Accelerating Image Generation with Sub-Path Linear Approximation Model

Chen Xu[1,2][†] , Tianhui Song[1,2][†] , Weixin Feng[2] , Xubin Li[2] ,
Tiezheng Ge[2] , Bo Zheng[2] , and Limin Wang[1,3,*]

[1] State Key Laboratory for Novel Software Technology, Nanjing University,
Nanjing, China
[2] Alibaba Group, Hangzhou, China   [3] Shanghai AI Lab, Shanghai, China

Abstract. Diffusion models have significantly advanced the state of the
art in image, audio, and video generation tasks. However, their applica-
tions in practical scenarios are hindered by slow inference speed. Draw-
ing inspiration from the consistency models, we propose the **S**ub-**P**ath
Linear **A**pproximation **M**odel (SPLAM), which can accelerate diffusion
models while maintaining high-quality image generation. SPLAM treats
the PF-ODE trajectory as a series of PF-ODE sub-paths divided by
sampled points, and harnesses sub-path linear (SL) ODEs to form a pro-
gressive and continuous error estimation along each individual PF-ODE
sub-path. The optimization on such SL-ODEs allows SPLAM to con-
struct denoising mapping with smaller cumulative approximated error.
An efficient distillation method is also developed to facilitate the incor-
poration of pre-trained diffusion models, such as latent diffusion mod-
els. The extensive experimental results demonstrate SPLAM achieves
remarkable training efficiency, requiring only 6 A100 GPU days to pro-
duce a high-quality generative model capable of 2 to 4-step generation.
Comprehensive evaluations on LAION, MS COCO 2014, and MS COCO
2017 datasets also illustrate that SPLAM surpasses the existing accel-
eration methods in few-step generation tasks, achieving state-of-the-art
performance both on FID and the quality of the generated images.

Keywords: Diffusion Models · Accelerating Diffusion Models · Diffusion
Model Distillation · Consistency Models.

1 Introduction

Diffusion models, also known as score-based generative models, have emerged
as a potent paradigm in generative computer vision, enabling the synthesis of
highly realistic images by progressively refining random noise into structured
visual content [9,27,29,42,43]. Despite their impressive ability, one of the primary
challenges associated with diffusion models lies in their computational intensity,
often requiring hundreds of iteration steps to produce a single image. This has
spurred a surge of research focused on accelerating diffusion models to retain

---

[†] Equal contributions. Interns at Alibaba Group. [*] Corresponding author.

high-quality outputs while significantly reducing the computation cost during the inference phase [19–22, 24, 33, 39, 41, 46, 47].

Within the spectrum of acceleration techniques, consistency models [24, 41] have garnered attention as they forge a consistent denoising mapping across points on Probability Flow (PF) ODE trajectories. The learning strategy brings consistency models a notable *consistency property* and could estimate the overall prediction errors as a summation of incremental errors, which are computed as the difference between the predicted results of adjacent trajectory points. In this paper, we recognize that the approximation of denoising mappings by consistency models is essentially a minimization process targeting the endpoints of sub-paths along ODE trajectories. We observe that the approximated performance is currently limited by the accumulation of errors that arise from either an overabundance of approximation operations, or the heightened challenge of optimizing individual sub-path errors as the skipping step size expands.

To address these challenges, we propose a novel approach in this paper, designated as the Sub-Path Linear Approximation Model (SPLAM). SPLAM adheres to the foundational concept of cumulative approximation of PF-ODE trajectories but innovates through its sustained learning from Sub-Path Linear (SL) ODEs. Specifically, we dissect the sub-path learning objective based on the noise prediction design [9, 13] into two interrelated aspects, and establish the SL-ODEs to give respective progressive or continuous estimation for each component, by a carefully designed linear interpolation between the endpoints of sub-paths. We then utilize the SL-ODEs to approximate the complete PF-ODE trajectories which allows a more nuanced optimization. Consequently, the prediction error of our approach is assessed through iterative solutions of all SL-ODEs, enabling a reduction of cumulative errors and an enhancement in image generation quality. Furthermore, we also develop an efficient distillation procedure for our SPLAM which enables the incorporation with pre-trained latent diffusion models [31] (*e.g.*, Stable Diffusion). Our contributions can be summarized as below:

1. We identify that the optimization process for consistency models essentially minimizes the cumulative approximated error along PF-ODE sub-path endpoints, and observe that the performance of such approximations is hindered by the proliferating number of approximations or the amplified difficulty in optimizing single sub-path errors for as skipping step size increases.
2. To address these challenges, we propose a novel approach as Sub-Path Linear Approximation Model (SPLAM). SPLAM employs Sub-Path Linear (SL) ODEs to continuously approximate the complete PF-ODE trajectories and progressively optimize the sub-path learning objectives, which could construct the denoising mappings with smaller cumulative approximated errors.
3. Leveraging the proposed SPLAM and SL-ODE framework, we put forth an efficient distillation method. When integrated with powerful pre-trained models like Stable Diffusion, our approach allows more efficient training and respectively attains impressive FIDs as 10.09, 10.06, 20.77 in LAION, MS COCO 2014, MS COCO 2017 datasets, achieving better performance and close inference latency to all previous accelerating approaches.
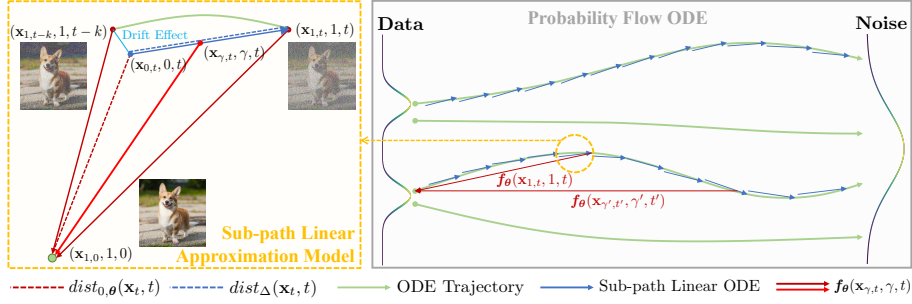
Fig. 1: Our Sub-Path Linear Approximation Model employs Sub-Path Linear ODEs to approximate the sub-paths on the PF-ODE trajectories, which is determined by the linear interpolation of corresponding endpoints. SPLAM is then trained based on the consistent mapping along SL-ODEs to minimize the approximated errors.

2   Related Work

Diffusion Models [1, 9, 13, 28, 31, 37, 43] have solidified their status as a cornerstone in the realm of generative models, outshining previous approaches in creating rich and detailed images. Song *et al.* [43] model this process from continuous-time perspective with a stochastic differential equation (SDE), which iteratively denoise an initial noise distribution leveraging the learned *score* of the data distribution to steer the process towards data points [9, 42, 43]. This reverse diffusion process has been verified to be particularly adept at capturing the intricate structures and variations inherent in complex data sets. They also demonstrate that there exists an ordinary differential equation (ODE), dubbed as Probability Flow (PF) ODE, which shares the marginal probability densities with the reverse-time SDE and thus yields a deterministic sampling trajectory [13, 43]. In contrast to other generative models like VAEs [14, 38] and GANs [6], diffusion models demonstrate remarkable robustness in training and excel in producing samples with substantial diversity and high fidelity, thereby offering a robust solution for modeling complex distributions in an ever-expanding array of generative tasks.

Accelerating Diffusion Models. While diffusion models have demonstrated their superiority in generating high-quality samples, the generation speed remains a major hindrance due to requiring thousands of sampling steps, which poses difficulties for practical and efficient applications. To address these issues, a surge of advancements has emerged aiming to accelerate the inference process. Some works concentrate on designing non-training fast diffusion samplers [2, 11, 13, 18, 21, 22, 43, 52], potentially cutting down the steps from one thousand to a modest 20-50. In the realm of distillation [8], efforts have been undertaken [3, 7, 23, 26, 33, 50, 52] to condense the inference steps of pre-trained diffusion models to fewer than 10. Progressive distillation (PD) [33] seeks to amortize the integration of PF-ODE into a new sampler that takes half as many sampling steps, displaying efficacy with as few as 2/4 steps. Consistency mod-

els [24, 25, 40, 41], as a nascent class of models, offer the promise of high-quality one-step generation by mapping any point along the PF-ODE trajectory back to the origin. Representing flow-based approaches [17, 19, 20, 44], InstaFlow [19, 20] propose a reflow technique to straighten the trajectories of probability flows and refine the coupling between noises and images, which achieves a one-step SD model. Concurrently, some strategies are exploring the inclusion of GAN-like objectives into diffusion models to afford fast generative capabilities [16, 34, 46, 47]. DMD [47] additionally proposes a distribution matching method that enables one-step high-quality image generation.

3   Preliminaries

Diffusion Models are a class of generative models that gradually transform data into a noisy state through Gaussian perturbations and subsequently learn to reverse this process to reconstruct the original data by progressively denoising it. Denote $\boldsymbol{x}_0$ as the data sampled from the original distribution $\boldsymbol{x}_0 \sim p_{data}(\boldsymbol{x})$ and $\alpha(t), \sigma(t)$ as functions that define a noise schedule. Diffusion models transition the data to a noise-corrupted marginal distribution, which can be expressed as:

$$p_t(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t|\alpha(t)\boldsymbol{x}_0, \sigma(t)^2 I), \tag{1}$$

for any time step $t \in [0, T]$.

Song *et al.* [43] describe the diffusion process using a stochastic differential equation (SDE):

$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt + g(t)d\boldsymbol{w}_t, \tag{2}$$

where $f(\cdot, \cdot)$ and $g(\cdot)$ denote the drift and diffusion coefficients, respectively, and $\boldsymbol{w}_t$ signifies the standard Brownian motion at time t. They also derive an ordinary differential equation (ODE) corresponding to this SDE, which defines the trajectories of solutions sampled at time t according to $p_t(\boldsymbol{x}_t)$:

$$d\boldsymbol{x}_t = \left[\boldsymbol{f}(\boldsymbol{x}_t, t) - \frac{1}{2}g(t)^2 \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) \right] dt, \tag{3}$$

referred to as the Probability Flow (PF) ODE. In the reverse denoising process, models are taught to learn a score function $\mathbf{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \approx \nabla \log p_t(\boldsymbol{x}_t)$, adhering to the PF-ODE. Therefore, diffusion models are also recognized as score-based generative models. Based on the diffusion process, latent diffusion models (LDMs) additionally employ a VAE encoder $\mathcal{E}(\cdot)$ and decoder $\mathcal{D}(\cdot)$ to compress the image $\boldsymbol{x}$ into latent space as $\boldsymbol{z} = \mathcal{E}(\boldsymbol{x})$ and reconstruct it by the decoder: $\hat{\boldsymbol{x}} = \mathcal{D}(\boldsymbol{z})$, and implement the diffusion process on the compressed vector $\boldsymbol{z}$ via latent space [31]. With the latent diffusion process, the pre-trained large-scale LDMs like Stable Diffusion (SD) Models could achieve more precise PF-ODE solutions and thus generate high-quality images.

Consistency Model has been proposed by Song *et al.* [41] as a novel paradigm within the family of generative models. Considering a solution trajectory of the PF-ODE $\{(\boldsymbol{x}_t, t)\}_{t \in [\epsilon, T]}$, consistency models comply with a *consistency function*

that projects every pair $(\boldsymbol{x}_t, t)$ along the trajectory back to the starting point: $\boldsymbol{F}(\boldsymbol{x}_t, t) \mapsto \boldsymbol{x}_\epsilon$, for any $t \in [\epsilon, T]$, to obtain a one-step generator. Here, ϵ represents a small positive constant, thereby making $\boldsymbol{x}_\epsilon$ a viable surrogate for $\boldsymbol{x}_0$. An important characteristic of the consistency models is the *self-consistency property*:

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

which is leveraged as the training constraint for the consistency models, whether when distilling knowledge from a pre-trained model or training from scratch. The model is parameterized as follows:

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

where $c_{\text{skip}}(t)$ and $c_{\text{out}}(t)$ are differentiable functions ensuring that $c_{\text{skip}}(\epsilon) = 1$ and $c_{\text{out}}(\epsilon) = 0$, guaranteeing that $\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{x}_\epsilon, \epsilon) \equiv \boldsymbol{x}_\epsilon$, and $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ is a deep neural network. For the distillation approach called as *Consistency Distillation*, the training objective is formulated as:

$$\mathcal{L}_{CD}(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \phi) = \mathbb{E}[d(\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_{n+1}}, t_{n+1}), \boldsymbol{F}_{\boldsymbol{\theta}^-}(\hat{\boldsymbol{x}}^{\Phi}_{t_n}, t_n))], \tag{6}$$

where $\hat{\boldsymbol{x}}^{\Phi}_{t_n} = \boldsymbol{x}_{t_{n+1}} + (t_{n+1} - t_n)\Phi(\boldsymbol{x}_{t_{n+1}}, t_{n+1}; \phi)$ serves as a one-step estimation of $\boldsymbol{x}_{t_n}$ based on $\boldsymbol{x}_{t_{n+1}}$ from $\Phi(\cdot; \phi)$, a update function of a one-step ODE solver, and $d(\cdot, \cdot)$ is a chosen distance metric. Consistency models also utilize the EMA strategy to stabilize the training, and $\boldsymbol{\theta}^-$ is the running average of $\boldsymbol{\theta}$. Latent Consistency Models (LCMs) [24] introduce consistency model into the distillation for latent diffusion models. To accelerate the training of consistency models, LCM employs a skipping step size k to ensure consistency between the current timestep and k-step away. With a conditional input c and a guidance scale w to achieve the CFG strategy [10], the modified learning objective for the *latent consistency distillation* is formulated as:

$$\mathcal{L}_{LCD}(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \phi) = \mathbb{E}[d(\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_{n+k}}, w, c, t_{n+k}), \boldsymbol{F}_{\boldsymbol{\theta}^-}(\hat{\boldsymbol{x}}^{\Phi}_{t_n}, w, c, t_n))]. \tag{7}$$

4 Methodology

4.1 Approximation Strategy for Denoiser

One-step Denoiser Parameterization. To synthesize an image from a sampled input $\boldsymbol{x}_t$ at a large time step t in one-step, a natural approach is to adopt the strategy from [9] that employs a neural network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ to predict a standard Gaussian distribution, which implements the denoising mapping parameterized as $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{\boldsymbol{x}_t - \sigma(t)\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{\alpha(t)}$. By redefining the target distribution for $(\boldsymbol{x}_t, t)$ as $\boldsymbol{x}^t_0 = \alpha(t)\boldsymbol{x}_0 \sim p_{data,t}(\alpha(t)\boldsymbol{x})$ and $\boldsymbol{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \alpha(t) * \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \boldsymbol{x}_t - \sigma(t)\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$, this predictive formulation can be recast into a canonical denoiser function defined in [13] that aims to minimize the denoising error as follows:

$$\mathcal{L}_D(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}^t_0 \sim p_{data,t}, \boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{x}^t_0, \sigma(t)^2 I)}[\|\boldsymbol{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \alpha(t)\boldsymbol{x}_0\|], \tag{8}$$

where $|\cdot|$ is an estimation of the error vector (*e.g.*, a L2 distance). *However, the Eq. (8) is hard to be optimized in practice.* For instance, when $\alpha(t)$ decreases over time step t which implies $\alpha(t)\boldsymbol{x}_0 \to \mathbf{0}$, the training is likely to collapse and the denoiser is taught to generally give a zero output.

Approximation Strategy in Consistency Models. We observe that, consistency models [24,41] provide a solution to the aforementioned issues by leveraging the *consistency property*. As we presume that we have obtained a good prediction result $\boldsymbol{f_\theta}(\boldsymbol{x}_{t-k}) \approx \boldsymbol{x}_0$, from a time step $t - k$ ahead of t for k steps, this property yields an approximated error estimation of Eq. (8) as :

$$\mathbb{E}[|\boldsymbol{D_\theta}(\boldsymbol{x}_t, t) - \alpha(t)\boldsymbol{f_\theta}(\boldsymbol{x}_{t-k}, t - k)|]. \tag{9}$$

By incorporating the expressions for $\boldsymbol{f_\theta}(\boldsymbol{x}_{t-k}, t-k)$ and $\boldsymbol{D_\theta}(\boldsymbol{x}_t, t)$, we derive the approximated error estimation based on $\boldsymbol{\epsilon_\theta}(\cdot, \cdot)$ as:

$$\mathcal{L}_{\text{Approx}}(\boldsymbol{\theta}) = \mathbb{E}[|\boldsymbol{x}_t - \frac{\alpha(t)}{\alpha(t-k)}\boldsymbol{x}_{t-k} + \frac{\alpha(t)}{\alpha(t-k)}\sigma(t-k)\boldsymbol{\epsilon_\theta}(\boldsymbol{x}_{t-k}, t-k) - \sigma(t)\boldsymbol{\epsilon_\theta}(\boldsymbol{x}_t, t)|], \tag{10}$$

where the mentioned impact on optimization is reduced as the coefficient is amplified by $\alpha(t-k)$. When k is limited to 1, the error between the mapping result $\boldsymbol{f_\theta}(\boldsymbol{x}_t, t)$ and the trajectory origin $\boldsymbol{x}_0$ can be quantified by the accumulation of incremental approximated errors [41]: $|\boldsymbol{x}_0 - \boldsymbol{f_\theta}(\boldsymbol{x}_t, t)| \leq \sum_{1 \leq t' \leq t} |\boldsymbol{f_\theta}(\boldsymbol{x}_{t'}, t') - \boldsymbol{f_\theta}(\boldsymbol{x}_{t'-1}, t'-1)|$. Ideally if the error of one single approximation can be bounded, we can reduce the cumulative error by decreasing the number of approximations. This technique, also called SKIPPING-STEP in LCM [24], extends to optimize the error for skipping sampled points on the trajectories as $|\boldsymbol{f_\theta}(\boldsymbol{x}_{t'}, t') - \boldsymbol{f_\theta}(\boldsymbol{x}_{t'-k}, t' - k)|$ for a fixed skipping step size k. However, our insights reveal this precondition does not hold for extended situations. Denote $\{\boldsymbol{x}_{t'}\}_{t' \in [t-k,t]}$ as the *sub-path* between $\boldsymbol{x}_{t-k}$ and $\boldsymbol{x}_t$ from the original PF-ODE trajectory, we discern that the learning objective in Eq. (10) for $\boldsymbol{\epsilon_\theta}(\boldsymbol{x}_t, t)$ can be decomposed into two complementary components: 1) $dist_\Delta(\boldsymbol{x}_{t-k}, \boldsymbol{x}_t, t) = \boldsymbol{x}_t - \frac{\alpha(t)}{\alpha(t-k)}\boldsymbol{x}_{t-k}$, which gauges the incremental distance from $\boldsymbol{x}_{t-k}$ to $\boldsymbol{x}_t$ attributable to the drift and diffusion processes, and 2) $dist_{0,\boldsymbol{\theta}}(\boldsymbol{x}_{t-k}, t - k, t) = \frac{\alpha(t)}{\alpha(t-k)}\sigma(t - k)\boldsymbol{\epsilon_\theta}(\boldsymbol{x}_{t-k}, t - k)$, which captures the denoising contribution from previous time steps that should be coherently propagated to subsequent time steps t. Thus we rewrite Eq. (10) as a *sub-path learning objective*:

$$\mathcal{L}_{\text{Sub-p}}(\boldsymbol{\theta}, k) = \mathbb{E}[|dist_\Delta(\boldsymbol{x}_t, \boldsymbol{x}_{t-k}, t) + dist_{0,\boldsymbol{\theta}}(\boldsymbol{x}_{t-k}, t - k, t) - \sigma(t)\boldsymbol{\epsilon_\theta}(\boldsymbol{x}_t, t)|]. \tag{11}$$

In Eq. (11), the learning process of $dist_\Delta$ equates to modeling the denoising distribution $p(\boldsymbol{x}_{t-k}|\boldsymbol{x}_t)$, which deviates from Gaussian for larger skipping step sizes and is found to be intractable to estimate [13, 21, 22, 45, 46]. Consequently, the approximated error escalates uncontrollably with increased k due to reliance on the flawed learning. Although LCM sets an empirical k of 20 to balance the pros and cons, the fundamental issues remain unaddressed and unexplored.

4.2   Sub-Path Linear Approximation Model

To improve the learning objective in Eq. (11), in this paper we introduce a new approach for accelerating diffusion models termed **S**ub-**p**ath **L**inear **A**pproximation **M**odel (SPLAM). SPLAM introduces Sub-Path Linear (SL) ODEs to approximate the sub-paths on the PF-ODE trajectories as a linear interpolation between the according sub-path endpoints. As the optimization based on such SL-ODEs gives a respectively progressive and continuous estimation for the decomposed two terms in Eq. (11), our SPLAM is trained based on the conducted SL-ODE learning objectives, and achieves smaller overall prediction errors and better generation quality. We also develop an efficient distillation procedure for latent diffusion models [31], with *Multiple Estimation* strategy which improves the estimated results of teacher models.

Sub-Path Linear ODE Based on the above analysis, in this paper, we introduce *Sub-Path Linear* (SL) ODEs to model approximated sub-paths in the original PF-ODE trajectories, which gives a progressive estimation for $dist_\Delta$. For a sampled sub-path $\{\boldsymbol{x}_{t'}\}_{t'\in[t-k,t]}$ on a solution trajectory dictated by Eq. (3), we interpolate a linear path from $(\boldsymbol{x}_{t-k}, t-k)$ to $(\boldsymbol{x}_t, t)$, guided by the vector direction of $dist_\Delta(\boldsymbol{x}_t, \boldsymbol{x}_{t-k}, t)$. To distinguish the impacts of $dist_\Delta$ and $dist_{0,\boldsymbol{\theta}}$, we account for the drift component in the linear approximated path, causing a shift on coefficient from $(\boldsymbol{x}_{t-k}, t-k)$ to $(\frac{\alpha(t)}{\alpha(t-k)}\boldsymbol{x}_{t-k}, t-k)$. The points on the approximated path $\{\boldsymbol{x}_{\gamma,t}\}_{\gamma\in[0,1]}$ are thus computed as:

$$
\begin{aligned}
\boldsymbol{x}_{\gamma,t} &= \frac{\alpha(t)}{\alpha(t-k)}\boldsymbol{x}_{t-k} + \gamma * dist_\Delta(\boldsymbol{x}_t, \boldsymbol{x}_{t-k}, t) \\
&= (1-\gamma)\frac{\alpha(t)}{\alpha(t-k)}\boldsymbol{x}_{t-k} + \gamma\boldsymbol{x}_t,
\end{aligned}
\tag{12}
$$

for a sampled $(\boldsymbol{x}_{t-k}, t-k)$ and $(\boldsymbol{x}_t, t)$.

Since $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-k}$ conform to distributions governed by the PF-ODE, our linear transformation effectively defines a linear ODE from distribution $\frac{\alpha(t)}{\alpha(t-k)}\boldsymbol{x}_{t-k} \sim p_{t-k,k}(\boldsymbol{x}_{t-k})$ to $\boldsymbol{x}_t \sim p_t(\boldsymbol{x}_t)$ over γ, where $p_{t,k}(\boldsymbol{x}_t)$ has the property $p_{t,k}(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\alpha(t+k)\boldsymbol{x}_0, \left[\frac{\alpha(t+k)\sigma(t)}{\alpha(t)}\right]^2 I)$:

$$
d\boldsymbol{x}_{\gamma,t} = [\gamma * dist_\Delta(\boldsymbol{x}_t, \boldsymbol{x}_{t-k}, t)]d\gamma.
\tag{13}
$$

We denote it as *Sub-Path Linear* (SL) ODE. To apply the approximation strategy on the SL-ODE, the Denoiser and generation function replacing $\boldsymbol{x}_t$ with $\boldsymbol{x}_{\gamma,t}$ are given by:

$$
\begin{aligned}
\boldsymbol{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_{\gamma,t}, \gamma, t) &= \boldsymbol{x}_{\gamma,t} - \sigma(\gamma, t)\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_{\gamma,t}, \gamma, t), \\
\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_{\gamma,t}, \gamma, t) &= \frac{\boldsymbol{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_{\gamma,t}, \gamma, t)}{\alpha(t)}.
\end{aligned}
\tag{14}
$$

Incorporating these into Eq. (11), we derive the sub-path learning objective for our SPLAM model as :

$$
\mathcal{L}_{\text{SPLAM}}(\boldsymbol{\theta}, k) = \mathbb{E}[\|\gamma * dist_\Delta(\boldsymbol{x}_t, \boldsymbol{x}_{t-k}, t) + dist_{0,\boldsymbol{\theta}}(\boldsymbol{x}_{t-k}, t, t-k) - \sigma(\gamma, t)\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_{\gamma,t}, \gamma, t)\|],
\tag{15}
$$

which gives a progressive estimation for the otherwise intractable $dist_\Delta$ objective. The value for $\sigma(\gamma, t)$ can be precisely estimated from the distribution $p_t(\boldsymbol{x}_t)$ and $p_{t-k}(\boldsymbol{x}_{t-k})$ but has a complex expression. Empirically we utilize an approximate result as $\sigma(\gamma, t) = (1-\gamma)\frac{\alpha(t)}{\alpha(t-k)}\sigma(t-k) + \gamma * \sigma(t)$. Compared to consistency models which adopt Eq. (10) or Eq. (11), our $\mathcal{L}$ maintains a progressive estimation for $dist_\Delta$ and a consistent estimation for $dist_{0,\boldsymbol{\theta}}$, which enables the learning for large skipping step size. The overall prediction error can still be assessed by the aggregate of approximated errors between the sub-path endpoints and the approximated error between these points is continuously optimized through the SL-ODEs. Consequently, the optimization for the approximated errors in our SPLAM could be significantly improved. Our approach could further benefit from the increased skipping step size, allowing our method to generate images of higher quality with reduced sampling steps in more efficient training.

Sub-Path Linear Approximation Distillation In this paper, we adopt pre-trained Stable Diffusion (SD) models [31] to obtain the solution PF-ODE trajectories which we build our SL-ODEs upon, and we call the approach **S**ub-**p**ath **L**inear **A**pproximation **D**istillation (SPLAD). To achieve conditional generation with the conditional input c, the noise prediction model is parameterized as $\epsilon_{\boldsymbol{\theta}}(\boldsymbol{z}_t, c, t)$ [21, 43]. We also introduce γ into the prediction models for solving our SL-ODEs, and leverage the $\gamma-$conditioned training where γ is converted to Fourier embeddings and fed into the models as an input. Specifically, to predict $\boldsymbol{z}_0$ in the latent space, the generation function for SPLAM is defined as:

$$\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{z}_{\gamma,t}, c, \gamma, t) = c_{\text{skip}}(t)\boldsymbol{z}_{\gamma,t} + c_{\text{out}}(t)\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}_{\gamma,t}, c, \gamma, t), \tag{16}$$

where $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}_{\gamma,t}, c, \gamma, t)$ mirrors Eq. (14) while replacing $\epsilon_{\boldsymbol{\theta}}(\boldsymbol{z}_{\gamma,t}, \gamma, t)$ with the conditional form $\epsilon_{\boldsymbol{\theta}}(\boldsymbol{z}_{\gamma,t}, c, \gamma, t)$. The functions c_{skip} and c_{out} are leveraged to ensure that $\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{z}_{1,0}, c, 1, 0) \equiv \boldsymbol{z}_0$ (we regard $F_{\boldsymbol{\theta}}$ as the same expression of $f_{\boldsymbol{\theta}}$ since $c_{\text{skip}}(t) \ll c_{\text{out}}(t)$ for most time steps). Integrating this with Eq. (9), our SPLAD approach minimizes the following objective:

$$\mathcal{L}_{\text{SPLAD}}(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \phi) = \mathbb{E}_{\boldsymbol{z}_0 \sim p_{data}, t \sim \mathcal{U}[k,T], \gamma \sim \mathcal{U}[0,1]}[|\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{z}_{\gamma,t}, c, \gamma, t) - \boldsymbol{F}_{\boldsymbol{\theta}^-}(\hat{\boldsymbol{z}}^\Phi_{1,t-k}, c, 1, t-k)|], \tag{17}$$

where $\mathcal{U}$ denotes the uniform distribution, and k is a pre-determined skipping step size. The $\alpha(t)$ in Eq. (9) is omitted due to its negligible effect on optimization in practice. The term $\hat{\boldsymbol{z}}^\Phi_{1,t-k} = \hat{\boldsymbol{z}}^\Phi_{t-k}$ is estimated using ODE solvers $\Phi(\cdots; \phi)$ derived from teacher models. In this paper DDIM [39] is employed as our choice from the advanced solvers of LDMs. Moreover, to improve the estimation of $\hat{\boldsymbol{z}}^\Phi_{t-k}$, we apply the *Multiple Estimation* which executes the solver $\Phi(\cdots, \phi)$ multiple times with a reduced skipping step size k_ϕ. Denoting $t_{\phi,i} = t - i * k_\phi$ and initializing $\hat{\boldsymbol{z}}^\Phi_{t_{\phi,0}} = \boldsymbol{z}_t$, the multiple estimation is iteratively executed as:

$$\hat{\boldsymbol{z}}^\Phi_{t_{\phi,i+1}} = \hat{\boldsymbol{z}}^\Phi_{t_{\phi,i}} + w\Phi(\hat{\boldsymbol{z}}^\Phi_{t_{\phi,i}}, t_{\phi,i}, t_{\phi,i+1}, c; \phi) + (1-w)\Phi(\hat{\boldsymbol{z}}^\Phi_{t_{\phi,i}}, t_{\phi,i}, t_{\phi,i+1}, \emptyset; \phi), \tag{18}$$

for $i = 0, 1, 2, ...; i \le \frac{k}{k_\phi} - 1$, where $\emptyset$ denotes no conditional inputs and w is a fixed guidance scale which controls the effect of conditional generation [10] from

---

Algorithm 1 Sub-Path Linear Approximation Distillation (SPLAD)

Input: dataset $\mathcal{D}$, initial model parameter $\boldsymbol{\theta}$, learning rate η, EMA decay rate μ, ODE solver $\Phi(\cdot, \cdot; \phi)$, distance estimation $|\cdot|$, a fixed guidance scale w, step size k, VAE encoder $\mathcal{E}(\cdot)$, noise schedule $\alpha(t), \sigma(t)$

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

repeat

    sample $(x, c) \sim \mathcal{D}, t \sim \mathcal{U}[k, T]$ and $\gamma \sim \mathcal{U}[0, 1]$

    convert x into latent space: $z = \mathcal{E}(x)$

    sample $\boldsymbol{z}_t \sim \mathcal{N}(\alpha(t)z, \sigma(z)^2 I)$

    $\hat{\boldsymbol{z}}_{t_{\phi,0}}^{\Phi} \leftarrow \boldsymbol{z}_t$, $i \leftarrow 0$

    repeat

        $\hat{\boldsymbol{z}}_{t_{\phi,i+1}}^{\Phi} \leftarrow \hat{\boldsymbol{z}}_{t_{\phi,i}}^{\Phi} + w\Phi(\hat{\boldsymbol{z}}_{t_{\phi,i}}^{\Phi}, t_{\phi,i}, t_{\phi,i+1}, c; \phi) + (1-w)\Phi(\hat{\boldsymbol{z}}_{t_{\phi,i}}^{\Phi}, t_{\phi,i}, t_{\phi,i+1}, \emptyset; \phi)$

        $i \leftarrow i + 1$

    until $k = i * k_\phi$

    $\boldsymbol{z}_{\gamma,t} \leftarrow (1-\gamma) * \frac{\alpha(t)}{\alpha(t-k)}\hat{\boldsymbol{z}}_{i-k}^{\Phi} + \gamma * \boldsymbol{z}_t$ ▷ Sample a point on the SL-ODE.

    $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \phi) \leftarrow |(\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{z}_{\gamma,t}, c, \gamma, t) - \boldsymbol{F}_{\boldsymbol{\theta}}(\hat{\boldsymbol{z}}_{1,t-k}^{\Phi}, c, 1, t-k))|$

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

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

until convergence

---

the conditional input c. The pseudo-code for SPLAD is presented in Algorithm 1. SPLAD shares a similar training pipeline with consistency models [24, 41] but can be distinguished as it optimizes the sub-path learning objectives based on the SL-ODEs and utilizes the γ-conditioned training. For a pair of input noise and time step $(\boldsymbol{z}_t, t)$, SPLAM gives the prediction of the denoised latent $\hat{\boldsymbol{z}}_0$ as:

$$\hat{\boldsymbol{z}}_0 = \boldsymbol{F}_{\boldsymbol{\theta}^-}(\boldsymbol{z}_{1,t}, c, 1, t), \tag{19}$$

for one-step generation, adhering strictly to the $\gamma = 1$ condition. We also use the same iterative sample strategy as illustrated in [41] which could improve the quality of the generated images. In practice, we set the γ-embedding to $\boldsymbol{0}$ for $\gamma = 1$, thereby allowing the weights associated with trained γ-embeddings to be discarded post-training. Thus our Sub-Path Linear Approximation Model (SPLAM) necessitates no additional parameters beyond the training phase and can be utilized the same as teacher models.

5   Experiments

In this section, we conduct experiments to examine the performance of our proposed Sub-Path Linear Approximation Model (SPLAM). Firstly, we describe the experiment configuration and implementation details, and evaluate our models comprehensively on the text-to-image task (Sec. 5.1). Secondly, we verify the effectiveness of our algorithm design through detailed ablation studies (Sec. 5.2). Finally, we present the qualitative results of our SPLAM. (Sec. 5.3).

Table 1: Quantitative results for SDv2.1-base with $w = 8$. The results of DDIM, DPM, DPM++ and LCM* for LAION test-set are derived from [24]. LCM (fix w) is our reproduction conducted as stated in the paper. The results on COCO-30k are evaluated by us.

Methods	LAION-Aesthetics-6+						COCO-30k					
	FID($\downarrow$)			CLIP-Score($\uparrow$)			FID($\downarrow$)			CLIP-Score($\uparrow$)		
	1 Step	2 Steps	4 Steps	1 Step	2 Steps	4 Steps	1 Step	2 Steps	4 Steps	1 Step	2 Steps	4 Steps
DDIM [39]	183.29	81.05	22.38	6.03	14.13	25.89	431.26	229.44	32.77	2.88	7.72	28.76
DPM Solver [21]	185.78	72.81	18.53	6.35	15.10	26.64	206.37	73.87	22.04	10.56	22.87	31.18
DPM Solver++ [22]	185.78	72.81	18.43	6.35	15.10	26.64	206.35	73.82	22.11	10.57	22.87	31.16
LCM* [24]	35.36	13.31	11.10	24.14	27.83	28.69	-	-	-	-	-	-
LCM (fix w) [24]	32.41	12.17	10.43	26.99	30.13	30.76	43.87	15.71	14.88	27.66	31.07	31.52
SPLAM	32.64	**12.06**	**10.09**	**27.13**	**30.18**	**30.76**	**40.52**	**14.59**	**13.81**	**27.83**	31.00	31.45

5.1   Text-to-Image Generation

Experimental Configuration On text-to-image generation task, we train two models with pre-trained Stable Diffusion-V1.5 (SDv1.5) and Stable Diffusion-V2.1-base (SDv2.1-base) as teacher models respectively. Following the setting of [24], the training dataset is one subset of LAION-5B [36]: LAION-Aesthetics-6+. We choose DDIM-Solver as the ODE solver ϕ at skipping step $k_\phi = 20$.

For evaluation, we adopt the commonly used FID and CLIP Score metrics. The results are reported on both SDv1.5 and SDv2.1-base backbones, thus verifying the generalizability of our method. For the experiment of distilling SDv2.1-base, we bench-mark our model on two test sets, LAION-Aesthetics-6+ as used in LCM [24] and MSCOCO2014-30k for zero-shot generalization. We also reproduce a SDv2.1-base LCM according to the training configuration outlined in [24] while replacing the w-condition with the fixed guidance scale, which has also improved the performance. We generally set the guidance scale for distilling SDv2.1-base to 8 and skipping step size to 20, which is consistent with [24]. For the experiment of distilling SDv1.5, we compare our model with state-of-the-art generative models including foundation diffusion models, GANs, and accelerated diffusion models. The guidance scale is set to 3 to obtain the optimal FID, and we adopt the huber [40] loss for our SPLAD metric. The skipping step size is set to 100 for SPLAM which has shown fast convergence. We examine our method on two commonly used benchmarks, MSCOCO2014-30k and MSCOCO2017-5k. More implementation details are provided in the supplementary materials.

Main Results The results for SDv2.1-base are presented in Tab. 1, we use DDIM [39], DPM [21], DPM++ [22] and LCM [24] as baselines. It reveals that our SPLAM surpasses baseline methods nearly across both test sets, at each step, and on both FID and CLIP Score metrics. We suppose that the close results on LAION are caused by overfitting, since the test set and train set are sourced from the same data collection. For SDv1.5 under the guidance scale $w = 3$, the quantitative results are demonstrated in Tab. 2a and Tab. 2b. Our model with 4 steps gets FID-30k of 10.06 and FID-5k of 20.77, which outperforms

Table 2: Quantitative results for SDv1.5. Baseline numbers are cited from [47] and [46]. All the results of LCM are our reproduction whose performance is aligned as stated in the paper. [†] Results are evaluated by us using the released models.

(a) Results on MSCOCO2014-30k, $w = 3$.

Family	Methods	Latency($\downarrow$)	FID($\downarrow$)
Unaccelerated	DALL-E [30]	-	27.5
	DALL-E2 [29]	-	10.39
	Parti-750M [48]	-	10.71
	Parti-3B [48]	6.4s	8.10
	Parti-20B [48]	-	7.23
	Make-A-Scene [5]	25.0s	11.84
	Muse-3B [4]	1.3	7.88
	GLIDE [27]	15.0s	12.24
	LDM [31]	3.7s	12.63
	Imagen [32]	9.1s	7.27
	eDiff-I [1]	32.0s	6.95
GANs	LAFITE [51]	0.02s	26.94
	StyleGAN-T [35]	0.10s	13.90
	GigaGAN [12]	0.13s	9.09
Accelerated Diffusion	DPM++ (4step) [22]	0.26s	22.36
	UniPC (4step) [49]	0.26s	19.57
	LCM-LoRA (4step) [25]	0.19s	23.62
	InstaFlow-0.9B [20]	0.09s	13.10
	InstaFlow-1.7B [20]	0.12s	11.83
	UFOGen [46]	0.09s	12.78
	DMD [47]	0.09s	11.49
	LCM (2step) [24]	0.12s	14.29
	SPLAM (2step)	0.12s	12.31
	LCM (4step) [24]	0.19s	10.68
	SPLAM (4step)	0.19s	**10.06**
Teacher	SDv1.5 [31][†]	2.59s	8.03

(b) Results on MSCOCO2017-5k, $w = 3$.

Methods	#Step	Latency($\downarrow$)	FID($\downarrow$)
DPM Solver++ [22][†]	4	0.21s	35.0
	8	0.34s	21.0
Progressive Distillation [33]	1	0.09s	37.2
	2	0.13s	26.0
	4	0.21s	26.4
CFG-Aware Distillation [15]	8	0.34s	24.2
InstaFlow-0.9B [20]	1	0.09s	23.4
InstaFlow-1.7B [20]	1	0.12s	22.4
UFOGen [46]	1	0.09s	22.5
LCM [24]	2	0.12s	25.22
	4	0.19s	21.41
SPLAM	2	0.12s	23.07
	4	0.19s	**20.77**

(c) Results on MSCOCO2014-30k, $w = 8$.

Family	Methods	Latency($\downarrow$)	FID($\downarrow$)
Accelerated Diffusion	DPM++ (4step)	0.26s	22.44
	UniPC (4step) [49]	0.26s	23.30
	LCM-LoRA (4step) [25]	0.19s	23.62
	DMD [47]	0.09s	14.93
	LCM (2step) [24] [24]	0.12s	15.56
	SPLAM (2step)	0.12s	14.50
	LCM (4step) [24] [24]	0.19s	14.53
	SPLAM (4step)	0.19s	**13.39**
Teacher	SDv1.5 [31][†]	2.59s	13.05

all other accelerated diffusion models, including flow-based method InstaFlow [20] and techniques that introduce GAN objectives such as UFOGen [46] and DMD [47]. Furthermore, SPLAM showcases commensurate results with state-of-the-art foundation generative models such as DALL-E2 [29]. Even in two steps, SPLAM has achieved a competitive performance of FID-30k 12.31 with parallel algorithms. In practical scenarios, a higher guidance scale w is typically favored to enhance the resultant image quality. Accordingly, we trained our SPLAM with w set to 8 and bench-mark it against a range of advanced diffusion methodologies, as delineated in Tab. 2c. In this regime, SPLAM also demonstrates significant advantages, achieving state-of-the-art performance with a four-step FID-30k of 13.39 which exceeds other models by a large margin and is close to the teacher model. Notably, the FID-30k of our model with only two steps reaches 14.50, surpassing the four-step LCM and DMD. While DMD training consumes over one hundred A100 GPU days, which is more than 16 times our training duration.

5.2   Ablation Study

Skipping Step Size & Training Cost Fig. 2a ablates the skipping step size during training, where we compare SPLAM with or without the multiple estimation strategy (Sec. 4.2) and LCM. We can observe that: 1) Without multiple estimation, when the skipping step size k is increasing, LCM suffers a more drastic decline in performance due to heightened optimization challenges for sub-path
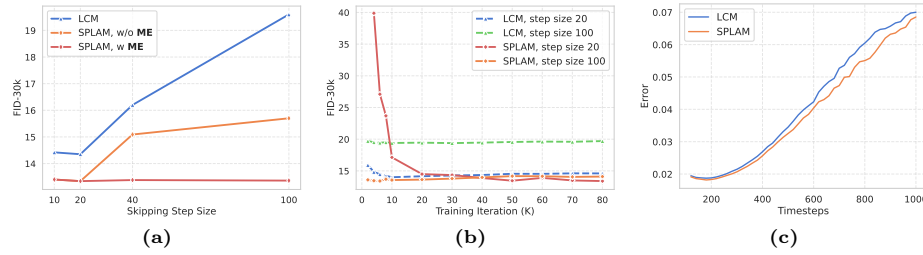
Fig. 2: (a) Ablations on skipping step size and skipping mechanism. **ME** denotes for our Multiple Estimation strategy. (b) Training curve comparing LCM and SPLAM. Our SPLAM with step size 100 is conducted with **ME**, which brings faster convergence. (c) Estimation of the error δ between consistency mapping values of two adjacent points through PF-ODE. SPLAM consistently outperforms LCM in terms of the error.
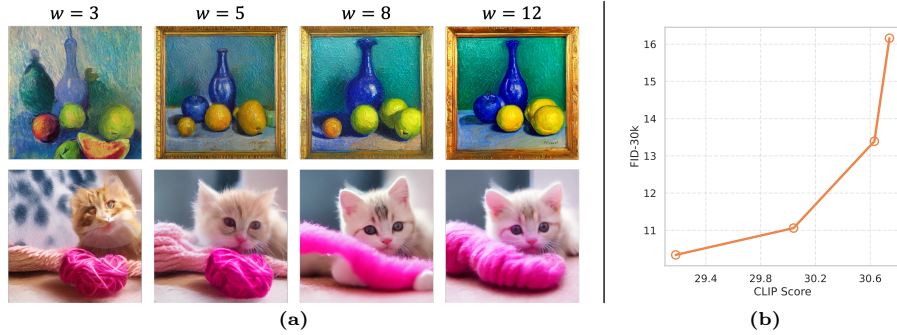

Fig. 3: (a) Visualization for different guidance scale w on SPLAM. (b) The trade-off curve of applying difference guidance scale. w increases from $\{3.0, 5.0, 8.0, 12.0\}$.

learning. Through leveraging our proposed Sub-Path Linear ODE, SPLAM can progressively learn the $dist_\Delta$ and effectively alleviate this collapse. 2) Equipped with the multiple estimation strategy, SPLAM is capable of stably maintaining high image fidelity with large steps. Moreover, we compare the convergence trends between our method and LCM during training, as depicted in Fig. 2b. When $k = 20$, although our metrics initially converge more slowly during the early stages, the performance of our method gradually surpasses LCM by a large margin. It indicates that our training strategy provides a more effective learning objective, enabling SPLAM to achieve a better result, while LCM quickly becomes overfitted. As k raised to 100, larger skipping step size brings SPLAM faster convergence that needs just 2K to 6K iterations which requires about only 6 A100 GPU days training, facilitating practical applications with fewer resources. Note that LCM needs 10k+ iterations for optimal performance which costs about 16 A100 GPU days and can not be applied to larger skipping step size due to the serious performance gap.

Approximated Error Estimation for SPLAM. To illustrate the efficacy of our approach, we directly estimate the denoising mapping error between two ad-
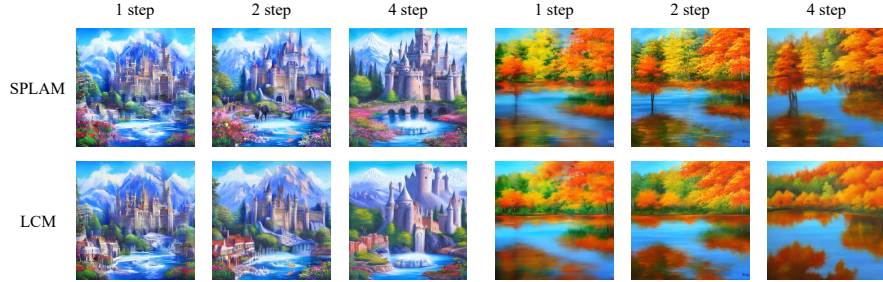
Fig. 4: Comparsion of our SPLAM and LCM [24] in 1,2 and 4-step generation. The results of LCM are based on our reproduction as illustrated in Sec. 5.1. SPLAM has generated consistently higher-quality images that are clearer and more detailed. Noteworthy is the remarkable performance of SPLAM in the 2-step generation, which aligns closely with the 4-step generation results of LCM, highlighting the efficiency and effectiveness of our approach in producing high-fidelity images with fewer generation steps.

jacent samples on the PF-ODE: $\delta(t, k) = \mathbb{E}[|\boldsymbol{f_\theta}(\boldsymbol{x}_{t_{n+k}}, t_{n+k}), \boldsymbol{f_\theta}(\boldsymbol{x}_{t_n}, t_n))|]$, which is firstly defined in Eq. (6). The results are shown in Fig. 2c. We randomly selected 1000 samples from the COCO dataset and simulated adjacent points on the ODE by adding the same noise with adjacent timesteps. We utilize $k = 20$ and the corresponding 50 timesteps for the DDIM scheduler, disregarding steps smaller than 100 due to their relatively larger simulation deviation. It can be seen that, especially at larger timesteps, the error δ of our SPLAM is further reduced (about 10% at $t = 800$). This observation substantiates that SPLAM indeed contributes to minimizing approximated errors, boosting the model's capacity for high-quality image generation.

The Effect of Guidance Scale w. The guidance scale w is a critical hyperparameter in Stable Diffusion [10,31], with its adjustment allowing users to alter the semantic alignment and the quality of the generated image. In this study, we also examine the impact of varying the guidance scale w for our SPLAM based on SDv1.5, which is visualized in Fig. 3. As well as vanilla Stable Diffusion, while a higher w value contributes to better sample quality as reflected by CLIP Scores, it concurrently leads to a degradation in FID performance and oversaturation.

5.3 Qualitative Results

To emphasize the boosted generation quality of our SPLAM, we display the 1,2 and 4-step generation results with the comparison to LCM [24] in Fig. 4. Moreover, we compare our SPLAM distilled from SDv1.5 [31] with the most advanced accelerating diffusion models in Fig. 5, which demonstrate that our SPLAM has achieved the best generation quality across the existing methods.

6   Conclusion

In this paper, we propose a novel approach Sub-Path Linear Approximation Models (SPLAM) for accelerating diffusion models. SPLAM leverages the ap-
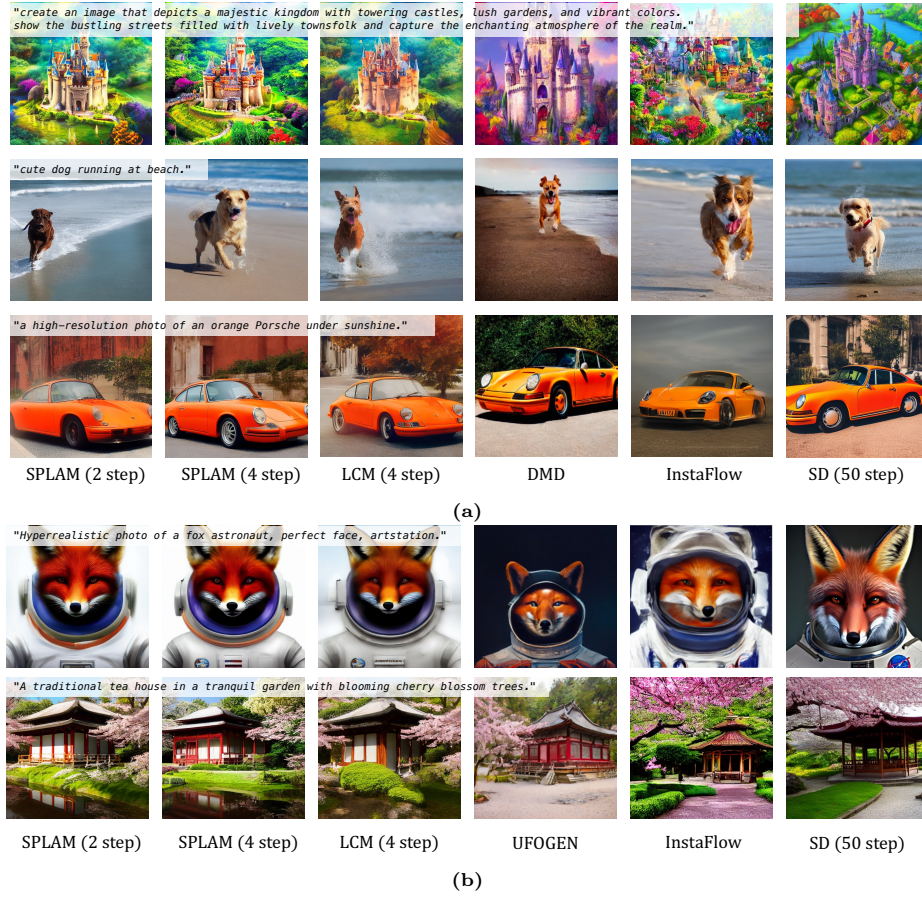
Fig. 5: Qualitative Results. The text prompts are selected from DMD [47] in (a) and UFOGEN [46] in (b), and the results of the two are also cited from respective papers. Clearly, SPLAM demonstrates the best generation quality in 4-step generation except for the SD models. When decreasing the sampling step to 2, SPLAM still maintains a comparable performance, which generates even better results than 4-step LCM [24].

proximation strategy in consistency models and considers the PF-ODE trajectories as a series of interconnected sub-paths delineated by sampled points. Guided by the optimization direction charted by each sub-path, Sub-Path Linear (SL) ODEs also enable our approach to progressively and continuously optimize the approximated learning objectives and thus construct the denoising mappings with smaller cumulative errors. We also develop an efficient distillation procedure for SPLAM to enable the incorporation of latent diffusion models. Extensive experiments on LAION, MS COCO 2014 and MS COCO 2017 datasets have consistently demonstrated the superiority of our method across existing accelerating diffusion approaches in a few-step generation with a fast training convergence.

Acknowledgments

References

1. Balaji, Y., Nah, S., Huang, X., Vahdat, A., Song, J., Kreis, K., Aittala, M., Aila, T., Laine, S., Catanzaro, B., et al.: ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. arXiv preprint arXiv:2211.01324 (2022)
2. Bao, F., Li, C., Zhu, J., Zhang, B.: Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. arXiv preprint arXiv:2201.06503 (2022)
3. Berthelot, D., Autef, A., Lin, J., Yap, D.A., Zhai, S., Hu, S., Zheng, D., Talbot, W., Gu, E.: Tract: Denoising diffusion models with transitive closure time-distillation. arXiv preprint arXiv:2303.04248 (2023)
4. Chang, H., Zhang, H., Barber, J., Maschinot, A., Lezama, J., Jiang, L., Yang, M.H., Murphy, K., Freeman, W.T., Rubinstein, M., et al.: Muse: Text-to-image generation via masked generative transformers. In: ICML (2023)
5. Gafni, O., Polyak, A., Ashual, O., Sheynin, S., Parikh, D., Taigman, Y.: Make-a-scene: Scene-based text-to-image generation with human priors. In: ECCV (2022)
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Nets. In: NIPS (2014)
7. Gu, J., Zhai, S., Zhang, Y., Liu, L., Susskind, J.M.: Boot: Data-free distillation of denoising diffusion models with bootstrapping. In: ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling (2023)
8. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: NeurIPS 2014 Deep Learning Workshop (2015)
9. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
10. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: arXiv preprint arXiv:2207.12598 (2022)
11. Jolicoeur-Martineau, A., Li, K., Piché-Taillefer, R., Kachman, T., Mitliagkas, I.: Gotta go fast when generating data with score-based models. arXiv preprint arXiv:2105.14080 (2021)
12. Kang, M., Zhu, J.Y., Zhang, R., Park, J., Shechtman, E., Paris, S., Park, T.: Scaling up gans for text-to-image synthesis. In: CVPR (2023)
13. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. In: NeurIPS (2022)
14. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
15. Li, Y., Wang, H., Jin, Q., Hu, J., Chemerys, P., Fu, Y., Wang, Y., Tulyakov, S., Ren, J.: Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. Advances in Neural Information Processing Systems **36** (2024)
16. Lin, S., Wang, A., Yang, X.: Sdxl-lightning: Progressive adversarial diffusion distillation. arXiv preprint arXiv:2402.13929 (2024)

17. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
18. Liu, L., Ren, Y., Lin, Z., Zhao, Z.: Pseudo numerical methods for diffusion models on manifolds. In: ICLR (2022)
19. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: ICLR (2023)
20. Liu, X., Zhang, X., Ma, J., Peng, J., Liu, Q.: Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. arXiv preprint arXiv:2309.06380 (2023)
21. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In: NeurIPS (2022)
22. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. In: arXiv preprint arXiv:2211.01095 (2022)
23. Luhman, E., Luhman, T.: Knowledge distillation in iterative generative models for improved sampling speed. arXiv preprint arXiv:2101.02388 (2021)
24. Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H.: Latent consistency models: Synthesizing high-resolution images with few-step inference. arXiv preprint arXiv:2310.04378 (2023)
25. Luo, S., Tan, Y., Patil, S., Gu, D., von Platen, P., Passos, A., Huang, L., Li, J., Zhao, H.: Lcm-lora: A universal stable-diffusion acceleration module. arXiv preprint arXiv:2310.04378 (2023)
26. Meng, C., Rombach, R., Gao, R., Kingma, D., Ermon, S., Ho, J., Salimans, T.: On distillation of guided diffusion models. In: CVPR (2023)
27. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In: ICML (2022)
28. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning. pp. 8162–8171. PMLR (2021)
29. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)
30. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: ICML (2021)
31. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
32. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. In: NeurIPS (2022)
33. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. In: ICLR (2022)
34. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation. arXiv preprint arXiv:2311.17042 (2023)
35. Sauer, A., Schwarz, K., Geiger, A.: Stylegan-xl: Scaling stylegan to large diverse datasets. In: SIGGRAPH (2022)
36. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. In: NeurIPS (2022)
37. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: ICML (2015)

38. Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. Advances in neural information processing systems **28** (2015)
39. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: ICLR (2021)
40. Song, Y., Dhariwal, P.: Improved techniques for training consistency models. arXiv preprint arXiv:2310.14189 (2023)
41. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models. In: ICML (2023)
42. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. In: NeurIPS (2019)
43. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: ICLR (2021)
44. Tong, A., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Fatras, K., Wolf, G., Bengio, Y.: Improving and generalizing flow-based generative models with mini-batch optimal transport. arXiv preprint arXiv:2302.00482 (2023)
45. Xiao, Z., Kreis, K., Vahdat, A.: Tackling the generative learning trilemma with denoising diffusion gans. In: ICLR (2022)
46. Xu, Y., Zhao, Y., Xiao, Z., Hou, T.: Ufogen: You forward once large scale text-to-image generation via diffusion gans. arXiv preprint arXiv:2311.09257 (2023)
47. Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T., Park, T.: One-step diffusion with distribution matching distillation. arXiv preprint arXiv:2311.18828 (2023)
48. Yu, J., Xu, Y., Koh, J.Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B.K., et al.: Scaling autoregressive models for content-rich text-to-image generation. arXiv preprint arXiv:2206.10789 **2**(3), 5 (2022)
49. Zhao, W., Bai, L., Rao, Y., Zhou, J., Lu, J.: Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. arXiv preprint arXiv:2302.04867 (2023)
50. Zheng, H., Nie, W., Vahdat, A., Azizzadenesheli, K., Anandkumar, A.: Fast sampling of diffusion models via operator learning. In: ICML (2023)
51. Zhou, Y., Zhang, R., Chen, C., Li, C., Tensmeyer, C., Yu, T., Gu, J., Xu, J., Sun, T.: Towards language-free training for text-to-image generation. In: CVPR (2022)
52. Zhou, Z., Chen, D., Wang, C., Chen, C.: Fast ode-based sampling for diffusion models in around 5 steps. arXiv preprint arXiv:2312.00094 (2023)