LucidDreamer: Towards High-Fidelity Text-to-3D Generation via Interval Score Matching

Yixun Liang*[1] Xin Yang*[1,2] Jiantao Lin[1] Haodong Li[1] Xiaogang Xu[3,4] Yingcong Chen**[1,2]
[1] HKUST (GZ) [2] HKUST [3] CUHK [4] Zhejiang University
yliang982@connect.hkust-gz.edu.cn xin.yang@connect.ust.hk jlin695@hkust-gz.edu.cn
hli736@connect.hkust-gz.edu.cn xiaogangxu00@gmail.com yingcongchen@ust.hk

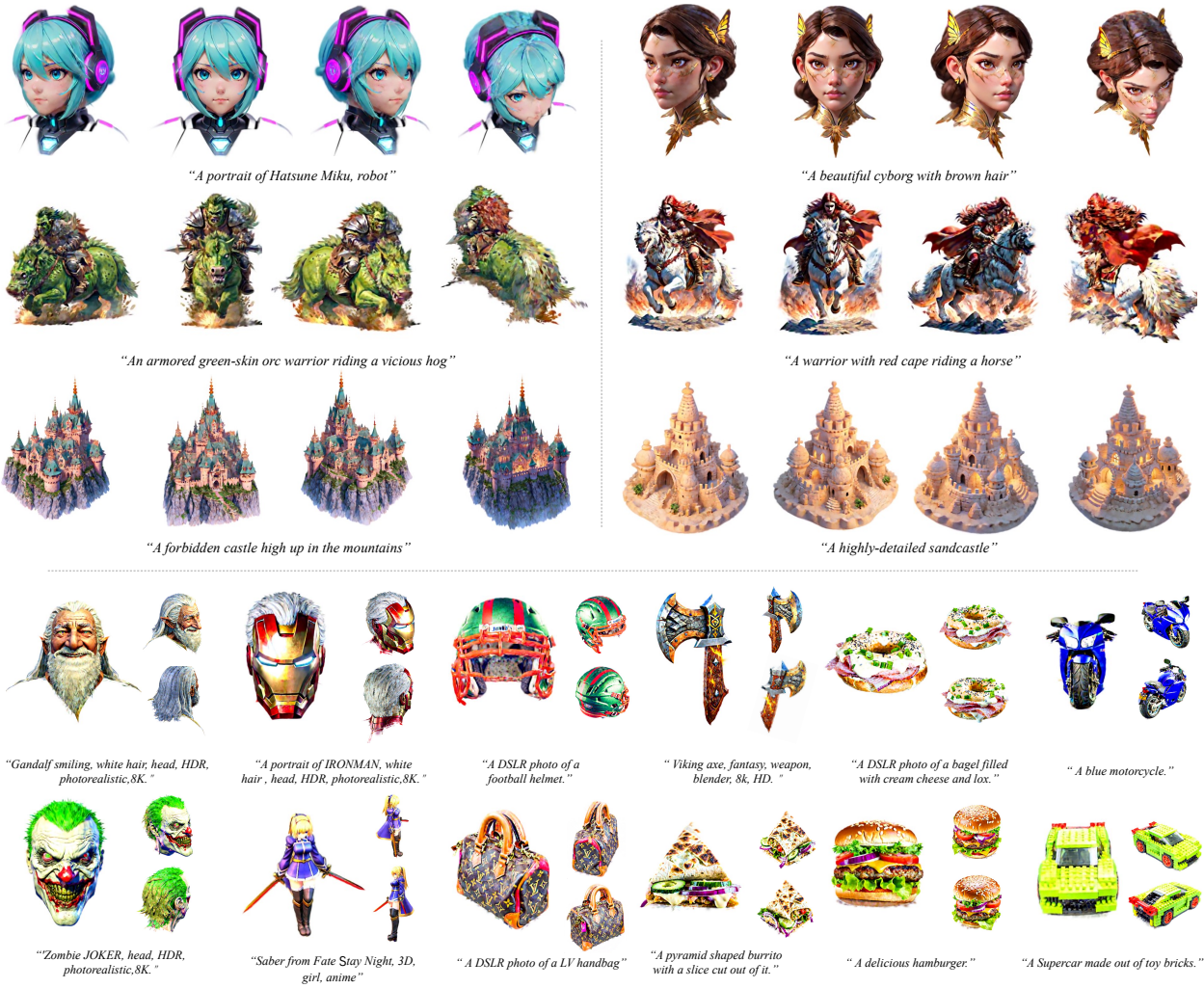

Figure 1. **Examples of text-to-3D content creations with our framework.** We present a text-to-3D generation framework, named the *LucidDreamer*, to distill high-fidelity textures and shapes from pretrained 2D diffusion models (detailed shows on Sec. 4) with a novel **Interval Score Matching** objective and an *Advanced 3D distillation pipeline*. Together, we achieve superior 3D generation results with photorealistic quality in a short training time. Please zoom in for details.

** Corresponding author.
*The first two authors contributed equally to this work.

* *Conceptualization*: **Yixun Liang**: 60%, **Xin Yang**: 40%,
Methodology: **Xin Yang**: 60%, **Yixun Liang**: 40%.

Abstract

The recent advancements in text-to-3D generation mark a significant milestone in generative models, unlocking new possibilities for creating imaginative 3D assets across various real-world scenarios. While recent advancements in text-to-3D generation have shown promise, they often fall short in rendering detailed and high-quality 3D models. This problem is especially prevalent as many methods base themselves on Score Distillation Sampling (SDS). This paper identifies a notable deficiency in SDS, that it brings inconsistent and low-quality updating direction for the 3D model, causing the over-smoothing effect. To address this, we propose a novel approach called Interval Score Matching (ISM). ISM employs deterministic diffusing trajectories and utilizes interval-based score matching to counteract over-smoothing. Furthermore, we incorporate 3D Gaussian Splatting into our text-to-3D generation pipeline. Extensive experiments show that our model largely outperforms the state-of-the-art in quality and training efficiency. Our code is available at: EnVision-Research/LucidDreamer

1. Introduction

Digital 3D asserts have become indispensable in our digital age, enabling the visualization, comprehension, and interaction with complex objects and environments that mirror our real-life experiences. Their impact spans a wide range of domains including architecture, animation, gaming, virtual and augmented reality, and is widely used in retail, online conferencing, education, etc. The extensive use of 3D technologies brings a significant challenge, i.e., generating high-quality 3D content is a process that needs a lot of time, effort, and skilled expertise.

This stimulates the rapid developments of 3D content generation approaches [5, 14, 16, 21–24, 28, 30, 33, 34, 40, 45]. Among them, text-to-3D generation [5, 14, 21, 28, 30, 33, 45, 51] stands out for its ability to create imaginative 3D models from mere text descriptions. This is achieved by utilizing a pretrained text-to-image diffusion model as a strong image prior to supervise the training of a neural parameterized 3D model, enabling for rendering 3D consistent images in alignment with the text. This remarkable capability is fundamentally grounded in the use of Score Distillation Sampling (SDS). SDS acts as the core mechanism that lifts 2D results from diffusion models to the 3D world, enabling the training of 3D models without images [4, 5, 16, 21, 28, 33, 49].

Despite its popularity, empirical observations have shown that SDS often encounters issues such as over-smoothing, which significantly hampers the practical application of high-fidelity 3D generation. In this paper, we thoroughly investigate the underlying cause of this problem. Specifically, we reveal that the mechanism behind SDS is to match the images

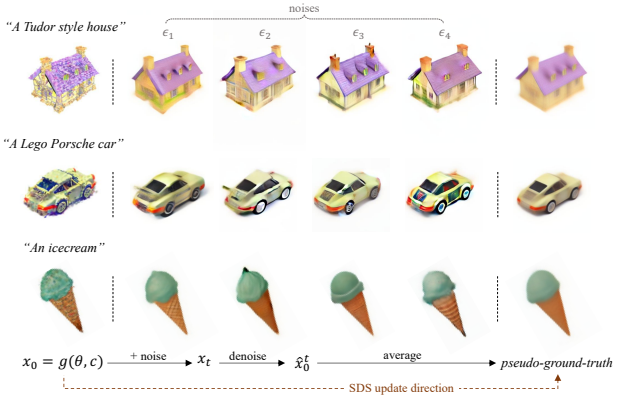

Figure 2. **Examples of SDS [33].** Let $t = 500$, we simulate the SDS distillation process by sampling x_t with same x_0 but different noises $\{\epsilon_1, ..., \epsilon_4\}$. We discover that the SDS distillation process produces overly-smoothed *pseudo-ground-truth* (i.e., $\hat{x}_0^t$) for x_0. First, the random noise and timestep sampling strategy of SDS drives x_0 towards the averaged $\hat{x}_0^t$ and eventually leads to the "feature-averaging" result. Second, SDS exploits the diffusion model for $\hat{x}_0^t$ estimation in one step, which results in low-quality guidance at large timesteps. Please refer to Sec. 3.1 for analysis.

rendered by the 3D model with the pseudo-Ground-Truth (pseudo-GT) generated by the diffusion model. However, as shown in Fig. 2, the generated pseudo-GTs are usually *inconsistent* and have *low visual quality*. Consequently, all update directions provided by these pseudo-GTs are subsequently applied to the same 3D model. Due to the average effect, the final results tend to be over-smooth and lack of details.

This paper aims to overcome the aforementioned limitations. We show that the unsatisfactory pseudo-GTs originated from two aspects. Firstly, these pseudo-GTs are one-step reconstruction results from the diffusion models, which have high reconstruction errors. Besides, the intrinsic randomness in the diffusion trajectory makes these pseudo-GTs semantically variant, which causes an averaging effect and eventually leads to over-smoothing results. To address these issues, we propose a novel approach called Interval Score Matching (ISM). ISM improves SDS with two effective mechanisms. Firstly, by employing DDIM inversion, ISM produces an invertible diffusion trajectory and mitigates the averaging effect caused by pseudo-GT inconsistency. Secondly, rather than matching the pseudo-GTs with images rendered by the 3D model, ISM conducts matching between two interval steps in the diffusion trajectory, which avoids one-step reconstruction that yields high reconstruction error. We show that our ISM loss consistently outperforms SDS by a large margin with highly realistic and detailed results. Finally, we also show that our ISM is not only compatible with the original 3D model introduced in [33], by utilizing a more advanced model – 3D Gaussian Splatting [20], our model achieves superior results compared to the state-of-

the-art approaches, including Magic3D [21], Fantasia3D [5], and ProlificDreamer [45]. Notably, these competitors require multi-stage training, which is not needed in our model. This not only reduces our training cost but also maintains a simple training pipeline. Overall, our contributions can be summarized as follows.

- We provide an in-depth analysis of Score Distillation Sampling (SDS), the fundamental component in text-to-3D generation, and identify its key limitations for providing inconsistent and low-quality pseudo-GTs. This provides an explanation of the over-smoothing effect that exists in many approaches.
- In response to SDS's limitations, we propose the Interval Score Matching (ISM). With invertible diffusion trajectories and interval-based matching, ISM significantly outperforms SDS with highly realistic and detailed results.
- By integrating with 3D Gaussian Splatting, our model achieves state-of-the-art performance, surpassing existing methods with less training costs.

2. Related Works

Text-to-3D Generation. One work can be categorized as text-to-3D generation [2, 5–7, 12, 17, 21, 29, 33, 37, 38, 40, 43, 47]. As a pioneer, DreamField [17] firstly train NeRF [31] with CLIP [36] guidance to achieve text-to-3D distillation. However, the results is unsatisfactory due to the weak supervision from CLIP loss. With the advance of diffusion model, Dreamfusion [33] introduces Score Distillation Sampling (SDS) to distill 3D assets from pre-trained 2D text-to-image diffusion models. SDS facilitates 3D distillation by seeking specific modes in a text-guide diffusion model, allowing for training a 3D model based on the 2D knowledge of diffusion models. This quickly motivates a great number of following works [5, 16, 21, 29, 33, 35, 49] and becomes a critical integration of them. These works improve the performance of text-to-3D in various ways. For example, some of them [5, 6, 12, 21, 29, 43, 47] improve the visual quality of text-to-3D distillation via modifying NeRF or introducing other advanced 3D representations. The other some [2, 6, 40] focus on addressing the Janus problems, e.g., MVDream [40] propose to fine-tune the pre-trained diffusion models to make it 3D aware. However, all these methods heavily rely on the Score Distillation Sampling. Albeit promising, SDS has shown over-smoothing effects in a lot of literatures [21, 30, 33, 49]. Besides, it need coupling with a large conditional guidance scale [12], leading to over-saturation results. There are also some very recent works [18, 18, 45, 48, 48, 51] target at improving SDS. e.g., ProlificDreamer [45] proposes VSD to model 3D representation as a distribution. Our work is intrinsically different in the sense that it provides a systematic analysis on the the inconsistency and low-quality pseudo-ground-truths in SDS. And by introducing the Interval Score Matching, it achieves superior results without increasing the computational burden.

Differentiable 3D Representations. Differentiable 3D representation is a crucial integration of text-guided 3D generation. Given a 3D representation with the trainable parameter θ, a differentiable rendering equation $\boldsymbol{g}(\theta, c)$ is used to render an image in camera pose c of that 3D representation. As the process is differentiable, we could train the 3D representation to fit our condition with backpropagation. Previously, various representations have been introduce to text-to-3D generations [3, 8, 31, 39, 44]. Among them, NeRF [21, 31, 40] is the most common representation in text-to-3D generation tasks. The heavy rendering process of implicit representations makes it challenging for NeRF to produce high-resolution images that match the diffusion's resolution during distillation. To address this, textual meshes s [39], known for their efficient explicit rendering, are now used in this field to create detailed 3D assets [5, 21, 45], leading to better performance. Meanwhile, 3D Gaussian Splatting (3DGS) [19], another effective explicit representation, demonstrates remarkable efficiency in reconstruction tasks. In this paper, we investigate 3DGS [19] as the 3D representation in our framework.

Diffusion Models. Another key component of text-to-3D generation is the diffusion model, which provides supervision for the 3D model. We briefly introduce it here to cover some notations. The Denoising Diffusion Probabilistic Model (DDPM) [13, 38, 42] has been widely adopted for text-guided 2D image generation for its comprehensive capability. DDPMs assume $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ as a diffusion process according to a predefined schedule β_t on timestep t, that:

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}). \tag{1}$$

And the posterior $p_\phi(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ is modelled with a neural network ϕ, where:

$$p_\phi(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mu_\phi(\boldsymbol{x}_t), (1-\bar{\alpha}_{t-1})\Sigma_\phi(\boldsymbol{x}_t)), \tag{2}$$

where $\bar{\alpha}_t := (\prod_1^t 1-\beta_t)$, and $\mu_\phi(\boldsymbol{x}_t)$, $\Sigma_\phi(\boldsymbol{x}_t)$ denote the predicted mean and variance given x_t, respectively.

3. Methodology

3.1. Revisiting the SDS

As mentioned in Sec. 2, SDS [33] pioneers text-to-3D generation by seeking modes for the conditional post prior in the DDPM latent space. Denoting $\boldsymbol{x}_0 := \boldsymbol{g}(\theta, c)$ as 2D views rendered from θ, the posterior of noisy latent x_t is defined as:

$$q^\theta(\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I}). \tag{3}$$

Meanwhile, SDS adopts pretrained DDPMs to model the conditional posterior of $p_\phi(\boldsymbol{x}_t|y)$. Then, SDS aims to distill 3D representation θ via seeking modes for such conditional

posterior, which can be achieved by minimizing the following KL divergence for all t:

$$\min_{\theta\in\Theta} \mathcal{L}_{\text{SDS}}(\theta) \coloneqq \mathbb{E}_{t,c}\left[\omega(t) D_{\text{KL}}(q^{\theta}(\boldsymbol{x}_t) \parallel p_{\phi}(\boldsymbol{x}_t|y))\right]. \quad (4)$$

Further, by reusing the weighted denoising score matching objective [13, 42] for DDPM training, the Eq. (4) is reparameterized as:

$$\min_{\theta\in\Theta} \mathcal{L}_{\text{SDS}}(\theta) \coloneqq \mathbb{E}_{t,c}\left[\omega(t)||\boldsymbol{\epsilon}_{\phi}(\boldsymbol{x}_t, t, y) - \boldsymbol{\epsilon}||_2^2\right], \quad (5)$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is the ground truth denoising direction of $\boldsymbol{x}_t$ in timestep t. And the $\boldsymbol{\epsilon}_{\phi}(\boldsymbol{x}_t, t, y)$ is the predicted denoising direction with given condition y. Ignoring the UNet Jacobian [33], the gradient of SDS loss on θ is given by:

$$\nabla_{\theta}\mathcal{L}_{\text{SDS}}(\theta) \approx \mathbb{E}_{t,\boldsymbol{\epsilon},c}\left[\omega(t)(\underbrace{\boldsymbol{\epsilon}_{\phi}(\boldsymbol{x}_t, t, y) - \boldsymbol{\epsilon}}_{\text{SDS update direction}})\frac{\partial \boldsymbol{g}(\theta, c)}{\partial \theta}\right]. \quad (6)$$

Analysis of SDS. To lay a clearer foundation for the upcoming discussion, we denote $\gamma(t) = \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}$ and equivalently transform Eq. (5) into an alternative form as follows:

$$\begin{aligned}\min_{\theta\in\Theta} \mathcal{L}_{\text{SDS}}(\theta) &\coloneqq \mathbb{E}_{t,\boldsymbol{\epsilon},c}\left[\frac{\omega(t)}{\gamma(t)}||\gamma(t)(\boldsymbol{\epsilon}_{\phi}(\boldsymbol{x}_t, t, y) - \boldsymbol{\epsilon}) + \frac{(\boldsymbol{x}_t - \boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}}||_2^2\right] \\ &= \mathbb{E}_{t,\boldsymbol{\epsilon},c}\left[\frac{\omega(t)}{\gamma(t)}||\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^t||_2^2\right].\end{aligned} \quad (7)$$

where $\boldsymbol{x}_t \sim q^{\theta}(\boldsymbol{x}_t)$ and $\hat{\boldsymbol{x}}_0^t = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\phi}(\boldsymbol{x}_t, t, y)}{\sqrt{\bar{\alpha}_t}}$. Similarly, we can also rewrite the gradient of SDS loss as:

$$\nabla_{\theta}\mathcal{L}_{\text{SDS}}(\theta) = \mathbb{E}_{t,\boldsymbol{\epsilon},c}\left[\frac{\omega(t)}{\gamma(t)}(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^t)\frac{\partial \boldsymbol{g}(\theta, c)}{\partial \theta}\right]. \quad (8)$$

In this sense, the SDS objective can be viewed as matching the view $\boldsymbol{x}_0$ of the 3D model with $\hat{\boldsymbol{x}}_0^t$ (i.e., the pseudo-GT) that DDPM estimates from $\boldsymbol{x}_t$ in a single-step. However, we have discovered that this distillation paradigm overlooks certain critical aspects of the DDPM. In Fig. 2, we show that the pretrained DDPM tends to predict feature-inconsistent pseudo-GTs, which are sometimes of low quality during the distillation process. However, all updating directions yielded by Eq. (8) under such undesirable circumstances would be updated to the θ, and inevitably lead to over-smoothed results. We conclude the reasons for such phenomena from two major aspects. First, it is important to note a key intuition of SDS: it generates pseudo-GTs with 2D DDPM by referencing the input view $\boldsymbol{x}_0$. And afterward, SDS exploits such pseudo-GTs for $\boldsymbol{x}_0$ optimization. As disclosed by Eq. (8), SDS achieves this goal by first perturbing $\boldsymbol{x}_0$ to $\boldsymbol{x}_t$ with random noises, then estimating $\hat{\boldsymbol{x}}_0^t$ as the pseudo-GT. However, we notice that the DDPM is very sensitive to its input, where minor fluctuations in $\boldsymbol{x}_t$ would change the features of pseudo-GT significantly. Meanwhile, we find that not only the randomness in the noise component of $\boldsymbol{x}_t$, but also the randomness in the camera pose of $\boldsymbol{x}_0$ could contribute to such fluctuations, which is inevitable during the distillation. Optimizing $\boldsymbol{x}_0$ towards inconsistent pseudo-GTs ultimately leads to feature-averaged outcomes, as depicted in the last column of Fig. 2.

Second, Eq. (8) implies that SDS obtains such pseudo-GTs with a single-step prediction for all t, which neglects the limitation of single-step-DDPM that are usually incapable of producing high-quality results. As we also show in the middle columns of Fig. 2, such single-step predicted pseudo-GTs are sometimes detail-less or blurry, which obviously hinders the distillation. Consequently, we believe that distilling 3D assets with the SDS objective might be less ideal. Motivated by such observations, we aim to settle the aforementioned issues in order to achieve better results.

3.2. Interval Score Matching

Note that the aforementioned problems originate from the fact that $\hat{\boldsymbol{x}}_0^t$, which serves as the *pseudo-ground-truth* to match with $\boldsymbol{x}_0 = \boldsymbol{g}(\theta, c)$, is inconsistent and sometimes low quality. In this section, we provide an alternative solution to SDS that significantly mitigates these problems.

Our core idea lies in two folds. First, we seek to obtain more consistent pseudo-GTs during distillation, regardless of the randomness in noise and camera pose. Then, we generate such pseudo-GTs with high visual quality.

DDIM Inversion. As discussed above, we seek to produce more consistent pseudo-GTs that are aligned with $\boldsymbol{x}_0$. Thus, instead of producing $\boldsymbol{x}_t$ stochastically with Eq. (3), we employ the DDIM inversion to predict the noisy latent $\boldsymbol{x}_t$. Specifically, DDIM inversion predicts a invertible noisy latent trajectory $\{\boldsymbol{x}_{\delta_T}, \boldsymbol{x}_{2\delta_T}, ..., \boldsymbol{x}_t\}$ in an iterative manner:

$$\begin{aligned}\boldsymbol{x}_t &= \sqrt{\bar{\alpha}_t}\hat{\boldsymbol{x}}_0^s + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\phi}(\boldsymbol{x}_s, s, \emptyset) \\ &= \sqrt{\bar{\alpha}_t}(\hat{\boldsymbol{x}}_0^s + \gamma(t)\boldsymbol{\epsilon}_{\phi}(\boldsymbol{x}_s, s, \emptyset)),\end{aligned} \quad (9)$$

where $s = t - \delta_T$, and $\hat{x}_0^s = \frac{1}{\sqrt{\bar{\alpha}_s}}\boldsymbol{x}_s - \gamma(s)\boldsymbol{\epsilon}_{\phi}(\boldsymbol{x}_s, s, \emptyset)$. With some simple computation, we organize $\hat{\boldsymbol{x}}_0^s$ as:

$$\begin{aligned}\hat{\boldsymbol{x}}_0^s = \boldsymbol{x}_0 &- \gamma(\delta_T)[\boldsymbol{\epsilon}_{\phi}(\boldsymbol{x}_{\delta_T}, \delta_T, \emptyset) - \boldsymbol{\epsilon}_{\phi}(\boldsymbol{x}_0, 0, \emptyset)] - \cdots \\ &- \gamma(s)[\boldsymbol{\epsilon}_{\phi}(\boldsymbol{x}_s, s, \emptyset) - \boldsymbol{\epsilon}_{\phi}(\boldsymbol{x}_{s-\delta_T}, s - \delta_T, \emptyset)],\end{aligned} \quad (10)$$

Thanks to the invertibility of DDIM inversion, we significantly increase the consistency of the pseudo-GT (i.e., the $\hat{\boldsymbol{x}}_0^t$) with $\boldsymbol{x}_0$ for all t, which is important for our subsequent operations, please refer to our supplement for more details.

Interval Score Matching. Another limitation of SDS is that it generates pseudo-GTs with a single-step prediction from x_t for all t, making it challenging to guarantee high-quality pseudo-GTs. On this basis, we further seek to improve the visual quality of the pseudo-GTs. Intuitively, this can be achieved by replacing the single-step estimated pseudo-GT $\hat{\boldsymbol{x}}_0^t = \frac{1}{\sqrt{\bar{\alpha}_t}}\boldsymbol{x}_t - \gamma(t)\boldsymbol{\epsilon}_{\phi}(\boldsymbol{x}_t, t, y)$ with a multi-step one,

denoted as $\tilde{\boldsymbol{x}}_0^t := \tilde{\boldsymbol{x}}_0$, following the multi-step DDIM denoising process, i.e., iterating

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

until $\tilde{\boldsymbol{x}}_0$. Note that different from the DDIM inversion (Eq. (9)), this denoising process is conditioned on y. This matches the behavior of SDS (Eq. (6)), i.e., SDS imposes unconditional noise $\boldsymbol{\epsilon}$ during forwarding and denoise the noisy latent with a conditional model $\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t, y)$.

Intuitively, by replacing $\hat{\boldsymbol{x}}_0^t$ in Eq. (8) with $\tilde{\boldsymbol{x}}_0^t$, we conclude a naive alternative of the SDS, where:

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_c\left[\tfrac{\omega(t)}{\gamma(t)}(\boldsymbol{x}_0 - \tilde{\boldsymbol{x}}_0^t)\tfrac{\partial \boldsymbol{g}(\theta,c)}{\partial \theta}\right]. \quad (12)$$

Although $\tilde{\boldsymbol{x}}_0^t$ might produce higher quality guidance, it is overly time-consuming to compute and limits the practicality of such an algorithm. This motivates us to delve deeper into the problem and search for a more efficient approach.

Initially, we investigate the denoising process of $\tilde{\boldsymbol{x}}_0^t$ jointly with the inversion process. We first unify the iterative process in Eq. (11) as

$$\begin{aligned}\tilde{\boldsymbol{x}}_0^t = \frac{\boldsymbol{x}_t}{\sqrt{\bar{\alpha}_t}} &- \gamma(t)\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t, y) + \gamma(s)[\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t, y) - \boldsymbol{\epsilon}_\phi(\tilde{\boldsymbol{x}}_s, s, y)] \\ &+ \cdots + \gamma(\delta_T)[\boldsymbol{\epsilon}_\phi(\tilde{\boldsymbol{x}}_{2\delta_T}, 2\delta_T, y) - \boldsymbol{\epsilon}_\phi(\tilde{\boldsymbol{x}}_{\delta_T}, \delta_T, y)].\end{aligned} \quad (13)$$

Then, combining Eq. (9) with Eq. (13), we could transform Eq. (12) as follows:

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{t,c}\left[\tfrac{\omega(t)}{\gamma(t)}(\gamma(t)[\underbrace{\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_s, s, \emptyset)}_{\text{interval scores}}] + \eta_t)\tfrac{\partial \boldsymbol{g}(\theta,c)}{\partial \theta}\right]. \quad (14)$$

where we summarize the bias term η_t as:

$$\begin{aligned}\eta_t = &+ \gamma(s)[\boldsymbol{\epsilon}_\phi(\tilde{\boldsymbol{x}}_s, s, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_{s-\delta_T}, s-\delta_T, \emptyset)] \\ &- \gamma(s)[\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_s, s, \emptyset)] \\ &+ \ldots \\ &+ \gamma(\delta_T)[\boldsymbol{\epsilon}_\phi(\tilde{\boldsymbol{x}}_{\delta_T}, \delta_T, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_0, 0, \emptyset)] \\ &- \gamma(\delta_T)[\boldsymbol{\epsilon}_\phi(\tilde{\boldsymbol{x}}_{2\delta_T}, 2\delta_T, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_{\delta_T}, \delta_T, \emptyset)].\end{aligned} \quad (15)$$

Notably, η_t includes a series of neighboring interval scores with opposing scales, which are deemed to cancel each other out. Moreover, minimizing η_t is beyond our intention since it contains a series of score residuals that are more related to δ_T, which is a hyperparameter that is unrelated to 3D representation. Thus, we propose to disregard η_t to gain a boost in the training efficiency without compromising the distillation quality. Please refer to our supplement for more analysis and experiments about η_t.

Consequently, we propose an efficient alternative to Eq. (12) by disregarding the bias term η_t and focusing on minimizing the interval score, which we termed Interval Score Matching (ISM). Specifically, with a given prompt y and the noisy latents $\boldsymbol{x}_s$ and $\boldsymbol{x}_t$ generated through DDIM inversion from x_0, the ISM loss is defined as:

$$\min_{\theta\in\Theta} \mathcal{L}_{\text{ISM}}(\theta) := \mathbb{E}_{t,c}\left[\omega(t)||\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_s, s, \emptyset)||^2\right]. \quad (16)$$

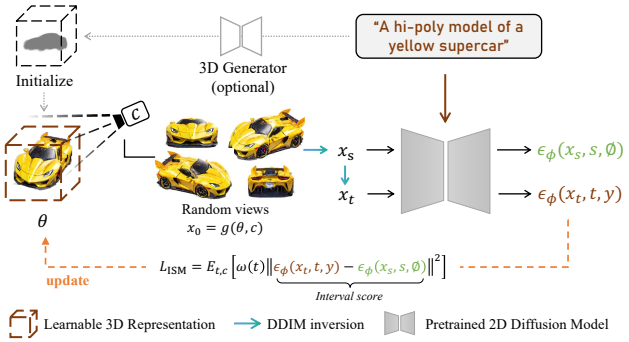

Figure 3. **An overview of LucidDreamer.** In our paper, we first initialize the 3D representation (i.e. Gaussian Splatting [20]) θ via the pretrained text-to-3D generator [32] with prompt y. Incorporate with pretrained 2D DDPM, we disturb random views $\boldsymbol{x}_0 = \boldsymbol{g}(\theta, c)$ to unconditional noisy latent trajectories $\{\boldsymbol{x}_0, ..., \boldsymbol{x}_s, \boldsymbol{x}_t\}$ via DDIM inversion [41]. Then, we update θ with the *interval score*. Please refer to Sec. 3.2 for details.

Following [33], the gradient of ISM loss over θ is given by:

$$\nabla_\theta \mathcal{L}_{\text{ISM}}(\theta) := \mathbb{E}_{t,c}\left[\omega(t)(\underbrace{\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_s, s, \emptyset)}_{\text{ISM update direction}})\tfrac{\partial \boldsymbol{g}(\theta,c)}{\partial \theta}\right]. \quad (17)$$

Despite omitting η_t from Equation (14), the core of optimizing the ISM objective still revolves around updating $\boldsymbol{x}_0$ towards pseudo-GTs that are *feature-consistent*, *high-quality*, yet *computationally friendly*. Hence, ISM aligns with the fundamental principles of SDS-like objectives [9, 33, 45] albeit in a more refined manner.

As a result, ISM presents several advantages over previous methodologies. Firstly, ISM provides consistent, high-quality pseudo-GTs, which leads to high-fidelity distillation outcomes with rich details and fine structure, eliminating the necessity for a large conditional guidance scale [12] and enhancing the flexibility for 3D content creation. Secondly, unlike the other works [26, 45], transitioning from SDS to ISM takes marginal computational overhead. Meanwhile, although ISM necessitates additional computation costs for DDIM inversion, it does not compromise the overall efficiency since 3D distillation with ISM usually converges in fewer iterations; more analysis is in our supplement.

Meanwhile, as the standard DDIM inversion usually adopts a fixed stride, it increases the cost for trajectory estimation linearly as t goes larger. However, it is usually beneficial to supervise θ at larger timesteps. Thus, instead of estimating the latent trajectory with a uniform stride, we propose to accelerate the process by predicting $\boldsymbol{x}_s$ with larger step sizes δ_S. We find such a solution reduces the training time dramatically without compromising the distillation quality. In addition, we present a quantitative analysis of the impact of δ_T and δ_S in Sec. 4.1. Overall, we summarize our proposed ISM in Fig. 3 and Algorithm 1.

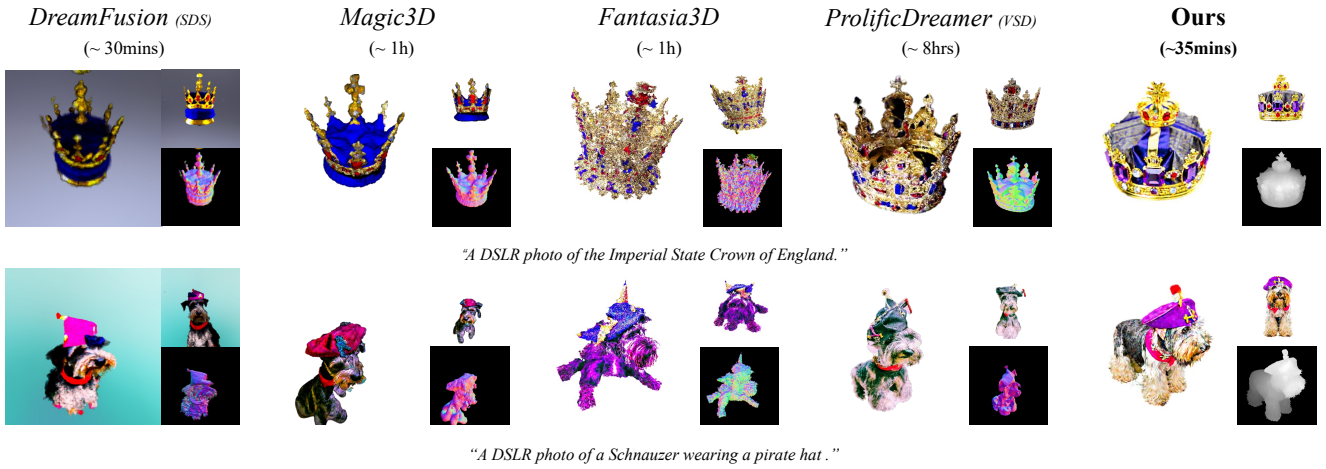

Figure 4. **Comparison with baselines methods in text-to-3D generation.** Experiment shows that our approach is capable of creating 3D content that matches well with the input text prompts with high fidelity and intricate details. The running time of our method is measured on a single A100 GPU with a view batch size of 4, $\delta_S = 200$. Please zoom in for details.

Algorithm 1 Interval Score Matching

Initialization: DDIM inversion step size δ_T and δ_S, the target prompt y
while θ is not converged **do**
  Sample: $\boldsymbol{x}_0 = g(\theta, c), t \sim \mathcal{U}(1, 1000)$
  let $s = t - \delta_T$ and $n = s/\delta_S$
  for $i = [0, ..., n-1]$ **do**
    $\hat{\boldsymbol{x}}_0^{i\delta_S} = \frac{1}{\sqrt{\bar{\alpha}_{i\delta_S}}}(\boldsymbol{x}_{i\delta_S} - \sqrt{1-\bar{\alpha}_{i\delta_S}}\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_{i\delta_S}, i\delta_S, \emptyset))$
    $\boldsymbol{x}_{(i+1)\delta_S} = \sqrt{\bar{\alpha}_{(i+1)\delta_S}}\hat{\boldsymbol{x}}_0^{i\delta_S} + \sqrt{1-\bar{\alpha}_{(i+1)\delta_S}}\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_{i\delta_S}, i\delta_S, \emptyset)$
  end for
  predict $\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_s, s, \emptyset)$, then step $\boldsymbol{x}_s \to \boldsymbol{x}_t$ via $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\hat{\boldsymbol{x}}_0^s + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_s, s, \emptyset)$
  predict $\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t, y)$ and compute ISM gradient $\nabla_\theta L_{\text{ISM}} = \omega(t)(\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_s, s, \emptyset))$
  update $\boldsymbol{x}_0$ with $\nabla_\theta L_{\text{ISM}}$
end while

3.3. The Advanced Generation Pipeline

We also explore the factors that would affect the visual quality of text-to-3D generation and propose an advanced pipeline with our ISM. Specifically, we introduce 3D Guassians Splatting (3DGS) as our 3D representation and 3D point cloud generation models for initialization.

3D Gaussian Splatting. Empirical observations of existing works indicate that increasing the rendering resolution and batch size for training would significantly improve the visual quality. However, most learnable 3D representations that have been adopted in the text-to-3D generation [33, 40, 45] are relatively time and memory-consuming. In contrast, 3D Gaussian Splatting [19] provides highly efficient in both rendering and optimizing. This drives our pipeline to achieve high-resolution rendering and large batch size even with more limited computational resources.

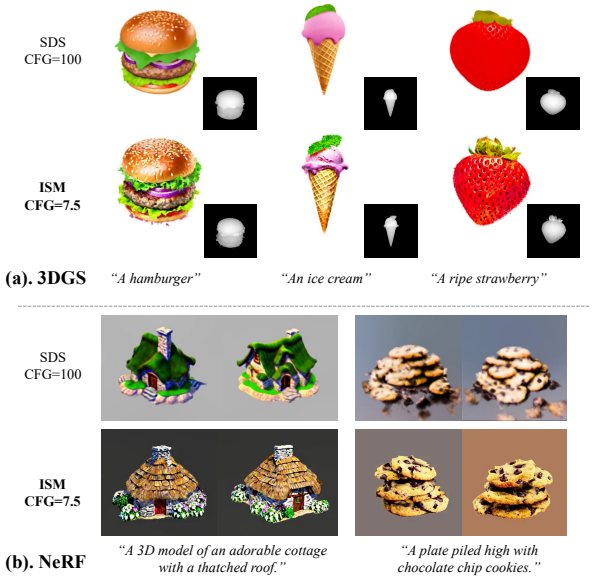

Figure 5. **A comparison of SDS [33] and ISM.** It shows that either using (a). 3DGS or (b). NeRF, the results of SDS tend to be smooth, whereas our ISM excels in generating realistic details.

Initialization. Most previous methods [5, 33, 40, 45] usually initialize their 3D representation with limited geometries like box, sphere, and cylinder, which could lead to undesired results on non-axial-symmetric objects. Since we introduce the 3DGS as our 3D representation, we can naturally adopt several text-to-point generative models [32] to generate the coarse initialization with humans prior, it greatly improves the convergence speed, as shown in Sec. 4.1.

4. Experiments

Text-to-3D Generation. We show the generated results of LucidDreamer in Fig. 1 with original stable diffu-

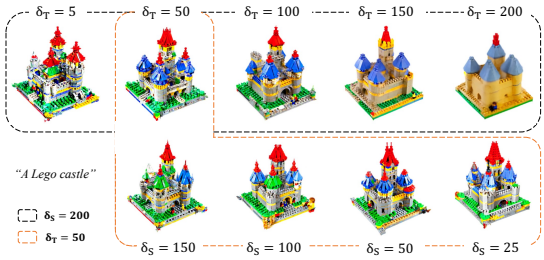

Figure 6. **ISM with Different δ_T and δ_S.** We fix $\delta_T = 50$ (orange dashed box) and $\delta_S = 200$ (black dashed box), respectively, to compare the influence of these hyperparameters qualitatively.

sion [37] (below the dashed line) and various fintune checkpoints [1, 27, 52][1] (above the dashed line). The results demonstrate that LucidDreamer is capable of generating 3D content that is highly consistent with the semantic cues of the input text. It excels in producing realistic and intricate appearances, avoiding issues of excessive smoothness or over-saturation, such as in the details of character portraits or hair textures. Furthermore, our framework is not only proficient in accurately generating common objects but also supports creative creations, like imagining unique concepts such as "Iron Man with white hair" (Fig. 1).

Generalizability of ISM. To evaluate the generalizability of ISM, we conduct a comparison with ISM and SDS in both explicit representation (3DGS [20]) and implicit representation (NeRF [31]). Notably, we follow the hyperparameter design of ProlificDreamer in the NeRF comparison. As shown in Fig 5, our ISM provides fined-grained details even with normal CFG (7.5) in both NeRF [31] and 3D Gaussian Splatting [20] (3DGS), which is significantly better than the SDS. This demonstrates the generalizability of our ISM.

Qualitative Comparison. We compare our model with current SoTA baselines [5, 21, 33, 45] reimplemented by Three-studio [11]. We all use the stable diffusion 2.1 for distillation and all experiments were conducted on A100 for fair comparison. As shown in Fig. 4, our method achieves results regarding high fidelity and geometry consistency with less time and resource consumption. For example, the Crown generated by our framework exhibits more precise geometric structures and realistic colors, contrasting sharply with the geometric ambiguity prevalent in other baseline methods. Compared to Schnauzer generated by other methods, our approach produces Schnauzer with hair texture and overall body shape that is closer to reality, showing a clear advantage. Meanwhile, since the Point Generator introduces the geometry prior, the Janus problem is also reduced.

User study. We conduct a user study to provide a comprehensive evaluation. Specifically, we select 28 prompts and generate objects using different Text-to-3D generation methods with each prompt. The users were asked to rank them based on the fidelity and the degree of alignment with the given text prompt. We show the average ranking to evaluate the users' preferences. As shown in Tab. 1, our framework gets the highest average ranking in 6 selective methods. Indicate that users consistently favored the 3D models generated by our framework. Please refer to our supplement for more details of the user study and more visual results. Also, we conduct GPTEval3D [46] to measure the "wining rate" of our method against the baseline as shown in Tab. 2.

[1]Term of Service: https://civitai.com/content/tos

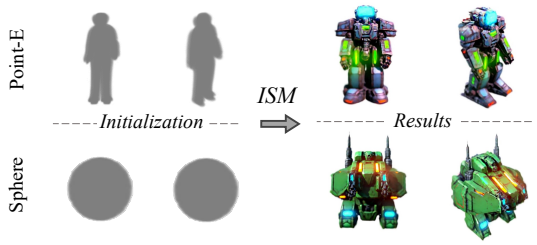

Figure 7. **LucidDreamer with Different initialization.** We compare the results of two different initializations to evaluate the effectiveness of the Point Generator in our advanced pipeline.

DreamFusion [33]	Magic3D [21]	Text2Mesh[30]	Fantasia3D [5]	ProlificDreamer [45]	**Ours**
3.28	3.44	4.76	4.53	2.37	**1.25**

Table 1. We survey the users' preference ranking (**the smaller, the better**) averaged on 28 sets of text-to-3D generation results produced by baselines and our method, respectively. Our result is preferred by most users.

In	Alignment	Plausibility	Color-Geo	Texture	Geometry
LucidDreamer v.s. ProlificDreamer	58%	63%	61%	63%	62%
LucidDreamer v.s. Magic3D	61%	68%	52%	72%	77%
LucidDreamer v.s. Fantasia3D	70%	83%	68%	68%	83%
LucidDreamer v.s. DreamFusion	84%	82%	76%	82%	88%

Table 2. Winning rate of LucidDreamer measured on 28 sets of generated 3D assets with GPT-4v. **the higher, the better**)

4.1. Ablation Studies

Effect of Interval Length. We explore the effect of interval length δ_T and δ_S during training in this section. In Fig. 6, we visualize the influence of δ_T and δ_S. For a fixed δ_T, an increasing δ_S takes marginal influence in the results but significantly saves the computational costs of DDIM inversion. Meanwhile, as the parameter δ_T increases, the results adopt a more natural color and simpler structure. However, this comes at the expense of detail. Thus, we conclude a trade-off in the selection of δ_T. For instance, at higher δ_T, castle walls appear smoother. Conversely, lower δ_T values enhance detail but can result in unnecessary visual anomalies, such as overly saturated color and the illusion of floating artifacts atop castle towers. We hypothesize such observation is caused by the gradients provided by small intervals containing more detailed features but less structural supervision. Thus, we propose annealing the interval with the intuitive process of initially constructing the overall structures and subsequently incorporating fine-grained features. Moreover, this hyperparameter allows the user to generate objects with different levels of smoothness according to their preferences.

Initialization with Point Generators We ablate the Point Generators in this section. Specifically, we train two 3D

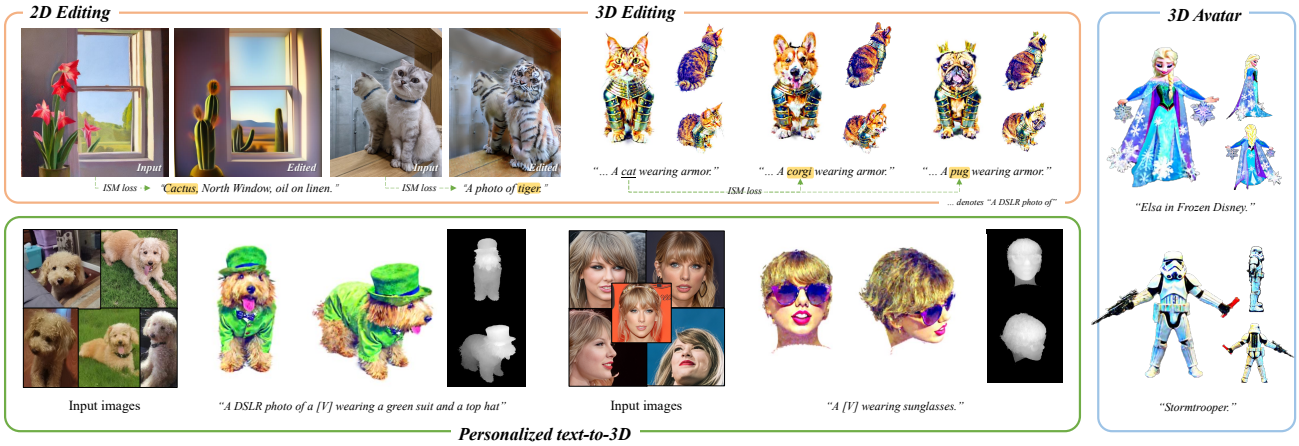

Figure 8. **Applications of ISM.** We explore several applications with our proposed ISM, including the *zero-shot 2D and 3D editing* (top left), *personalized text-to-3D generation* with LoRA (bottom left), and *3D avatar generation*. Generally, our proposed ISM as well as the Advanced 3D generation pipeline performs surprisingly well across various tasks. Please refer to our paper for more details.

Gaussians from a random initialization and starting from a generated raw point cloud with a given prompt, respectively. In Fig. 7, we compare the distillation results with the same prompts but different. With the parameter and random seed guaranteed to be constant, 3D Gaussian with point initialization has a better result in geometry.

5. Applications

This section further explores the applications of LucidDreamer. Specifically, we combine our framework with advanced conditioning techniques and achieve some real-world applications.

Zero-shot Avatar Generation. We expand our framework to produce pose-specific avatars by employing the Skinned Multi-Person Linear Model (SMPL) [25] as a geometry prior to initializing the 3D Gaussian point cloud. Then, we rely on ControlNet [50] conditioned on DensePose [10] signals to offer more robust supervision. Specifically, we render the 3D human mesh into a 2D image using pytorch3d based on sampled camera parameters and subsequently input it into the pre-trained DensePose model to acquire the human body part segmentation map as a DensePose condition. A more detailed framework is shown in the supplement. Following such an advanced control signal, we can achieve a high-fidelity avatar as shown in Fig. 8.

Personalized Text-to-3D. We also combine our framework with personalized techniques, LoRA [15]. Using such techniques, our model can learn to tie the subjects or styles to an identifier string and generate images of the subjects or styles. For text-to-3D generation, we can use the identifier string for 3D generation of specific subjects and styles. As shown in Fig. 8, our method can generate personalized humans or things with fine-grained details. This also shows the great potential of our method in controllable text-to-3D generation by combining it with advanced personalized techniques.

Zero-shot 2D and 3D Editing. While our framework is primarily designed for text-to-3D generation tasks, extending ISM to editing is feasible due to the similarities in both tasks. Effortlessly, we can edit a 2D image or 3D representation in a conditional distillation manner, as ISM provides consistent update directions based on the input image, guiding it towards the target condition, as demonstrated in Fig. 8. Owing to space limitations, we reserve further customization of ISM for 2D/3D editing tasks for future exploration.

6. Conclusions

In this paper, we have presented a comprehensive analysis of the over-smoothing effect inherent in Score Distillation Sampling (SDS), identifying its root cause in the inconsistency and low quality of pseudo ground truth. Addressing this issue, we introduced Interval Score Matching (ISM), a novel approach that offers consistent and reliable guidance. Our findings demonstrate that ISM effectively overcomes the over-smoothing challenge, yielding highly detailed results without extra computational costs. Notably, ISM's compatibility extends to various applications, including NeRF and 3DGS for 3D generation and editing, as well as 2D editing tasks, showcasing its exceptional versatility. Building upon this, we have developed *LucidDreamer*, a framework that combines ISM with 3D Gaussian Splatting. Extensive experiments established that *LucidDreamer* significantly surpasses current SoTA methodologies. Its superior performance paves the way for a broad spectrum of practical applications, ranging from text-to-3D generation and editing to zero-shot avatar creation and personalized Text-to-3D conversions, among others.

7. Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 62206068 and No.92370204), and Natural Science Foundation of Zhejiang Province, China (No. LD24F020002).

References

[1] 7whitefire7. Realcartoon-pixar. https://civitai.com/models/107289/realcartoon-pixar, 2023. 7

[2] Mohammadreza Armandpour, Huangjie Zheng, Ali Sadeghian, Amir Sadeghian, and Mingyuan Zhou. Re-imagine the negative prompt algorithm: Transform 2d diffusion into 3d, alleviate janus problem and beyond. *arXiv*, 2023. 3

[3] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021. 3

[4] Yukang Cao, Yan-Pei Cao, Kai Han, Ying Shan, and Kwan-YeeK. Wong. Dreamavatar: Text-and-shape guided 3d human avatar generation via diffusion models. 2023. 2

[5] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *ICCV*, 2023. 2, 3, 6, 7

[6] Zilong Chen, Feng Wang, and Huaping Liu. Text-to-3d using gaussian splatting. *arXiv*, 2023. 3

[7] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. 3

[8] Wenhang Ge, Tao Hu, Haoyu Zhao, Shu Liu, and Ying-Cong Chen. Ref-neus: Ambiguity-reduced neural implicit surface learning for multi-view reconstruction with reflection. *ICCV*, 2023. 3

[9] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. In *NeurIPS*, 2022. 5

[10] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018. 8

[11] Yuan-Chen Guo, Ying-Tian Liu, Ruizhi Shao, Christian Laforte, Vikram Voleti, Guan Luo, Chia-Hao Chen, Zi-Xin Zou, Chen Wang, Yan-Pei Cao, and Song-Hai Zhang. threestudio: A unified framework for 3d content generation. https://github.com/threestudio-project/threestudio, 2023. 7

[12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 3, 5

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 3, 4

[14] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. 2022. 2

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv*, 2021. 8

[16] Yukun Huang, Jianan Wang, Ailing Zeng, He Cao, Xianbiao Qi, Yukai Shi, Zheng-Jun Zha, and Lei Zhang. Dreamwaltz: Make a scene with complex 3d animatable avatars. *arXiv*, 2023. 2, 3

[17] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *CVPR*, 2022. 3

[18] Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. Noise-free score distillation. *arXiv*, 2023. 3

[19] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ToG*, 2023. 3, 6

[20] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ToG*, 2023. 2, 5, 7

[21] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *CVPR*, 2023. 2, 3, 7

[22] Yukang Lin, Haonan Han, Chaoqun Gong, Zunnan Xu, Yachao Zhang, and Xiu Li. Consistent123: One image to highly consistent 3d asset using case-aware diffusion priors. 2023.

[23] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Zexiang Xu, Hao Su, et al. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *arXiv*, 2023.

[24] Ruoshi Liu, Rundi Wu, BasileVan Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. 2023. 2

[25] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023. 8

[26] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models, 2023. 5

[27] Lykon. 3d animation diffusion. https://civitai.com/models/118086/3d-animation-diffusion, 2023. 7

[28] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. 2

[29] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *CVPR*, 2023. 3

[30] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13492–13502, 2022. 2, 3, 7

[31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 2021. 3, 7

[32] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv*, 2022. 5, 6

[33] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 2, 3, 4, 5, 6, 7

[34] Senthil Purushwalkam and Nikhil Naik. Conrad: Image constrained radiance fields for 3d generation from a single image. *arXiv*, 2023. 2

[35] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv*, 2023. 3

[36] 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. Learning transferable visual models from natural language supervision, 2021. 3

[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3, 7

[38] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. 3

[39] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In *NeurIPS*, 2021. 3

[40] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv*, 2023. 2, 3, 6

[41] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. 5

[42] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 3, 4

[43] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arxiv*, 2023. 3

[44] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *CVPR*, 2022. 3

[45] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv*, 2023. 2, 3, 5, 6, 7

[46] Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas Guibas, Dahua Lin, and Gordon Wetzstein. Gpt-4v (ision) is a human-aligned evaluator for text-to-3d generation. *arXiv preprint arXiv:2401.04092*, 2024. 7

[47] Taoran Yi, Jiemin Fang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. *arXiv*, 2023. 3

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

[49] Huichao Zhang, Bowen Chen, Hao Yang, Liao Qu, Xu Wang, Li Chen, Chao Long, Feida Zhu, Kang Du, and Min Zheng. Avatarverse: High-quality & stable 3d avatar creation from text and pose. 2023. 2, 3

[50] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 8

[51] Joseph Zhu and Peiye Zhuang. Hifa: High-fidelity text-to-3d with advanced diffusion guidance. *arXiv*, 2023. 2, 3

[52] Zovya. A-zovya rpg artist tools. https://civitai.com/models/8124/a-zovya-rpg-artist-tools, 2023. 7