See Further When Clear: Curriculum Consistency Model

Yunpeng Liu[1,2] Boxiao Liu[3] Yi Zhang[3] Xingzhong Hou[1,2] Guanglu Song[3] Yu Liu[3] Haihang You[1,*]
[1]Institute of Computing Technology, Chinese Academy of Sciences
[2]School of Computer Science and Technology, University of Chinese Academy of Sciences
[3]SenseTime Research
{liuyunpeng22b, houxingzhong, youhaihang}@ict.ac.cn
{liuboxiao, zhangyi17, songguanglu}@sensetime.com
liuyuisanai@gmail.com

Abstract

Significant advances have been made in the sampling efficiency of diffusion and flow matching models, driven by Consistency Distillation (CD), which trains a student model to mimic the output of a teacher model at a later timestep. However, we found that the knowledge discrepancy between student and teacher varies significantly across different timesteps, leading to suboptimal performance in CD. To address this issue, we propose the Curriculum Consistency Model (CCM), which stabilizes and balances the knowledge discrepancy across timesteps. Specifically, we regard the distillation process at each timestep as a curriculum and introduce a metric based on the Peak Signal-to-Noise Ratio (PSNR) to quantify the knowledge discrepancy of this curriculum, then ensure that the curriculum maintains consistent knowledge discrepancy across different timesteps by having the teacher model iterate more steps when the noise intensity is low. Our method achieves competitive single-step sampling Fréchet Inception Distance (FID) scores of 1.64 on CIFAR-10 and 2.18 on ImageNet 64x64. Moreover, we have extended our method to large-scale text-to-image models and confirmed that it generalizes well to both diffusion models (Stable Diffusion XL) and flow matching models (Stable Diffusion 3). The generated samples demonstrate improved image-text alignment and semantic structure since CCM enlarges the distillation step at large timesteps and reduces the accumulated error.

1. Introduction

Diffusion Models (DM) and Flow Matching (FM) are two leading methods for generative image synthesis. DM [9],[35],[37] generates samples by iteratively reversing a diffusion process, i.e., Stochastic Differential Equation (SDE), whereas FM [19],[39] constructs explicit probability paths, known as Probability Flow Ordinary Differential Equations (PF-ODE), between noise and data, incorporating the reversed diffusion process as a special case. Despite the ability to produce high-quality images of DM and FM, their performances in sampling efficiency are not satisfactory and often require a lot of function evaluations. With the introduction of Consistency Models (CM) [38], the Number of Function Evaluations (NFEs) required for sampling has been significantly reduced by enforcing self-consistency. In common, as shown in Figure 1, CM encourages the student model at timestep t (where $t \in [0, 1)$) to mimic the output of the teacher model at timestep u (where $u \in (t, 1]$). Latent consistency models (LCM) [26] employ self-consistency in the latent space, significantly reducing computational costs and extending CM to high-resolution text-to-image syntheses, thereby promoting the widespread application of CM.

We found a critical problem in CM that differences between student and teacher outputs are highly unstable across different timesteps, resulting in inefficient training. Specifically, we regard the distillation process that student learn from teacher as a curriculum and use knowledge discrepancy to evaluate the curriculum difficulty. Easy curriculum leads to unsatisfactory generation of details ($t \to 1$) and high-level features such as semantic and structural features ($t \to 0$). We visualize the issue in Figure 2, where we quantify the knowledge discrepancy based on the Peak Signal-to-Noise Ratio (PSNR) between the student and teacher outputs at different timesteps. The results indicate that the knowledge discrepancy of curriculums decreases gradually as t progresses from smaller values (corresponding to near-pure noise) to larger values (closer to the final image). However, most studies [38], [26] suffer from the instability of knowledge discrepancy, as they sample uniformly along the timesteps and use a fixed distillation step $l = u - t$ for the CM. As a result, the student model struggles to learn effectively from easy curriculums, which affects the semantic structure and details in the diffusion process. Recent works,

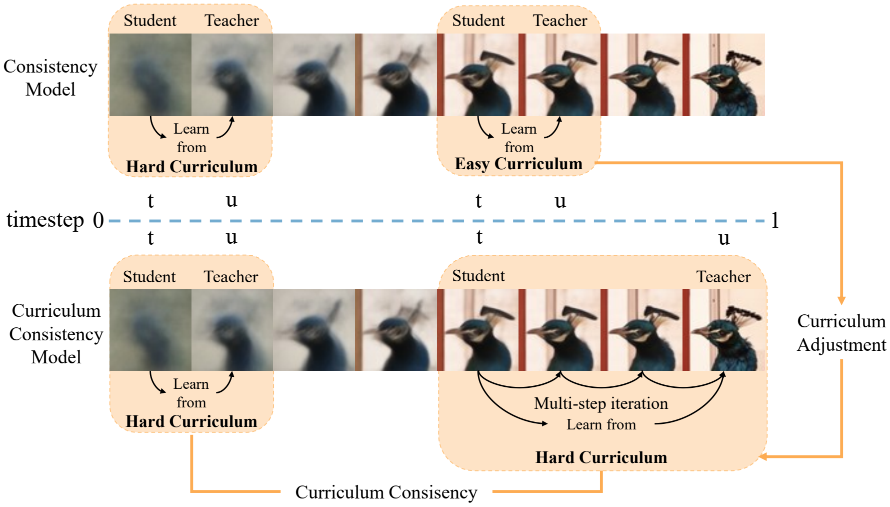

Figure 1. Comparison between Consistency Models (CM) and Curriculum Consistency Model (CCM). CM encourages the student model at timestep t to learn from the teacher model at timestep u, but the knowledge discrepancy (curriculum difficulty) at a larger timestep is small. CCM maintains curriculum consistency by dynamically adjusting the teacher model to a more challenging timestep through multi-step iteration.

iCT [36] and ECM [6], have also tackled similar instabilities in CMs. However, their focus is on addressing error accumulation, known as the ”Curse of Consistency” [6]. iCT progressively reduces the distillation step following a power-law schedule during training, while ECM refines this reduction process to achieve a smoother transition from diffusion models to consistency models. As shown in Figure 3, decreasing the distillation step reduces knowledge discrepancy, which makes the training inefficiency more obvious.

To address these issues, we propose an adaptive training method that stabilizes and balances the knowledge discrepancy under varying noise intensities, as shown in Figure 1. We first measure the Knowledge Discrepancy of the Curriculum (KDC) based on PSNR at the current timestep. Then our approach dynamically adjusts the learning targets to construct a hard curriculum with reasonable knowledge discrepancy. To ensure high-quality teacher outputs, we efficiently adopt a multi-step iterative generation strategy.

In summary, we propose the Curriculum Consistency Model (CCM) to perform the consistency distillation for the diffusion models and flow matching models. Our main contributions are as follows:

- We identify the instability in knowledge discrepancy during consistency distillation, which significantly impacts text-to-image alignment and the generation of semantic structures in the diffusion process.
- We assess curriculum difficulty based on PSNR and design a more effective adaptive noise schedule to maintain curriculum consistency across different training samples.
- Our method achieves high-quality few-step generation. Specifically, we obtain one-step sampling Fréchet Inception Distance (FID) scores of 1.64 on CIFAR-10 and 2.18 on ImageNet 64x64.
- CCM generalizes well and has been extended to both large-scale diffusion models (Stable Diffusion XL [28]) and flow matching models (Stable Diffusion 3 [5]) for high-resolution image generation. Our results show that the introduction of curriculum consistency leads to lower FID, higher CLIP scores, and significantly improved image-text alignment and semantic structure in the generated images.

2. Related Works

Diffusion Models (DM). Diffusion models have become a leading approach in high-fidelity image generation [32], [10]. This type of model relies on Stochastic Differential Equations (SDEs) to find trajectories from noise to data. Recent work focuses on improving sample quality [9], optimizing density estimation [37], and accelerating the sampling process [38], [27]. Some studies explore the underlying mechanisms and design space of DMs [13], while others scale up DMs for text-conditioned image synthesis [28] or improve sampling efficiency through methods in the latent space[35].

Flow Matching (FM). Flow matching models learn a vector field that generates an Ordinary Differential Equation (ODE) for a desired trajectory from noise to data, without requiring computationally intensive simulations [19]. This flexibility has led to various efforts to improve trajectory properties, particularly straightness, which enables efficient simulation with fewer steps. Methods like rectified flow

[22], [21], multi-sample FM [29], and minibatch OT-CFM [39] aim to straighten trajectories, but the computation costs and sample efficiency are still unsatisfied.

Consistency Models (CM). Consistency models [38] represent a new family of generative models that ensures all points along the ODE trajectory converge to the same solution, often surpassing diffusion models in performance and significantly improving the sample efficiency. Consistency Trajectory Model (CTM) [15] introduces trajectory consistency and further allows unlimited traversal along the PF-ODE between arbitrary starting and ending points during the diffusion process, offering a flexible framework. Latent diffusion models (LCM) [26] employ consistency distillation in the latent space and extend the models to high-resolution text-to-image synthesis. Phased Consistency Model (PCM) [40] identifies key limitations in LCM and addresses them by phasing the ODE trajectory and enforcing the self-consistency property on each sub-trajectory. iCT [36] improves the training of CM by removing the EMA of the teacher, adopting Pseudo-Huber loss, adjusting the discretization and noise schedule, etc. Inspired by iCT, ECM [6] studies discretization interval deeply and proposes adaptive scaling discretization interval and continuous time scheduling schemes. Methods like LCM and PCM employ consistency in different contexts, but don't address the knowledge discrepancy problem identified in this work. iCT and ECM focus on reducing distillation steps to address error accumulation, whereas CCM uniquely approaches the problem by maintaining consistent knowledge discrepancy across timesteps.

3. Method

3.1. Preliminaries

Consistency models [38] aim to simplify multiple function evaluations by directly learning an Ordinary Differential Equation (ODE) that maps any point x on the ODE trajectory to the same output at the endpoint. Specifically, suppose that 0 means noise and 1 means image, the objective of consistency distillation is to align the neural mapping $\boldsymbol{f}_\theta$ with the true mapping $\boldsymbol{f}$ by ensuring $\boldsymbol{f}_\theta(\boldsymbol{x}_t, t, 1) \approx \boldsymbol{f}(\boldsymbol{x}_t, t, 1), \forall t \in [0, 1)$. We can train $\boldsymbol{f}_\theta$ by comparing it with the numerical solution of the pre-trained ODE solver.

$$\boldsymbol{f}_\theta(\boldsymbol{x}_t, t, 1) \approx \mathrm{Solver}(\boldsymbol{x}_t, t, 1; \phi) \approx \boldsymbol{f}(\boldsymbol{x}_t, t, 1) \tag{1}$$

where ϕ means a perfect teacher model. To simplify the training process, local consistency [15] is often performed and formulated in Eq. 2, which compares the student's prediction with the result obtained by solving the ODE over the interval (t, u) using the teacher model, followed by mapping to timestep 1:

$$\boldsymbol{f}_\theta(\boldsymbol{x}_t, t, 1) \approx \boldsymbol{f}_{\theta^-}(\mathrm{Solver}(\boldsymbol{x}_t, t, u; \phi), u, 1) \tag{2}$$

where u is randomly sampled from $(t, 1)$, and θ^- denotes the exponential moving average (EMA) of the parameters, $\theta^- \leftarrow \mathit{stopgrad}(\mu\theta^- + (1-\mu)\theta)$. Local consistency ensures that the student model effectively distills information from the teacher model over the interval (t, u). After training, the generation process begins by sampling $\boldsymbol{x}_0 \sim \mathcal{N}(0, I)$, and then directly obtaining $\boldsymbol{x}_1$ through $\boldsymbol{f}_\theta(\boldsymbol{x}_0, 0, 1)$.

Consistency Distillation in Diffusion Models. In diffusion models, the inverse of the diffusion process can be represented by a deterministic ODE which is given by[37] :

$$\mathrm{d}\boldsymbol{x} = \left[-\tfrac{1}{2}\boldsymbol{\beta}_\sigma \boldsymbol{x}_\sigma - \tfrac{1}{2}\boldsymbol{\beta}_\sigma \mathbf{s}_\theta(\boldsymbol{x}_\sigma, \sigma)\right] \mathrm{d}\sigma \tag{3}$$

where $\sigma \in [\epsilon, T]$ means noise-to-signal ratio and ϵ is a small positive value to ensure numerical stability, $\boldsymbol{\beta}$ is variance and $\mathbf{s}_\theta$ is score function. Note that the noise-to-signal ratio can be transfered into timestep through $\sigma = \frac{1-t}{t}$, so the neural mapping $\boldsymbol{f}_\theta$ in diffusion models can be described by σ: $\boldsymbol{f}_\theta(\boldsymbol{x}_\sigma, \sigma, \epsilon) \approx \boldsymbol{x}_\epsilon$.

A practical solution is to enforce consistency between two adjacent points (timesteps) on the ODE trajectory. By discretizing the interval $[\epsilon, T]$ into N steps, $\sigma_i = \left(\epsilon^{1/\rho} + \frac{i-1}{N-1}(T^{1/\rho} - \epsilon^{1/\rho})\right)^\rho$ [13], we can approximate $\hat{\boldsymbol{x}}_\phi(\sigma_n)$ using Euler's method, and the resulting loss function is:

$$\begin{split}\mathcal{L}_{\mathrm{CD}}^N(\theta, \theta^-; \phi) = \mathbb{E}_{n \sim \mathcal{U}[1, N-1]}\Big[\boldsymbol{\lambda}(\sigma_n) d\left(\boldsymbol{f}_\theta(\boldsymbol{x}_{\sigma_{n+1}}, \sigma_{n+1}, \epsilon),\right. \\ \left.\boldsymbol{f}_{\theta^-}(\hat{\boldsymbol{x}}_{\phi, \sigma_n}, \sigma_n, \epsilon)\right)\Big]\end{split} \tag{4}$$

where $\lambda(\sigma_n) = 1$ and $d(\cdot, \cdot)$ is a distance metrics.

Consistency Distillation in Flow Matching. Continuous Normalizing Flow (CNF) $\boldsymbol{\psi}_t(\boldsymbol{x})$ transforms a probability density from $\boldsymbol{p}_0$ to $\boldsymbol{p}_1$ [2], which is a time-dependent diffeomorphic map induced by vector field $\boldsymbol{u}_t(x)$, can be derived using the ODE:

$$\mathrm{d}\boldsymbol{\psi}_t(\boldsymbol{x}) = \boldsymbol{u}_t(\boldsymbol{\psi}_t(\boldsymbol{x}))\mathrm{d}t, \quad \boldsymbol{\psi}_0(\boldsymbol{x}_0) = \boldsymbol{x}_0 \tag{5}$$

Conditional Flow Matching (CFM) [19] is a simplified simulation-free framework for training CNFs by regressing onto a target vector field $\boldsymbol{u}_t(\boldsymbol{x})$. A specific choice of the ODE trajectory is the optimal transport displacement interpolant and the corresponding trajectory points $\boldsymbol{x}_t = \boldsymbol{\psi}_t(\boldsymbol{x}_0|\boldsymbol{x}_1) = (1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1$. Then we can implement consistency distillation based on Eq. 2. Specific consistency distillation in flow matching has not been extensively studied, which has also been deeply explored in this paper.

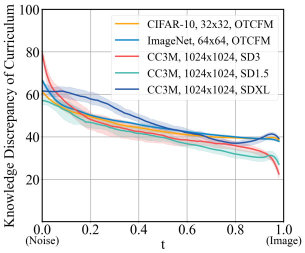

Figure 2. Knowledge Discrepancy Investigation: Analysis of the KDC over (t, u) across different timesteps on various datasets for both flow matching models and diffusion models.

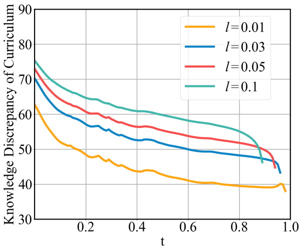

Figure 3. The relationship of KDC with different distillation steps l. In iCT [36] and ECM [6], reducing l during training iterations leads to smaller KDC, resulting in inefficient learning.

3.2. Problem Analysis

In generative models based on denoising, the varying levels of noise in the input can lead to different signal-to-noise ratios (SNR) during the denoising process, as discussed in [7, 13]. Consequently, at different training timesteps, the difficulty that generative models learn varies, which in turn affects the model's convergence rate and the quality of the generated results. The core of the knowledge discrepancy lies in the magnitude of the difference between the model's predicted results and the ground truth. Inspired by this phenomenon, we conducted an in-depth examination of the knowledge discrepancy during the consistency model learning process by comparing the outputs of the student model with those of the teacher model.

In this article, we regard the distillation information over the interval (t, u) as a curriculum and propose a metric based on the Peak Signal-to-Noise Ratio (PSNR) to access knowledge discrepancy of the curriculum, as PSNR is widely used to measure the difference between a denoised image and its original counterpart. Specifically, according to Eq. 2, given the outputs of the student model, $\boldsymbol{x}_{\text{est}} = \boldsymbol{f}_\theta(\boldsymbol{x}_t, t, 1)$, and those of the teacher model, $\boldsymbol{x}_{\text{target}} = \boldsymbol{f}_{\theta^-}(\text{Solver}(\boldsymbol{x}_t, t, u; \phi), u, 1)$, Knowledge Discrepancy of the Curriculum (KDC) over the interval (t, u) is defined as KDC_t^u and calculated using the following formula:

$$\text{KDC}_t^u = 100 - \text{PSNR}(\boldsymbol{x}_{\text{est}}, \boldsymbol{x}_{\text{target}}) = 100- \\ 10\log_{10}(\frac{(2^n-1)^2}{\text{MSE}(\boldsymbol{f}_\theta(\boldsymbol{x}_t, t, 1), \boldsymbol{f}_{\theta^-}(\text{Solver}(\boldsymbol{x}_t, t, u; \phi), u, 1)}) \tag{6}$$

n represents the bit depth of the image. A large KDC means large difference between $\boldsymbol{x}_{\text{est}}$ and $\boldsymbol{x}_{\text{target}}$, and vice versa.

We conducted measurements on both diffusion models (SD 1.5 [32], SDXL [28]) and flow matching models (SD3 [5], OTCFM [39]) and select 3 classic datasets (CIFAR-10, ImageNet, and CC3M) covering both low and high resolutions (32x32, 64x64, and 1024x1024) to ensure reliability and robustness. The mean and variance of KDC between the student and teacher model outputs on t are shown in Figure 2. KDC shows similar trends and close values across different datasets and models, demonstrating that it is a stable and intuitive indicator for measuring knowledge discrepancy during consistency distillation. We observe that the KDC value consistently decreases as t progresses from 0 to 1, indicating a gradual reduction in the knowledge discrepancy of curriculums. This aligns with our intuition: when t is near 0, the KDC is typically around 60, as the input is heavily mixed with noise, leading to a large knowledge discrepancy. At this stage, the model is prone to confusion, causing instability and slow convergence. Conversely, when t approaches 1, the KDC is usually less than 40, indicating that the knowledge discrepancy is too small, resulting in reduced learning efficiency. We argue that this instability and inefficiency hinder the overall learning process of the CM.

We further explored the effect of distillation step $l = u - t$ in CM, and the results are presented in Figure 3. It can be observed that KDC decreases as l decreases. Consequently, in iCT [36] and ECM [6], where l reduces over training iterations, the progressively smaller differences between student and teacher model outputs are more prone to cause inefficient learning. Especially, inefficient learning due to small differences becomes even more significant in consistency distillation, as a pre-trained teacher model is easier to follow than ground truth.

Can we mitigate this imbalance in knowledge discrepancy to enhance the effectiveness of CM learning? In this paper, we attempt to present a feasible solution by proposing an adaptive method named the Curriculum Consistency Model (CCM) which will be elaborated in the following section.

3.3. Curriculum Consistency Model

Our goal is to design an algorithm that ensures a stable and balanced knowledge discrepancy for the model at different timesteps (i.e., under different noise intensities) and various training iterations. To achieve this, we should **see further when clear**, thus, we propose the Curriculum Consistency Model (CCM). CCM incorporates three key designs, which are 1. A reliable metric Knowledge Discrepancy of the Curriculum (KDC) for measuring the difference between student and teacher over the interval (t, u), 2. Dynamic adjustment of learning objectives based on the KDC, and 3. Multi-step iterative generation to ensure the quality of learning objectives.

Measuring the knowledge discrepancy. We propose KDC based on PSNR to measure the knowledge discrepancy in Eq. 6. We have analyzed and shown the stability and generalizability of KDC across different datasets, different timesteps, and different training iterations in Section 3.

Dynamic adjustment of learning objectives. To maintain the consistency of knowledge discrepancy across different timesteps and training iterations, we change the learning objective $\boldsymbol{x}_{\text{target}}$ to $\boldsymbol{x}_{\text{target}}^{\text{KDC}}$. At each timestep, we cycle between estimating the knowledge discrepancy and modifying u until the knowledge discrepancy exceeds a certain fixed value. At different values of t and during various training iterations, we may obtain different values of u, showing the adaptive nature of CCM. Dynamic adjustment becomes effective at larger timesteps during the early stages of training, and extends across all timesteps in the later stages as the model progresses. Limited knowledge discrepancy results in a larger distillation step $l = u - t$ and allows the student to step further, avoiding cumulative errors from many small-step distillations and achieving improved image details, image-text alignment, and semantic structure.

Multi-step iterative generation. Since the teacher model ϕ will remain the same in the training process, the pivotal issue for generating the learning objective at timestep t: $\boldsymbol{x}_{\text{target}} = \boldsymbol{f}_{\theta^-}(\text{Solver}(\boldsymbol{x}_t, t, u; \phi), u, 1)$ is to determine $\boldsymbol{x}_u = \text{Solver}(\boldsymbol{x}_t, t, u; \phi)$. There are various methods to compute $\boldsymbol{x}_u$ and a straightforward approach is to estimate $\boldsymbol{x}_u$ directly from $\boldsymbol{x}_t$ through one-step iteration without regard for the magnitude of the distillation step $l = u - t$. However, CCM may select a u that is significantly greater than t to ensure a stable knowledge discrepancy, which could lead to the teacher model making inaccurate predictions due to a large timestep size s. Consequently, this may result in the student model learning targets that are vague or inaccurate. Therefore, we propose a multi-step iterative generation method where the teacher model will iterate one small timestep size s forward each time until the estimated knowledge discrepancy meets the requirements, which are currently unknown. As shown in Figure 4, in contrast to iCT [36], CCM will increase the l as training progresses. Unlike the multi-step sampling in Scott [20], where a large distillation step is subdivided and the relative positions of u and t remain fixed to reduce cumulative error, CCM determines u by iterating forward from t. CCM allows the relative positions of u and t to vary dynamically across different timesteps and training iterations, ensuring the consistency of KDC. For clarity, we have written the CCM algorithm's procedure in pseudocode and presented it in Algo1.

3.4. Unified Distillation Loss of CCM

CCM focuses on addressing general issues in CM, thus making it applicable to a variety of common denoising-based generative models, including diffusion models and flow matching models. Suppose the ODE is defined on the time interval [0, 1) with 0 and 1 corresponding to noise and ground truth respectively, we can express the consistency distillation loss of CCM in a general form:

Algorithm 1 KDC-Adjusted Target Computation

```
Input: noisy input $\boldsymbol{x}_t$, timestep size $s$, threshold $T_{\text{KDC}}$, teacher model $\phi$, target model $\boldsymbol{f}_{\theta^-}$, student model $\boldsymbol{f}_\theta$
Output: KDC-Adjusted target $\boldsymbol{x}_{\text{target}}^{\text{KDC}}$
Sample $t \sim \mathcal{U}(0, 1)$
Calculate $\boldsymbol{x}_{\text{est}} = \boldsymbol{f}_\theta(\boldsymbol{x}_t, t, 1)$
repeat
    Update $u \leftarrow \min(t + s, 1)$
    Calculate $\boldsymbol{x}_u = \text{Solver}(\boldsymbol{x}_t, t, u; \phi)$
    Compute $\boldsymbol{x}_{\text{target}}^{\text{KDC}} = \boldsymbol{f}_{\theta^-}(\boldsymbol{x}_u, u, 1)$
    Compute $\text{KDC}_t^u = 100 - \text{PSNR}(\boldsymbol{x}_{\text{est}}, \boldsymbol{x}_{\text{target}}^{\text{KDC}})$
    Update $t \leftarrow u$, $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_u$
until $T_{\text{KDC}} < \text{KDC}_t^u$ or $u == 1$
```

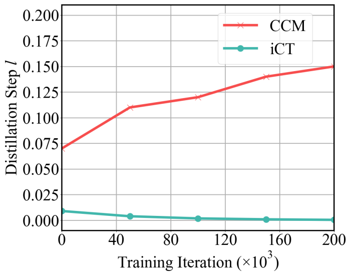

Figure 4. Distillation step vs. training iterations in CCM and iCT.

$$\mathcal{L}_{\text{CCM}}(\theta; \phi) := \mathbb{E}_{t \in [0,1)} \mathbb{E}_{u \in (t,1]} \mathbb{E}_{\boldsymbol{x}_1} \mathbb{E}_{\boldsymbol{x}_t | \boldsymbol{x}_1} [d(\boldsymbol{f}_\theta(\boldsymbol{x}_t, t, 1), \boldsymbol{x}_{\text{target}}^{\text{KDC}}(u, 1))]. \tag{7}$$

where t and u are two timesteps of different noise intensities, $d(\cdot, \cdot)$ is a distance metric which can be L1, L2 or LPIPS. The difference between $\mathcal{L}_{\text{CCM}}$ and standard consistency distillation loss is that the learning target $\boldsymbol{x}_{\text{target}}^{\text{KDC}}(u, 1)$ is obtained through a multi-step iteration according to Algo 1.

CCM with diffusion models. In diffusion models, it is customary to describe the denoising process using noise-to-signal ratio $\sigma \in [\epsilon, T]$, which can be transformed to timestep in Eq. 7 through $t = \frac{1}{\sigma+1}$. The interval $[\epsilon, T]$ will be discretized firstly and standard consistency distillation loss can be calculated based on σ_n and adjacent σ_{n+1} as shown in Eq 4. CCM tends to calculate loss based on σ_n and σ_{n+m}, where m is the number of iteration steps according to Algo 1.

CCM with flow matching models. In flow matching models, a direct approach is to transform the noise-to-signal ratio σ into discrete timesteps t for consistency distillation, where t becomes discrete within the range $0 < t =$

$\frac{1}{\sigma+1} < 1$. We adopt an approach starting from vanilla flow matching, where t is chosen uniformly within $[0, 1)$ [19]. This approach leverages t as a continuous variable, allowing consistency distillation to span a broader range of the ODE trajectory compared to discretized methods in diffusion models. Moreover, distillation at $t = 0$ aligns with inference since generation begins from pure noise. Recent work in [24] also explores continuous-time consistency models. However, the selection of u remains an open question. CCM offers a straightforward method to determine u through adaptive iteration using a base timestep size s. In the following sections, we discuss the choice of s, u, and extra computational cost due to multi-step iterations.

3.5. Adversarial Losses

In generative modeling, student models derived from distillation often produce lower-quality samples compared to their teacher models, as they rely solely on distillation losses. To improve the student's performance and potentially surpass the teacher in quality, we incorporate adversarial training into our framework. Previous work, such as [4] and [15], has demonstrated that combining reconstruction and adversarial losses significantly enhances image generation quality.

Our Curriculum Consistency Model (CCM) framework integrates both KDC-adjusted distillation loss and adversarial losses into a unified training objective:

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(\theta, \eta) =&\mathbb{E}_{\boldsymbol{x}_1}(\log \boldsymbol{d}_\eta(\boldsymbol{x}_1) + \\ &\mathbb{E}_{t\in[0,1)}\mathbb{E}_{\boldsymbol{x}_1}\mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_1}[\log(1 - \boldsymbol{d}_\eta(\boldsymbol{x}_{\text{est}}(\boldsymbol{x}_t, t, 1))]\end{aligned} \tag{8}$$

$$\min_\theta \max_\eta \mathcal{L}(\theta, \eta) = \mathcal{L}_{\text{CCM}}(\theta; \phi) + \boldsymbol{\lambda}_{\text{GAN}}\mathcal{L}_{\text{GAN}}(\theta, \eta) \tag{9}$$

where $\boldsymbol{d}_\eta$ represents the discriminator network and $\boldsymbol{\lambda}_{\text{GAN}}$ is an adaptive weighting. Details are in [15].

4. Experiments

To verify the reliability and generalization of the method, our experiments cover classical datasets with different resolutions, and studies are carried out on diffusion models and flow matching models.

4.1. Experimental Details

Datasets. For low-resolution image generation, we train and evaluate models on CIFAR-10 [16] and ImageNet 64x64 [3] datasets. For high-resolution image generation, we train LoRA weights [11] on the CC3M [1] dataset and evaluate on COCO-2014 [18] with 30K split.

Models. We verify the image generation based on both flow matching and diffusion models, including Optimal Transport Conditional Flow Matching (OT-CFM) [39], Stable Diffusion 3 [5], and Stable Diffusion XL [28]. Our code implementation is based on torchcfm and phased consistency model [40].

Evaluation Metrics. We report the FID [8] and CLIP Score [30] of the generated images and the validation 30K-sample splits. We also comprehensively evaluate the compositionality of CCM on T2I-CompBench [12].

Our experimental parameters are shown in the Appendix.

4.2. Experimental Results and Analysis

(a) Performance comparisons on CIFAR-10

Model Type	**Method**	**Params**	**NFE** (↓)	**FID** (↓)
GAN	StyleGAN-XL([34])	>100M	1	1.85
Diffusion Models	DDPM([9])	-	1000	3.17
	DDIM([35])	-	100	4.16
	Score SDE([37])	-	2000	2.20
	EDM([13])	56.4M	35	2.01
	2-Rectified Flow([23])	-	1	4.85
	DMD([42])	-	1	3.77
	ECM([6])	55.7M	1	3.60
	CD([38])	56.4M	1	3.55
	iCT([36])	56.4M	1	2.83
	CCM w/o GAN	56.4M	1	2.79
	CD w/ GAN([25])	56.4M	1	2.65
	CTM([15])	-	1	1.98
	SiD([43])	-	1	1.92
Flow Matching Models	OT-CFM([39])	34.09M	100	4.49
	PCM([40])	-	8	1.94
	CD(retrained)	34.09M	1	10.5
	CCM w/o GAN	34.09M	1	7.42
	CD w/ GAN(retrained)	34.09M	1	4.08
	CCM(ours)	34.09M	1	**1.64**

(b) Performance comparisons on ImageNet 64×64

Model Type	**Method**	**NFE** (↓)	**FID** (↓)
Diffusion Models	EDM([13])	79	2.44
	CD([38])	1	6.20
	ECM([6])	1	4.05
	iCT([36])	1	4.02
	Moment Matching([33])	1	3.00
	CTM([15])	1	1.92
	SiD([43])	1	1.52
	DMD2([41])	1	1.28
	PaGoDA([14])	1	**1.21**
Flow Matching Models	OT-CFM(retrained)	100	5.36
	CCM(ours)	1	**2.18**

(c) Performance comparisons on COCO2014-30K

Base Model	**Method**	**CLIP Score** (↑)	**FID** (↓)
SD3	Original	26.18	86.84
	LCM([26])	31.27	25.44
	PCM([40])	31.06	28.52
	CCM w/o GAN	31.32	21.83
	CCM(ours)	**31.41**	**21.49**
SDXL	Hyper-SD([31])	31.30	30.87
	PCM([40])	31.63	21.15
	CCM w/o GAN	31.63	21.08
	CCM(ours)	**31.73**	**20.47**

Table 1. Performance comparisons on different datasets.

According to Table 1, we conduct a performance analysis of CCM compared to existing approaches. On the CIFAR-10 dataset, CCM achieves an impressive unconditional FID of 1.64 with only one function evaluation

(NFE=1), indicating that CCM not only surpasses these methods in sampling efficiency but also achieves superior image quality. **For flow matching models**, CCM w/o GAN outperforms CD, CCM significantly supasses CD w/ GAN and even exceeds GAN. **For diffusion models**, CCM w/o GAN outperforms CD, ECM and iCT. Therefore, vanilla CCM is also effective. Due to limitations in the teacher model and model capacity, flow matching models underperform diffusion models. On the ImageNet 64×64 dataset, CCM also performed strongly: CCM's FID (NFE=1) reaches 2.18 on conditional generation, which is also competitive with the mainstream generated models. Although the performance of a student model heavily depends on its teacher, CCM (2.18) demonstrates a more substantial improvement over its teacher model, OT-CFM (5.36), than CTM (1.92) does over its teacher model, EDM (2.44). We have also included some non-CM-based state-of-the-art distillation methods (Moment Matching, SiD, DMD2 and PaGoDA) for a more complete comparison. The samples generated by CCM (NFE=1) are shown in Figure 5 and are comparable in quality to those generated by OT-CFM (NFE=100), indicating that CCM shows excellent acceleration with at least 50x faster in inference.

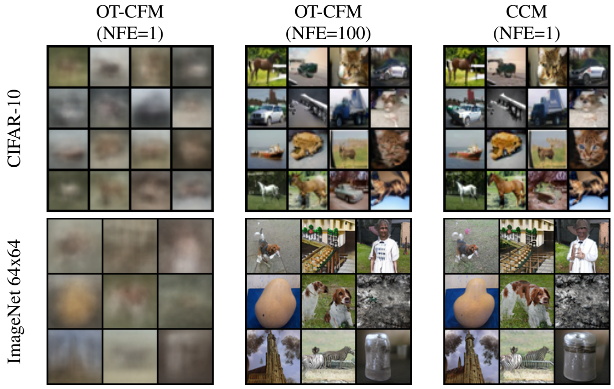

Figure 5. Samples generated by OT-CFM and CCM on CIFAR-10 and ImageNet 64x64.

When scaled to large-scale methods and high resolution, CCM can still maintain advantages. According to Table 1(c), CCM has achieved lower FID and higher CLIP scores on both diffusion models and flow matching models. On T2I-Compbench [12], CCM-4Step outperforms both LCM and PCM across all six metrics, achieving results comparable to SD3-28Step. Additionally, CCM based on SDXL performs well in color, texture, non-spatial, and complex attributes. We compare the samples generated by different methods and find that CCM performs better image-text alignment (Figure. 6) and semantic structure (Figure. 7). Further, we conduct a user study and Figure 8 affirms the good performance of CCM. These results demonstrate the strong generalization capabilities of CCM.

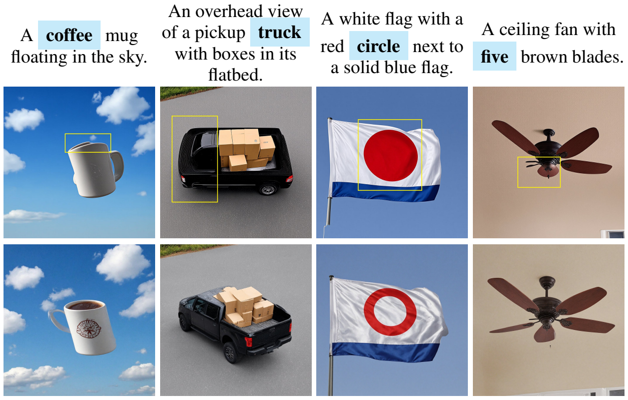
A coffee mug floating in the sky.
An overhead view of a pickup truck with boxes in its flatbed.
A white flag with a red circle next to a solid blue flag.
A ceiling fan with five brown blades.

Figure 6. Semantic comparison of images generated by LCM (up) and CCM (down). CCM shows better image-text alignment and generates images that better fit the text.

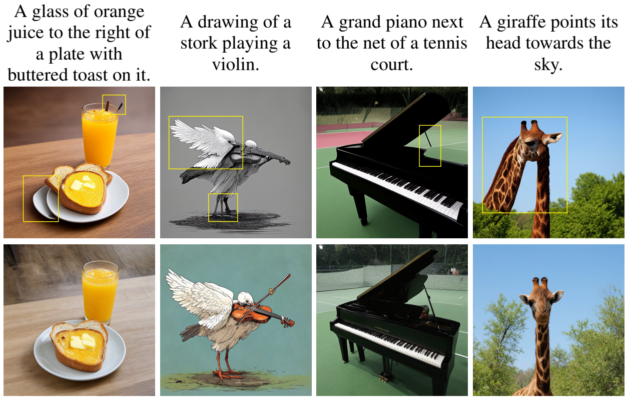
A glass of orange juice to the right of a plate with buttered toast on it.
A drawing of a stork playing a violin.
A grand piano next to the net of a tennis court.
A giraffe points its head towards the sky.

Figure 7. Structure comparison of images generated by LCM (up) and CCM (down). Both models correctly understand the text, but the structures generated by CCM are more reasonable.

4.3. Ablation Studies

We perform thorough ablation studies to evaluate the impact of different modules in the method. All ablation experiments are based on CIFAR-10 without adversarial losses and are performed over 100K iterations.

Static vs. Dynamic. We first compared different target selection strategies to study the effect of the distillation step $l = u - t$, numbers of iterative steps n, and timestep sizes s. The three key variables have the following relation $l = \sum_{i=1}^{n} s_i$. Strategies fall into two categories: static strategies that l, s, n are fixed and dynamic strategies that at least one variable in l, s, n varies with t. From Table 3, we can observe that CCM surpasses all other strategies. Moreover, when n increases from 1 to 3 with fixed $s = 0.03$, the model's performance improves. Similarly, increasing the distillation step $l = u - t$ also exhibits a similar phenomenon, but a larger value of l with fewer iterative steps n can be detrimental($l = 0.1, n = 1$). Furthermore, we ex-

Base Model	Method	Attribute Binding			Object Relationship		Complex (↑)
		Color (↑)	Shape (↑)	Texture (↑)	Spatial (↑)	Non-Spatial (↑)	
SD3	Original	0.813	0.590	0.759	0.343	0.311	0.479
	LCM ([26])	0.705	0.482	0.587	0.225	0.309	0.346
	PCM ([40])	0.702	0.480	0.599	0.212	0.305	0.346
	CCM(ours)	**0.733**	**0.493**	**0.633**	**0.245**	**0.310**	**0.358**
SDXL	Original	0.587	0.468	0.529	0.213	0.311	0.323
	LCM ([26])	0.604	0.407	0.497	0.172	0.310	0.337
	PCM ([40])	0.606	0.420	0.497	0.202	0.311	0.332
	Lightning ([17])	0.581	**0.437**	0.499	**0.221**	0.311	0.325
	CCM(ours)	**0.614**	0.427	**0.511**	0.207	**0.312**	**0.338**

Table 2. Quantitative Results on T2I-CompBench. CCM provides consistent improvements in all categories for both SD3 and SDXL. Blue means the reference results from the original models (28 steps for SD3 and 40 steps for SDXL). Other models use 4 inference steps.

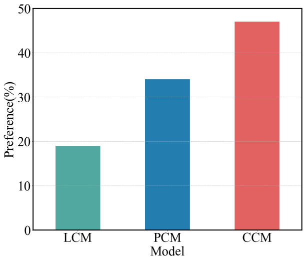

Figure 8. User study. Subjects were shown generated images and asked for preference.

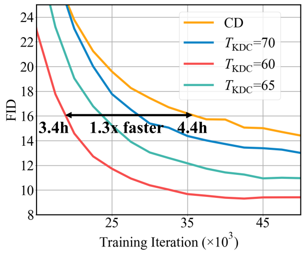

Figure 9. Comparisons of different T_{KDC}.

perimented with varying the timestep size s in accordance with the changes in t. Increasing l proportionally as t increases is not a good choice since it is almost impossible to learn when both t and timestep size s are very small, which also reminds us to balance knowledge discrepancy and model ability. A special case of the opposite is learning ground truth directly, i.e., $l = s = 1 - t$, which also lags behind CCM. Last, I-CCM, which uses an opposite strategy to CCM, not only performs worse than CCM but is also inferior to some static methods.

Strategy	l	n	s	**FID** (↓)
Static	0.01	1	0.01	14.06
	0.03	1	0.03	11.38
	0.1	1	0.1	16.2
	0.06	2	0.03	10.15
	0.09	3	0.03	9.89
Dynamic	$0.1t$	1	0.1t	27.19
	I-CCM	-	0.03	12.66
	$1 - t$	1	1-t	10.67
	CCM	-	0.03	**9.32**

Table 3. Comparison between static and dynamic strategies. For CCM, $T_{\mathrm{KDC}} = 60$. I-CCM adopts the opposite strategy of CCM.

Method	s	**FID** (↓)
Single-step	-	46.82
Multi-steps	0.01	9.96
	0.03	**9.32**
	0.05	9.78

Table 4. Comparisons among strategies of determining $\boldsymbol{x}_{\mathrm{target}}$.

Strategies of determining $\boldsymbol{x}_{\mathrm{target}}$. We tested various methods for determining $\boldsymbol{x}_{\mathrm{target}}$, including single-step iteration and multiple-steps with different timestep sizes s in Table 4. The effect of directly generating $\boldsymbol{x}_u$ from $\boldsymbol{x}_t$ is poor compared to the effect of multi-step generation. This may be because the quality of the directly generated $\boldsymbol{x}_u$ is relatively low, which affects the effectiveness of CM learning. We also found that after using CCM, the model is no longer sensitive to timestep sizes, with $s = 0.03$ slightly outperforming other choices.

The choice of T_{KDC}. Different T_{KDC} determine the dynamically selected number of iterative steps during the training process, which is a hyperparameter in the methods presented in this paper. We conducted experiments with different values of T_{KDC}, as shown in Figure 9. It can be observed that within the range of 60-70, the FID results are better than CD, indicating that our method is not very sensitive to T_{KDC}. Moreover, although CCM will lead to an increase in the time of a single iteration, the convergence rate is accelerated at the same time. Based on the same FID, CCM achieves 1.3× faster convergence than the vanilla CD and achieves a lower FID, bringing significant benefits.

5. Conclusion

In this article, we introduce the knowledge discrepancy to measure the difficulty in the CM learning process, and have discovered that the distribution of difficulty is highly imbalanced under different noise intensities. To alleviate this issue, we propose Curriculum Consistency Model (CCM), an efficient method for training models based on ODEs. We design an adaptive noise schedule to maintain the consistency of curriculum difficulty and verify the rationality and validity of the design. Our method achieves comparable single-step sampling FID results on CIFAR-10 (1.64) and ImageNet64x64 (2.18). More importantly, our approach works on diffusion models and flow matching models as well and we have successfully extended the proposed method to large-scale models, such as SDXL and SD3.

Acknowledgements: This work is partially supported by the Postdoctoral Fellowship Program (Grade B) of China Postdoctoral Science Foundation Grant GZB20240760.

References

[1] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021. 6

[2] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018. 3

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[4] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 6

[5] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 2, 4, 6

[6] Zhengyang Geng, Ashwini Pokle, William Luo, Justin Lin, and J Zico Kolter. Consistency models made easy. *arXiv preprint arXiv:2406.14548*, 2024. 2, 3, 4, 6

[7] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. *arXiv preprint arXiv:2303.09556*, 2023. 4

[8] 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. *Advances in neural information processing systems*, 30, 2017. 6

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2, 6

[10] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023. 2

[11] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 6

[12] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023. 6, 7

[13] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022. 2, 3, 4, 6

[14] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Yuhta Takida, Naoki Murata, Toshimitsu Uesaka, Yuki Mitsufuji, and Stefano Ermon. Pagoda: Progressive growing of a one-step generator from a low-resolution diffusion teacher. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 6

[15] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *Advances in neural information processing systems*, 2023. 3, 6

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[17] Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024. 8

[18] 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. 6

[19] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. 1, 2, 3, 6

[20] Hongjian Liu, Qingsong Xie, Zhijie Deng, Chen Chen, Shixiang Tang, Fueyang Fu, Zheng-jun Zha, and Haonan Lu. Scott: Accelerating diffusion models with stochastic consistency distillation. *arXiv preprint arXiv:2403.01505*, 2024. 5

[21] Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022. 3

[22] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 3

[23] Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. 6

[24] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024. 6

[25] Haoye Lu, Yiwei Lu, Dihong Jiang, Spencer Ryan Szabados, Sun Sun, and Yaoliang Yu. Cm-gan: Stabilizing gan training with consistency models. In *ICML 2023 Workshop*, 2023. 6

[26] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 1, 3, 6, 8

[27] Hao Phung, Quan Dao, and Anh Tran. Wavelet diffusion models are fast and scalable image generators. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10199–10208, 2023. 2

[28] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, 2024. 2, 4, 6

[29] Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky TQ Chen. Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2304.14772*, 2023. 3

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6

[31] Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis. *arXiv preprint arXiv:2404.13686*, 2024. 6

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 4

[33] Tim Salimans, Thomas Mensink, Jonathan Heek, and Emiel Hoogeboom. Multistep distillation of diffusion models via moment matching. *Advances in Neural Information Processing Systems*, 37:36046–36070, 2024. 6

[34] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 6

[35] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020. 1, 2, 6

[36] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023. 2, 3, 4, 5, 6

[37] 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, 6

[38] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pages 32211–32252. PMLR, 2023. 1, 2, 3, 6

[39] Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2023. 1, 3, 4, 6

[40] Fu-Yun Wang, Zhaoyang Huang, Alexander William Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, Hongsheng Li, and Xiaogang Wang. Phased consistency model, 2024. 3, 6, 8

[41] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in Neural Information Processing Systems*, 37: 47455–47487, 2024. 6

[42] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6613–6623, 2024. 6

[43] Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *Forty-first International Conference on Machine Learning*, 2024. 6