DreamLCM: Towards High Quality Text-to-3D Generation Via Latent Consistency Model

Anonymous Author(s)
Submission Id: 1610

ABSTRACT

Recently, the text-to-3D task has developed rapidly due to the appearance of the SDS method. However, the SDS method always generates 3D objects with poor quality due to the over-smooth issue. This issue is attributed to two factors: 1) the DDPM single-step inference produces poor guidance gradients; 2) the randomness from the input noises and timesteps averages the details of the 3D contents. In this paper, to address the issue, we propose DreamLCM which incorporates the Latent Consistency Model (LCM). DreamLCM leverages the powerful image generation capabilities inherent in LCM, enabling generating consistent and high-quality guidance, i.e., predicted noises or images. Powered by the improved guidance, the proposed method can provide accurate and detailed gradients to optimize the target 3D models. In addition, we propose two strategies to enhance the generation quality further. Firstly, we propose a guidance calibration strategy, utilizing Euler solver to calibrate the guidance distribution to accelerate 3D models to converge. Secondly, we propose a dual timestep strategy, increasing the consistency of guidance and optimizing 3D models from geometry to appearance in DreamLCM. Experiments show that DreamLCM achieves state-of-the-art results in both generation quality and training efficiency.

CCS CONCEPTS

• **Information systems** → **Multimedia content creation**.

KEYWORDS

Text-to-3D Generation, Diffusion Model, Gaussian Splatting, Latent Consistency Model

1 INTRODUCTION

Recent advancements in Diffusion Models (DMs) [34, 35, 37] have made progress in satisfying the needs of synthesizing high-quality images given text descriptions. Besides, by training on large-scale image datasets [38] where images are coupled with detailed texts, DMs achieve powerful ability in generating all kinds of 3D contents conditioned on the given text prompts. Existing works [3, 17, 22, 23, 27, 32, 44, 47] have been proposed to apply well-trained diffusion models to the task of text-to-3D generation. Conditioned on text prompts, DMs can generate guidance information in the latent space. This latent guidance can be utilized to supervise the carving process of the target 3D objects. Thus, this alternative approach tackles the challenge of 3D object generation without large-scale 3D models for training. For example, the Score Distillation Sampling (SDS) objective is introduced in DreamFusion [32] to leverage the robust prior knowledge acquired by text-to-image diffusion models [35, 37]. The SDS backpropagates the gradients from the 2D diffusion model to 3D objects and bridges the gap between the diffusion models and the 3D representations, as shown in Fig. 1(a). The acquired prior knowledge is utilized to optimize the 3D objects represented by Neural Radiance Fields (NeRF) [28] conditioned on a single text prompt.

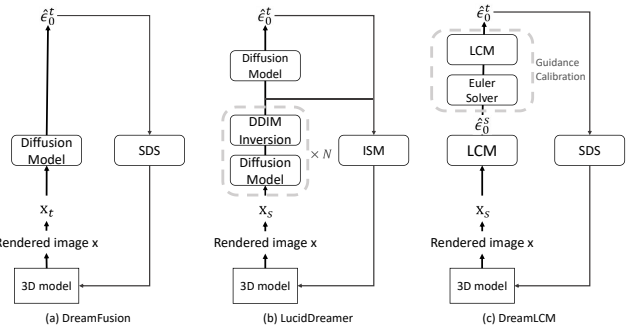

Figure 1: Illustration of different guidance generation approaches. x and $\hat{\epsilon}$ indicates the rendered image and the guidance, respectively. (a) SDS generates guidance via a single diffusion model while producing over-smooth results. (b) LucidDreamer utilizes the DDIM inversion technique, forwarding Diffusion Models multiple times where $N = \{2, 3, 4, 5\}$. (c) The proposed DreamLCM method incorporates LCM as the guidance model. We also propose a guidance calibration strategy that uses Euler Solver to refine the guidance $\hat{\epsilon}_0^s$ to $\hat{\epsilon}_0^t$. Our method generates higher-quality guidance compared to (a) and (b).

However, SDS is limited in generating fine details as it produces over-smooth results. This effect has been noted by previous works [17, 22, 45]. These works attempt to improve SDS and achieve good results in increasing the quality of 3D models. For instance, ProlificDreamer [45] optimizes multiple 3D models simultaneously. These models are mutually benefited and merged by finetuning a LoRA model [12]. The LoRA thus reserves the details of the 3D model. Nevertheless, extra resources are needed to regenerate the lost details. In this paper, we think that the problem of the over-smooth issue stems from two factors. Firstly, the guidance of SDS is derived by the DDPM [9] single-step inference, which leads to *low-quality guidance* and blurred details. Secondly, the rendered images of the target 3D object act as conditions and are fed into DMs

Unpublished working draft. Not for distribution.

ACM MM, 2024, Melbourne, Australia

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

after adding random noises. Besides, DMs need to sample timesteps randomly for the diffusion process. The randomness from both the added noises and timesteps directly results in the *inconsistent guidance* between different iterations. This inconsistency ultimately averages the details of 3D models and leads to the over-smooth issue.

This paper endeavors to tackle the over-smooth issue by handling the factors above accordingly. We propose a novel approach to generate clear guidance for the *low-quality guidance issue*. Inspired by Latent Consistency Model (LCM) [26], we propose **DreamLCM** by incorporating LCM as a guidance model to fully utilize the capability to generate high-quality guidance in a single-step inference. Notably, LCM generates high-quality images in a single-step inference, rather than gradually approaching to the origin along the probably flow ODE (PF-ODE) trajectory [42] by multi-step inference like DDIM [40]. Therefore, DreamLCM merely predicts guidance of a rendered image by directly denoising the noisy latent from an arbitrary timestep along the PF-ODE trajectory to keep fine details of the target 3D models. For the *inconsistency issue*, we observe that LCM generates consistent guidance regardless of the randomness. To solve the issue, a similar method has been proposed in LucidDreamer [22], which uses the DDIM inversion technique [40] to improve the consistency of the guidance. However, different from LucidDreamer, the proposed DreamLCM method provides two merits: 1) DreamLCM only needs a single-step inference to compute the guidance while LucidDreamer forwards the U-Net [36] in DM multiple times; 2) DreamLCM keeps the original SDS loss. Since LCM can resolve the two issues causing the over-smooth issue, there is no need to change the loss forms. On the contrary, LucidDreamer utilizes a complex objective function to adapt the DDIM Inversion method. The difference is shown in Fig. 1(b)(c).

In addition, to further improve generation quality, we propose two novel strategies, i.e., *Guidance Galibration* and *Dual Timestep Strategy*. For *Guidance Calibration*, we propose a two-stage strategy that repeats the perturbing and denoising steps to calibrate the distribution of the guidance. In this way, the disturbing information can be gradually removed. In the first stage, we perturb a rendered image and predict the corresponding guidance. This guidance is consistent with the rendered image, i.e., the 3D object, as the added noise and timestep are small. In the second stage, we run a discretization step of a numerical ODE solver, where we use the Euler Solver to obtain a latent with relatively large noises. The large noises and timestep can provide a more reasonable optimization direction. Consequently, the calibrated guidance is ensured to be consistent with both the rendered image and the highest data density conditioned on the text prompt, effectively improving the guidance's quality. We calculate SDS loss and back-propagate the gradient using the calibrated guidance to optimize the 3D models. For *Dual Timestep Strategy*, we utilize the timestep sampling strategy to enable dreamLCM to optimize the geometry and appearance of 3D objects in separate phases. In particular, in the initial phase, we apply large timesteps to guide the 3D model in producing large deformations. In this case, DreamLCM tends to optimize geometry, where the position of Gaussian Splatting is greatly updated. In the refinement phase, we use small timesteps to optimize the appearance because small timesteps help DreamLCM generate guidance with fine details. Besides, we sample monotonically decreasing timesteps to increase the consistency of the guidance. Overall, our proposed dual timestep strategy is the combination of the timestep strategy in HiFA [49] and ProlificDreamer [45].

We apply the Gaussian Splatting [18] as the 3D representation to form the 3D target objects. The proposed DreamLCM achieves the state-of-the-art results. As shown in Fig. 3, we can see that DreamLCM generates high-quality 3D objects with fine details. Besides, our model trains end-to-end, reducing training costs and maintaining a streamlined training pipeline. Overall, our contributions can be summarized as follows:

- We resolve the over-smooth issue of SDS in a new perspective by proposing DreamLCM. We analyze the two weaknesses in the generated guidance of diffusion models, i.e., low quality and low consistency. In response to the two issues, we incorporate LCM as our guidance model to make full use of the ability in LCM and generate high-quality, consistent guidance in a single inference step.
- We propose two novel strategies to further improve the quality of the guidance for 3D generation. A guidance calibration strategy is proposed, using Euler solver to obtain an improved sample, which subsequently generates calibrated guidance to help 3D models converge accurately. Besides, a dual timestep strategy is proposed, enabling DreamLCM to optimize the geometry and the appearance in two phases. We prove the effectiveness of the two strategies in Sec.6.4.
- We conduct experiments to demonstrate that DreamLCM significantly outperforms the state-of-the-art methods in terms of both quality and training efficiency.

2 RELATED WORK

2.1 Diffusion Models

Diffusion Models(DMs) have emerged as powerful tools for image generation [10, 29, 31, 34, 40, 42], excelling in denoising noise-corrupted data and estimating data distribution scores. The stable ability of DMs for generating high-quality images led to multiple applications in various domains, including video [8, 11, 19] and 3D [32, 44], *etc*. During inference, these models employ reverse diffusion processes to gradually denoise data points and generate samples. In comparison to Variational Autoencoders (VAEs) [20, 39] and Generative Adversarial Networks (GANs) [7], diffusion models offer enhanced training stability and likelihood estimation. However, their sampling efficiency is often hindered. Discretizing reverse-time SDE [6, 42] or ODE [42] are proposed to handle the challenge, various techniques such as ODE solvers [24, 25, 40, 48], adaptive step size solvers [14], and predictor-corrector methods [42] have been proposed. Notably, the Latent Diffusion Model(LDM) [35] conducts forward and reverse diffusion processes in the latent data space, leading to more efficient computation. The Consistency model [26, 41] demonstrates promising potential as a rapid sampling generative model for generating high-quality images in a single-step inference. In this paper, we transfer the ability of LCM to text-to-3D task to generate high-quality guidance. Meanwhile, we use Euler Solver as the numerical ODE Solver to further calibrate the guidance for higher quality.

2.2 Text-to-3D Generation.

This task targets generating 3D contents from given text prompts. The 3D contents are parameterized by various 3D representations, including implicit representations [1, 2, 5, 16], 3D Gaussians [4, 21, 22, 43, 46], *etc*. Existing methods includes 3D generative methods [15, 30]. However, these methods can only generate objects within limited categories due to the lack of large-scale datasets. Our method uses DMs to guide the 3D optimization. DreamField [13] represent pioneering efforts in training Neural Radiance Fields (NeRF) [28] with guidance from CLIP [33]. Dreamfusion [32] firstly employs Score Distillation Sampling (SDS) to distil 3D assets from pretrained text-to-image diffusion models. SDS has become integral to subsequent works, with endeavours aiming at enhancing 3D representations, addressing inherent challenges such as the Janus problem, and mitigating the over-smooth effect observed in SDS. Recent studies like ProlificDreamer [45], HiFA [49], and LucidDreamer [22] have made significant strides in improving the SDS loss. Concurrent methods such as CSD [47] and NFSD [17] provide empirical solutions to enhance SDS. In our novel approach, DreamLCM, we resolve the over-smooth problem in a new perspective of the guidance model, showing that it is possible to generate high-quality 3D models without any alterations to SDS.

3 REVISITING CONSISTENCY MODELS

The core idea of the Consistency Model (CM) and Latent Consistency Model (LCM) is to learn a function that maps any points on a trajectory of PF-ODE [42] to that trajectory origin, i.e., the solution of PF-ODE. The trajectory origin indicates the real data distribution region, which has the highest data density. Besides, LCM extends the denoising process to the latent space. LCM predicts the solution of PF-ODE by introducing a consistency function in a single-step inference. LCM is a text-to-image DM $\boldsymbol{f}_\phi$ parameterized by ϕ. The objective is to fulfill the mapping: $\boldsymbol{f}_\phi(\boldsymbol{x}_t, \boldsymbol{y}, t) \rightarrow \boldsymbol{x}_0$, where $\boldsymbol{x}_t$ is the noisy latent while $\boldsymbol{y}$ is the text prompt. The self-consistency property of LCM is expressed in Eq. (1) as

$$\boldsymbol{f}_\phi(\boldsymbol{x}_t, \boldsymbol{y}, t) = \boldsymbol{f}_\phi(\boldsymbol{x}_{t'}, \boldsymbol{y}, t'), \forall t, t' \in [\delta, T], \tag{1}$$

where δ is a fixed small positive number. The formula shows the consistency of the perturbed images between different timesteps.

Overall, LCM has two benefits: 1) generating $\boldsymbol{f}_\phi(\boldsymbol{x}_t, \boldsymbol{y}, t)$ with high quality in a single-step inference. 2) different $\boldsymbol{x}_t$ between different t generate consistent guidance. We attribute the over-smooth issue in SDS loss to two factors in Sec. 1. The first is the *low-quality guidance* issue, which can be resolved by utilizing LCM to generate high-quality guidance. The second is the *inconsistency issue*. The issue is mitigated because the guidance generated via LCM is consistent between different timesteps. Therefore, we incorporate LCM into the text-to-3D task as the guidance model. The guidance generated by LCM exhibits high quality and high consistency.

4 METHOD

In this section, we present DreamLCM for high-quality text-to-3D synthesis. First, we formulate the entire 3D generation process and analyze the issues in recent works. Then, we propose DreamLCM, Guidance Calibration, and Dual Timestep Strategy. We explain how these methods benefit the generation quality.

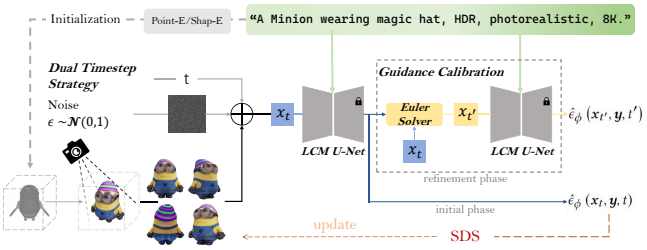

Figure 2: Illustration of DreamLCM. DreamLCM initializes the 3D model θ via text-to-3D generator [15, 30]. We utilize the proposed timestep strategy to divide the training into two phases. In the initial phase, we directly generate guidance via a single LCM network. In the refinement phase, we utilize another LCM network and an Euler Solver to calibrate the guidance. We calculate the original SDS loss to update θ.

4.1 The Problem Definition

Dreamfusion [32] proposes a general framework for the text-to-3D generation task. It has two important components. The first is a 3D representation, e.g., NeRF [28], 3D Gaussian Splatting [18], that is parameterized by θ for depicting the target 3D object Θ. The second is a pretrained text-to-image diffusion model for providing guidance information and supervising the target Θ. To bridge the 3D object and its guidance model, a differentiable renderer $\boldsymbol{g}$ is utilized to obtain the rendered image $\boldsymbol{x}$, which is formulated as $\boldsymbol{x} = \boldsymbol{g}(\theta, c)$ with camera pose c. Then $\boldsymbol{x}$ is fed into a VAE encoder [20] and perturbed by noise ϵ. Here, we denote the latent embedding as $\boldsymbol{x}$ for simplification. Given noisy latent $\boldsymbol{x}_t$, timestep t, and text prompt $\boldsymbol{y}$ as inputs, the guidance model predicts guidance gradients for updating the 3D object. The guidance prediction of DM can be expressed by two equivalent forms, i.e., ϵ-prediction $\hat{\epsilon}_\phi(\boldsymbol{x}_t, \boldsymbol{y}, t)$ and x-prediction $\hat{\boldsymbol{x}}_0^t$ in Eq (2) following SDS [32].

$$\begin{aligned} \nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, \boldsymbol{g}) &= \mathbb{E}_{t,\epsilon,c}\left[\omega(t)\left(\hat{\epsilon}_\phi\left(\boldsymbol{x}_t, \boldsymbol{y}, t\right) - \epsilon\right)\frac{\partial \boldsymbol{g}(\theta,c)}{\partial \theta}\right], \\ &= \mathbb{E}_{t,\epsilon,c}\left[\frac{\omega(t)}{\gamma(t)}\left(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^t\right)\frac{\partial \boldsymbol{g}(\theta,c)}{\partial \theta}\right]. \end{aligned} \tag{2}$$

Here $\hat{\boldsymbol{x}}_0^t = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon}_\phi(\boldsymbol{x}_t, \boldsymbol{y}, t)}{\sqrt{\bar{\alpha}_t}}$, $\sqrt{\bar{\alpha}_t}$ is the noise weight,which shows that $\hat{\epsilon}_\phi(\boldsymbol{x}, \boldsymbol{y}, t)$ and $\hat{\boldsymbol{x}}_0^t$ are equivalent, and we consider them both as guidance. $\omega(t)$ is a weighting function that depends on the timestep t. $\gamma(t) = \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}$. The gradient leads the 3D model closer to the corresponding text prompt.

Previous works ProlificDreamer [45] and LucidDreamer [22] observe that SDS generates over-smooth 3D models. We attribute this issue to two factors: 1) DMs [35, 37] generate low quality guidance because $\hat{\boldsymbol{x}}_0^t$ are obtained from DDPM single-step inference [9], as shown in Fig. 1(a); 2) DMs are sensitive to the randomness in noise ϵ and timestep t. Especially, a large t is hard to generate $\hat{\boldsymbol{x}}_0^t$ consistent with $\boldsymbol{x}_0$, which averages the appearance of the 3D models during optimization. Overall, the weakness in DMs generates poor guidance, resulting in over-smooth outcomes.

4.2 DreamLCM

The proposed DreamLCM method aims at resolving the aforementioned over-smooth issue by incorporating LCM and further enhancing the generation quality by proposing two effective strategies, i.e., Guidance Calibration and a dual timestep strategy. The overall framework is shown in Fig. 2. The entire DreamLCM approach is depicted in Algorithm 1.

For the *low-quality guidance issue*, DreamLCM utilizes the powerful ability of LCM to generate high-quality guidance. LCM trains a function $\boldsymbol{f}_\phi$ in Eq. 1, which can be seen as guidance, to map any $\boldsymbol{x}_t$ to its PF-ODE trajectory origin, i.e.. Consequently, LCM is capable of generating high-quality guidance in a single-step inference.

For the *inconsistency issue*, we utilize the important property of LCM in Eq. 1, highlighting the consistency of guidance between different timesteps. When guided by DMs, rendered image $\boldsymbol{x_0}$ is added random noise ϵ, which is further weighted by different timesteps t. The randomness in ϵ and t directly results in inconsistent $\hat{\boldsymbol{x}}_0^t$, eventually resulting in a *feature-average* outcome. However, due to LCM's property, LCM can generate consistent $\hat{\boldsymbol{x}}_0^t$ regardless of the randomness.

Overall, given a timestep s, we integrate LCM and SDS by generating guidance $\hat{\epsilon}_\phi\left(\boldsymbol{x}_s, \boldsymbol{y}, s\right)$ via LCM to calculate the the SDS loss :

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, \boldsymbol{g}) = \mathbb{E}_{s,\epsilon,c}\left[w(s)\left(\hat{\epsilon}_\phi\left(\boldsymbol{x}_s, \boldsymbol{y}, s\right) - \epsilon\right)\frac{\partial \boldsymbol{g}(\theta, c)}{\partial \theta}\right], \quad (3)$$

where $\hat{\epsilon}_\phi\left(\boldsymbol{x}_s, \boldsymbol{y}, s\right)$ is obtained in a single-step inference with high quality and fine details. It enables the 3D model to have fine details, mitigate the over-smooth issue, and save training costs.

Unfortunately, it is difficult to resolve the two issues perfectly due to the nature of diffusion models. The images generated by LCM with a single-step inference are always blurred, and the high-quality images are derived from four-step inferences iteratively. This fact is also stated in LCM [26]. For the first issue, if we directly utilize LCM's single-step inference results, the guidance would be blurred and not conducive to generating high-quality 3D models. Therefore, we further resolve this weakness by proposing a *guidance calibration* strategy. For the second issue, during the training of LCM, the two noisy latents in Eq. (1) are limited to be on the same PF-ODE trajectory, rather than two arbitrary noisy latents. We follow this protocol during the inference by fixing the noise ϵ' to perturb the rendered image, reducing the randomness in noise. Besides, we propose a decreasing timestep strategy, where we utilize monotonically decreasing timesteps during training, to reduce the randomness in timesteps.

Guidance Calibration. We further dive into the principle when LCM generates guidance. We first review that in DMs, the denoising process follows a reverse stochastic differential equation(SDE):

$$\mathrm{d}\mathbf{x} = -\dot{\sigma}_t \sigma_t \nabla \log p_t(\mathbf{x})\mathrm{d}t + \sqrt{\dot{\sigma}_t \sigma_t}\,\mathrm{d}\mathbf{w}, \quad (4)$$

where $p_t(\boldsymbol{x}_t) \sim \mathcal{N}(\boldsymbol{x}_0, \sigma_t^2\mathbf{I})$, σ_t varies along timestep t, $\dot{\sigma}_t$ is the time derivative of σ_t, $\mathbf{w}$ is the standard Wiener process and $\nabla \log p_t(\mathbf{x})$ is the score function which indicates the direction towards highest data density. And there exists a corresponding reverse ordinary deterministic equation(ODE) defined below:

$$\mathrm{d}\mathbf{x} = -\dot{\sigma}_t \sigma_t \nabla \log p_t(\mathbf{x})\mathrm{d}t. \quad (5)$$

where $\nabla \log p_t(\mathbf{x})$ is estimated as $-\frac{1}{\sqrt{1-\bar{\alpha}_t}}\hat{\epsilon}_0^t$. LCM can map any $\boldsymbol{x}_t$ on a trajectory of the PF-ODE to the origin $\boldsymbol{x}_0^*$, which indicates the highest-data-density region. However, the mapped origin, i.e., $\hat{\boldsymbol{x}}_0^t$ derived from single-step inference is always shifted. We attribute the shifting to the insufficient training of $\boldsymbol{f}_{\boldsymbol{\phi}}$. Our goal is to calibrate $\hat{\boldsymbol{x}}_0^t$ to get closer to $\boldsymbol{x}_0^*$. We consider the insufficient training in LCM and rationally assume that the denoising process of LCM follows a smooth PF-ODE trajectory with a small slope. This assumption allows us to calibrate the guidance from the perspective of PF-ODE.

Based on the analysis, we propose our guidance calibration strategy, which is a two-stage strategy. We repeat the perturbing and denoising steps to calibrate the guidance distribution. In the first stage, given a perturbed sample $\boldsymbol{x}_s$ at timestep s, we first predict guidance $\hat{\epsilon}_\phi(\boldsymbol{x}_s, \boldsymbol{y}, s)$, where s is set to a small number to make the guidance more consistent with $\boldsymbol{x}_0$ than large s. In the second stage, since the denoising process of LCM follows PF-ODE, we run a discretization step of a numerical ODE solver. we use Euler Solver to get another sample $\boldsymbol{x}_t$:

$$\boldsymbol{x}_t = \frac{\sqrt{1-\sigma_s^2}\boldsymbol{x}_s + (\sigma_t - \sigma_s)\hat{\epsilon}_\phi(\boldsymbol{x}_s, \boldsymbol{y}, s)}{\sqrt{1-\sigma_t^2}} \quad (6)$$

where $t > s$. We then fed $\boldsymbol{x}_t$ to LCM network to obtain the final calibrated guidance $\hat{\epsilon}_\phi(\boldsymbol{x}_t, \boldsymbol{y}, t)$. Compared to $\hat{\epsilon}_\phi(\boldsymbol{x}_s, \boldsymbol{y}, s)$, the guidance $\hat{\epsilon}_\phi(\boldsymbol{x}_t, \boldsymbol{y}, t)$ has two advantages: 1) it is consistent with the original rendered image $\boldsymbol{x}_0$ because the Euler solver makes $\boldsymbol{x}_t$ and $\boldsymbol{x}_s$ on the same PF-ODE trajectory; 2) large timestep t provides more reasonable optimization direction conditioned on $\boldsymbol{y}$, leading $\hat{\boldsymbol{x}}_0^t$ closer to $\boldsymbol{x}_0^*$. Overall, $\hat{\boldsymbol{x}}_0^t$ is ensured to be consistent with both the rendered image and the PF-ODE trajectory origin conditioned on the text prompt, effectively improving the quality of the guidance. We optimize θ using the final guidance $\hat{\boldsymbol{x}}_0^t$ to calculate SDS loss following Eq. (7) as

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, \boldsymbol{g}) = \mathbb{E}_{t,\epsilon,c}\left[w(t)\left(\hat{\epsilon}_\phi\left(\boldsymbol{x}_t, \boldsymbol{y}, t\right) - \epsilon\right)\frac{\partial \boldsymbol{g}(\theta, c)}{\partial \theta}\right]. \quad (7)$$

Dual Timestep Strategy. In this paper, we incorporates 3D Gaussians [18] as the 3D representation, which requires initialization models, i.e., PointE [30], Shap-E [15] to initialize the geometry. However, these models sometimes initialize badly, especially when given complex text prompts. Thus, properly updating the geometry positions of the 3D model is crucial in 3D generation. We propose a two-phase strategy, optimizing the geometry and appearance of 3D objects in separate phases. In the initial phase, we use large timesteps to predict large deformations to the 3D model since large timesteps keep less information in the rendered image. As a result, the guidance includes global geometry features, leading DreamLCM to optimize the geometry, where the position of Gaussian Splatting is greatly updated. In the refinement phase, we use small timesteps to optimize the appearance because small timesteps keep more information on the rendered image, generating guidance with fine local features.

We propose a dual timestep strategy combining the decreasing timestep strategy with the two-phase strategy. Specifically, we define a cut-off iteration T_{cut} and a cut-off timestep t_{cut}, in each

Algorithm 1 DreamLCM

Initialization: 3D model parameters θ, training iteration n, LCM network ϕ denoising timestep from N_{min} to N_{max}, cut-off iteration T_{cut} and timestep t_{cut}, text prompt $\boldsymbol{y}$, fixed noise ϵ'.
for $i = [0, ..., n-1]$ **do**
  if $i \leq T_{cut}$ **then**
    $t_{max} \leftarrow N_{max}, t_{min} \leftarrow t_{cut}, t_{interval} \leftarrow T_{cut}, id \leftarrow i$
  else
    $t_{max} \leftarrow t_{cut}, t_{min} \leftarrow N_{min}, t_{interval} \leftarrow n - T_{cut}, id \leftarrow i - t_{cut}$
  end if
  Sample: *camera pose* $c, \boldsymbol{x}_0 = g(\theta, c)$
  $s \leftarrow t_{\max} - (t_{\max} - t_{\min}) \sqrt{id/t_{interval}}, t \leftarrow 2s$
  $\boldsymbol{x}_s \leftarrow \boldsymbol{x}_0 + \sigma_s \epsilon'$
  predict $\hat{\epsilon}_\phi(\boldsymbol{x}_s, \boldsymbol{y}, s)$
  if $i \leq T_{cut}$ **then**
    calculate SDS loss:
    $\nabla_\theta L_{\text{SDS}} = \omega(s) \left(\hat{\boldsymbol{\epsilon}}_\phi \left(\boldsymbol{x}_s, \boldsymbol{y}, s \right) - \epsilon \right)$, update θ.
  else
    use Euler Solver to obtain x_t.
    predict $\hat{\epsilon}_\phi(\boldsymbol{x}_t, \boldsymbol{y}, t)$ then calculate SDS loss:
    $\nabla_\theta L_{\text{SDS}} = \omega(t) \left(\hat{\boldsymbol{\epsilon}}_\phi \left(\boldsymbol{x}_t, \boldsymbol{y}, t \right) - \epsilon \right)$, update θ.
  end if
end for

stage, the timestep is calculated as follows :

$$t = t_{\max} - (t_{\max} - t_{\min}) \sqrt{T/N}, \tag{8}$$

where T and N are the current iteration and total iteration. For the first T_{cut} iterations, we optimize geometry using timesteps larger than t_{cut}. For the remaining iterations, we use timesteps less than T_{cut} to optimize appearance. Overall, we can see that the timestep strategy in HiFA [49] and ProlificDreamer [45] are two special cases of our timestep strategy. Experiments demonstrate that the strategy can generate high-quality 3D models with fine geometry and appearance, as shown in Fig. 5.

5 DISCUSSION

Similar to the proposed DreamLCM method, ProlificDreamer [45] and LucidDreamer [22] targets resolving the over-smooth issue in SDS. They refine the SDS loss with different loss functions to alleviate the over-smoothed and over-saturated results based on SDS. We will revisit these two losses to show the relationship between DreamLCM and the two works and demonstrate that our work is more effective than theirs.

ProlificDreamer is based on the SDS loss. It handles the over-smooth issue by training an additional LoRA [12] network denoted as ϵ_{LoRA}. ProlificDreamer optimizes multiple 3D models simultaneously. It aggregates and estimates their distributions by finetuneing ϵ_{LoRA}. The VSD loss for the i_{th} 3D model is as follows:

$$\nabla_{\theta^{(i)}} \mathcal{L}_{\text{VSD}}(\theta^{(i)}) = \mathbb{E}_{t,\boldsymbol{\epsilon},c} \left[\omega(t) \left(\hat{\epsilon}_\phi \left(\boldsymbol{x}_t^{(i)}, \boldsymbol{y}, t \right) - \hat{\epsilon}_{LoRA} \left(\boldsymbol{x}_t^{(i)}, \boldsymbol{y}, t, c \right) \right) \frac{\partial \boldsymbol{g}(\theta^{(i)}, c)}{\partial \theta^{(i)}} \right], \tag{9}$$

where $\boldsymbol{x}_t^{(i)}$ is the rendered image of the i_{th} 3D model and c is the camera condition. $\hat{\epsilon}_{LoRA} \left(\boldsymbol{x}_t^{(i)}, \boldsymbol{y}, t, c \right)$ indicates the distribution of the rendered image. When ProlificDreamer optimizes one 3D model, the distribution of the rendered image can be estimated as ϵ, where ϵ is the noise added to the rendered image. We consider the SDS gradient as the vector starting from ϵ and $\hat{\epsilon}_{LoRA}$ to $\hat{\epsilon}_\phi$. Since $\hat{\epsilon}_{LoRA}$ contains information from multiple 3D models, $\hat{\epsilon}_{LoRA}$ is a steadier and more robust starting point than ϵ, averaging the random and inconsistent features in the optimization process of each 3D model.

We observe that the essential problem is the randomness and inconsistency when optimizing a single 3D model. Besides, $\hat{\epsilon}_{LoRA}$ introduces extra parameters and trains several 3D models simultaneously, leading to high training costs. However, the proposed DreamLCM method incorporates LCM as the guidance model, greatly mitigating the inconsistent issue when optimizing one 3D model. As a result, there is no need for DreamLCM to train another $\hat{\epsilon}_{LoRA}$ to decrease the training costs for generating high-quality 3D models.

LucidDreamer proposes ISM [22] loss, which employs DDIM Inversion to enhance the quality and consistency of the guidance. Specifically, it predicts a invertible noisy latent trajectory $\{\boldsymbol{x}_{\delta_T}, \boldsymbol{x}_{2\delta_T}, \ldots, \boldsymbol{x}_t\}$, iteratively following Eq. (10),

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t} \hat{\boldsymbol{x}}_0^s + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}_\phi \left(\boldsymbol{x}_s, \emptyset, s \right), \tag{10}$$

where $s = t - \delta t$. The guidance is obtained by a multi-step DDIM denoising process i.e., iterating

$$\tilde{\boldsymbol{x}}_{t-\delta_T} = \sqrt{\bar{\alpha}_{t-\delta_T}} \left(\hat{\boldsymbol{x}}_0^t + \gamma \left(t - \delta_T \right) \boldsymbol{\epsilon}_\phi \left(\boldsymbol{x}_t, \boldsymbol{y}, t \right) \right), \tag{11}$$

where $\eta(t) = \frac{1-\sqrt{\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}$. Next, by replacing $\hat{\boldsymbol{x}}_0^t$ in Eq. (2) with $\tilde{\boldsymbol{x}}_0$, the SDS loss can be rewrote as $\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_c \left[\frac{\omega(t)}{\gamma(t)} \left(\boldsymbol{x}_0 - \tilde{\boldsymbol{x}}_0^t \right) \frac{\partial \boldsymbol{g}(\theta, c)}{\partial \theta} \right]$. LucidDreamer then unifies the iterative process in Eq. (10) and Eq. (11), proposing ISM loss as follows:

$$\begin{aligned} \nabla_\theta \mathcal{L}(\theta) &= \mathbb{E}_{t,c} \left[\frac{\omega(t)}{\gamma(t)} \left(\gamma(t) \left[\boldsymbol{\epsilon}_\phi \left(\boldsymbol{x}_t, \boldsymbol{y}, t \right) - \boldsymbol{\epsilon}_\phi \left(\boldsymbol{x}_s, \emptyset, s \right) \right] + \eta_t \right) \frac{\partial \boldsymbol{g}(\theta, c)}{\partial \theta} \right] \\ &\approx \mathbb{E}_{t,c} \left[\omega(t) \left(\boldsymbol{\epsilon}_\phi \left(\boldsymbol{x}_t, \boldsymbol{y}, t \right) - \boldsymbol{\epsilon}_\phi \left(\boldsymbol{x}_s, \emptyset, s \right) \right) \frac{\partial \boldsymbol{g}(\theta, c)}{\partial \theta} \right]. \end{aligned} \tag{12}$$

where η_t includes a series of neighboring interval scores with opposing scales, which can be disregarded. ISM essentially substitutes DDPM single-step inference [9] for DDIM multi-step inference [40] to generate high-quality and high-fidelity guidance $\tilde{\boldsymbol{x}}_0^t$.

However, the multi-step inference needs to forward the U-Net [36] in DMs multiple times, increasing training costs. Moreover, we can see that a key improvement in ISM is the quality and consistency of the guidance. Compared to LucidDreamer, DreamLCM is capable of generating high-quality and high-consistency guidance in a single-step inference. Consequently, DreamLCM is more effective with fewer training costs.

To sum up, we observe that the principal cause of the over-smooth issue in SDS is the inadequate quality of the guidance. These two methods tackle the issue by utilizing extra resources, e.g., training multiple NeRFs and DDIM Inversions, which is time-consuming. Differently, DreamLCM can resolve the over-smooth issue by taking full advantage of LCM, while saving training costs.

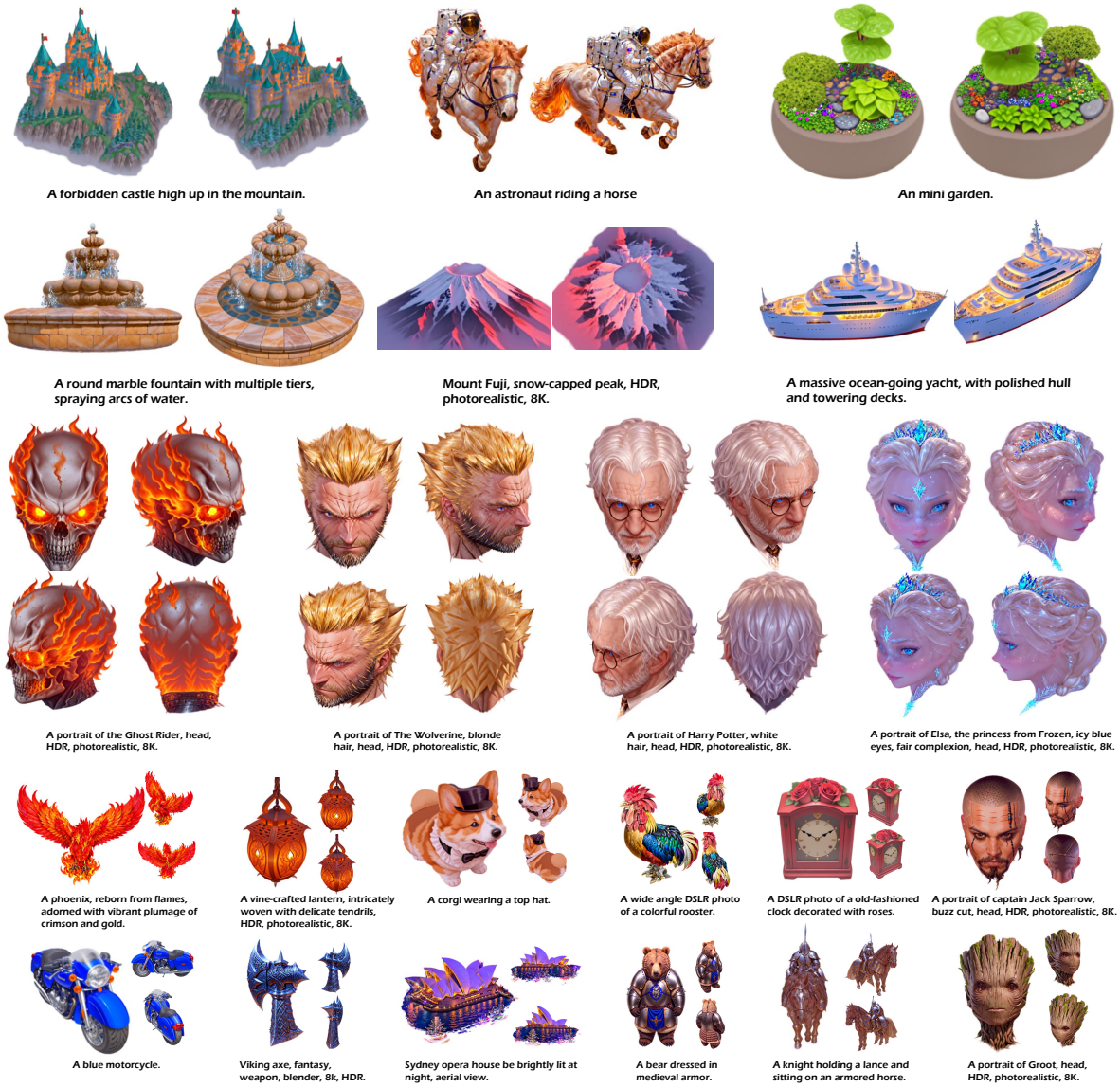

Figure 3: Examples generated by DreamLCM. We incorporate the Latent Consistency Model (LCM) as a guidance model, with two proposed strategies to further enhance the generation quality (See section 4 for details). DreamLCM generates high-quality results with fine details.

6 EXPERIMENTS

6.1 Implementation Details

We train our end-to-end network for 5000 iterations overall. We employ 3D Gaussian Splatting [18] as our 3D representation and 3D point cloud generation models Shap-E [15] and Point-E [30] for initialization. The rendering resolution is 512×512. As for the guidance calibration strategy, we use it in appearance optimization. We practically consider $s = 350$ as the cut-off timestep. Unless stated otherwise, we train the first 1000 iters for geometry optimization using timesteps s fulfilling $350 \leq s \leq 980$ and the remaining 4000 iters for appearance optimization using timesteps s fulfilling $20 \leq s \leq 350$. Since we assume that LCM follows a smooth PF-ODE, the interval between s and t is less limited. Practically, we choose $t = 2s$. We use SDS with a normal CFG scale of 7.5. All experiments are performed and measured with an RTX 3090 (24G) GPU. We train about 50 min per sample.

6.2 Text-to-3D Generation.

In Fig. 4, we show the generated results of DreamLCM. We generate all examples using the original LCM without LoRA and any finetuned checkpoints. We can see that DreamLCM can generate photo-realistic 3D objects with fine details. The 3D objects are creative and highly consistent with the text prompts. Especially, we

Figure 4: Comparison with the state-of-the-art text-to-3D generation methods with Gaussian Splatting as 3D representations. Experiments show that the proposed DreamLCM generates photo-realistic 3D objects with high quality and fine details. The models generated by DreamLCM are more consistent with the text prompt. The training time is measured with a single RTX 3090 GPU.

can see that DreamLCM can generate different and amazing avatar heads conditioned on text prompts, such as *"A portrait of Harry Potter, white hair, head, HDR, photorealistic, 8K"*. Besides, the proposed DreamLCM method is good at generating objects conditioned on complex text prompts, like *"the fuji mountain"*, *"the massive yacht"*, and *"the fountain"*. These examples demonstrate that DreamLCM well resolves the over-smooth issue in SDS. Besides, these examples show great potential in generating all kinds of complex objects with different LCM finetuned checkpoints.

6.3 Qualitative Comparison

We compare our method with the current SoTA baselines which generate 3D Gaussian Objects [22, 43, 46]. As shown in Fig. 4, our model generates more photo-realistic results than other works, exhibiting high quality and fine details. For example, *"A portrait of a unicorn"* is more photo-realistic, and the fur is more silky than the results from the other three approaches. As for the results conditioned on the text *"A Spanish galleon"*, our model generates the most intact galleon, and the details of the hull are the finest

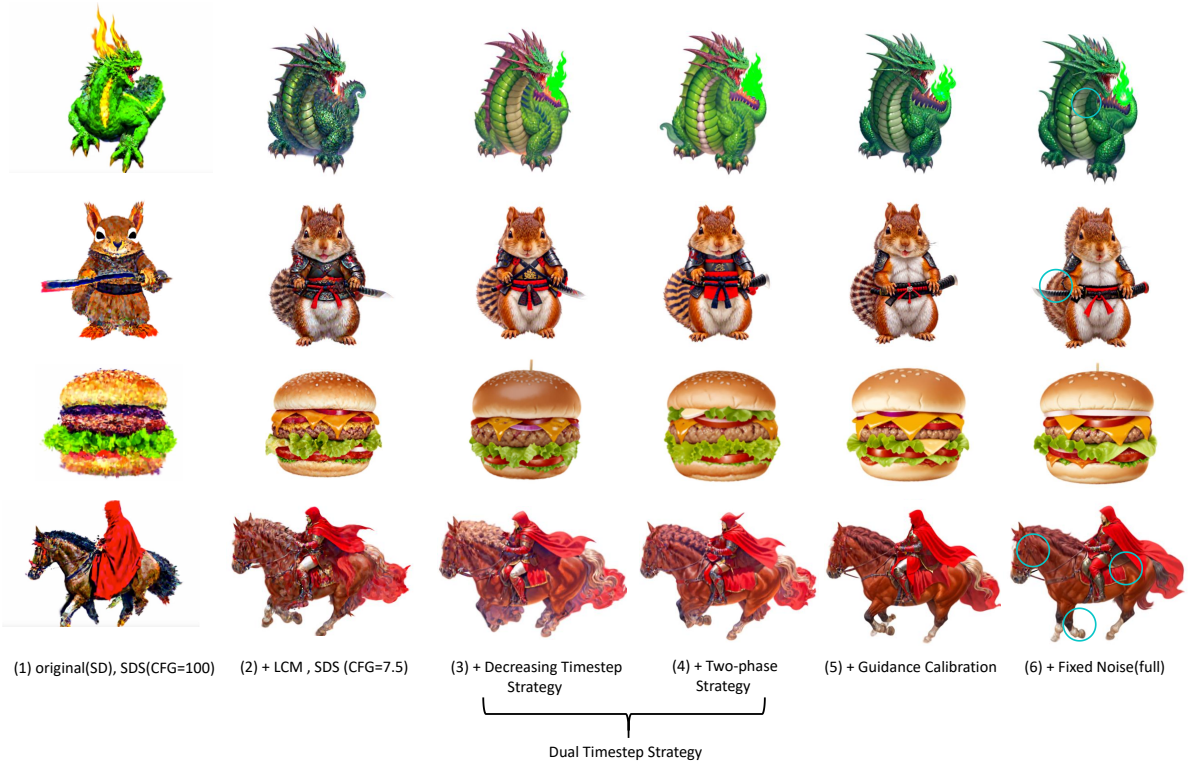

Figure 5: Ablation Study of DreamLCM. The proposed components are effective and can improve the text-to-3D generation quality. (1) The results of SDS with a large CFG scale of 100. (2) We incorporate LCM as a guidance model with a small CFG of 7.5. (3)(4) The results after adding the Dual Timestep Strategy. It includes two parts, the Decreasing Timestep Strategy to reduce the randomness in timesteps and the Two-phases Strategy to improve geometry. Both parts are effective. (5) The results after adding Guidance Calibration to further improve the generation quality. (6) We use fixed noise to perturb the samples to reduce the randomness in noises to improve the details. We highlight some improved details in cyan. The prompts corresponding to the four examples are *"a green dragon breathing fire"*, *"a squirrel in samurai armor wielding a katana"*, *"a delicious hamburger"* and *"A warrior with red cape riding a horse"*.

among all the shown methods. Notably, compared to LucidDreamer, we generate higher quality objects with less training costs.

6.4 Ablation Study

Fig. 5 depicts the ablation experiments of different baseline methods. In Fig. 5(b) and (c), we utilize timesteps between 20 and 500 to generate high-quality images. Other settings are the same as the final settings 6.1. Starting from the original SDS loss, guided by Stable Diffusion [35], with a large CFG scale(100). We first incorporate LCM as the guidance model to demonstrate that LCM is a superior guidance model to DMs [35]. We can see that LCM makes a huge improvement in generation quality. We then add our Dual Timestep Strategy. We divide the strategy into two parts. We demonstrate the effectiveness of Decreasing Time Strategy and the Two-phase Strategy, as shown in Fig. 5(c) and (d). We can see that the hamburger adding Decreasing Time strategy shows a quality improvement. Based on (c), the hamburger adding the two-phase strategy shows a geometric advancement of the bread at the bottom of the hamburger. Besides, due to the two-phase strategy, the worrier example is deformed to be less close to the horse, since this 3D model is initialized by the prompt *"a wolf"*. Then, we add the guidance calibration strategy which smooths the appearance and improves the details, making the objects more photo-realistic. Finally, we add fixed noises to improve the consistency of guidance between different timesteps. As shown in Fig. 5, the ability to improve details is demonstrated in cyan, such as the eye, legs, and cushion on the horse back in the worrier sample, the katana in the squirrel sample and the shadow in the dragon sample.

7 CONCLUSION

In this paper, we propose DreamLCM method to improve the text-to-3D object task. We incorporate LCM as a guidance model to generate high-quality guidance to resolve the two factors that cause the over-smooth issue. Besides, we introduce two techniques, i.e., Guidance Calibration and Dual Timestep Strategy, to further improve the generation quality. Experiments show superior performance of our method. Our method achieves state-of-the-art results in both generation and training efficiency.

REFERENCES

[1] Titas Anciukevičius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J. Mitra, and Paul Guerrero. 2023. RenderDiffusion: Image Diffusion for 3D Reconstruction, Inpainting and Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12608–12618.

[2] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. 2023. Single-Stage Diffusion NeRF: A Unified Approach to 3D Generation and Reconstruction. arXiv:2304.06714

[3] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. 2023. Fantasia3D: Disentangling Geometry and Appearance for High-quality Text-to-3D Content Creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 22246–22256.

[4] Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. 2024. Text-to-3D using Gaussian Splatting. arXiv:2309.16585

[5] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander Schwing, and Liangyan Gui. 2023. SDFusion: Multimodal 3D Shape Completion, Reconstruction, and Generation. arXiv:2212.04493

[6] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. 2022. Score-Based Generative Modeling with Critically-Damped Langevin Diffusion. arXiv:2112.07068

[7] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. arXiv:1406.2661

[8] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. 2022. Imagen Video: High Definition Video Generation with Diffusion Models. arXiv:2210.02303

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. 6840–6851.

[10] Jonathan Ho and Tim Salimans. 2022. Classifier-Free Diffusion Guidance. arXiv:2207.12598

[11] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. 2022. Video Diffusion Models. arXiv:2204.03458

[12] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[13] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. 2021. Zero-Shot Text-Guided Object Generation with Dream Fields. *arXiv* (2021).

[14] Alexia Jolicoeur-Martineau, Ke Li, Remi Piche-Taillefer, Tal Kachman, and Ioannis Mitliagkas. 2021. Gotta Go Fast When Generating Data with Score-Based Models. *ArXiv* abs/2105.14080 (2021).

[15] Heewoo Jun and Alex Nichol. 2023. Shap-E: Generating Conditional 3D Implicit Functions. arXiv:2305.02463

[16] Animesh Karnewar, Niloy J. Mitra, Andrea Vedaldi, and David Novotny. 2023. HoloFusion: Towards Photo-realistic 3D Generative Modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 22976–22985.

[17] Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. 2023. Noise-Free Score Distillation. arXiv:2310.17590

[18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. arXiv:2308.04079

[19] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. 2023. Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators. arXiv:2303.13439

[20] Diederik P Kingma and Max Welling. 2022. Auto-Encoding Variational Bayes. arXiv:1312.6114

[21] Xinhai Li, Huaibin Wang, and Kuo-Kun Tseng. 2023. GaussianDiffusion: 3D Gaussian Splatting for Denoising Diffusion Probabilistic Models with Structured Noise. arXiv:2311.11221

[22] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. 2023. LucidDreamer: Towards High-Fidelity Text-to-3D Generation via Interval Score Matching. arXiv:2311.11284

[23] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2023. Magic3D: High-Resolution Text-to-3D Content Creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 300–309.

[24] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. 2022. DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. *ArXiv* abs/2206.00927 (2022).

[25] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. 2023. DPM-Solver++: Fast Solver for Guided Sampling of Diffusion Probabilistic Models. arXiv:2211.01095

[26] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. 2023. Latent Consistency Models: Synthesizing High-Resolution Images with Few-Step Inference. arXiv:2310.04378

[27] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. 2023. Latent-NeRF for Shape-Guided Generation of 3D Shapes and Textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12663–12673.

[28] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

[29] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. arXiv:2112.10741

[30] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. 2022. Point-E: A System for Generating 3D Point Clouds from Complex Prompts. arXiv:2212.08751

[31] Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved Denoising Diffusion Probabilistic Models. In *Proceedings of the 38th International Conference on Machine Learning*, Vol. 139. 8162–8171.

[32] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2022. DreamFusion: Text-to-3D using 2D Diffusion. *arXiv* (2022).

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020

[34] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents. arXiv:2204.06125

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10684–10695.

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.04597

[37] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In *Advances in Neural Information Processing Systems*, Vol. 35. 36479–36494.

[38] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 2022. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. 25278–25294.

[39] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28.

[40] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising Diffusion Implicit Models. *arXiv:2010.02502* (2020).

[41] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. 2023. Consistency Models. arXiv:2303.01469

[42] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020. Score-Based Generative Modeling through Stochastic Differential Equations. arXiv:2011.13456

[43] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. 2024. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation. arXiv:2309.16653

[44] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. 2023. Score Jacobian Chaining: Lifting Pretrained 2D Diffusion Models for 3D Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12619–12629.

[45] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan LI, Hang Su, and Jun Zhu. 2023. ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation. In *Advances in Neural Information Processing Systems*, A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. 8406–8441.

[46] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu/, Qi Tian, and Xinggang Wang. 2023. GaussianDreamer: Fast Generation from Text to 3D Gaussians by Bridging 2D and 3D Diffusion Models. arXiv:2310.08529

[47] Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. 2023. Text-to-3d with classifier score distillation. *arXiv preprint arXiv:2310.19415* (2023).

[48] Qinsheng Zhang, Molei Tao, and Yongxin Chen. 2023. gDDIM: Generalized denoising diffusion implicit models. arXiv:2206.05564

[49] Junzhe Zhu, Peiye Zhuang, and Sanmi Koyejo. 2024. HiFA: High-fidelity Text-to-3D Generation with Advanced Diffusion Guidance. arXiv:2305.18766

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009