Single-fold Distillation for Diffusion models

Chi Hong[1] (✉), Jiyue Huang[1], Robert Birke[2], Dick Epema[1], Stefanie Roos[3], and Lydia Y. Chen[1,4]

[1] TU Delft, The Netherlands {c.hong, j.huang-4, D.H.J.Epema}@tudelft.nl
[2] University of Turin, Italy robert.birke@unito.it
[3] RPTU Kaiserslautern-Landau, Germany stefanie.roos@cs.rptu.de
[4] University of Neuchatel, Switzerland lydiaychen@ieee.org

Abstract. While diffusion models effectively generate remarkable synthetic images, a key limitation is the inference inefficiency, requiring numerous sampling steps. To accelerate inference and maintain high-quality synthesis, teacher-student distillation is applied to compress the diffusion models in a progressive and binary manner by retraining, e.g., reducing the 1024-step model to a 128-step model in 3 folds. In this paper, we propose a single-fold distillation algorithm, SFDDM, which can flexibly compress the teacher diffusion model into a student model of any desired step, based on reparameterization of the intermediate inputs from the teacher model. To train the student diffusion, we minimize not only the output distance but also the distribution of the hidden variables between the teacher and student model. Extensive experiments on four datasets demonstrate that our student model trained by the proposed SFDDM is able to sample high-quality data with steps reduced to less than 1%, thus, trading off inference time. Our remarkable performance highlights that SFDDM effectively transfers knowledge in single-fold distillation, achieving semantic consistency and meaningful image interpolation.

Keywords: knowledge distillation · diffusion models · inference efficiency.

1 Introduction

Diffusion models [23,8,7] have emerged as generative models for images of exceptionally high quality without the necessity of conducting adversarial training.A diffusion model constitutes a Markov chain of forward steps of slowly adding random noise to data, followed by a reverse denoising process that gradually reconstructs the data from the noise via trained neural networks. These underlying networks typically use the UNet architecture to better connect the forward steps to the corresponding denoising step. However, such models require large numbers of sampling steps, e.g., 1000 in DDPM [7], which leads to high sampling/inference[5] times, limiting their applications in latency-sensitive applications.

Prior art explored diverse directions to reduce the sampling time of diffusion models and maintain the image synthesis quality. One approach is to reduce the

[5] We interchangly use sampling and inference time.

computing complexity by compressing the UNet [11,25] and leverage the acceleration technologies of modern GPUs. Another approach is to reduce the number of sampling steps, i.e., the required UNet inferences. [21] skips intermediate steps by generalizing the original Markovian process via a class of non-Markovian diffusion steps. These two approaches do not change the original training procedure. Differently, progressive distillation [19,15] introduces a teacher-student framework to reduce a trained teacher diffusion model of T steps into a student diffusion of T' steps, where $T' \ll T$, via multiple binary foldings by retraining. For instance, distilling a 1024-step diffusion model into a 128-step model needs first to train a 512-step intermediate model, then a 256-step, to finally arrive to the 128-step model. The distillation objective is to minimize the output differences between the teacher and student models. Progressive distillation better maintains the synthesis quality than step-skipping, but it incurs high distillation time and must comply with specific values of T and T' due to progressive halvings.

Our objective is to design an effective distillation method that achieves high quality in (any) small sampling step and concurrently incurs low distillation time. We propose SFDDM, a single-fold distillation framework able to reduce a T-step teacher diffusion model into a T'-step student diffusion model in a single fold. Thanks to its single-fold nature, SFDDM offers the flexibility to distill the teacher model into a student with any number of steps. To such an end, we first define a new student model, which can extract knowledge of the teacher diffusion model, by an arbitrary steps sub-sequence. Our forward process definition solves the challenge of aligning the variables of teacher and student models, enabling flexible single-fold distillation. Secondly, when training the reverse denoise process of the student model, we minimize not only the difference in the model outputs but also in the hidden variables at each step to better preserve the image synthesis quality.

To demonstrate the effectiveness, we evaluate SFDDM against sampling-skip [21] and progressive distillation [19] on CIFAR-10, CelebA, LSUN-Church, LSUN- Bedroom image datasets and 2D Swiss Roll tabular dataset. We compare their synthesis quality in terms of Fréchet Inception Distance (FID) [6] of distilling a 1024-step diffusion model into 128-step and 16-step models. SFDDM achieves the lowest FID as well as the best perceptual quality, also with flexibility on the numerical relation between the teacher and the student, i.e., works for both 1024 to 128 or 100 steps. Further, the distillation effectiveness is validated on semantic input-output consistency and image interpolation.

We summarize the contributions of this paper as follows:

- We propose a novel single-fold distillation algorithm, SFDDM, which can agilely compress teacher diffusion into a student diffusion model of any step in one fold.
- We define a new forward process for student diffusion, which aligns and approximates student and teacher Markovian variables, enabling flexible single-fold distillation.
- We design effective training for student diffusion by minimizing the difference of output and hidden variables with respect to the teacher diffusion.

– We experimentally demonstrate superiority of SFDDM in achieving high-quality sampling data by less than 1% number of steps.

2 Related studies

Diffusion models [12,23,8,7,5,17] first step-wise destroy in a forward process the training data structure and then learn how to restore the data structure from noise in a reverse process. DDPM [7] proposed the first stable and effective implementation capable of high-quality image synthesis. However, diffusion models, including DDPM, suffer from slow inference stemming from immense intermediate hidden variables, each the size of the synthetic output, as well as the complex architecture. This sparked research on how to accelerate data synthesis with related work exploring three main directions.

Fast sampling. Diffusion models mostly rely on an UNet [18] architecture combining cross-attention and ResNet blocks for denoising. Fast sampling [11,25] facilitates the reverse process by optimizing the computations of UNets. [11] proposes an efficient UNet by identifying the redundancy of the original model and reducing the computation, while [25] further comprehensively analyzes and simplifies each component. These optimizations are orthogonal to other acceleration techniques.

Sampling step skipping. DDIM [21] focuses on generalizing the Markovian diffusion of DDPM via a family of non-Markovian processes. These are deterministic and thus faster. Accordingly, it is able to reduce the required number of sampling steps on the trained DDPM model, without necessity of retraining. A noticeable limitation is that DDIM trades off the quality of generated data. as DDIM sampling approximates the procedure of the original model with skipped intermediate steps.

Knowledge distillation. Previously explored for GANs [4,13], distillation [1,3] allows to transfer knowledge from a large trained teacher model to a smaller student model for faster inference. To train a student model, progressive distillation [19] halves repeatedly the steps of a teacher model until the desired number of steps has been reached. [15] further extends the folding optimization to classifier-free guided diffusion implementation of Text-to-Image tasks. Although progressive distillation delivers increasingly efficient inference, each halving requires to train a new student model which multiples the training effort and impacts the output quality due to added approximation noise at each folding. Consistency models are proposed to improve the sample quality with few steps [22]. A consistency model can be directly trained or obtained by distilling a trained teacher model.

3 Single-fold distillation

We rethink knowledge distillation of diffusion models to reduce the number of sampling steps by proposing single-fold distillation. Instead of progressive multiple folds [19], which introduces distortion and costs extra training effort at

each fold, SFDDM directly distills the teacher model to a student model with a given number of steps in a single fold. One crucial challenge is the alignment from the teacher's to the student's hidden variables, as the Markov chain is defined by every two consecutive variables but the student has much fewer steps.

We first introduce the preliminaries of a DDPM model used as a teacher model, including the definition of the forward/reverse process and the training objective. Then we present SFDDM which defines the forward and reverse process of the student by aligning and matching the hidden variables of the teacher. Finally, we design the distillation algorithm for deployment, which shows the training procedure of the student accordingly. For ease of presentation, we set the number of steps in the student model as a divisor of the number of steps in the teacher model, but the method is valid for an arbitrary number of student steps, i.e. fractional teacher/student step ratios.

3.1 Preliminary

DDPM is composed of a reverse (denoising) and a forward (noising) process, through T steps. Its objective is to learn the denoising process via a given forward process.

Reverse process: The optimization objective of diffusion models [20] is derived by variational inference. Given the observed data $\boldsymbol{x}_0$, the diffusion model is a probabilistic model which specifies the joint distribution $p_\theta(\boldsymbol{x}_{0:T})$, where $\boldsymbol{x}_1, ..., \boldsymbol{x}_T$ are latent variables with the same dimensions as $\boldsymbol{x}_0$, and θ are learnable model parameters. $p_\theta(\boldsymbol{x}_{0:T})$ is a Markov chain that samples from $\boldsymbol{x}_T$ to $\boldsymbol{x}_0$, $p_\theta\left(\boldsymbol{x}_{0:T}\right) := p_\theta\left(\boldsymbol{x}_T\right) \prod_{t=1}^{T} p_\theta\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right)$. DDPM assumes that $p_\theta\left(\boldsymbol{x}_T\right) = \mathcal{N}\left(\boldsymbol{x}_T; \mathbf{0}, \boldsymbol{I}\right)$ and

$$p_\theta\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right) = \mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta\left(\boldsymbol{x}_t, t\right), \sigma_t^2 \boldsymbol{I}\right),$$

where $\sigma_t \in \mathbb{R}_{\geq 0}$. $p_\theta(\boldsymbol{x}_{0:T})$ is called the *reverse process*. It gradually denoises a noise $\boldsymbol{x}_T \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{I}\right)$.

Forward process: To derive a lower bound on the log likelihood of the observed data, diffusion models introduce the approximate posterior $q\left(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0\right)$ which is a Markov chain, $q\left(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0\right) := \prod_{t=1}^{T} q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}\right)$, that samples from $\boldsymbol{x}_1$ to $\boldsymbol{x}_T$. Then the log data likelihood can be decomposed as

$$\mathbb{E}\left[\log p_\theta\left(\boldsymbol{x}_0\right)\right] = \mathbb{E}_q\left[\log \frac{p_\theta\left(\boldsymbol{x}_{0:T}\right)}{q\left(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0\right)}\right] + D_{\mathrm{KL}}\left(q\left(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0\right) \| p_\theta\left(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0\right)\right).$$

Thus, we have the lower bound $\mathbb{E}\left[\log p_\theta\left(\boldsymbol{x}_0\right)\right] \geq \mathbb{E}_q\left[\log \frac{p_\theta(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)}\right]$. When training the diffusion model, to maximize $\mathbb{E}\left[\log p_\theta\left(\boldsymbol{x}_0\right)\right]$, the parameters θ are learned to minimize the negative evidence lower bound:

$$\arg\min_\theta \mathbb{E}_q\left[\log q\left(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0\right) - \log p_\theta\left(\boldsymbol{x}_{0:T}\right)\right]. \tag{1}$$

The Markov chain $q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)$ is called the *forward process*. It progressively turns the data x_0 into noise. The conditional distribution in each forward step is defined as:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) := \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{\frac{\alpha_t}{\alpha_{t-1}}} x_{t-1}, \left(1 - \frac{\alpha_t}{\alpha_{t-1}}\right) \boldsymbol{I}\right), \tag{2}$$

where $\alpha_t \in (0, 1]$ and $\alpha_1, ..., \alpha_T$ is a decreasing sequence, which ensures that the values on the diagonal of the covariance matrix are positive. Reparameterizing using the definition in Eq. (2), an important property of the forward process is that:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_0, (1 - \alpha_t)\boldsymbol{I}). \tag{3}$$

Training and sampling: According to the definition of the reverse p and forward q processes, α_t and σ_t for all t are not learnable parameters. Thus, DDPM simplifies Eq. (1) as:

$$\arg\min_{\theta} \sum_t \mathbb{E}_q[D_{\mathrm{KL}}(q(x_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \| p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t))]. \tag{4}$$

DDPM further chooses the form of $\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t)$ to be:

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\frac{\alpha_t}{\alpha_{t-1}}}} \left(\boldsymbol{x}_t - \frac{1 - \frac{\alpha_t}{\alpha_{t-1}}}{\sqrt{1 - \alpha_t}} \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right), \tag{5}$$

where $\boldsymbol{\epsilon}_\theta$ is a function with trainable parameters θ. According to the above definitions of the forward and reverse processes and applying the parameterization shown in Eq. (5), Eq. (4) can be simplified as $\arg\min_\theta L(\theta)$, where:

$$L(\theta) := \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\epsilon}_t}\left[\gamma_t \left\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta\left(\sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1 - \alpha_t}\boldsymbol{\epsilon}_t, t\right)\right\|^2\right] \tag{6}$$

with $\gamma_t = \frac{(\alpha_{t-1} - \alpha_t)^2}{2\sigma_t^2 \alpha_t \alpha_{t-1}(1 - \alpha_t)}$ and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. It is important to note that when deriving this loss, DDPM reparameterize $\boldsymbol{x}_t$ as

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1 - \alpha_t}\boldsymbol{\epsilon}_t, \tag{7}$$

using the property in Eq. (3). DDPM further simplifies L by setting $\gamma_t = 1$ independent of $\alpha_{1:T}$.

The number of sampling steps T decides the generalization capability and sampling cost of diffusion models. A big T leads to a reverse process with high generalization capability that better captures the pattern of x_0. However, it also increases the sampling time and makes the sampling from DDPMs significantly slower than other generative models, e.g, GANs. Such inefficiency promotes our design of SFDDM to reduce the number of sampling steps via knowledge distillation.

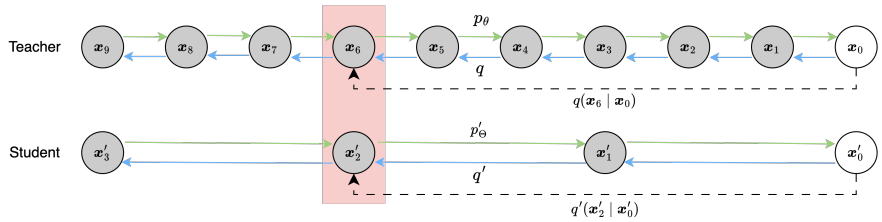


Fig. 1: Single-Fold Distillation of Diffusion Model (SFDDM). The student accelerates the inference by a small number of steps T' instead of a large T. We use $T = 9$ and $T' = 3$ in the figure for readability. To align the teacher and student Markov chains, we propose to match the intermediate hidden variables to make, e.g., $q'(\boldsymbol{x}'_2 = \boldsymbol{x}_6 | \boldsymbol{x}'_0 = \boldsymbol{x}_0)$ equal to $q(\boldsymbol{x}_6 | \boldsymbol{x}_0)$.

3.2 Single-fold Distilled Diffusion (SFDDM)

Instead of multiple folds like progressive distillation, which introduces distortion at every fold by retraining, we want to extract knowledge from the teacher model through a single fold. The first consideration is aligning steps from a T-step teacher to steps of a given smaller T'-step student. Here, our goal is to distill the knowledge of the teacher model by mimicking its hidden variables $(\boldsymbol{x}_1, ..., \boldsymbol{x}_T)$ by a compressed student model[6] with hidden variables $(\boldsymbol{x}'_1, ..., \boldsymbol{x}'_{T'})$, where $T' \ll T$.

A crucial challenge stems from the need to map a subset of multiple consecutive steps at the teacher into one single step at the student (see Fig. 1). For example, given an index[7] $c \cdot t$, where $c = T/T'$ and $c \in \mathbb{Z}$, according to Sec. 3.1, the distributions we can explicitly obtain from the teacher are $q(\boldsymbol{x}_{c \cdot t} \mid \boldsymbol{x}_{c \cdot t-1})$ (see Eq. 2). However, mapping $\boldsymbol{x}'_t$ with $\boldsymbol{x}_{c \cdot t}$ from the student to the teacher does not hold for the next step, i.e., $\boldsymbol{x}'_{t-1}$ does not correspond to $\boldsymbol{x}_{c \cdot t-1}$, which is supposed to map $\boldsymbol{x}_{c(t-1)}$. Thus, when distilling the DDPM-like Markov chains, it is not reasonable to simply and straightforwardly simulate $q'(\boldsymbol{x}'_t | \boldsymbol{x}'_{t-1})$ as $q(\boldsymbol{x}_{c \cdot t} | \boldsymbol{x}_{c \cdot t-1})$ while correspondingly estimating $p'_\Theta(\boldsymbol{x}'_t | \boldsymbol{x}'_{t-1})$ as $p_\theta(\boldsymbol{x}_{c \cdot t} | \boldsymbol{x}_{c \cdot t-1})$, where the notations q' and p'_Θ are used to represent the forward process and reverse process of the student model, respectively.

To overcome this challenge, we define a novel student model as follows. Note that the key definition for diffusion models is the forward process, as it determines the variational inference and dominates the training of the model. Therefore, to better distill the knowledge from the teacher, we design the forward process of the student $q'(\boldsymbol{x}'_{1:T'} \mid \boldsymbol{x}'_0)$ by extracting the teacher's forward process $q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)$. Specifically, to ensure correspondence between the latent variables in the two diffusion models, given any $t \in [1, T']$, the proposed distillation algorithm **aims** to make $q'(\boldsymbol{x}'_t = \boldsymbol{x}_{c \cdot t} \mid \boldsymbol{x}'_0 = \boldsymbol{x}_0)$ equal to $q(\boldsymbol{x}_{c \cdot t} | \boldsymbol{x}_0)$, which has a close-form solution (see Eq. 3). After fixing q' by such an approximation in the forward process, the

[6] Hereon we use '′' in symbols to indicate the corresponding student variables.

[7] For simplicity, we assume that T is divisible by T' but this is not necessary (see Sec. 3.6).

reverse process of the student is constructed accordingly. In the following, we show in detail how to define and train a student model.

3.3 The forward process of the student model

To distill the DDPM teacher, we also assume that the student model has a Markovian forward process. Thus $q'\left(\boldsymbol{x}'_{1:T'} \mid \boldsymbol{x}'_0\right)$ is a Markov chain that can be factorized as

$$q'\left(\boldsymbol{x}'_{1:T'} \mid \boldsymbol{x}'_0\right) := \prod_{t=1}^{T'} q'\left(\boldsymbol{x}'_t \mid \boldsymbol{x}'_{t-1}\right).$$

As shown in Eq. (2), the forward process of the teacher is defined by a decreasing sequence $\alpha_1, ..., \alpha_T$. As for the student, different from the Gaussian distribution shown in Eq. (2), we set the student's forward process to the following form:

$$q'\left(\boldsymbol{x}'_t \mid \boldsymbol{x}'_{t-1}\right) := \mathcal{N}\left(\boldsymbol{x}'_t; \sqrt{\frac{\alpha_{c\cdot t}}{\alpha_{c\cdot t-c}}}\boldsymbol{x}'_{t-1}, \left(1 - \frac{\alpha_{c\cdot t}}{\alpha_{c\cdot t-c}}\right)\boldsymbol{I}\right), \tag{8}$$

for all $t \in [1, T']$ based on the elements of the sequence $\{\alpha_t\}_{t=1}^{T}$ (the hyper-parameters of the given teacher). Then, according to this forward process definition of the student, we have the property:

$$q'(\boldsymbol{x}'_t \mid \boldsymbol{x}'_0) = \mathcal{N}\left(\boldsymbol{x}'_t; \sqrt{\alpha_{c\cdot t}}\boldsymbol{x}'_0, \left(1 - \alpha_{c\cdot t}\right)\boldsymbol{I}\right). \tag{9}$$

This **ensures** that $q'(\boldsymbol{x}'_t = \boldsymbol{x}_{c\cdot t} \mid \boldsymbol{x}'_0 = \boldsymbol{x}_0) = q(\boldsymbol{x}_{c\cdot t}|\boldsymbol{x}_0)$, where $\boldsymbol{x}'_t$ corresponds to $\boldsymbol{x}_{c\cdot t}$. Thus, the student's Markov chain $q'\left(\boldsymbol{x}'_{1:T'} \mid \boldsymbol{x}'_0\right)$ can be regarded as a simplified copy of the teacher's forward process $q\left(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0\right)$.

Before introducing the reverse process, we derive some important forward process posteriors. Using Bayes' rule $q'\left(\boldsymbol{x}'_{t-1} \mid \boldsymbol{x}'_t, \boldsymbol{x}'_0\right) = q'\left(\boldsymbol{x}'_t \mid \boldsymbol{x}'_{t-1}, \boldsymbol{x}'_0\right) \frac{q'\left(\boldsymbol{x}'_{t-1}|\boldsymbol{x}'_0\right)}{q'\left(\boldsymbol{x}'_t|\boldsymbol{x}'_0\right)}$, and the Markov chain property that $q'\left(\boldsymbol{x}'_t \mid \boldsymbol{x}'_{t-1}, \boldsymbol{x}'_0\right) = q'\left(\boldsymbol{x}'_t \mid \boldsymbol{x}'_{t-1}\right)$, we have the posteriors:

$$q'\left(\boldsymbol{x}'_{t-1} \mid \boldsymbol{x}'_t, \boldsymbol{x}'_0\right) = \mathcal{N}(\boldsymbol{x}'_{t-1}; \frac{(1-\alpha_{c\cdot t-c})\sqrt{\alpha_{c\cdot t}}}{(1-\alpha_{c\cdot t})\sqrt{\alpha_{c\cdot t-c}}}\boldsymbol{x}'_t + \frac{\alpha_{c\cdot t-c} - \alpha_{c\cdot t}}{(1-\alpha_{c\cdot t})\sqrt{\alpha_{c\cdot t-c}}}\boldsymbol{x}'_0, \sigma'_t\boldsymbol{I}), \tag{10}$$

where $\sigma'_t = \frac{(1-\alpha_{c\cdot t-c})(\alpha_{c\cdot t-c}-\alpha_{c\cdot t})}{(1-\alpha_{c\cdot t})\alpha_{c\cdot t-c}}$.

3.4 The reverse process of the student model

In the following, we define the reverse process of the student model according to its forward process. Similarly, it is also a Markov chain represented as $p'_\Theta(\boldsymbol{x}'_{0:T'}) := p'_\Theta\left(\boldsymbol{x}'_T\right)\prod_{t=1}^{T'} p'_\Theta\left(\boldsymbol{x}'_{t-1} \mid \boldsymbol{x}'_t\right)$, where $p'_\Theta\left(\boldsymbol{x}'_T\right) = \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{I}\right)$ and Θ represents the learnable parameters of the student.

Then, we decide the form of $p'_\Theta\left(\boldsymbol{x}'_{t-1} \mid \boldsymbol{x}'_t\right)$. Referring to Eq. (9), we can reparameterize $\boldsymbol{x}'_t$ as a linear combination of $\boldsymbol{x}'_0$ and $\epsilon_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ that is $\boldsymbol{x}'_t =$

$\sqrt{\alpha_{c\cdot t}}\boldsymbol{x}'_0 + \sqrt{1-\alpha_{c\cdot t}}\epsilon_t$. Let the model $\epsilon'_\Theta(\boldsymbol{x}'_t, t)$ predict ϵ_t, we have a prediction of $\boldsymbol{x}'_0$ given $\boldsymbol{x}'_t$:

$$f_\Theta^{(t)}\left(\boldsymbol{x}'_t\right) := \left(\boldsymbol{x}'_t - \sqrt{1-\alpha_{c\cdot t}}\cdot\epsilon'_\Theta\left(\boldsymbol{x}'_t, t\right)\right)/\sqrt{\alpha_{c\cdot t}} \tag{11}$$

According to Eq. (4), we know that in the student diffusion model, $p'_\Theta\left(\boldsymbol{x}'_{t-1} \mid \boldsymbol{x}'_t\right)$ is a distribution that predicts $q'(\boldsymbol{x}'_{t-1} \mid \boldsymbol{x}'_t, \boldsymbol{x}'_0)$. Thus, we define that for all $t \in [1, T']$,

$$p'_\Theta\left(\boldsymbol{x}'_{t-1} \mid \boldsymbol{x}'_t\right) = q'(\boldsymbol{x}'_{t-1} \mid \boldsymbol{x}'_t, f_\Theta^{(t)}\left(\boldsymbol{x}'_t\right)). \tag{12}$$

Then, based on Eq. (10) and Eq. (12), by replacing $\boldsymbol{x}'_0$ as the predictor $f_\Theta^{(t)}\left(\boldsymbol{x}'_t\right)$, we have:

$$p'_\Theta\left(\boldsymbol{x}'_{t-1} \mid \boldsymbol{x}'_t\right) = \mathcal{N}\left(\boldsymbol{x}'_{t-1}; \boldsymbol{\mu}'_\Theta\left(\boldsymbol{x}'_t, t\right), \sigma'_t\boldsymbol{I}\right), \tag{13}$$

where the mean is:

$$\boldsymbol{\mu}'_\Theta\left(\boldsymbol{x}'_t, t\right) = \frac{(1-\alpha_{ct-c})\sqrt{\alpha_{ct}}}{(1-\alpha_{ct})\sqrt{\alpha_{ct-c}}}\boldsymbol{x}'_t + \frac{\alpha_{ct-c}-\alpha_{ct}}{(1-\alpha_{ct})\sqrt{\alpha_{ct-c}}}f_\Theta^{(t)}\left(\boldsymbol{x}'_t\right). \tag{14}$$

3.5 Distillation procedure

Having defined the forward and reverse processes, here we design the algorithm for training the student model. The reparameterization of $\boldsymbol{\mu}'_\Theta\left(\boldsymbol{x}'_t, t\right)$ shown in Eq. (14) can be applied to derive the training loss of the student. For maximizing the log data likelihood of the observed data $\{\boldsymbol{x}'_0\}$ on the student[8], referring to Eq. (4), it is equivalent to minimize the Kullback-Leibler divergence between Eq. (10) and Eq. (13). Then by using Eq. (10), (13), (14) and (11), we have the following loss for training the student:

$$L(\Theta) := \sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}'_0,\boldsymbol{\epsilon}'_t}\left[\gamma'_t\left\|\boldsymbol{\epsilon}'_t - \boldsymbol{\epsilon}_\Theta\left(\sqrt{\alpha_{c\cdot t}}\boldsymbol{x}'_0 + \sqrt{1-\alpha_{c\cdot t}}\boldsymbol{\epsilon}'_t, t\right)\right\|^2\right], \tag{15}$$

where $\gamma'_t = \frac{(\alpha_{c\cdot t-c}-\alpha_{c\cdot t})^2}{2\sigma'^2_t\alpha_{c\cdot t}\alpha_{c\cdot t-c}(1-\alpha_{c\cdot t})}$. By the loss (Eq. 15), we can train the defined student model from scratch using the observed data $\{\boldsymbol{x}'_0\}$. However, in order to extract the knowledge from the trained teacher, in the following we connect the training of the student with the trained teacher.

In the derivation of the loss $L(\Theta)$ (Eq. 15), we use the following reparameterization:

$$\boldsymbol{x}'_t = \sqrt{\alpha_{c\cdot t}}\boldsymbol{x}'_0 + \sqrt{1-\alpha_{c\cdot t}}\boldsymbol{\epsilon}'_t. \tag{16}$$

According to Eq. (7), we know that $\boldsymbol{x}_{c\cdot t} = \sqrt{\alpha_{c\cdot t}}\boldsymbol{x}_0 + \sqrt{1-\alpha_{c\cdot t}}\boldsymbol{\epsilon}_{c\cdot t}$. In our distillation, we want to make the hidden variable $\boldsymbol{x}'_t$ of the student equal to its corresponding hidden variable $\boldsymbol{x}_{c\cdot t}$ of the teacher. As the student and the teacher use the same observed data, $\boldsymbol{x}'_0 = \boldsymbol{x}_0$, by Eq. (16), if we assume $\boldsymbol{\epsilon}'_t = \boldsymbol{\epsilon}_{c\cdot t}$, then

[8] In distillation, student and teacher observe the same training samples. Thus, $\boldsymbol{x}'_0$ always equals $\boldsymbol{x}_0$ and $\{x'_0\}$ is equivalent to $\{x_0\}$. To avoid confusion, we use $\{x'_0\}$ to represent the observed data when training the student.

we can have $\boldsymbol{x}'_t = \boldsymbol{x}_{c\cdot t}$. By the training loss of the teacher $L(\theta)$ (Eq. 6), we know that the output $\boldsymbol{\epsilon}_\theta\left(\sqrt{\alpha_{c\cdot t}}\boldsymbol{x}_0 + \sqrt{1-\alpha_{c\cdot t}}\boldsymbol{\epsilon}_{c\cdot t}, c\cdot t\right)$ from the trained function $\boldsymbol{\epsilon}_\theta$ of the teacher is a good predictor for $\boldsymbol{\epsilon}_{c\cdot t}$ (also for $\boldsymbol{\epsilon}'_t$). Thus, we naturally rewrite the student loss (Eq. 15) as

$$\begin{aligned} L(\Theta) := \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{x}'_0, \boldsymbol{\epsilon}'_t}[\gamma'_t \| \boldsymbol{\epsilon}_\theta\left(\sqrt{\alpha_{c\cdot t}}\boldsymbol{x}'_0 + \sqrt{1-\alpha_{c\cdot t}}\boldsymbol{\epsilon}'_t, c\cdot t\right) \\ - \boldsymbol{\epsilon}_\Theta\left(\sqrt{\alpha_{c\cdot t}}\boldsymbol{x}'_0 + \sqrt{1-\alpha_{c\cdot t}}\boldsymbol{\epsilon}'_t, t\right) \|^2]. \end{aligned} \tag{17}$$

Training: For distillation, we train the student according to the loss (Eq. 17). During the implementation in Sec. 4, we simplify the loss by setting $\gamma'_t = 1$, a simpler approach shown beneficial for sample quality.

Sampling: We need an input noise $\boldsymbol{x}'_{T'}$ when sampling images from the trained student. The straightforward way is to draw the noise by $\boldsymbol{x}'_{T'} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Reducing the number of sampling steps can negatively impact a model's generalization ability. To avoid the risk of generating poor-quality images from certain input sample points when performing direct random sampling from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, we propose a method to reduce the sampling space of $\boldsymbol{x}'_{T'}$. This approach ensures that the student model maintains strong performance even with fewer sampling steps (e.g., 4 steps), preserving high-quality outputs while improving efficiency. For a distilled student model, a high-quality input noise set $\{\hat{\boldsymbol{x}}'\}$ can be obtained by $\arg\min_{\hat{\boldsymbol{x}}'} \|StudentDiffusion(\hat{\boldsymbol{x}}') - \boldsymbol{x}'_0\|$. A random linear weighted combination of elements from this set will be used for the random sampling of $\boldsymbol{x}'_{T'}$.

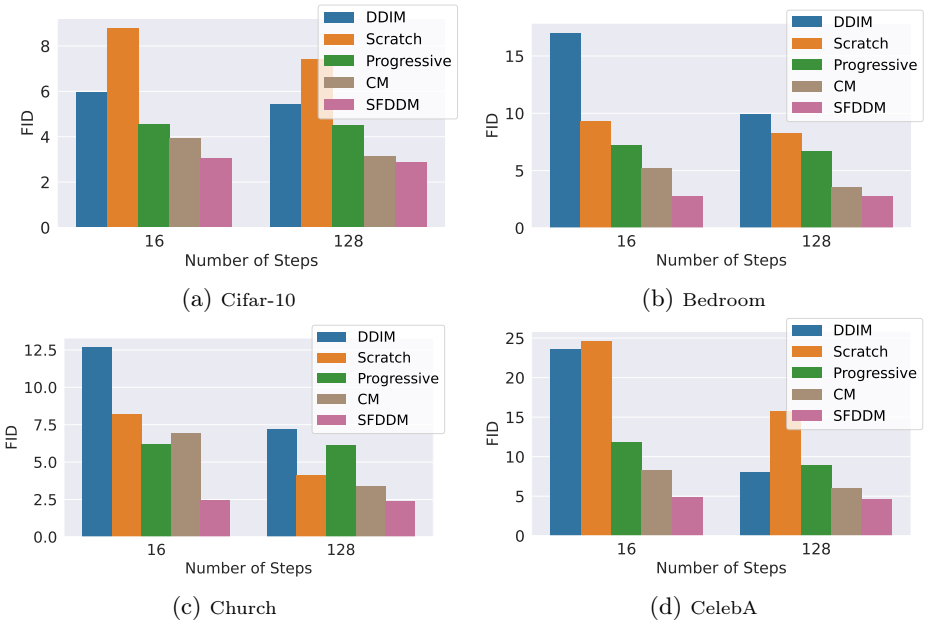
Since the reverse process of the student is defined by Eq. (13), given the input $\boldsymbol{x}'_{T'}$, we can steadily sample all variables from $\boldsymbol{x}'_{T'-1}$ to $\boldsymbol{x}'_0$. Note that the student model is also a DDPM model. Any sampling algorithm compatible with DDPM can be applied to the distilled student.

3.6 Distillation on flexible sub-sequence

In the previous derivation, the student extracts the knowledge from a special variable subset $\{\boldsymbol{x}_{c\cdot t}\}$ of the teacher where $t \in [1, T']$. Indeed, our proposed method can be extended to a more general case where we distill the knowledge from any given subset $\{\boldsymbol{x}_{\phi_0}, ..., \boldsymbol{x}_{\phi_{T'}}\}$ of the teacher where ϕ is an increasing sub-sequence of $\{0, ..., T\}$, $\phi_0 = 0$ and $\phi_{T'} = T$. In this general case, the Gaussian mean of $p'_\Theta\left(\boldsymbol{x}'_{t-1} \mid \boldsymbol{x}'_t\right)$ (see Eq. 13) is

$$\boldsymbol{\mu}'_\Theta\left(\boldsymbol{x}'_t, t\right) = \frac{(1-\alpha_{\phi_{t-1}})\sqrt{\alpha_{\phi_t}}}{(1-\alpha_{\phi_t})\sqrt{\alpha_{\phi_{t-1}}}}\boldsymbol{x}'_t + \frac{\alpha_{\phi_{t-1}} - \alpha_{\phi_t}}{(1-\alpha_{\phi_t})\sqrt{\alpha_{\phi_{t-1}}}}\mathcal{F}_\Theta^{(t)}\left(\boldsymbol{x}'_t\right),$$

where $\mathcal{F}_\Theta^{(t)}\left(\boldsymbol{x}'_t\right) := \left(\boldsymbol{x}'_t - \sqrt{1-\alpha_{\phi_t}} \cdot \epsilon'_\Theta\left(\boldsymbol{x}'_t, t\right)\right) / \sqrt{\alpha_{\phi_t}}$ is a prediction of $\boldsymbol{x}'_0$ given $\boldsymbol{x}'_t$. Remarkably, T no longer needs to be divisible by T'. This extension is beneficial as it greatly expands the applicability of our distillation under various steps of teacher diffusion models.



(a) Cifar-10 (b) Bedroom

(c) Church (d) CelebA

Fig. 2: FID under different number of sampling steps from the teacher $T = 1024$, on four datasets.

4 Evaluation

SFDDM distills the knowledge of any arbitrary DDPM-like teacher models with efficiency and high sampling quality. We demonstrate its effectiveness by four image benchmark datasets: CIFAR-10 [10], CelebA [14], LSUN-Church and LSUN-Bedroom [24], as well as one tabular dataset: 2D Swiss Roll following the diffusion model settings [16]. For each image dataset, we distill the same teacher diffusion model of DDPM into 16, 100, or 128 student steps. Note that although the original DDPM contains 1000 sampling steps, in this paper, we set it as 1024 steps in order to compare with progressive distillation [19], which requires a number of steps that is a power of 2 for progressive halving. For the 2D Swiss Roll dataset, we distill a teacher model with 500 steps into a student with 50 steps by SFDDM. The evaluation metric applied is FID together with perceptual visualization of sampled images.

4.1 Experimental setups

Our evaluation is carried out by Alienware-Aurora-R13 with Ubuntu 20.04. The machine is equipped with 64G memory, 4× GeForce RTX 3090 GPU and 16-core Intel i9 CPU. Each of the 8 P-cores has two threads, hence each machine contains 24 logical CPU cores in total. We consider various image generation benchmarks (CIFAR-10, CelebA, LSUN-Bedroom, LSUN-Church), with resolution varying

from 32×32 to 128×128. All experiments for the teacher diffusion model use the sigmoid schedule, particularly good for large images, following the settings of [9] and all models use a UNet architecture same as DDPM [7]. Our schedule of the student model is defined in Sec. 3.3 accordingly. Our training setup closely matches the open source code by DDPM. For the training of the student model, we choose Adam optimizer with learning rate fixed to 2×10^{-5} while other hyper-parameters remain the same as the default setting of PyTorch Adam. An interesting observation on SFDDM is that experimentally using l_1 norm on L_Θ leads to faster convergence comparing to l_2 norm. FID scores are computed across 10K images.

4.2 Sampling quality and efficiency

To demonstrate the quality and efficiency of SFDDM, we report the FID in Fig. 2 on all four image datasets comparing against DDIM, progressive distillation ("Progressive"), Consistency model (CM) [22], and training directly on the smaller model with the same dataset as the teacher model ("From scratch").

In general, our SFDDM achieves the best FID over different datasets and different small T'. From the results, we observe that the sampling data quality increases with the increase of T', as more sampling steps match more hidden variables of the teacher model, better approximating the teacher distributions.

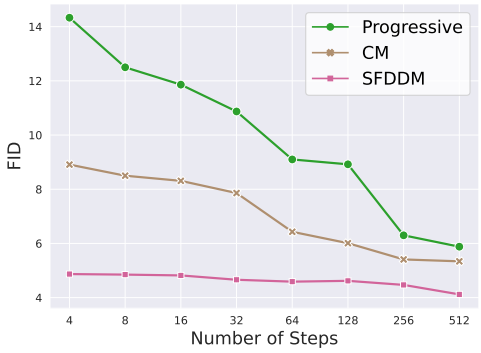


Fig. 3: FID of the methods with different number of sampling steps on CelebA-HQ.

Comparing the baselines, DDIM achieves mostly the lowest quality in sampled images as shown by high FID scores. This stems from the nature that DDIM **does not** retrain a student model but focuses on improving the inference efficiency of the original model. Thus, it benefits from simplicity but falls short in terms of quality, when skipping too many intermediate steps during sampling, e.g., $T' = 16$. On the other hand, training from scratch on a small diffusion model comes in as

the second worst in terms of FID, due to the difficulty of capturing image features with a small number of steps. Progressive distillation yields marginal improvement as multiple folding and retraining increases distortion from teacher knowledge due to noises it introduces at each folding. An interesting finding emerges in the context of the LSUN-Church dataset with 128 sampling steps. Here, training from scratch surpasses progressive distillation in FID performance. This can be attributed to the fact that, with a relatively large number of steps, direct training exhibits superior quality compared to progressive distillation, where systematic distortion occurs fold by fold. This is consistent with the conclusion in [19]. We also compare the student FID for progressive distillation, CM, and SFDDM with varying sampling steps from 4 to 512 in Fig. 3. Overall, SFDDM consistently outperforms the baselines. We also visualize the corresponding student model output of step 4 and 16 in Fig. 4, demonstrating limited quality loss of small sampling steps by SFDDM.

Fig. 4: Visualizied images generated by student model of SFDDM on CelebA-HQ.

4.3 Distillation with different sub-sequences

In accordance with Sec. 3.6, our algorithm can be extended to accommodate flexible sub-sequences of the student model. Here, we compare FID values for different choices to demonstrate their impact. Specifically, the different cases of flexible sub-sequence are designed by various degrees of concentration of mapped steps around the midpoint element. We define it by the percentage of elements distributed uniformly within a 5% range near the midpoint element (i.e., 512) while the others are uniformly distributed in the remaining range of steps. Moreover, In our results presented in Tab. 1, we include the concentration degrees of 40%, 20%, plus Scattered. "Scattered" refers to the student model matching a sparse sub-sequence spread across the full teacher Markov chain. Scattered allows to cover more knowledge of the noising/denoising procedure form the teacher. Yet, concentrated choices, in which elements are nearby and centered in a partial part of the teacher chain, are still able to distill high-quality diffusion models, as shown by similarity in FID scores.

Table 1: Distilling the same teacher with different sub-sequences on CelebA-HQ with $T' = 16$.

sub-sequence	Concentrated (40%)	Concentrated (20%)	Scattered
FID	4.85	4.69	4.82

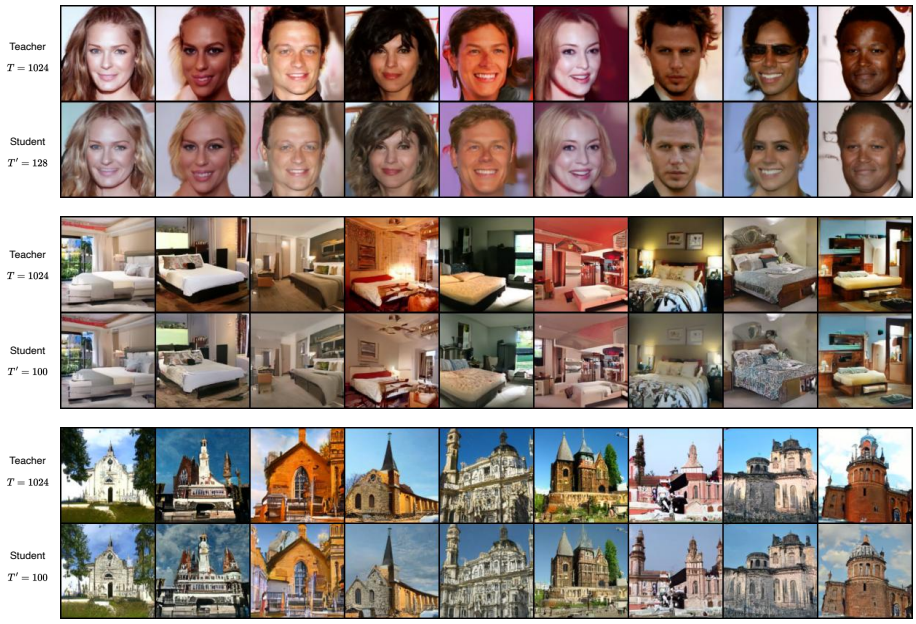


Fig. 5: Consistency on CelebA-HQ, LSUN-Bedroom and LSUN-Church: inputing the same noise.

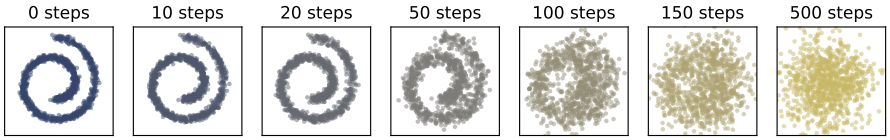
4.4 Consistency between teacher and student

We zoom into the consistency between images sampled by the teacher and student models when inputting identical noise. It is important to note that for a fair comparison of the distillation results, we use the DDIM sampling method (also applied in Sec. 4.5) for both the DDPM teacher and our SFDDM student. This is because the output of the original DDPM sampling is not solely determined by the input noise, owing to the introduced random factor during the stochastic generative process. In contrast, DDIM sampling ensures pair correspondence, i.e., same input, same output, facilitating a clear and accurate comparison. The outcomes across three distinct datasets are illustrated in Fig. 5. It is noteworthy that we employ different sampling steps for the student among different datasets, thereby validating the flexibility of our approach under varying numbers of sub-sequences, where the number of steps is not a power of 2. As evidenced by the images, it is clearly observed that inputting identical noise leads to similar

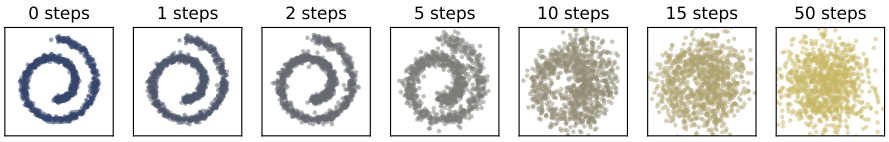
outputs, showing our effectiveness in transferring knowledge from the teacher to a student having only approximately 1/10 of the steps.

4.5 Interpolation on the teacher and the student

Fig. 6: Interpolation on the distilled student and original teacher.



(a) Teacher diffusion forward process of 500 steps on 2D Swiss Roll.



(b) Student diffusion forward process of 50 steps on 2D Swiss Roll.

Fig. 7: Forward process of teacher and SFDDM student model on 2D Swiss Roll dataset.

We further assess the efficacy of knowledge transfer through measuring the similarity of semantic interpolation between the teacher and student models. We evenly interpolate between two given noises to showcase the intermediate sampled data in Fig. 6. The results reveal a stable visual interpolation in the generated images since the input noise encodes distinctive high-level features of the image. Consequently, the interpolation data implicitly captures the features between the two noises into perceptually similar outputs. When comparing the output of the teacher and student, we observe similarity in each interpolation, showcasing the effectiveness of distillation as the student successfully inherits a stable and consistent sampling capability from the teacher.

4.6 Distillating tabular data

To assess the generality and applicability of SFDDM among different types of data, we apply SFDDM on the tabular 2D Swiss Roll dataset, visualized in Fig. 7. On this tabular DDPM, the forward process turns Swiss Roll-like points into randomly distributed 2D points. Contrarily, the reverse process constructs a Swiss Roll distribution according to randomly distributed points. Fig. 7 presents the forward process of both teacher and student models. The results show that our proposed algorithm SFDDM works on DDPM for distilling the generation of tabular data by demonstrating the similar capability of generating 2D Swiss Roll points with student sampling steps 10x fewer than the teacher diffusion.

5 Conclusion

In this paper, we propose a novel and effective single-fold distillation method for diffusion models, SFDDM. In contrast to the prior study of progressive distillation, SFDDM is able to compress any T-step teacher model into any T'-step student model in single-fold distillation. The key enabling features are (i) new derivation of the forwarding process of the student model, which leverages the reparameterization of the teacher model and approximation of their Markovian states; and (ii) optimization of the denoise process of the student model by minimizing the difference of model outputs and distribution of the hidden variables. Our evaluation results on five datasets show that SFDDM achieves remarkable quality on FID, allowing to strike better quality-compute tradeoffs.

Acknowledgments. This research is part of the Priv-GSyn project, 200021E_229204 of Swiss National Science Foundation and the DEPMAT project, P20-22 / N21022, of the research programme Perspectief which is partly financed by the Dutch Research Council (NWO). This research was partly funded by the Spoke "FutureHPC & BigData" of the ICSC-Centro Nazionale di Ricerca in "High Performance Computing, Big Data and Quantum Computing", funded by the European Union - NextGenerationEU, and by the DYMAN project funded by the European Union - European Innovation Council under G.A. n. 101161930.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Asadi, N., Davari, M., Mudur, S., Aljundi, R., Belilovsky, E.: Prototype-sample relation distillation: Towards replay-free continual learning. In: International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA. Proceedings of Machine Learning Research, vol. 202, pp. 1093–1106. PMLR (2023)
2. Bishop, C.: Pattern recognition and machine learning. Springer google schola **2**, 531–537 (2006)

3. Cao, S., Li, M., Hays, J., Ramanan, D., Wang, Y., Gui, L.: Learning lightweight object detectors via multi-teacher progressive distillation. In: International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA. Proceedings of Machine Learning Research, vol. 202, pp. 3577–3598. PMLR (2023)
4. Cazenavette, G., Wang, T., Torralba, A., Efros, A.A., Zhu, J.: Generalizing dataset distillation via deep generative prior. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023. pp. 3739–3748. IEEE (2023)
5. Han, J., Feng, S., Zhou, M., Zhang, X., Ong, Y.S., Li, X.: Diffusion model in normal gathering latent space for time series anomaly detection. In: ECML PKDD 2024, Vilnius, Lithuania, September 9-13, 2024, Proceedings, Part III. Springer
6. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 6626–6637 (2017)
7. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
8. Ho, J., Saharia, C., Chan, W., Fleet, D.J., Norouzi, M., Salimans, T.: Cascaded diffusion models for high fidelity image generation. The Journal of Machine Learning Research **23**(1), 2249–2281 (2022)
9. Jabri, A., Fleet, D.J., Chen, T.: Scalable adaptive computation for iterative generation. In: International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA. Proceedings of Machine Learning Research, vol. 202, pp. 14569–14589. PMLR (2023)
10. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
11. Li, Y., Wang, H., Jin, Q., Hu, J., Chemerys, P., Fu, Y., Wang, Y., Tulyakov, S., Ren, J.: Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. arXiv preprint arXiv:2306.00980 (2023)
12. Liu, J., Yi, X., Wu, S., Yin, X., Zhang, T., Huang, X., Jin, S.: Continuous geometry-aware graph diffusion via hyperbolic neural PDE. In: ECML PKDD 2024, Vilnius, Lithuania, September 9-13, 2024, Proceedings, Part III. Springer (2024)
13. Liu, X., Liu, A., den Broeck, G.V., Liang, Y.: Understanding the distillation process from deep generative models to tractable probabilistic circuits. In: International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA. Proceedings of Machine Learning Research, vol. 202, pp. 21825–21838. PMLR (2023)
14. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (December 2015)
15. Meng, C., Rombach, R., Gao, R., Kingma, D., Ermon, S., Ho, J., Salimans, T.: On distillation of guided diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14297–14306 (2023)
16. Nagel, J.: `https://github.com/joseph-nagel/diffusion-demo/blob/main/notebooks/swissroll.ipynb`
17. Ninniri, M., Podda, M., Bacciu, D.: Classifier-free graph diffusion for molecular property targeting. In: ECML PKDD 2024, Vilnius, Lithuania, September 9-13, 2024, Proceedings, Part III. Springer

18. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., III, W.M.W., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III. Lecture Notes in Computer Science, vol. 9351, pp. 234–241. Springer (2015). https://doi.org/10.1007/978-3-319-24574-4_28, https://doi.org/10.1007/978-3-319-24574-4_28
19. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. In: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net (2022)
20. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. pp. 2256–2265. PMLR (2015)
21. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021)
22. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA. PMLR (2023)
23. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. ICLR 2021 (2020)
24. Yu, F., Zhang, Y., Song, S., Seff, A., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)
25. Zhao, Y., Xu, Y., Xiao, Z., Hou, T.: Mobilediffusion: Subsecond text-to-image generation on mobile devices. arXiv preprint arXiv:2311.16567 (2023)