Latent Consistency Models - Synthesizing High-Resolution Images with Few-Step Inference

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, Hang Zhao
arXiv 2023
Tsinghua University

Suk-Hwan Lee

Artificial Intelligence

Creating the Future

Dong-A University

Division of Computer Engineering & Artificial Intelligence

References

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, Hang Zhao
Latent Consistency Models: Synthesizing High-Resolution Images with Few-Step Inference
arXiv:2310.04378

https://latent-consistency-models.github.io/
https://github.com/luosiallen/latent-consistency-model

huggingface
https://huggingface.co/docs/diffusers/api/pipelines/latent_consistency_models
https://huggingface.co/SimianLuo/LCM_Dreamshaper_v7

[Paper Review blog]
https://kimjy99.github.io/논문리뷰/latent-consistency-model/
http://dmqm.korea.ac.kr/activity/seminar/434
Accelerating Diffusion Models: Consistency Models and Hybrid Approach
https://www.youtube.com/watch?v=OT3JWNz0Il8&t=235s

Latent Consistency Models

Abstract (Paper)

- **Latent Diffusion models (LDMs)**
  - ✓ Remarkable results in **synthesizing high-resolution images**.
  - ✓ However, **Iterative sampling process -** Computationally intensive and leads to slow generation.

- Inspired by **Consistency Models** (song et al.),
- Propose **Latent Consistency Models (LCMs)**,
  - ✓ Enabling **swift Inference** with minimal steps on any **pre-trained LDMs**, including **Stable Diffusion** (rombach et al).

- Viewing **the guided reverse diffusion process** as **solving an augmented probability flow ODE (PF-ODE)**,
- LCMs are designed to directly predict the solution of such ODE in latent space, mitigating the need for numerous iterations and allowing rapid, high-fidelity sampling.

- Efficiently **distilled from pre-trained classifier-free guided diffusion models**, a high-quality **768 x 768 2~4-step LCM** takes only **32 A100 GPU hours for training**.
- Furthermore, introduce **Latent Consistency Fine-tuning (LCF)**, a novel method that is tailored for fine-tuning LCMs on customized image datasets.

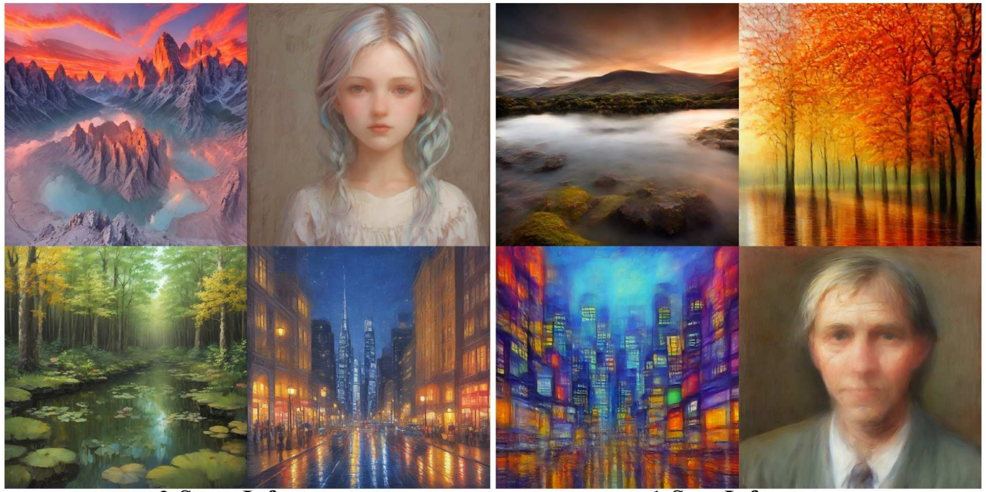
- Evaluation on the LAION-5B-Aesthetics dataset demonstrates that LCMs achieve **state-of-the-art text-to-image generation** performance with **few-step inference**.

- Project Page: https://latent-consistency-models.github.io/

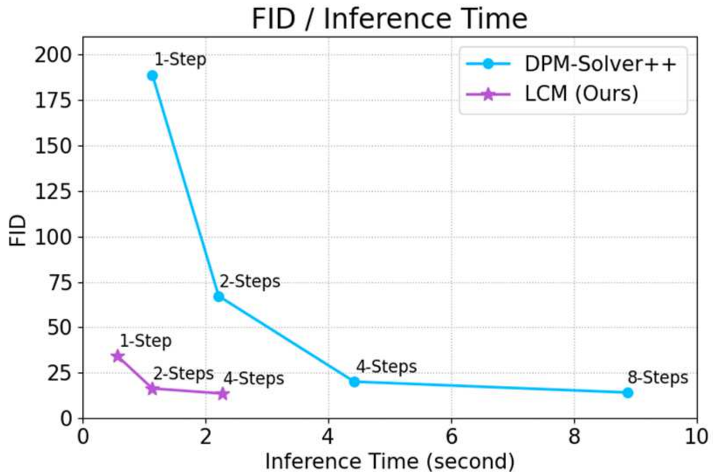
Latent Consistency Models

https://latent-consistency-models.github.io/

4-Steps Inference



2-Steps Inference 1-Step Inference



Few-Step Generated Images :

LCMs can be distilled from any pre-trained Stable Diffusion (SD) in only 4,000 training steps (~32 A100 GPU Hours) for generating high quality 768 x 768 resolution images in 2~4 steps or even one step, significantly accelerating text-to-image generation. We employ LCM to distill the Dreamshaper-V7 version of SD in just 4,000 training iterations.

Latent Consistency Models

1 Introduction & 2. Related Works

➢ Diffusion Models

- Compared to VAEs and GANs, diffusion models enjoy the benefit of **training stability** and **better likelihood estimation**.
- Be trained to **denoise the noise-corrupted data to estimate the score of data distribution**.
- During inference, draw samples by running the reverse diffusion process to gradually denoise the data point.
- Bottlenecked by their **slow generation speed**

- Ho et al., 2020, **DDPM**, Denoising Diffusion Probabilistic Models
- Song et al., 2020a, **DPIM**, Denoising Diffusion Implicit Models
- Nichol & Dhariwal, 2021, **iDDPM**, Improved Denoising Diffusion Probabilistic Models
- Ramesh et al., 2022, **DALL-E2**, Hierarchical Text-Conditional Image Generation with Clip Latents
- Rombach et al., 2022, **Stable Diffusion**, High-Resolution Image Synthesis with Latent Diffusion Models
- Song & Ermon, 2019. Generative Modeling by Estimating Gradients of the Data Distribution

➢ Accelerating DMs

○ Training-free methods such as

- **ODE solvers** : Song et al., 2020a, **DPIM**; Lu et al., 2022a;b, DPM-Solver, **DPM-Solver++**
- **Adaptive step size solvers** : Jolicoeur-Martineau et al., 2021, Gotta go fast when generating data with score-based models.
- **Predictor-corrector methods** : Song et al., 2020b, Score-based generative modeling through stochastic differential equations

○ Training-based approaches include

- **Optimized discretization** : Watson et al., 2021, Learning to efficiently sample from diffusion probabilistic models
- **Truncated diffusion** : Lyu et al., 2022; Zheng et al., 2022,
- **Neural operator** : Zheng et al., 2023, Fast sampling of diffusion models via operator learning
- **Distillation** : Salimans & Ho, 2022, Progressive distillation for fast sampling of diffusion models; Meng et al., 2023, **On distillation of guided diffusion models**

○ More recently, new generative models for faster sampling have also been proposed (Liu et al., 2022; 2023).

1 Introduction & 2. Related Works

➢ **Latent Diffusion Models** (LDMs)

- Synthesizing high-resolution text-to-images.
- Ex) Stable Diffusion (SD) performs forward and reverse diffusion processes in the data latent space, resulting in more efficient computation.

- Rombach et al., 2022, **Stable Diffusion**, High-Resolution Image Synthesis with Latent Diffusion Models

➢ **Consistency Models** (CMs)

- A new type of generative model for **faster sampling** while **preserving generation quality**.
- CMs adopt **consistency mapping** to **directly map any point in ODE trajectory** to its origin, **enabling fast one-step generation**.
- CMs can be **trained by distilling pre-trained diffusion models** or as standalone **generative models**.

- Song et al., 2023, Consistency Models

- CMs **constrained to pixel space image generation tasks**, making it **unsuitable for synthesizing high-resolution images**.
- Moreover, the applications to the **conditional diffusion model** and the incorporation **of classifier-free guidance** have **not been explored**, rendering their methods unsuitable for text-to-image generation synthesis.

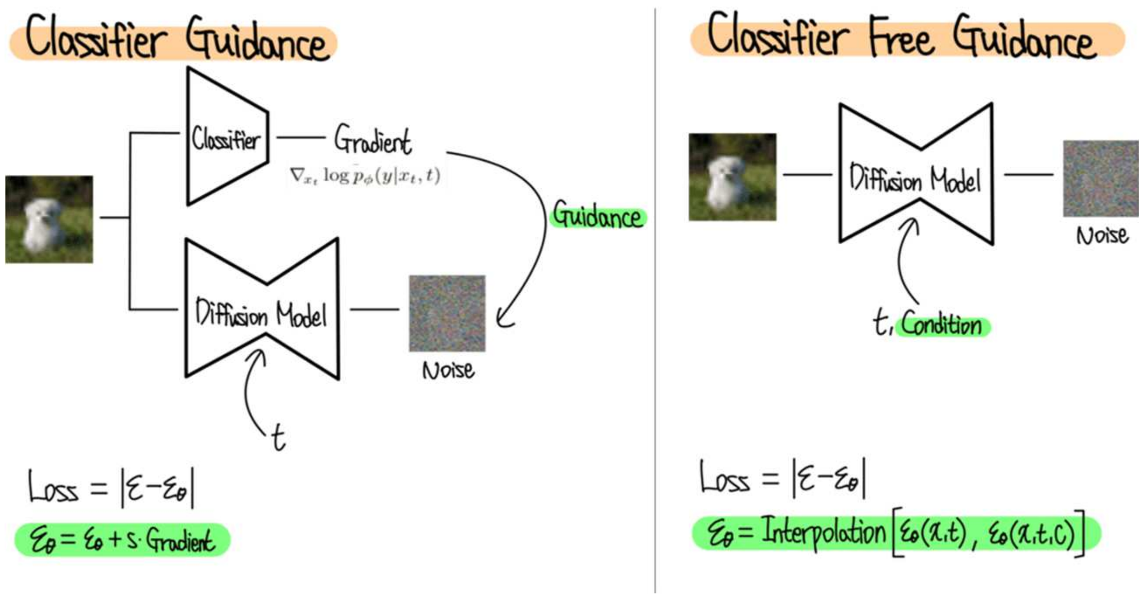
1 Introduction & 2. Related Works

❖ Classifier Free Guidance : Classifier-free diffusion guidance

- Jonathan Ho, Tim Salimans, Classifier-Free Diffusion Guidance, NIPS2021, Google Research, Brain team

- Diffusion Models Beat GANs on Image Synthesis 논문 :
  - ✓ 추가 classifier를 학습하여 샘플의 품질을 향상시키는 classifier guidance가 제안됨. **Classifier guidance는 diffusion model의 score 추정치와 classifier의 로그 확률의 입력 기울기를 혼합함.**
  - ✓ Classifier gradient의 강도를 변경하여 Inception Score (IS)와 FID (또는 precision과 recall)를 절충할 수 있음

- 어떠한 classifier도 사용하지 않는 classifier-free guidance를 제안함
- **Classifier-free guidance**는 이미지 classifier의 기울기 방향으로 샘플링 하는 대신 **conditional diffusion model**과 함께 학습된 **unconditional diffusion model**의 **score** 추정치를 혼합함.
- 혼합 가중치를 사용하여 classifier guidance에서 얻은 것과 유사한 FID/IS tradeoff를 얻는다. 또한 pure generative diffusion model이 다른 유형의 생성 모델과 함께 매우 높은 fidelity의 샘플을 합성하는 것이 가능함



[Source] https://ffighting.net/deep-learning-paper-review/diffusion-model/classifier-free-guidance/

Latent Consistency Models

Main Contributions

- **Latent Consistency Models (LCMs)** for fast, high-resolution image generation.
  - LCMs employ **consistency models** in the **image latent space**, enabling **fast few-step or even one-step high-fidelity sampling** on **pre-trained latent diffusion models** (e.g., Stable Diffusion (SD)).
- Provide a simple and efficient **one-stage guided consistency distillation method**
  - to distill SD for few-step (2~4) or even 1-step sampling
  - to efficiently convert a **pre-trained guided diffusion model** into **a latent consistency model** by solving an **augmented PF-ODE**.
  - Propose the ***SKIPPING-STEP*** technique to further accelerate the convergence.
  - For 2- and 4-step inference, our method costs only 32 A100 GPU hours for training and achieves state-of-the-art performance
- Propose **Latent Consistency Finetuning**, which allows fine-tuning a pre-trained LCM to support few-step inference on customized image datasets while preserving the ability of fast inference.

Latent Consistency Models

3. PRELIMINARIES

❖ Diffusion Models

○ Diffusion models, or score-based generative models

- Progressively inject Gaussian noises into the data, and then generate samples from noise via a reverse denoising process

- Define a **forward process** transitioning the origin data distribution $p_{data}(x)$ to marginal distribution $q_t(\boldsymbol{x}_t)$, via transition kernel:

$$q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) = N(\boldsymbol{x}_t|\alpha(t)\boldsymbol{x}_0, \sigma^2(t)\boldsymbol{I}),$$

- $\alpha(t)$, $\sigma(t)$ specify the noise schedule.

○ In continuous time perspective, the forward process can be described by a **stochastic differential equation (SDE)** for t ∈ [0, T]: (☞ Song et al. (2020b); Lu et al. (2022a); Karras et al. (2022))

$$d\boldsymbol{x}_t = f(t)\boldsymbol{x}_t dt + g(t) d\boldsymbol{w}_t, \qquad \boldsymbol{x}_0 \sim p_{data}(\boldsymbol{x}_0)$$

- $w(t)$ is the standard Brownian motion

$$f(t) = \frac{\mathrm{d}\log\alpha(t)}{\mathrm{d}t}, \quad g^2(t) = \frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t} - 2\frac{\mathrm{d}\log\alpha(t)}{\mathrm{d}t}\sigma^2(t). \tag{1}$$

○ By considering the reverse time SDE,

- the marginal distribution $q_t(\boldsymbol{x})$ satisfies the following ordinary differential equation, called the **Probability Flow ODE (PF-ODE)** (☞ Song et al., 2020b; Lu et al., 2022a):

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = f(t)\boldsymbol{x}_t - \frac{1}{2}g^2(t)\nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}_t), \ \ \boldsymbol{x}_T \sim q_T(\boldsymbol{x}_T). \tag{2}$$

○ In diffusion models, we train the noise prediction model $\epsilon_\theta(\boldsymbol{x}_t, t)$ to fit $-\nabla\log q_t(\boldsymbol{x}_t)$ (called the ***score function***).

- Approximating the score function by the noise prediction model in 21, one can obtain the following **empirical PF-ODE** for sampling:

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = f(t)\boldsymbol{x}_t + \frac{g^2(t)}{2\sigma_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t), \quad \boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \tilde{\sigma}^2\boldsymbol{I}). \tag{3}$$

3. PRELIMINARIES

❖ Diffusion Models

○ For **class-conditioned diffusion models**,

- **Classifier-Free Guidance (CFG)** (Ho & Salimans, 2022) :

✓ An effective technique to significantly improve the quality of generated samples

✓ Has been widely used in several large-scale diffusion models including **GLIDE** Nichol et al. (2021), **Stable Diffusion** (Rombach et al., 2022), **DALL·E 2** (Ramesh et al., 2022) and **Imagen** (Saharia et al., 2022).

- Given a CFG scale ω, the original noise prediction is replaced by a linear combination of conditional and unconditional noise prediction:

$$\tilde{\epsilon}_{\boldsymbol{\theta}}(z_t, \omega, \boldsymbol{c}, t) = (1+\omega)\epsilon_{\boldsymbol{\theta}}(z_t, \boldsymbol{c}, t) - \omega\epsilon_{\boldsymbol{\theta}}(z, \varnothing, t)$$

3. PRELIMINARIES

❖ Consistency Models

○ A new family of generative models

- Enables one-step or few-step generation.
- Core idea : **Learn the consistency function** that **maps any points on a trajectory of the PF-ODE to that trajectory's origin** (i.e., the solution of the PF-ODE)

$$\boldsymbol{f} : (\boldsymbol{x}_t, t) \longmapsto \boldsymbol{x}_\epsilon$$

ϵ is a fixed small positive number.

- The consistency function should satisfy the ***self-consistency property***:

$$\boldsymbol{f}(\boldsymbol{x}_t, t) = \boldsymbol{f}(\boldsymbol{x}_{t'}, t'), \forall t, t' \in [\epsilon, T]. \quad (4)$$

- (Song et al., 2023) Key idea for learning a consistency model $\boldsymbol{f}_\theta$: To learn a consistency function from data by effectively enforcing the self-consistency property
- Ensure that $\boldsymbol{f}_\theta(\boldsymbol{x}, \epsilon) = \boldsymbol{x}$, the consistency model $\boldsymbol{f}_\theta$ is parameterized as:

$$\boldsymbol{f}_\theta(\boldsymbol{x}, t) = c_{\text{skip}}(t)\boldsymbol{x} + c_{\text{out}}(t)\boldsymbol{F}_\theta(\boldsymbol{x}, t), \quad (5)$$

$c_{skip}(t)$ and $c_{out}(t)$: Differentiable functions with $c_{skip}(t, \epsilon) = 1$ and $c_{out}(\epsilon) = 0$

$\boldsymbol{F}_\theta(\boldsymbol{x}, t)$: a deep neural network

- A CM can be either distilled from a pre-trained diffusion model (known as *Consistency Distillation*) or trained from scratch

3. PRELIMINARIES

❖ Consistency Models

- To enforce the self-consistency property, we maintain a target model $\boldsymbol{\theta}^-$, updated with exponential moving average (EMA) of the parameter $\boldsymbol{\theta}$, we intend to learn; $\boldsymbol{\theta}^- \leftarrow \mu\boldsymbol{\theta}^- + (1-\mu)\boldsymbol{\theta}$
- Define the consistency loss

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \Phi) = \mathbb{E}_{\boldsymbol{x},t}\left[d\left(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_{n+1}}, t_{n+1}), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\hat{\boldsymbol{x}}^{\phi}_{t_n}, t_n)\right)\right], \qquad (6)$$

$d(\cdot)$: a chosen metric function for measuring the distance between two samples; $d(x, y) = \|x - y\|^2$

$\hat{x}^{\phi}_{t_n}$: a one-step estimation of x_{t_n} from $x_{t_{n+1}}$

$$\hat{\boldsymbol{x}}^{\phi}_{t_n} \leftarrow \boldsymbol{x}_{t_{n+1}} + (t_n - t_{n+1})\Phi(\boldsymbol{x}_{t_{n+1}}, t_{n+1}; \phi). \qquad (7)$$

- Φ denotes the one-step ODE solver applied to PF-ODE in Eq. 24. (Song et al., 2023) used Euler (Song et al., 2020b) or Heun solver (Karras et al., 2022) as the numerical ODE solver

- Pseudo-code for consistency distillation

Algorithm 2 Consistency Distillation (CD) (Song et al., 2023)

Input: dataset $\mathcal{D}$, initial model parameter $\boldsymbol{\theta}$, learning rate η, ODE solver $\Phi(\cdot,\cdot,\cdot)$, distance metric $d(\cdot,\cdot)$, and EMA rate μ

$\boldsymbol{\theta}^- \leftarrow \boldsymbol{\theta}$
repeat
  Sample $\boldsymbol{x} \sim \mathcal{D}$ and $n \sim \mathcal{U}[1, N-1]$
  Sample $\boldsymbol{x}_{t_{n+1}} \sim \mathcal{N}(\boldsymbol{x}; t^2_{n+1}\mathbf{I})$
  $\hat{\boldsymbol{x}}^{\phi}_{t_n} \leftarrow \boldsymbol{x}_{t_{n+1}} + (t_n - t_{n+1})\Phi(\boldsymbol{x}_{t_{n+1}}, t_{n+1}, \phi)$
  $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \Phi) \leftarrow d(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_{n+1}}, t_{n+1}), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\hat{\boldsymbol{x}}^{\phi}_{t_n}, t_n))$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \Phi)$
  $\boldsymbol{\theta}^- \leftarrow \text{stopgrad}(\mu\boldsymbol{\theta}^- + (1-\mu)\boldsymbol{\theta})$
until convergence

4. LATENT CONSISTENCY MODELS

❖ Summary

Consistency Models (CMs) (Song et al., 2023)

- Only focused on image generation tasks on ImageNet 64×64 (Deng et al., 2009) and LSUN 256×256 (Yu et al., 2015).
- Unexplore to generate higher-resolution text-to-image tasks

Latent Consistency Models (LCMs)

- Adopt a **consistency model** in **the image latent space**, similar to LDMs
- Choose the **Stable Diffusion (SD)** as the underlying diffusion model to **distill** from.
- Aim to achieve **few-step (2~4) and even one-step inference on SD without compromising image quality.**
- The **classifier-free guidance (CFG)** (Ho & Salimans, 2022) is an effective technique to further **improve sample quality** and is widely used in SD.

- Propose a **simple one-stage guided distillation method** in Sec 4.2 that solves an **augmented PF-ODE**, **integrating CFG into LCM** effectively.
- Propose **SKIPPING-STEP** technique to **accelerate the convergence of LCMs** in Sec. 4.3.
- Finally, propose **Latent Consistency Fine-tuning** to finetune a pre-trained LCM for few-step inference on a customized dataset in Sec 4.4.

Latent Consistency Models

4.1 LCM : Consistency Distillation in the Latent Space

- **LDM** : **Stable Diffusion (SD)** - Image Latent Space (Rombach et al., 2022)
  - Utilizing **image latent space** in large-scale diffusion models has effectively **enhanced image generation quality** and **reduced computational load**.
  - SD
    - An autoencoder (E, D) is first trained to compress high-dim image data into low-dim latent vector $z = E(x)$, which is then decoded to reconstruct the image as $x = D(z)$.
    - **Training diffusion models in the latent space** greatly reduces the computation costs compared to pixel-based models and speeds up the inference process;
  - LDMs make it possible to generate high-resolution images on laptop GPUs.

- For **LCMs**, we leverage the advantage of **the latent space for consistency distillation**, **contrasting with the pixel space used in CMs** (Song et al., 2023).
- Termed **Latent Consistency Distillation (LCD)** is applied to **pre-trained SD**, allowing the synthesis of high-resolution (e.g., 768×768) images in 1~4 steps.
  - We focus on **conditional generation**.

4.1 LCM : Consistency Distillation in the Latent Space

➢ Recall that the **PF-ODE** of the reverse diffusion process (Song et al., 2020b; Lu et al., 2022a)

$$\frac{\mathrm{d}z_t}{\mathrm{d}t} = f(t)z_t + \frac{g^2(t)}{2\sigma_t}\epsilon_\theta(z_t, c, t), \quad z_T \sim \mathcal{N}(\mathbf{0}, \tilde{\sigma}^2 \boldsymbol{I}), \tag{8}$$

- z_t are image latents
- $\epsilon_\theta(z_t, c, t)$ is the noise prediction model
- c is the given condition (e.g text)

- Samples can be drawn by solving the PF-ODE from T to 0.

➢ To perform **LCD**, introduce the **consistency function** $\boldsymbol{f_\theta}: (\boldsymbol{z_t}, \boldsymbol{c}, \boldsymbol{t}) \mapsto \boldsymbol{z_0}$ to **directly predict the solution of PF-ODE** (Eq. 8) **for $t = 0$.**

- We **parameterize $\boldsymbol{f_\theta}$** by **the noise prediction model $\hat{\epsilon}_\theta$**,

$$\boldsymbol{f}_\theta(\boldsymbol{z}, \boldsymbol{c}, t) = c_{\text{skip}}(t)\boldsymbol{z} + c_{\text{out}}(t)\left(\frac{\boldsymbol{z} - \sigma_t \hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{z}, \boldsymbol{c}, t)}{\alpha_t}\right), \quad (\epsilon\text{-Prediction}) \tag{9}$$

- $c_{\text{skip}}(0) = 1$ and $c_{\text{out}}(0) = 0$
- $\hat{\epsilon}_\theta(z_t, c, t)$: a noise prediction model that initializes with the same parameters as the teacher diffusion model

- $\boldsymbol{f_\theta}$ can be parameterized in various ways, depending on the teacher diffusion model parameterizations of predictions ☞ Appendix D.

4.1 LCM : Consistency Distillation in the Latent Space

➢ Recall that the **PF-ODE** of the reverse diffusion process

$$\frac{\mathrm{d}\boldsymbol{z}_t}{\mathrm{d}t} = f(t)\boldsymbol{z}_t + \frac{g^2(t)}{2\sigma_t}\boldsymbol{\epsilon}_\theta\left(\boldsymbol{z}_t, \boldsymbol{c}, t\right), \quad \boldsymbol{z}_T \sim \mathcal{N}\left(\mathbf{0}, \tilde{\sigma}^2 \boldsymbol{I}\right), \tag{8}$$

- Assume that an efficient **ODE solver** $\Psi(z_t, t, s, c)$ is available for approximating the integration of the right-hand side of Eq (8) from time t to s.
- In practice, we can use DDIM (Song et al., 2020a), DPM-Solver (Lu et al., 2022a) or DPM-Solver++ (Lu et al., 2022b) as $\Psi(\cdot,\cdot,\cdot,\cdot)$.
- Note that we only use these solvers in training/distillation, not in inference

❖ Discuss these solvers further when we introduce the **SKIPPING-STEP** technique in Sec. 4.3

➢ **LCM** aims to **predict the solution of the PF-ODE** by **minimizing the consistency distillation loss** (Song et al., 2023):

$$\mathcal{L}_{\mathcal{CD}}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \Psi\right) = \mathbb{E}_{\boldsymbol{z}, \boldsymbol{c}, n}\left[d\left(\boldsymbol{f}_\theta(\boldsymbol{z}_{t_{n+1}}, \boldsymbol{c}, t_{n+1}), \boldsymbol{f}_{\theta^-}(\hat{\boldsymbol{z}}^{\Psi}_{t_n}, \boldsymbol{c}, t_n)\right)\right] \tag{10}$$

- $\hat{z}^{\Psi}_{t_n}$: an estimation of the evolution of the PF-ODE from $t_{n+1} \rightarrow t_n$ using ODE solver Ψ:

$$\hat{\boldsymbol{z}}^{\Psi}_{t_n} - \boldsymbol{z}_{t_{n+1}} = \int_{t_{n+1}}^{t_n} \left(f(t)\boldsymbol{z}_t + \frac{g^2(t)}{2\sigma_t}\boldsymbol{\epsilon}_\theta\left(\boldsymbol{z}_t, \boldsymbol{c}, t\right)\right) \mathrm{d}t \tag{11}$$
$$\approx \Psi(\boldsymbol{z}_{t_{n+1}}, t_{n+1}, t_n, \boldsymbol{c}),$$

- The solver $\Psi(\cdot,\cdot,\cdot,\cdot)$ is used to approximate the integration from $t_{n+1} \rightarrow t_n$.

4.2 One-Stage Guided Distillation by Solving Augmented PF-ODE

➢ **Classifier-free guidance (CFG)** (Ho & Salimans, 2022)

- Crucial for **synthesizing high-quality textaligned images in SD**, typically needing a **CFG scale ω over 6**. Thus, **integrating CFG into a distillation** method becomes **indispensable**.

- Previous method **Guided-Distill** (Meng et al., 2023) introduces a **two-stage distillation** to support few-step sampling from a guided diffusion model. However, it is computationally intensive (e.g. at least **45 A100 GPUs Days for 2-step inference**, estimated in (Liu et al., 2023)).
- An **LCM** demands merely **32 A100 GPUs Hours training for 2-step inference**, as depicted in Figure 1.

- Furthermore, the **two-stage guided distillation** might result in **accumulated error**, leading to suboptimal performance.
- **LCM**s adopt efficient **one-stage guided distillation** by **solving an augmented PF-ODE**.

➢ **CFG used in reverse diffusion process**

$$\tilde{\epsilon}_\theta\left(z_t, \omega, c, t\right) := (1+\omega)\epsilon_\theta\left(z_t, c, t\right) - \omega\epsilon_\theta\left(z_t, \varnothing, t\right), \tag{12}$$

- The original noise prediction is replaced by the linear combination of conditional and unconditional noise
- ω is called the *guidance scale*

➢ **Augmented PF-ODE**

- To sample from the guided reverse process, we need to solve the following augmented PF-ODE: (i.e., augmented with the terms related to ω)

$$\frac{\mathrm{d}z_t}{\mathrm{d}t} = f(t)z_t + \frac{g^2(t)}{2\sigma_t}\tilde{\epsilon}_\theta\left(z_t, \omega, c, t\right), \quad z_T \sim \mathcal{N}\left(\mathbf{0}, \tilde{\sigma}^2 \boldsymbol{I}\right). \tag{13}$$

4.2 One-Stage Guided Distillation by Solving Augmented PF-ODE

➤ Augmented consistency function $\boldsymbol{f}_{\boldsymbol{\theta}}$

- To efficiently perform **one-stage guided distillation**, we introduce an augmented consistency function

$$\boldsymbol{f}_{\boldsymbol{\theta}} : (\boldsymbol{z}_t, \omega, \boldsymbol{c}, t) \mapsto \boldsymbol{z}_0$$

to directly predict the solution of augmented PF-ODE (Eq. 13) for $t = 0$.

- We parameterize the $\boldsymbol{f}_{\boldsymbol{\theta}}$ in the same way as in Eq. 9, except that $\hat{\epsilon}_\theta(z_t, c, t)$ is replaced by $\hat{\epsilon}_\theta(z_t, \omega, c, t)$, which is **a noise prediction model initializing with the same parameters** as **the teacher diffusion model**, but also contains additional trainable parameters for conditioning on ω.

- The **consistency loss** is the same as Eq. 10 except that we use augmented consistency function $\boldsymbol{f}_{\boldsymbol{\theta}}(z_t, \omega, c, t)$.

$$\mathcal{L}_{\mathcal{CD}}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \Psi\right) = \mathbb{E}_{\boldsymbol{z},\boldsymbol{c},\omega,n}\left[d\left(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}_{t_{n+1}}, \omega, \boldsymbol{c}, t_{n+1}), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\hat{\boldsymbol{z}}^{\Psi,\omega}_{t_n}, \omega, \boldsymbol{c}, t_n)\right)\right] \quad (14)$$

4.2 One-Stage Guided Distillation by Solving Augmented PF-ODE

➢ Consistency Loss

- The **consistency loss** is the same as Eq. 10 except that we use augmented consistency function $\boldsymbol{f_\theta}(z_t, \omega, c, t)$.

$$\mathcal{L}_{\mathcal{CD}}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{-}; \Psi\right) = \mathbb{E}_{\boldsymbol{z},\boldsymbol{c},\omega,n}\left[d\left(\boldsymbol{f_\theta}(\boldsymbol{z}_{t_{n+1}}, \omega, \boldsymbol{c}, t_{n+1}), \boldsymbol{f_{\theta^-}}(\hat{\boldsymbol{z}}_{t_n}^{\Psi,\omega}, \omega, \boldsymbol{c}, t_n)\right)\right] \quad (14)$$

- ω and n are uniformly sampled from interval $[\omega_{\text{min}}, \omega_{\text{max}}]$ and {1, . . . ,N−1} respectively.
- $\hat{z}_{t_n}^{\Psi,\omega}$ is estimated using the new noise model $\hat{\epsilon}_\theta(z_t, \omega, c, t)$

$$\begin{aligned}
\hat{\boldsymbol{z}}_{t_n}^{\Psi,\omega} - \boldsymbol{z}_{t_{n+1}} &= \int_{t_{n+1}}^{t_n}\left(f(t)\boldsymbol{z}_t + \frac{g^2(t)}{2\sigma_t}\tilde{\boldsymbol{\epsilon}}_\theta\left(\boldsymbol{z}_t, \omega, \boldsymbol{c}, t\right)\right)\mathrm{d}t \\
&= (1+\omega)\int_{t_{n+1}}^{t_n}\left(f(t)\boldsymbol{z}_t + \frac{g^2(t)}{2\sigma_t}\boldsymbol{\epsilon}_\theta\left(\boldsymbol{z}_t, \boldsymbol{c}, t\right)\right)\mathrm{d}t - \omega\int_{t_{n+1}}^{t_n}\left(f(t)\boldsymbol{z}_t + \frac{g^2(t)}{2\sigma_t}\boldsymbol{\epsilon}_\theta\left(\boldsymbol{z}_t, \varnothing, t\right)\right)\mathrm{d}t \quad (15) \\
&\approx (1+\omega)\Psi(\boldsymbol{z}_{t_{n+1}}, t_{n+1}, t_n, \boldsymbol{c}) - \omega\Psi(\boldsymbol{z}_{t_{n+1}}, t_{n+1}, t_n, \varnothing).
\end{aligned}$$

- We can use **DDIM** (Song et al., 2020a), **DPM-Solver** (Lu et al., 2022a) or **DPM-Solver++** (Lu et al., 2022b) as the **PF-ODE solver $\Psi(\cdot, \cdot, \cdot, \cdot)$.**

Latent Consistency Models

4.3 Accelerating Distillation with Skipping Time Steps

➢ **Discretization Schedule (or Time Schedule)**

- **Discrete diffusion models** (Ho et al., 2020; Song & Ermon, 2019) typically **train noise prediction models** with **a long time-step schedule** $\{t_i\}_i$ (also called discretization schedule or time schedule) to achieve **high quality generation results**.
  - ✓ **Stable Diffusion (SD)** has a time schedule of length **1,000**.
- Directly applying **Latent Consistency Distillation (LCD)** to **SD** with such **an extended schedule** can be **problematic**.
  - ✓ The model needs to **sample across all 1,000 time steps**.
  - ✓ The **consistency loss** attempts to **aligns the prediction of LCM model** $f_\theta(z_{t_{n+1}}, c, t_{n+1})$ with **the prediction** $f_\theta(z_{t_n}, c, t_n)$ **at the subsequent step along the same trajectory**.
  - ✓ Since $t_n - t_{n+1}$ **is tiny**, z_{t_n} **and** $z_{t_{n+1}}$ **(and thus** $f_\theta(z_{t_{n+1}}, c, t_{n+1})$ **and** $f_\theta(z_{t_n}, c, t_n)$**) are already close to each other**, incurring **small consistency loss** and hence leading to **slow convergence**.

➢ **Skipping-Step**

- To address this issues, we introduce the **SKIPPING-STEP** method to considerably **shorten the length of time schedule** (**from thousands to dozens**) to achieve **fast convergence** while **preserving generation quality**
- **Consistency Models (CMs)** (Song et al., 2023)
  - ✓ Use the **EDM** (Karras et al., 2022) **continuous time schedule**, and the **Euler**, or **Heun Solver** as **the numerical continuous PF-ODE solver**.
- For **LCM**s, in order to adapt to the **discrete-time schedule** in **SD,**
  - ✓ We utilize **DDIM** (Song et al., 2020a), **DPM-Solver** (Lu et al., 2022a), or **DPM-Solver++** (Lu et al., 2022b) as **the ODE solver**. (Lu et al., 2022a) shows that these advanced solvers can solve the PF-ODE efficiently in Eq. 8.

4.3 Accelerating Distillation with Skipping Time Steps

➢ Skipping-Step Method in Latent Consistency Distillation (LCD)

- Instead of ensuring consistency between adjacent time steps $\boldsymbol{t_{n+1}} \rightarrow \boldsymbol{t_n}$, LCMs aim to ensure consistency between the current time step and k-step away, $\boldsymbol{t_{n+k}} \rightarrow \boldsymbol{t_n}$.
- SKIPPING-STEP method is crucial in accelerating the LCD process.

→

- $k = 1$ reduces to the original schedule in (Song et al., 2023), leading to slow convergence, and very large k may incur large approximation errors of the ODE solvers.
- In our main experiments, we set $\boldsymbol{k = 20}$, drastically reducing the length of time schedule from thousands to dozens.

- Consistency distillation loss in Eq. 14 is modified to ensure consistency from $\boldsymbol{t_{n+k}}$ to $\boldsymbol{t_n}$:

$$\mathcal{L}_{\mathcal{CD}}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{-}; \Psi\right) = \mathbb{E}_{\boldsymbol{z}, \boldsymbol{c}, \omega, n}\left[d\left(\boldsymbol{f}_{\boldsymbol{\theta}}(z_{t_{n+k}}, \omega, \boldsymbol{c}, t_{n+k}), \boldsymbol{f}_{\boldsymbol{\theta}^{-}}(\hat{z}_{t_n}^{\Psi,\omega}, \omega, \boldsymbol{c}, t_n)\right)\right] \tag{16}$$

$\hat{z}_{t_n}^{\psi,\omega}$ being an estimate of z_{t_n} using numerical ***augmented PF-ODE*** solver Ψ

$$\hat{\boldsymbol{z}}_{t_n}^{\Psi,\omega} \longleftarrow \boldsymbol{z}_{t_{n+k}} + (1+\omega)\Psi(z_{t_{n+k}}, t_{n+k}, t_n, \boldsymbol{c}) - \omega\Psi(z_{t_{n+k}}, t_{n+k}, t_n, \varnothing) \tag{17}$$

4.3 Accelerating Distillation with Skipping Time Steps

➢ Skipping-Step Method in Latent Consistency Distillation (LCD)

- Consistency distillation loss in Eq. 14 is modified to ensure consistency from $\boldsymbol{t_{n+k}}$ to $\boldsymbol{t_n}$:

$$\mathcal{L}_{\mathcal{CD}}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{-}; \Psi\right) = \mathbb{E}_{\boldsymbol{z}, \boldsymbol{c}, \omega, n}\left[d\left(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}_{t_{n+k}}, \omega, \boldsymbol{c}, t_{n+k}), \boldsymbol{f}_{\boldsymbol{\theta}^{-}}(\hat{\boldsymbol{z}}_{t_n}^{\Psi,\omega}, \omega, \boldsymbol{c}, t_n)\right)\right] \quad (16)$$

$\hat{z}_{t_n}^{\psi,\omega}$ being an estimate of z_{t_n} using numerical ***augmented PF-ODE*** solver Ψ

$$\hat{\boldsymbol{z}}_{t_n}^{\Psi,\omega} \longleftarrow \boldsymbol{z}_{t_{n+k}} + (1+\omega)\Psi(\boldsymbol{z}_{t_{n+k}}, t_{n+k}, t_n, \boldsymbol{c}) - \omega\Psi(\boldsymbol{z}_{t_{n+k}}, t_{n+k}, t_n, \varnothing) \quad (17)$$

- For LCM, we use three possible ODE solvers here: DDIM (Song et al., 2020a), DPM-Solver (Lu et al., 2022a), DPM-Solver++ (Lu et al., 2022b), and we compare their performance in Sec 5.2.
- In fact, DDIM (Song et al., 2020a) is the first-order discretization approximation of the DPM-Solver (Proven in (Lu et al., 2022a)). Here we provide the detailed formula of the DDIM PF-ODE solver Ψ_{DDIM} from from $\boldsymbol{t_{n+k}}$ to $\boldsymbol{t_n}$.

$$\Psi_{\mathrm{DDIM}}(\boldsymbol{z}_{t_{n+k}}, t_{n+k}, t_n, \boldsymbol{c}) = \underbrace{\frac{\alpha_{t_n}}{\alpha_{t_{n+k}}}\boldsymbol{z}_{t_{n+k}} - \sigma_{t_n}\left(\frac{\sigma_{t_{n+k}} \cdot \alpha_{t_n}}{\alpha_{t_{n+k}} \cdot \sigma_{t_n}} - 1\right)\hat{\boldsymbol{\epsilon}}_{\theta}(\boldsymbol{z}_{t_{n+k}}, \boldsymbol{c}, t_{n+k})}_{\text{DDIM Estimated } \boldsymbol{z}_{t_n}} - \boldsymbol{z}_{t_{n+k}} \quad (18)$$

4.3 Accelerating Distillation with Skipping Time Steps

➢ **Pseudo-code for LCD with CFG and Skipping-Step techniques**

- The modifications from the original Consistency Distillation (CD) algorithm in Song et al. (2023) are highlighted in blue.
- LCM sampling algorithm 3 is provided in Appendix B.

Algorithm 1 Latent Consistency Distillation (LCD)

Input: dataset $\mathcal{D}$, initial model parameter $\boldsymbol{\theta}$, learning rate η, ODE solver $\Psi(\cdot,\cdot,\cdot,\cdot)$, distance metric $d(\cdot,\cdot)$, EMA rate μ, noise schedule $\alpha(t), \sigma(t)$, guidance scale $[w_{\min}, w_{\max}]$, skipping interval k, and encoder $E(\cdot)$
Encoding training data into latent space: $\mathcal{D}_z = \{(\boldsymbol{z}, \boldsymbol{c}) | \boldsymbol{z} = E(\boldsymbol{x}), (\boldsymbol{x}, \boldsymbol{c}) \in \mathcal{D}\}$
$\boldsymbol{\theta}^- \leftarrow \boldsymbol{\theta}$
repeat
  Sample $(\boldsymbol{z}, \boldsymbol{c}) \sim \mathcal{D}_z$, $n \sim \mathcal{U}[1, N-k]$ and $\omega \sim [\omega_{\min}, \omega_{\max}]$
  Sample $\boldsymbol{z}_{t_{n+k}} \sim \mathcal{N}(\alpha(t_{n+k})\boldsymbol{z}; \sigma^2(t_{n+k})\mathbf{I})$
  $\hat{\boldsymbol{z}}_{t_n}^{\Psi,\omega} \leftarrow \boldsymbol{z}_{t_{n+k}} + (1+\omega)\Psi(\boldsymbol{z}_{t_{n+k}}, t_{n+k}, t_n, \boldsymbol{c}) - \omega\Psi(\boldsymbol{z}_{t_{n+k}}, t_{n+k}, t_n, \varnothing)$
  $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \Psi) \leftarrow d(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}_{t_{n+k}}, \omega, \boldsymbol{c}, t_{n+k}), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\hat{\boldsymbol{z}}_{t_n}^{\Psi,\omega}, \omega, \boldsymbol{c}, t_n))$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-)$
  $\boldsymbol{\theta}^- \leftarrow \text{stopgrad}(\mu\boldsymbol{\theta}^- + (1-\mu)\boldsymbol{\theta})$
until convergence

4.4 Latent Consistency Fine-Tuning For Customized Dataset

☞ Appendix C

- **Stable Diffusion** excel in diverse text-to-image generation tasks but often **require fine-tuning on customized datasets** to meet the requirements of downstream tasks.

➢ Propose **Latent Consistency Fine-tuning (LCF)**, **a fine-tuning method for pretrained LCM**.

- Inspired by **Consistency Training (CT)** (Song et al., 2023), LCF enables **efficient few-step inference on customized datasets without relying on a teacher diffusion model trained on such data.**

Note that

- This method can also utilize the skipping-step technique to speedup the convergence.
- LCF is independent of the pre-trained teacher model, facilitating direct fine-tuning of a pre-trained LCM model without reliance on the teacher diffusion model

- Randomly select two time steps t_n and t_{n+k} that are k time steps apart and apply the same Gaussian noise ϵ to obtain the noised data z_{t_n}, $z_{t_{n+k}}$ as follows

$$\boldsymbol{z}_{t_{n+k}} = \alpha(t_{n+k})\boldsymbol{z} + \sigma(t_{n+k})\boldsymbol{\epsilon} \quad , \quad \boldsymbol{z}_{t_n} = \alpha(t_n)\boldsymbol{z} + \sigma(t_n)\boldsymbol{\epsilon}.$$

- Directly calculate the consistency loss for these two time steps to enforce self-consistency property in Eq.4.

Algorithm 4 Latent Consistency Fine-tuning (LCF)

Input: customized dataset $\mathcal{D}^{(s)}$, pre-trained LCM parameter $\boldsymbol{\theta}$, learning rate η, distance metric $d(\cdot,\cdot)$, EMA rate μ, noise schedule $\alpha(t), \sigma(t)$, guidance scale $[w_{\text{min}}, w_{\text{max}}]$, skipping interval k, and encoder $E(\cdot)$

Encode training data into the latent space: $\mathcal{D}_z^{(s)} = \{(\boldsymbol{z}, \boldsymbol{c}) | \boldsymbol{z} = E(\boldsymbol{x}), (\boldsymbol{x}, \boldsymbol{c}) \in \mathcal{D}^{(s)}\}$

$\boldsymbol{\theta}^- \leftarrow \boldsymbol{\theta}$

repeat

  Sample $(\boldsymbol{z}, \boldsymbol{c}) \sim \mathcal{D}_z^{(s)}$, $n \sim \mathcal{U}[1, N-k]$ and $w \sim [w_{\text{min}}, w_{\text{max}}]$

  Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

  $\boldsymbol{z}_{t_{n+k}} \leftarrow \alpha(t_{n+k})\boldsymbol{z} + \sigma(t_{n+k})\boldsymbol{\epsilon} \quad , \quad \boldsymbol{z}_{t_n} \leftarrow \alpha(t_n)\boldsymbol{z} + \sigma(t_n)\boldsymbol{\epsilon}$

  $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-) \leftarrow d(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}_{t_{n+k}}, t_{n+k}, \boldsymbol{c}, w), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\boldsymbol{z}_{t_n}, t_n, \boldsymbol{c}, w))$

  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-)$

  $\boldsymbol{\theta}^- \leftarrow \text{stopgrad}(\mu\boldsymbol{\theta}^- + (1-\mu)\boldsymbol{\theta})$

until convergence

5. EXPERIMENT

- Employ **latency consistency distillation** to **train LCM** on two subsets of LAION-5B.

5.1 Text-to-Image Generation

➢ **Datasets**

- LAION-5B (Schuhmann et al., 2022): LAION-Aesthetics-6+ (12M) and LAION-Aesthetics-6.5+ (650K) for text-to-image generation

- Consider two resolutions
- 512x512 resolution : Use LAION-Aesthetics-6+, which comprises 12M text-image pairs with predicted aesthetics scores higher than 6.
- 768x768 resolution : Use LAION-Aesthetics-6.5+, with 650K text-image pairs with aesthetics score higher than 6.5.

➢ **Model Configuration**

- 512x512 resolution : Use the **pre-trained Stable Diffusion-V2.1-Base** (Rombach et al., 2022) as the **teacher model**, which was originally trained on resolution 512×512 with ϵ-Prediction (Ho et al., 2020).
- 768x768 resolution : Use the widely used **pre-trained Stable Diffusion-V2.1**, originally trained on resolution 768×768 with v-Prediction (Salimans & Ho, 2022).
- Train LCM with 100K iterations and a batch size of 72 for (512×512) setting, and 16 for (768×768) setting, the same learning rate 8e-6 and EMA rate μ = 0.999943 as used in (Song et al., 2023).
- For *augmented PF-ODE solver* Ψ and skipping step k in Eq. 17,
  - Use DDIM-Solver (Song et al., 2020a) with skipping step k = 20.
  - Set the guidance scale range $[\omega_{\min}, \omega_{\max}] = [2,14]$, consistent with (Meng et al., 2023)

Latent Consistency Models

5.1 Text-to-Image Generation

➢ Baselines

- Baselines : DDIM (Song et al., 2020a), DPM (Lu et al., 2022a), DPM++ (Lu et al., 2022b), Guided-Distill (Meng et al., 2023)
- **DDIM, DPM, DPM++** : Taining-free samplers requiring m**ore peak memory** per step with **classifier-free guidance**.
- **Guided-Distill** : Requires **two stages of guided distillation**. Due to the limited resource (Meng et al. (2023) used a large batch size of 512, requiring at least 32 A100 GPUs), reduce the batch size to 72 and trained for the same 100K iterations.
- LCM achieves faster convergence and superior results under the same computation cost.

➢ Evaluation

- Generate **30K images** from **10K text prompts** in the test set (3 images per prompt)
- Adopt **FID** and **CLIP** scores to evaluate the diversity and quality of the generated images.
- Use ViT-g/14 for evaluating CLIP scores

Latent Consistency Models

5.1 Text-to-Image Generation

➢ **Results**

Model (512 × 512) Reso	FID ↓				CLIP Score ↑			
	1 Step	2 Steps	4 Steps	8 Steps	1 Steps	2 Steps	4 Steps	8 Steps
DDIM (Song et al., 2020a)	183.29	81.05	22.38	13.83	6.03	14.13	25.89	29.29
DPM (Lu et al., 2022a)	185.78	72.81	18.53	12.24	6.35	15.10	26.64	29.54
DPM++ (Lu et al., 2022b)	185.78	72.81	18.43	12.20	6.35	15.10	26.64	**29.55**
Guided-Distill (Meng et al., 2023)	108.21	33.25	15.12	13.89	12.08	22.71	27.25	28.17
LCM (Ours)	**35.36**	**13.31**	**11.10**	**11.84**	**24.14**	**27.83**	**28.69**	28.84

Table 1: Quantitative results with $\omega = 8$ at 512×512 resolution. LCM significantly surpasses baselines in the 1-4 step region on LAION-Aesthetic-6+ dataset. For LCM, DDIM-Solver is used with a skipping step of $k = 20$.

Model (768 × 768) Reso	FID ↓				CLIP Score ↑			
	1 Step	2 Steps	4 Steps	8 Steps	1 Steps	2 Steps	4 Steps	8 Steps
DDIM (Song et al., 2020a)	186.83	77.26	24.28	15.66	6.93	16.32	26.48	29.49
DPM (Lu et al., 2022a)	188.92	67.14	20.11	**14.08**	7.40	17.11	27.25	29.80
DPM++ (Lu et al., 2022b)	188.91	67.14	20.08	14.11	7.41	17.11	27.26	**29.84**
Guided-Distill (Meng et al., 2023)	120.28	30.70	16.70	14.12	12.88	24.88	28.45	29.16
LCM (Ours)	**34.22**	**16.32**	**13.53**	14.97	**25.32**	**27.92**	**28.60**	28.49

Table 2: Quantitative results with $\omega = 8$ at 768×768 resolution. LCM significantly surpasses the baselines in the 1-4 step region on LAION-Aesthetic-6.5+ dataset. For LCM, DDIM-Solver is used with a skipping step of $k = 20$.

- DDIM, DPM, DPM++ require more peak memory per sampling step with CFG
- LCM requires only one forward pass per sampling step, saving both time and memory.

- Guided-Distill : two-stage distillation procedure
- LCM : one-stage guided distillation, which is much simpler and more practical.

5.1 Text-to-Image Generation

➢ **Results**

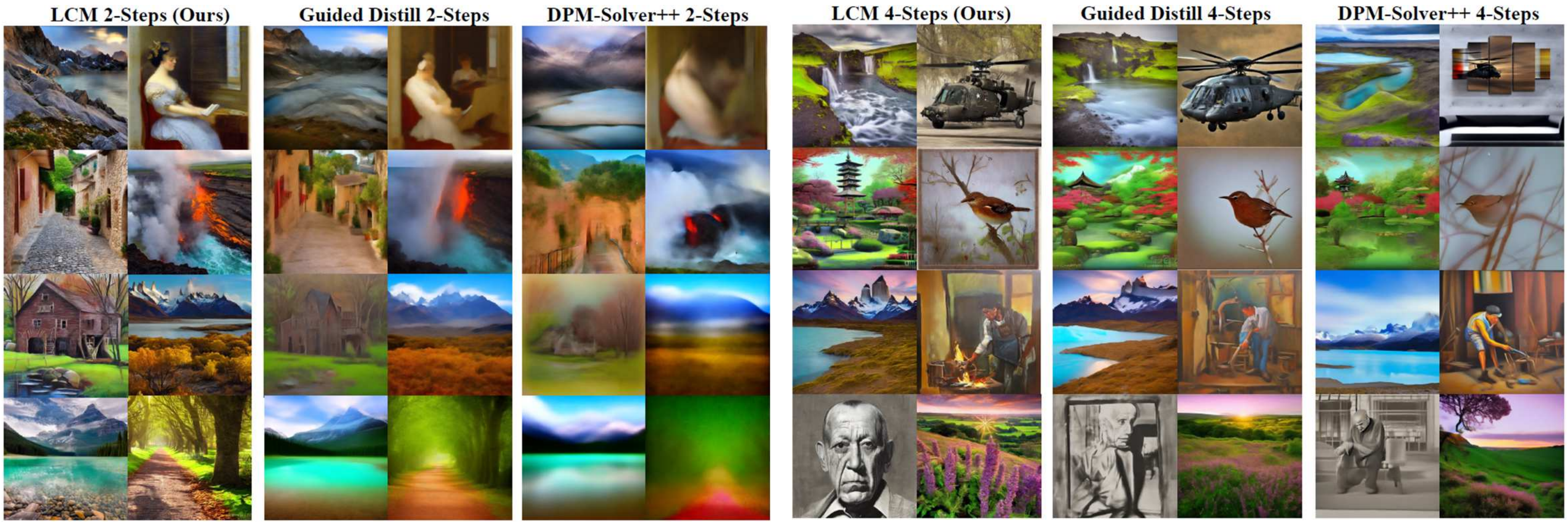


Figure 2: Text-to-Image generation results on LAION-Aesthetic-6.5+ with 2-, 4-step inference. Images generated by LCM exhibit superior detail and quality, outperforming other baselines by a large margin.

5.2 Ablation Study

➢ ODE Solvers & Skipping-Step Schedule

- Compare various solvers Ψ (DDIM, DPM, DPM++ for solving the ***augmented PF-ODE*** specified in Eq 17, and explore different skipping step schedules with different k.

1) Using **SKIPPING-STEP** techniques (see Sec 4.3), LCM achieves fast convergence within **2,000 iterations** in the **4-step** inference setting.
2) **DPM** and **DPM++** solvers perform **better at a larger skipping step** (k = 50) compared to the **DDIM solver** which suffers from **increased ODE approximation error with larger k**.
3) Very small k values (1 or 5) result in slow convergence and very large ones (e.g., 50 for DDIM) may lead to inferior results

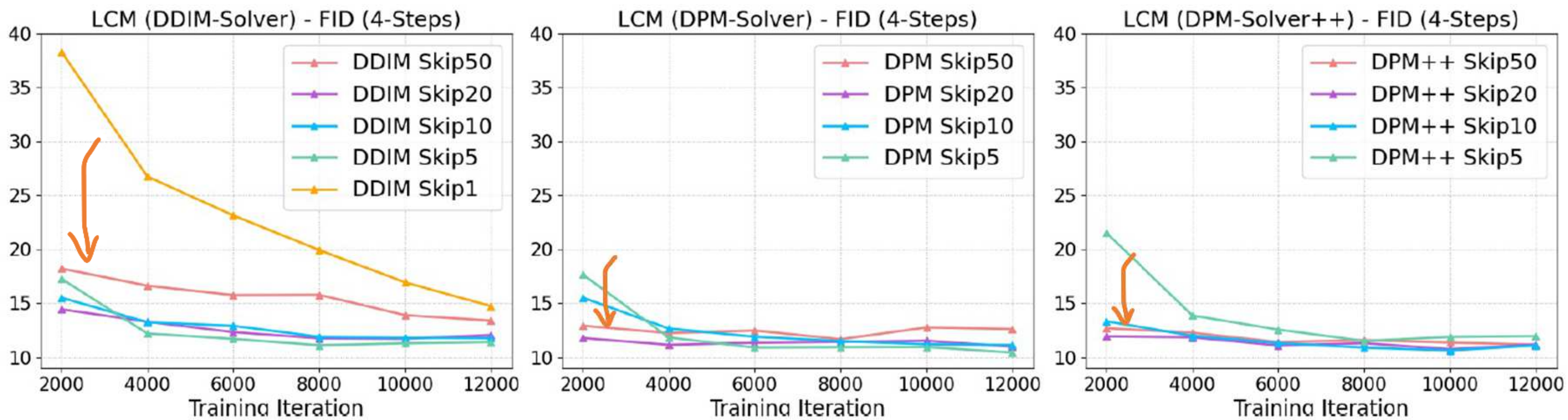


Figure 3: Ablation study on different ODE solvers and skipping step k. Appropriate skipping step k can significantly accelerate convergence and lead to better FID within the same number of training steps.

- We choose k = 20, which provides competitive performance for all three solvers.

5.2 Ablation Study

➢ The Effect of Guidance Scale ω

- Examine the effect of using different **CFG scales** ω in LCM.
- ω balances sample **quality** and **diversity**
  - ✓ **A larger ω** generally tends to **improve sample quality** (indicated by **CLIP**), but may **compromise diversity** (measured by **FID**).
  - ✓ An increased ω yields better CLIP scores at the expense of FID.

1) Using large ω enhances sample quality (CLIP Scores) but results in relatively inferior FID.
2) The performance gaps across 2, 4, and 8 inference steps are negligible, highlighting LCM's efficacy in 2~8 step regions. However, a noticeable gap exists in one-step inference.

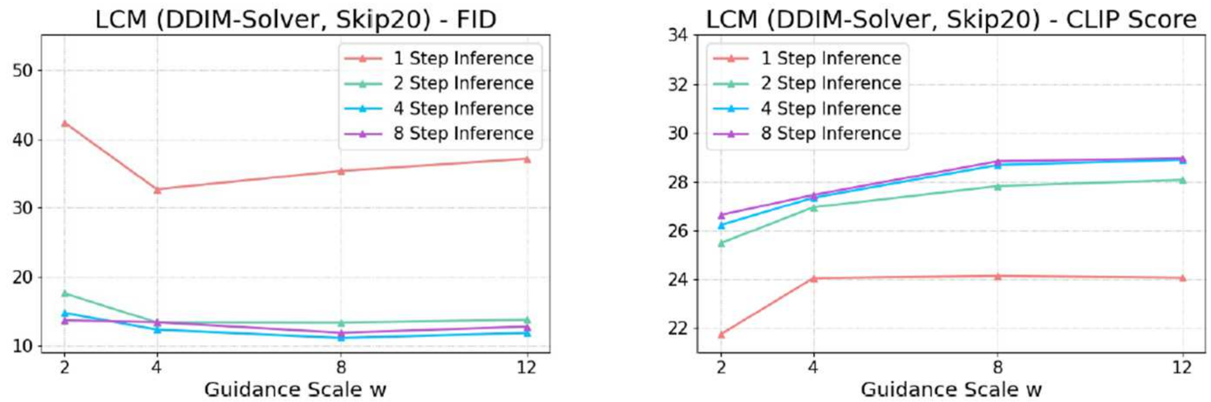


Figure 4: Ablation study on different classifier-free guidance scales ω. Larger ω leads to better sample quality (CLIP Scores). The performance gaps across 2, 4, and 8 steps are minimal, showing the efficacy of LCM.

5.2 Ablation Study

➢ The Effect of Guidance Scale ω

A larger ω enhances sample quality, verifying the effectiveness of our one-stage guided distillation method.

Figure 5: 4-step LCMs using different CFG scales ω. LCMs utilize one-stage guided distillation to directly incorporate CFG scales ω. Larger ω enhances image quality.

5.3 Downstream Consistency Fine-tuning Results

- 2 customized image datasets, Pokemon dataset (Pinkney, 2022) and Simpsons dataset (Norod78, 2022), that **90%** is used for **fine-tuning** and the rest **10%** for **testing**.
- For **LCF**, we utilize **pretrained LCM** that was originally trained at the resolution of 768×768 used in Table 2. We fine-tune the pre-trained LCM for **30K iterations** with a learning rate 8e-6.

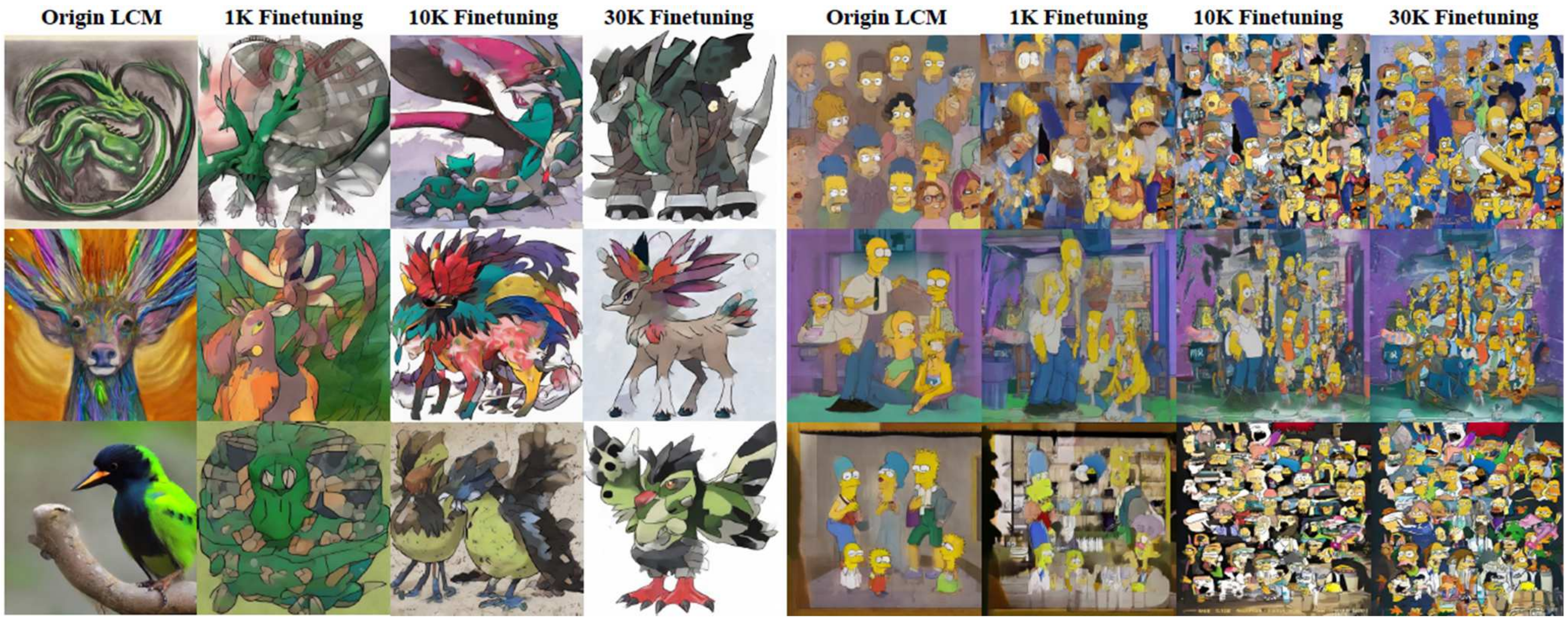


Figure 6: 4-step LCMs using Latent Consistency Fine-tuning (LCF) on two customized datasets: Pokemon Dataset (left), Simpsons Dataset (right). Through LCF, LCM produces images with customized styles.