are somewhat slow. Several tens of steps (20 to 50) are typically used. Anything lower would generate low-quality samples.
- Salimans and Ho proposed a technique to distill unconditional (and also class-conditional) DDPMs so that one can generate a sample in as few as 4 steps [SH22]. However, distilling models trained with classifier-free guidance has not been researched on before the present paper.
- The paper requires a pre-trained conditional DDPM that can evaluate conditional and unconditional scores $\nabla_{\mathbf{z}} \log p_t(\mathbf{z})$.
  - Oddly enough, the paper says that they require two models: one unconditional and the other unconditional.
  - I think what it really means is that every sampling steps requires two evaluations: one with conditioning information and the other without the conditioning information.
- The paper proposes a two-stage distillation process.
  - In the first stage, a single conditional model (different formulation from vanilla conditional DDPM) is trained on so that it can predict the output that would result from classifier-free guidance in one network evaluation.
  - In the second stage, the resulting model from the first stage is distilled with the progressive distillation algorithm of Salimans and Ho [SH22].
- Summary of results.
  - The model works on DDPMs trained on (1) the pixel space directly and (2) the latent space of an autoencoder (i.e., latent diffusion models [RBL+21]).
  - Pixel space results.
    - Experiments on ImageNet 64x64 and CIFAR-10.
    - Comparable visually to teacher model in 4 sampling steps.
    - Comparable FID/IS scores to teacher model in 4 to 16 sampling steps.
  - Latent spacce results.
    - Experiments on ImageNet 256x256 and LAION 512x512.
    - Comparable visually in 1 to 4 sampling steps.
    - Matching FID scores in 2 to 4 sampling steps.

2 Background

2.1 Diffusion Model

- A data item is represented by $\mathbf{x}$, and it comes from the distribution $p_{\text{data}}(\mathbf{x})$. With conditioning information c, the distribution becomes $p_{\text{data}}(\mathbf{x}|c)$.
- To make the exposition easier, we say that the unconditional distribution $p_{\text{data}}(\cdot)$ is a special case of the conditonional distribute $p_{\text{data}}(\cdot|c)$ with $c = \emptyset$. So, from now on, the data distribution is always conditional.
- A diffusion model works on latent variables $\{\mathbf{z}_t : t \in [0, 1]\}$, which is a stochastic process derived from the data item $\mathbf{x}$.
  - The forward process has two parameters, α_t and σ_t, collectively known as the **noise schedule**. They are functions of signature $[0, 1] \to [0, \infty]$.
  - The logarithm of the signal-to-noise ratio (SNR),
  $$\lambda_t = \log(\alpha^2/\sigma^2),$$
  should be monotonically decreasing as t increase. It should be a very high value (goes up to $+\infty$) at $t = 0$ and a very low value (goes down to $-\infty$) at $t = 1$.
  - We require that, for any $0 \leq s < t \leq 1$,
  $$\begin{aligned} p(\mathbf{z}_t|\mathbf{x}) &= \mathcal{N}(\mathbf{z}_t; \alpha_t\mathbf{x}, \sigma_t^2 I), \\ p(\mathbf{z}_t|\mathbf{z}_s) &= \mathcal{N}(\mathbf{z}_t; \alpha_{t|s}\mathbf{z}_s; \sigma_{t|s}^2 I) \end{aligned}$$
  where
  $$\begin{aligned} \alpha_{t|s} &= \frac{\alpha_t}{\alpha_s} \\ \sigma_{t|s}^2 &= \sigma_t^2 - \alpha_{t|s}^2\sigma_s^2. \end{aligned}$$
- It can be shown that, for any $0 \leq s < t \leq 1$, we have that
$$p(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x}) = \mathcal{N}\bigg(\mathbf{z}_s; \quad \frac{\alpha_{t|s}\sigma_s^2}{\sigma_t^2}\mathbf{z}_t + \frac{\alpha_s\sigma_{t|s}^2}{\sigma_t^2}\mathbf{x}, \quad \frac{\sigma_{t|s}^2\sigma_s^2}{\sigma_t^2} I\bigg).$$
- Integrating over $\mathbf{x}$, we have that the "marginal" distribution of $\mathbf{z}_t$ is
$$p_t(\mathbf{z}_t|c) = \int_{\mathbb{R}^d} p(\mathbf{z}_t|\mathbf{x}) p_{\text{data}}(\mathbf{x}|c)\, \mathrm{d}\mathbf{x}.$$
- A diffusion model is a network $\hat{\mathbf{x}}_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$ trained to predict $\mathbf{x}$ from $\mathbf{z}_t$. In other words, if we sample $\mathbf{x} \sim p_{\text{data}}(\mathbf{x}|c)$ and $\mathbf{z}_t \sim p(\mathbf{z}_t|\mathbf{x})$, then we want to have
$$\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t, c) \approx \mathbf{x}.$$
(In practical implementations, the network would also have to take some form of information about the time, but we drop the t parameter to avoid clutter.)
- We can use the following functions to train such a network:
$$E_{t\sim\mathcal{U}([0,1]), \mathbf{x}\sim p_{\text{data}}(\mathbf{x}|c), \mathbf{z}_t\sim\mathcal{N}(\alpha_t\mathbf{x}, \sigma_t^2 I)}[w(\lambda_t)\|\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t, c) - \mathbf{x}\|^2]$$
where $\mathcal{U}([0, 1])$ is the uniform distribution over $[0, 1]$, and $w(\lambda_t)$ is a pre-specified weight function. Once the model is trained, we have that
$$\nabla_{\mathbf{z}} \log p_t(\mathbf{z}|c) \approx \frac{\alpha_t\hat{\mathbf{x}}(\mathbf{z}_t, c) - \mathbf{z}_t}{\sigma_t^2}.$$

- We pix α_t and σ_t so that $\alpha_t^2 + \sigma_t^2 = 1$. (In other words, we use the variance-preserving formulation of DDPM.) We also set $\alpha_t = 0$ so that $p_1(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, I)$.

- With the noise schedule above, one can approximate $\mathbf{z}_s$ given $\mathbf{z}_t$ as follows:

$$\mathbf{z}_s = \alpha_s \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t, c) + \sigma_s \frac{\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t, c)}{\sigma_t}.$$

  This is the so called DDIM update rule [SSDK+21].

- From the DDIM update rule, a sampler can be create from it as follows.
  - Discretize the time in $[0, 1]$. We will only work with $t = i/N$ where N is the number of steps in the sampling process, and $i \in \{1, 2, \ldots, N\}$.
  - Start with $t = 1$ and sample $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, I)$.
  - For $i = N-1, N-2, \ldots, 0$, set $s = i/N$ and $t = (i+1)/N$. We can compute $\mathbf{z}_s$ from $\mathbf{z}_t$ using the DDIM update rule.
  - Output $\mathbf{z}_0$ as the generated sample.

2.2 Classifier-Free Guidance

- Classifier-free guidance allows one to improve the sample quality of a conditional DDPM by trading diversity for quality.

- The process needs a guidance weight parameter $\gamma \geq -1$.

- With the guidance weight, the denoise data item is given by

$$\hat{\mathbf{x}}^{\gamma}_{\boldsymbol{\theta}}(\mathbf{z}_t, c) = (1+\gamma)\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t, c) - \gamma \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t, \emptyset).$$

  We then use this value in the DDIM update rule during sampling.

- We can see that:
  - When $\gamma = 0$, we get the normal conditional DDIM sampling routine.
  - When $\gamma = -1$, we get the normal unconditional DDIM sampling routine.

- We can also see that, at each sampling step, one has to evaluate the DDPM two times.

3 Proposed Algorithm

- The input to the algorithm is a conditional DDPM $\hat{\mathbf{x}}_{\boldsymbol{\theta}}$, which is called the **teacher model**.

- It is assumed that the teacher model is a continous time model.

3.1 Stage 1 Distillation

- The output of this stage is a student model $\hat{\mathbf{x}}_{\boldsymbol{\eta}_1}$ such that

$$\hat{\mathbf{x}}_{\boldsymbol{\eta}_1}(\mathbf{z}_t, c, \gamma) \approx \hat{\mathbf{x}}^{\gamma}_{\boldsymbol{\theta}}(\mathbf{z}_t, c)$$

  for all guidance weight $\gamma \in [\gamma_{\max}, \gamma_{\min}]$. Note that the student model needs to take γ as an extra conditioning information.

- To get such a model, we optimize the model with respect to the following loss function

$$E_{\gamma\sim\mathcal{U}([\gamma_{\min},\gamma_{\max}]),t\sim\mathcal{U}[0,1],(\mathbf{x},c)\sim p_{\text{data}},\mathbf{z}_t\sim\mathcal{N}(\alpha_t\mathbf{x},\sigma_t^2 I)}[w(\lambda_t)\|\hat{\mathbf{x}}_{\boldsymbol{\eta}_1}(\mathbf{z}_t,w,c)-\hat{\mathbf{x}}^{\gamma}_{\boldsymbol{\theta}}(\mathbf{z}_t,c)\|^2]$$

- Model details.
  - The paper uses the basic architecture as the teacher model, but it adds a branch of conditional information intake that allows guidance strength γ to be given to the model.
  - The architecture is a U-Net like many other previous works.
  - The student model is initialized with the parameters of the teacher model, except for the part that deals with the guidance strength.
  - Fourier embedding [TSM+20] is applied to γ before being passed to the model. In other words, it is treated pretty much like the time step information.

- The training algorithm is as follows.

Phase-One-Distillation

while not converged
    $(\mathbf{x},c)\sim p_{\text{data}}$
    $\gamma\sim\mathcal{U}([\gamma_{\min},\gamma_{\max}])$
    $t\sim\mathcal{U}([0,1])$
    $\boldsymbol{\xi}\sim\mathcal{N}(\mathbf{0},I)$
    $\mathbf{z}_t\leftarrow\alpha_t\mathbf{x}+\sigma_t\boldsymbol{\xi}$
    $\tilde{\mathbf{x}}\leftarrow(1+\gamma)\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t,c)-\gamma\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t,\emptyset)$
    $\lambda_t\leftarrow\log(\alpha_t^2/\sigma_t^2)$
    $L_{\boldsymbol{\eta}_1}\leftarrow w(\lambda_t)\|\tilde{\mathbf{x}}-\hat{\mathbf{x}}_{\boldsymbol{\eta}_1}(\mathbf{z}_t,c,\gamma)\|_2^2$
    Update $\boldsymbol{\eta}_1$ according to $\nabla_{\boldsymbol{\eta}_1}L_{\boldsymbol{\eta}_1}$.

3.2 Stage 2 Distillation

- Stage 2 takes the model $\hat{\mathbf{x}}_{\boldsymbol{\eta}_1}$ from Stage 1 and progressively distill it with the algorithm proposed by Salimans and Ho [SH22].
- The distillation algorithm is as follows.

STAGE-TWO-DISTILLATION($\boldsymbol{\eta}_1, N$)

// $\boldsymbol{\eta}_1$ denotes the parameters of the resulting model from Stage 1.
// N is the number of iterations that the teacher model takes to sample.

for K iterations
    $N \leftarrow N/2$
    $\boldsymbol{\eta}_2 \leftarrow \boldsymbol{\eta}_1$
    while not converged
        $(\mathbf{x}, c) \sim p_{\mathrm{data}}$
        $\gamma \sim \mathcal{U}([\gamma_{\min}, \gamma_{\max}])$
        $i \sim \mathcal{U}(\{1, 2, \dots, N\})$
        $t \leftarrow i/N$
        $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, I)$
        // 2 steps of DDIM sampling with teacher model
        $t' \leftarrow t - 0.5/N; \quad t'' \leftarrow t - 1/N$
        $\mathbf{z}_{t'} \leftarrow \alpha_{t'} \hat{\mathbf{x}}_{\boldsymbol{\eta}_1}(\mathbf{z}_t, c, \gamma) + \frac{\sigma_{t'}}{\sigma_t}\big(\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_{\boldsymbol{\eta}_1}(\mathbf{z}_t, c, \gamma)\big)$
        $\mathbf{z}_{t''} \leftarrow \alpha_{t''} \hat{\mathbf{x}}_{\boldsymbol{\eta}_1}(\mathbf{z}_{t'}, c, \gamma) + \frac{\sigma_{t''}}{\sigma_{t'}}\big(\mathbf{z}_{t'} - \alpha_{t'} \hat{\mathbf{x}}_{\boldsymbol{\eta}_1}(\mathbf{z}_{t'}, c, \gamma)\big)$
        $\tilde{\mathbf{x}} \leftarrow \frac{\mathbf{z}_{t''} - (\sigma_{t''}/\sigma_t)\mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t)\alpha_t}$
        $\lambda_t \leftarrow \log(\alpha_t^2/\sigma_t^2)$
        $L_{\boldsymbol{\eta}_2} \leftarrow w(\lambda_t)\|\tilde{\mathbf{x}} - \hat{\mathbf{x}}_{\boldsymbol{\eta}_2}(\mathbf{z}_t, c, \gamma)\|^2$
        Update $\boldsymbol{\eta}_2$ according to $\nabla_{\boldsymbol{\eta}_2} L_{\boldsymbol{\eta}_2}$
    // Make the converged student the teacher of the next round.
    $\boldsymbol{\eta}_1 \leftarrow \boldsymbol{\eta}_2$

- The paper observes that, once $\hat{\mathbf{x}}_{\boldsymbol{\eta}_2}$ is trained, one can perform stochastic sampling with it using an algorithm similar to that proposed by Karras et al. [KAAL22].

- In each iteration of the stochastic sampling algorith:

    1. We apply one deterministic reverse-time step with two-times the original step length (i.e., same as a $N/2$-step deterministic sampler).
    2. We then perform one stochastic step forward (i.e., perturb with noise) using the original step length.

  More concretely, if we want to go from $t = i/N$ with $i > 1$ to $s = (i-1)/N$. Then, letting $u = (i-2)/N$, we compute

$$\mathbf{z}_u = \alpha_u \hat{\mathbf{x}}_{\boldsymbol{\eta}_2}(\mathbf{z}_t, c, \gamma) + \sigma_u \frac{\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_{\boldsymbol{\eta}_2}(\mathbf{z}_t, c, \gamma)}{\sigma_t}.$$

  Then, we compute

$$\mathbf{z}_s = \frac{\alpha_s}{\alpha_u}\mathbf{z} + \sigma_{s|u}\boldsymbol{\xi}$$

  where $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, I)$.

- Note that the procedure above does not work for $t = 1/N$. When it is the case, we must do the usual reverse-time step.

- In order for the above model to work, we need to train the student model so that, for $t > i/N$, it do the reverse-time step of length $2/N$. Moreover, for $t = 1/N$, it must perform a step of length $1/N$. This procedure needs a new training algorithm, which is given in the paper.

    – I feel that this is an uninteresting technical details. I will skip the exposition in this note.

4 Experiments

4.1 Pixel-Space Guided Models

- The paper uses two datasets: ImageNet 64x64 and CIFAR-10.
- The guidance weight range is $[\gamma_{\min}, \gamma_{\max}] = [0, 4]$.
- The teacher model is a 1024 step conditional DDIM model. (So, 2048 evaluations to sample one data item.)
- The teacher model is a U-Net trained to predict $\mathbf{v} := \alpha_t \boldsymbol{\xi} - \sigma_t \mathbf{x}$.
- It's quite unclear what is the conditioning information in this case. I presume it's the class label of each image.
- The paper observes that their distilled models is able to achieve FID scores close to those of the teacher model when $N = 4$. See Table 1 and Figure 5 in the paper.
- One interesting thing to note is that, in many cases, stochastic sampling with the distilled model performed the best.

4.2 Latent-Space Guided Models

- The paper uses the open-source latent diffusion model [RBL$^+$21]. They fine-tuned it to do $\mathbf{v}$-prediction rather than $\boldsymbol{\xi}$-prediction.
- They then examined performance of distilled models on three tasks.
  - Class-conditional generation.
  - Text-guided image generation.
  - Text-guided image-to-image translation.
  - Image inpaiting.
- For the last two tasks, only qualitative comparisons are available in the paper, so I will not do an exposition here.

4.2.1 Class-Conditional Generation

- The experiment is done on ImageNet 256x256.
- The teacher model is a DDIM with 512 sampling steps.
- The guidance weight range is $[\gamma_{\min}, \gamma_{\max}] = [0, 14]$.
- The distilled model matched the FID scores of the teacher model with 2 to 4 sampling steps. It also performed much better than DDIM sampling when the number of steps is 1 to 4.
- Samples obtained using only 1 sampling step were still satisfying.

4.2.2 Text-Guided Image Generation

- The dataset used is LAION-5B at the 512×512 resolution.
- The guidance weight range is $[\gamma_{\min}, \gamma_{\max}] = [2, 14]$.
- Distillation process.
  - Batch size = 512.
  - Stage 1 takes 3,000 gradient updates.
  - Stage 2 takes 2,000 gradient updates per iteration. However, then $i = 1, 2, 4$, the paper used 20,000 gradient updates.
- In terms of FID/IS scores, the distilled model performed much better than the DDIM sampler when $N \leq 4$.
- When $N \geq 8$, the paper did not observed significant differences in the scores. However, visual inspection showed that images generated by the distilled model were sharper and more coherent.

References

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020.

[HS22] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022.

[KAAL22] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models, 2022.

[MRG+22] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models, 2022.

[RBL+21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[SH22] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *CoRR*, abs/2202.00512, 2022.

[SSDK+21] 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.

[TSM+20] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *CoRR*, abs/2006.10739, 2020.