# SegmentDreamer: Towards High-fidelity Text-to-3D Synthesis with Segmented Consistency Trajectory Distillation

Jiahao Zhu[1] Zixuan Chen[1] Guangcong Wang[2] Xiaohua Xie[1,3,4*] Yi Zhou[1*]
[1]Sun Yat-sen University [2]Great Bay University [3]Pazhou Lab (Huangpu)
[4]Guangdong Province Key Laboratory of Information Security Technology
{zhujh59,chenzx3}@mail2.sysu.edu.cn, wanggc3@gmail.com, {xiexiaoh6,zhouyi}@mail.sysu.edu.cn


Figure 1. **Examples of 3D assets generated by *SegmentDreamer*.** Our framework addresses the improper conditional guidance issues of recent Consistency Distillation (CD)-based methods and theoretically provides a significantly tighter upper bound on distillation error, enabling high-fidelity text-to-3D generation within a reasonable timeframe ($\sim$32 minutes with classifier-free guidance [9] and $\sim$38 minutes combined with Perp-Neg [1]) on a single A100 GPU through 3D Gaussian Splatting (3DGS). Project Page: https://zjhJOJO.github.io/segmentdreamer

## Abstract

*Recent advancements in text-to-3D generation improve the visual quality of Score Distillation Sampling (SDS) and its variants by directly connecting Consistency Distillation (CD) to score distillation. However, due to the imbalance between self-consistency and cross-consistency, these CD-based methods inherently suffer from improper conditional guidance, leading to sub-optimal generation results. To address this issue, we present* ***SegmentDreamer****, a novel framework designed to fully unleash the potential of consistency models for high-fidelity text-to-3D generation. Specifically, we reformulate SDS through the proposed Segmented Consistency Trajectory Distillation (SCTD), effectively mitigating the imbalance issues by explicitly defining the relationship between self- and cross-consistency. Moreover,* ***SCTD*** *partitions the Probability Flow Ordinary Differential Equation (PF-ODE) trajectory into multiple sub-trajectories and ensures consistency within each segment, which can theoretically provide a significantly tighter upper bound on distillation error. Additionally, we propose a distillation pipeline for a more swift and stable generation. Extensive experiments demonstrate that our* ***SegmentDreamer*** *outperforms state-of-the-art methods in visual quality, enabling high-fidelity 3D asset creation through 3D Gaussian Splatting (3DGS).*


## 1. Introduction

Text-to-3D generation aims to create 3D assets that accurately align with text descriptions. It plays a vital role in various domains such as virtual reality, 3D gaming, and film production. One compelling solution is to directly "distill" 3D assets from 2D text-to-image Diffusion Models (DM) [34] through Score Distillation Sampling (SDS) [31, 46], which leverages pre-trained text-to-image DMs as the 3D priors to guide the optimization of differentiable 3D representations, such as Neural Radiance Fields (NeRF) [29] and 3D Gaussian Splatting (3DGS) [17]. Subsequent research focuses on improving the performance of SDS in diversity [11, 43, 47], visual quality [3, 16, 22, 24, 25, 47, 48, 51, 53, 57, 58], and multi-view consistency [5, 10, 12, 35, 37].

Recent methods [22, 48] have improved SDS and its variants by incorporating Consistency Distillation (CD) [18, 40] into their distillation loss. However, due to an imbalance between self-consistency and cross-consistency in their losses, these methods inherently suffer from improper conditional guidance when combined with Classifier-Free Guidance (CFG) for generation, ultimately leading to sub-optimal outcomes (see Figure 2; a detailed explanation is provided in Section 4.1). Additionally, the distillation error


*Corresponding author.


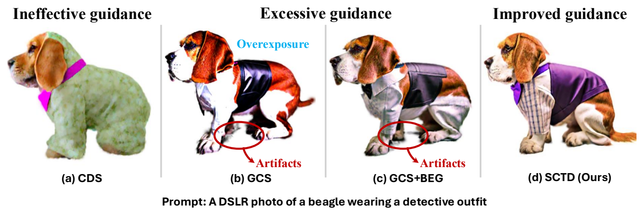


Figure 2. Visual examples of existing CD-based methods: CDS [48], GCS [22], and our *SegmentDreamer* with a Classifier-Free Guidance (CFG) scale of 7.5. As shown, improper conditional guidance leads to suboptimal results. **(a)** Ineffective guidance produces semantically inconsistent details, while **(b)** excessive guidance results in overexposure and artifacts. Although **(c)** Brightness-Equalized Generation (BEG) [22] can mitigate overexposure, artifacts persist. In contrast, our *SegmentDreamer* enhances conditional generation, producing high-fidelity 3D assets.

upper bound of these methods remains large, limiting their generation quality.

To address the aforementioned limitations, we present a novel framework – *SegmentDreamer*, which aims to fully unleash the potential of consistency models for high-fidelity text-to-3D generation. Specifically, by analyzing existing CD-based methods, we draw inspiration from segmented consistency models [33, 44] to reformulate SDS through the proposed Segmented Consistency Trajectory Distillation (SCTD), which effectively mitigates the imbalance between self-consistency and cross-consistency by explicitly defining their relationship. Moreover, unlike CD-based methods that directly enforce consistency across the entire Probability Flow Ordinary Differential Equation (PF-ODE) trajectory, *SCTD* partitions the PF-ODE trajectory into multiple sub-trajectories, ensuring consistency within each segment. This approach theoretically provides a significantly tighter upper bound on distillation error. In addition, we propose a distillation pipeline for a more swift and stable generation. We also introduce 3DGS as our 3D representation. Extensive experimental results demonstrate that our *SegmentDreamer* outperforms state-of-the-art methods in visual quality. Moreover, the proposed *SCTD* can be seamlessly applied to various 3D generative tasks (*e.g.*, 3D avatar generation and 3D portrait generation), facilitating the creation of high-fidelity 3D assets.

The main contributions are summarized as follows:

- We present *SegmentDreamer*, a novel framework for high-fidelity text-to-3D generation, which addresses the imbalance issue of existing CD-based methods.
- We reformulate SDS through the proposed *SCTD*, effectively mitigating the imbalance between self- and cross-consistency. *SCTD* also theoretically provides a significantly tighter upper bound on distillation error. We further propose a distillation pipeline for more swift and stable optimization.

- Experimental results demonstrate that our *Segment-Dreamer* outperforms state-of-the-art methods in visual quality, enabling high-fidelity 3D asset generation through 3DGS.

## 2. Related Work

### 2.1. Text-to-3D Generation

Creating 3D objects from text descriptions is challenging in the fields of computer vision and graphics. To achieve this goal, a series of end-to-end text-to-3D pipelines have been proposed [7, 8, 13, 21, 30, 49], but these methods require large-scale labeled 3D datasets for training. To reduce the reliance on 3D training data, a compelling solution is to directly "distill" 3D assets from 2D text-to-image Diffusion Models (DM) [34]. A groundbreaking contribution in this domain is DreamFusion [31], which introduced the Score Distillation Sampling (SDS) loss to optimize parametric 3D representations. Follow-up works focus on refining and extending SDS in various aspects. For example, several studies [11, 43, 47] improve SDS in diversity through prompting modifications, while others [3, 16, 22, 24, 25, 47, 48, 51, 53, 57, 58] enhance the fidelity of the generated 3D objects. Other advancements aim to achieve multi-view consistency [5, 10, 12, 35, 37] and faster generation [42, 52]. Recently, Consistency Distillation (CD) [18, 40] has received widespread attention as it enables high-quality generation through minimal sampling steps. Inspired by CD, Wu *et al.* [48] and Li *et al.* [22] incorporate CD into their distillation loss, which addresses some limitations brought by SDS. However, these CD-based methods inherently suffer from improper conditional guidance, leading to sub-optimal results. In this work, We aim to unleash CD's potential in generating high-fidelity 3D assets and push text-to-3D synthesis boundaries.

### 2.2. Differentiable 3D Representations

**Neural Radiance Field (NeRF)** [29] aims to construct implicit 3D representations using a multi-layer perceptron (MLP), leveraging standard volumetric rendering [14] and alpha compositing techniques [32]. Subsequent research has explored adapting NeRF to various applications, including sparse-view reconstruction [45], anti-aliasing [2], and medical image super-resolution [4].

**3D Gaussian Splatting (3DGS)** [17] has emerged as a leading approach for 3D representation, offering rapid optimization and high-speed rendering. Unlike NeRF, 3DGS explicitly models scenes using a set of 3D Gaussians and renders views through splatting [20]. Its versatility has enabled applications across various domains, including avatar generation [36, 55], single-view reconstruction [41], anti-aliasing [54], digital watermarking [6], and SLAM [28, 50].

## 3. Preliminaries

**Diffusion Model (DM)** [15, 38, 39] consists of a diffusion process and a denoising process. The diffusion process can be modeled as the solution of an Itô stochastic differential equation (SDE) with the transition kernel $\mathbb{P}(\mathbf{z}_t|\mathbf{z}_0) := \mathcal{N}(\mathbf{z}_t; \alpha_t\mathbf{z}_0, \sigma_t^2\mathbf{I})$. By reversing the diffusion process, $\mathbf{z}_0$ can be estimated by solving a Probability Flow Ordinary Differentiable Equation (PF-ODE), which is defined by $\mathrm{d}\mathbf{z}_t = [f(t)\mathbf{z}_t - \frac{1}{2}g(t)^2\nabla_{\mathbf{z}_t}\log\mathbb{P}(\mathbf{z}_t)]\mathrm{d}t$. A neural network $\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{z}, t)$ can be trained to approximate the scaled score function $-\sigma_t\nabla_{\mathbf{z}_t}\log\mathbb{P}(\mathbf{z}_t)$, offering an empirical formulation of the PF-ODE: $\mathrm{d}\mathbf{z}_t = [f(t)\mathbf{z}_t + \frac{g(t)^2}{2\sigma_t}\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{z}_t, t)]\mathrm{d}t$.

**Consistency Model** [27, 40] aims to achieve few-step inference by distilling a pre-trained DM $\boldsymbol{\phi}$ into a student model $\boldsymbol{\phi}_{stu}$. Given a PF-ODE trajectory $\{\mathbf{z}_t\}_{t\in[0,T]}$ and a condition $\mathbf{y}$, a consistency model $\boldsymbol{f}_{\boldsymbol{\phi}_{stu}}$ builds the mapping $\boldsymbol{f}_{\boldsymbol{\phi}_{stu}}(\mathbf{z}_t, t, \mathbf{y}) = \mathbf{z}_0$ between any points along this trajectory and the endpoint $\mathbf{z}_0$ by minimizing the following Consistency Distillation (CD) loss:

$$\mathcal{L}_{\text{CD}} = \mathbb{E}[\lambda(t)||\boldsymbol{f}_{\boldsymbol{\phi}_{stu}}(\mathbf{z}_t, t), \boldsymbol{f}_{\boldsymbol{\phi}_{stu}^-}(\hat{\mathbf{z}}_s^{\boldsymbol{\Phi}}, s)||_2^2], t > s, \quad (1)$$

where $\mathbf{z}_t = \alpha_t\mathbf{z}_0 + \sigma_t\boldsymbol{\epsilon}$ for $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{\phi}_{stu}^-$ is updated with the exponential moving average [27, 40]. $\hat{\mathbf{z}}_s^{\boldsymbol{\Phi}}$ can be obtained by:

$$\begin{aligned} \hat{\mathbf{z}}_s^{\boldsymbol{\Phi}} &= \boldsymbol{\Phi}(\mathbf{z}_t, t, s, \mathbf{y}) \\ &= \mathbf{z}_t + \int_t^s [f(u)\mathbf{z}_u + \frac{g(u)^2}{2\sigma_u}\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{z}_u, u, \mathbf{y})]\mathrm{d}u. \end{aligned} \quad (2)$$

The consistency function $\boldsymbol{f}_{\boldsymbol{\phi}_{stu}}$ can be parameterized in various ways [27, 40], and the solver $\boldsymbol{\Phi}(\cdot,\cdot,\cdot,\cdot)$ is used to calculate the point at $s$, which can be implemented using DDIM [38] or DPM-solver [26].

**Score Distillation Sampling (SDS) Loss**. Given a 3D representation $\boldsymbol{\theta}$, one can render a view image $\mathbf{z}_0 := \boldsymbol{g}(\mathbf{c}, \boldsymbol{\theta})$ with camera pose $\mathbf{c}$, SDS distills a 3D object from a 2D DM $\boldsymbol{\phi}$ by minimizing the following SDS loss *w.r.t.* $\boldsymbol{\theta}$:

$$\mathcal{L}_{\text{SDS}} := \mathbb{E}_{t,\boldsymbol{\epsilon}}[\omega(t)||\underbrace{\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{z}_t, t, \mathbf{y}) - \boldsymbol{\epsilon}}_{\text{noise residual}}||_2^2], \quad (3)$$

where $\mathbf{z}_t = \alpha_t\mathbf{z}_0 + \sigma_t\boldsymbol{\epsilon}$. Poole *et al.* [31] omit the Jacobian term $\frac{\partial\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{z}_t,t,\mathbf{y})}{\partial\mathbf{z}_t}$ in gradient calculation for speedup.

## 4. Our Method

In this section, we first review the mechanism of existing Consistency Distillation (CD)-based methods [22, 48] and reveal that they inherently suffer from improper conditional guidance when combined with Classifier-free Guidance (CFG). Subsequently, we address these limitations by reformulating SDS through the lens of the proposed Segmented Consistency Trajectory Distillation (SCTD). Moreover, building on this insight, we propose an efficient SCTD sampling method for high-fidelity text-to-3D generation. We further tailor a distillation pipeline for the proposed SCTD sampling to achieve more swift and stable optimization. Finally, we provide a theoretical proof for our distillation error upper bound. The overall framework of our *SegmentDreamer* is depicted in Fig. 3, with the algorithm detailed in Algorithm 1.

### 4.1. Reviewing CD-based Losses

The consistency function *w.r.t.* the 3D representation $\boldsymbol{\theta}$ is parameterized as $\boldsymbol{F_\theta}(\mathbf{z}_t, t, \mathbf{y}) = \frac{\mathbf{z}_t - \sigma_t \boldsymbol{\epsilon_\phi}(\mathbf{z}_t, t, \mathbf{y})}{\alpha_t}$ [22, 48]. Before proceeding, we clarify two key concepts: *self-consistency* and *cross-consistency*. The former ensures that all points on the same ODE trajectory consistently map to their original points, as in consistency models [18, 27, 40]. The latter enforces alignment between unconditional and conditional ODE trajectories [22].

**Consistency Distillation Sampling (CDS)** loss [48] is defined as

$$\mathcal{L}_{\text{CDS}}(\boldsymbol{\theta}) = \mathbb{E}_{t,s}[c(t)||\boldsymbol{F_\theta}(\mathbf{z}_t, t, \mathbf{y}) - \boldsymbol{F_\theta}(\hat{\mathbf{z}}_s^{\boldsymbol{\Phi}}, s, \mathbf{y})||_2^2], \quad (4)$$

where $s < t$, $c(t) = \omega(t)(\frac{\alpha_t}{\sigma_t})^2$, and $\hat{\mathbf{z}}_s^{\boldsymbol{\Phi}}$ is estimated using single-step DDIM sampling. Eq. (4) reveals that CDS prioritizes self-consistency in the whole ODE trajectory while omitting cross-consistency in its loss function, which can be interpreted as the absence of the "classifier score". As reported in CSD [53], this missing component, which is crucial for conditional generation, hinders CDS from producing semantically consistent content (see Fig. 5 for details). Moreover, the reliance on $\boldsymbol{F}_\theta$ may introduce a significant distillation error upper bound $\mathcal{O}(\Delta_t)T$, where $\Delta_t = \max|t - s|$. This issue is particularly pronounced in text-to-3D settings due to the typically large $|t - s|$.

**Guided Consistency Sampling (GCS)** [22] leverages the DPM-Solver-1 [26] as its consistency function:

$$\boldsymbol{G_\theta}(\mathbf{z}_t, t, s, \mathbf{y}) = \frac{\alpha_s}{\alpha_t}\mathbf{z}_t - \alpha_s \boldsymbol{\epsilon_\phi}(\mathbf{z}_t, t, \mathbf{y}) \int_{\lambda_t}^{\lambda_s} e^{-\lambda} \mathrm{d}\lambda, \quad (5)$$

where $\lambda_t = \ln \frac{\alpha_t}{\sigma_t}$. Based on Eq. (5), GCS proposes a more compact consistency (CC) loss $\mathcal{L}_{\text{CC}}$ defined as

$$\mathcal{L}_{\text{CC}}(\boldsymbol{\theta}) = \mathbb{E}_{t,s,e}[\left\|\boldsymbol{G_\theta}(\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}}, t, e, \emptyset) - \boldsymbol{G_\theta}(\hat{\mathbf{z}}_s^{\boldsymbol{\Phi}}, s, e, \emptyset)\right\|_2^2], \quad (6)$$

where $e<s<t$. GCS also employs a conditional guidance loss $\mathcal{L}_{\text{CG}}$ and a pixel-wise constraint $\mathcal{L}_{\text{CP}}$ to enforce cross-consistency (see Supp. A for details).

GCS ensures self-consistency within sub-trajectory, but it still suffers from improper conditional guidance due to two key issues: (1) $\boldsymbol{G_\theta}$ used to enforce cross-consistency is inherently flawed because $\boldsymbol{\epsilon_\phi}(\mathbf{z}_t, t, \mathbf{y})$ lacks a target timestep, as detailed in App. B. (2) GCS employs $\boldsymbol{F_\theta}$ (implemented via one-step DDIM) to enforce cross-consistency throughout the entire ODE trajectory in both latent and pixel spaces. However, as reported in its paper, it suffers from unstable generation due to the excessive conditional guidance, leading to overexposure and artifacts (more experimental analysis is provided in App. Sec. C). Furthermore, this approach results in a less stringent upper bound on the distillation error, expressed as $\mathcal{O}(\Delta_t)(T - e)$, which constrains its generation quality.

**Summary:** The improper conditional guidance in CDS and GCS stems mainly from an imbalance between self-consistency and cross-consistency. Additionally, their reliance on $\boldsymbol{F_\theta}$ to enforce consistency across the entire trajectory exacerbates this issue. Tab. 1 clarifies the differences between CDS, GCS, and ours.

Table 1. Differences between CDS, GCS, and SCTD."traj." denotes trajectory.

| | CDS [48] | GCS [22] | SCTD (Ours) |
|---|---|---|---|
| **Self-consistency** | ✓(whole *traj.*) | ✓ (sub *traj.*) | ✓ (sub *traj.*) |
| **Cross-consistency** | ✗ | ✗(whole *traj.*) | ✓(sub *traj.*) |
| **Error bound** | $\mathcal{O}(\Delta_t)T$ | $\mathcal{O}(\Delta_t)(T-e)$ | $\mathcal{O}(\Delta_t)(s_{m+1}-s_m)$ |

### 4.2. Reframing SDS through SCTD

We address the previous limitations by reframing SDS through the lens of SCTD. Specifically, we first define our consistency function as follows: as existing segmented consistency models [33, 44] do, we partition the entire timestep range $[0, T]$ into $N_s$ sub-intervals (*i.e.*, $N_s + 1$ edge points), represented by $\{[s_m, s_{m+1})|m \in \{0, \cdots, N_s - 1\}, s_0 = 0, s_{N_s} = T\}$ (the segmentation strategy is discussed in Sec. 4.3). We define our consistency function $\boldsymbol{G_\theta}$ using Eq. (5), aiming to enforce consistency within each sub-trajectory, *i.e.*, $\boldsymbol{G_\theta}(\mathbf{z}_t, t, s_m, \mathbf{y}) = \boldsymbol{G_\theta}(\mathbf{z}_s, s, s_m, \mathbf{y})$ for $s_m \leq s < t \leq s_{m+1}$. For brevity, we abbreviate $\boldsymbol{G_\theta}(\mathbf{z}_t, t, s_m, \mathbf{y})$ as $\boldsymbol{G_\theta^m}(\mathbf{z}_t, t, \mathbf{y})$. Notably, our consistency function $\boldsymbol{G_\theta^m}$ avoids the issues inherent in GCS, as analyzed in Sec. 4.1 and Supp. B, because at each training stage, the target time step $s_m$ is always uniquely determined.

When CFG [9] with scale $\omega$ is used, the noise residual term in Eq. (3) can be transformed into

$$\begin{aligned}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}}(\mathbf{z}_t, t, \mathbf{y}) - \boldsymbol{\epsilon} = \boldsymbol{\epsilon_\phi}(\mathbf{z}_t, t, \mathbf{y}) - \boldsymbol{\epsilon} \\ + \omega(\boldsymbol{\epsilon_\phi}(\mathbf{z}_t, t, \mathbf{y}) - \boldsymbol{\epsilon_\phi}(\mathbf{z}_t, t, \emptyset)).\end{aligned} \quad (7)$$

Based on Eq. (7), $\mathcal{L}_{\text{SDS}}$ can be expressed in the form of SCTD:

$$\mathcal{L}_{\text{SDS}}(\boldsymbol{\theta}) = \mathbb{E}_{t,s}[b(t)||\underbrace{\boldsymbol{G_\theta^m}(\hat{\mathbf{z}}_s^{\boldsymbol{\Phi}}, s, \mathbf{y}) - \boldsymbol{G_\theta^m}(\mathbf{z}_t, t, \mathbf{y})}_{\text{self-consistency constraint}} + \omega\underbrace{(\boldsymbol{G_\theta^m}(\mathbf{z}_t, t, \emptyset) - \boldsymbol{G_\theta^m}(\mathbf{z}_t, t, \mathbf{y}))}_{\text{cross-consistency constraint}} + \underbrace{\mathbf{z}_{s_m} - \boldsymbol{G_\theta^m}(\hat{\mathbf{z}}_s^{\boldsymbol{\Phi}}, s, \mathbf{y})}_{\text{generative prior}}||_2^2]. \quad (8)$$

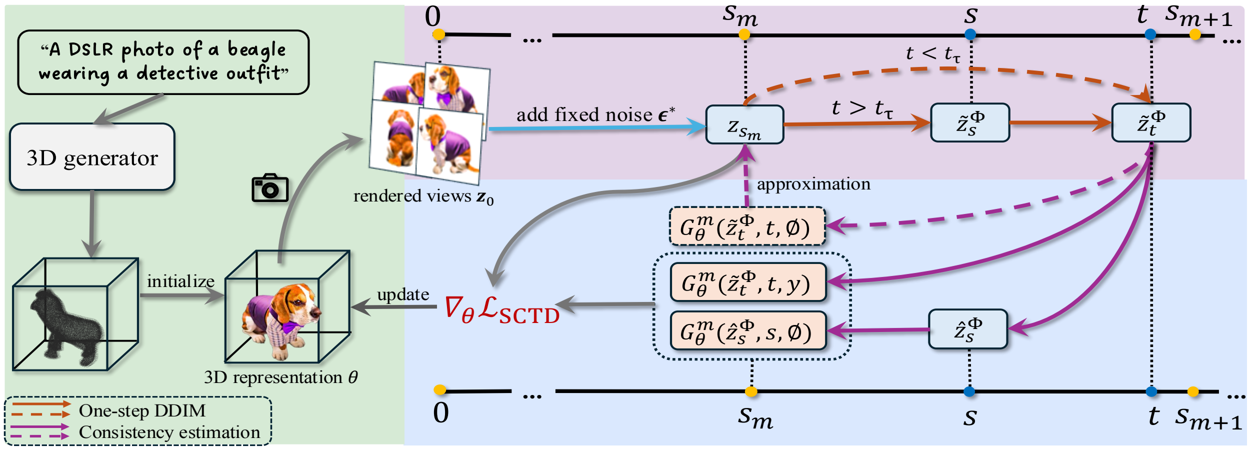


Figure 3. An overview of **SegmentDreamer**: We begin by initializing a 3D representation $\boldsymbol{\theta}$ using a 3D generator, such as Point-E [30]. In each iteration, we randomly render a batch of camera views $\mathbf{z}_0$ from $\boldsymbol{\theta}$ and diffuse them into $\mathbf{z}_{s_m}$ with fixed noise $\boldsymbol{\epsilon}^*$. Next, we transform $\mathbf{z}_{s_m}$ into $\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}}$ using either one-step or two-step unconditional deterministic sampling. During the denoising process, we first estimate $\hat{\mathbf{z}}_s^{\boldsymbol{\Phi}}$ through one-step conditional deterministic sampling from $\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}}$. Subsequently, we compute two consistency functions and utilize them to derive the loss $\mathcal{L}_{\text{SCTD}}$, which is ultimately employed to optimize $\boldsymbol{\theta}$.

where $b(t) = \frac{\omega(t)}{(\alpha_{s_m}\int_{\lambda_t}^{\lambda_{s_m}} e^{-\lambda}\mathrm{d}\lambda)^2}$ and $\hat{\mathbf{z}}_s^{\boldsymbol{\Phi}} = \boldsymbol{\Phi}(\mathbf{z}_t, t, s, \mathbf{y})$. The derivation is given in App. D. The first term on the r.h.s. of Eq. (8) enforces self-consistency across the conditional denoising sub-trajectory, referred to as the "self-consistency constraint." The second term enforces cross-consistency between the unconditional and conditional denoising sub-trajectories, termed the "cross-consistency constraint." The third term adheres to the naming conventions in CSD [53]. As observed, Eq. (8) explicitly defines the relationship between the self-consistency constraint and the cross-consistency constraint. Furthermore, as demonstrated in Sec. 4.3, SCTD provides a tighter upper bound on the distillation error.

## 4.3. SCTD Sampling

**Sampling Design**. A natural question arises: according to Eq. (8), does minimizing $\mathcal{L}_{SDS}$ necessarily enforce both self- and cross-consistency? The answer is no. This is because one cannot necessarily achieve $x_1 = x_2$ and $y_1 = y_2$ when $||x_1 - x_2 + \omega(y_1 - y_2)||_2^2 = 0$. Additionally, we observe that the condition $\mathbf{y}$ appears in all three loss terms in Eq. (8), which increases the computational cost, particularly when using Perp-Neg [1] or ControlNet [56].

To improve the computational efficiency, we perform the following equivalent transformation on Eq. (8):

$$\mathcal{L}_{\text{SDS}}(\boldsymbol{\theta}) = \mathbb{E}_{t,s}[b(t)||\underbrace{\boldsymbol{G}_{\boldsymbol{\theta}}^m(\hat{\mathbf{z}}_s^{\boldsymbol{\Phi}}, s, \emptyset) - \boldsymbol{G}_{\boldsymbol{\theta}}^m(\mathbf{z}_t, t, \emptyset)}_{\text{self-consistency constraint}} + (\omega+1)\underbrace{(\boldsymbol{G}_{\boldsymbol{\theta}}^m(\mathbf{z}_t, t, \emptyset) - \boldsymbol{G}_{\boldsymbol{\theta}}^m(\mathbf{z}_t, t, \mathbf{y}))}_{\text{cross-consistency constraint}} + \underbrace{\mathbf{z}_{s_m} - \boldsymbol{G}_{\boldsymbol{\theta}}^m(\hat{\mathbf{z}}_s^{\boldsymbol{\Phi}}, s, \emptyset)}_{\text{generative prior}}||_2^2]. \tag{9}$$

---

**Algorithm 1** Text-to-3D Generation via SCTD Loss

---

**Initialization:** threshold $t_\tau$, segmentation number $N_s$, and target prompt $\mathbf{y}$
Initialize $\boldsymbol{\theta}$ with a pre-trained 3D generator
**while** $\boldsymbol{\theta}$ is not converged **do**
  Sample $\mathbf{z}_0 = \boldsymbol{g}(\mathbf{c}, \boldsymbol{\theta})$ and $t \sim \mathcal{U}(0, T)$
  Obtain $s_m = max\{s_i | s_i \leq t, i = 0, \cdots, N_s - 1\}$
  Obtain $\mathbf{z}_{s_m} = \alpha_{s_m}\mathbf{z}_0 + \sigma_{s_m}\boldsymbol{\epsilon}^*$
  Sample $s \sim \mathcal{U}(s_m, t)$
  Predict $\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}}$ with Eq. (11) and $\hat{\mathbf{z}}_s^{\boldsymbol{\Phi}} = \boldsymbol{\Phi}(\tilde{\mathbf{z}}_s^{\boldsymbol{\Phi}}, t, s, \mathbf{y})$
  Obtain $\boldsymbol{G}_{\boldsymbol{\theta}}^m(\hat{\mathbf{z}}_s^{\boldsymbol{\Phi}}, s, \emptyset)$ and $\boldsymbol{G}_{\boldsymbol{\theta}}^m(\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}}, t, \mathbf{y})$ with Eq. (5)
  Obtain $\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{SCTD}}$ with Eq. (10) and update $\boldsymbol{\theta}$.

---

The derivation is given in Supp. E.

To enforce self- and cross-consistency, we follow the suggestion from CSD [53] to omit the generative prior from Eq. (9), and then impose a stricter constraint on Eq. (8):

$$\mathcal{L}_{\text{SCTD}}(\boldsymbol{\theta}) = \mathbb{E}_{t,s}[b(t)[||\underbrace{sg(\boldsymbol{G}_{\boldsymbol{\theta}}^m(\hat{\mathbf{z}}_s^{\boldsymbol{\Phi}}, s, \emptyset)) - \boldsymbol{G}_{\boldsymbol{\theta}}^m(\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}}, t, \emptyset)}_{\text{self-consistency constraint}}||_2^2 + ||(\omega+1)\underbrace{(\boldsymbol{G}_{\boldsymbol{\theta}}^m(\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}}, t, \emptyset) - sg(\boldsymbol{G}_{\boldsymbol{\theta}}^m(\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}}, t, \mathbf{y})))}_{\text{cross-consistency constraint}}||_2^2]]. \tag{10}$$

where $\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}}$ is obtained through multi-step deterministic sampling from $\mathbf{z}_0$, as discussed in Sec. 4.4. $\mathbf{z}_{s_m} = \alpha_{s_m}\mathbf{z}_0 + \sigma_{s_m}\boldsymbol{\epsilon}^*$, with $\boldsymbol{\epsilon}^*$ fixed throughout the entire training. $\hat{\mathbf{z}}_s^{\boldsymbol{\Phi}} = \boldsymbol{\Phi}(\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}}, t, s, \mathbf{y})$, where $\boldsymbol{\Phi}$ is implemented by a first-order ODE solver in this work. $sg(\cdot)$ represents the stopping of gradient backpropagation.

**Trajectory Segmentation Strategy**. We propose two heuristic trajectory segmentation strategies: equal segmentation and monotonically increasing segmentation. In the first strategy, the entire time-step range $[0,T]$ is uniformly divided into $N_s$ segments, *i.e.*, $s_{m+1}-s_m=\frac{T}{N_s}$. In the second strategy, we adopt a monotonically increasing interval segmentation method, where the $N_s$ intervals satisfy that $s_{m+1}-s_m=t_\tau+\frac{2m(T-N_s t_\tau)}{N_s(N_s-1)}$ for $\in\{0,\cdots,N_s-1\}$. A comparison of the two strategies is presented in Sec. 6.

### 4.4. Swift and Stable Optimization Pipeline

In previous work [22, 23, 48], $\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}}$ is derived from deterministic sampling with a fixed number of steps, lacking flexibility. In this paper, we adopt the following dynamic adjustment strategy: .

$$\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}}=\begin{cases}\boldsymbol{\Phi}(\boldsymbol{\Phi}(\mathbf{z}_{s_m},s_m,s,\emptyset),s_m,t,\emptyset), & \text{if } t>t_\tau\\ \boldsymbol{\Phi}(\mathbf{z}_{s_m},s_m,t,\emptyset), & \text{if } t\le t_\tau\end{cases}, \tag{11}$$

This strategy enables us to make a trade-off between generation quality and optimization speed, as evidenced in Sec. 6.

**Consistency function Approximation**. For faster optimization, we propose approximating $\boldsymbol{G}_{\boldsymbol{\theta}}^m(\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}},t,\emptyset)$ as $\mathbf{z}_{s_m}$, leveraging the theoretical invertibility of the unconditional PF-ODE trajectory defined in Eq. (11). However, since both $\boldsymbol{\Phi}$ and $\boldsymbol{G}_{\boldsymbol{\theta}}^m$ are implemented using a first-order ODE solver in this paper, perfect invertibility cannot be guaranteed. Nevertheless, we find that this approximation strategy not only eliminates the need to compute the U-Net Jacobian—reducing computational cost—but also improves optimization stability, as demonstrated in Sec. 6.

### 4.5. Theoretical Analysis of Distillation Error

We theoretically prove that our SCTD loss leads to a tighter upper bound on the distillation error, which explains why SCTD achieves superior generative results. Specifically, the SDS loss can be reformulated as:

$$\begin{aligned}\mathcal{L}_{\text{SDS}}&=\mathbb{E}_t[\omega(t)||\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}},t,\mathbf{y})-\boldsymbol{\epsilon}^*||_2^2]\\&=\mathbb{E}_t[b(t)||\mathbf{z}_{s_m}-\boldsymbol{G}_{\boldsymbol{\theta}}^m(\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}},t,\mathbf{y})||_2^2].\end{aligned} \tag{12}$$

Based on Eq. (12), we have the following theorem:

**Theorem 1.** *Given a sub-trajectory $[s_m,s_{m+1}]$, let $\Delta t=max_{t,s\in(s_m,s_{m+1}]}\{|t-s|\}$. We assume $\boldsymbol{G}_{\boldsymbol{\theta}}^m$ satisfies the Lipschitz condition. If there exists a $\mathbf{z}_0$ satisfying $\boldsymbol{G}_{\boldsymbol{\theta}}^m(\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}},t,\mathbf{y})=\boldsymbol{G}_{\boldsymbol{\theta}}^m(\hat{\mathbf{z}}_s^{\boldsymbol{\Phi}},s,\mathbf{y})$ for $\forall t,s\in[s_m,s_{m+1}]$ and a true image $\mathbf{z}^{data}\sim\mathbb{P}_{data}(\mathbf{z})$ (corresponding to $\mathbf{z}_0$), we have*

$$\sup_{t,s\in[s_m,s_{m+1})}||\mathbf{z}_0-\mathbf{z}^{data}||=\mathcal{O}(\Delta t)(s_{m+1}-s_m). \tag{13}$$

*Proof.* The full proof, inspired by consistency model-related works [22, 27, 44, 48], is provided in App. F. □

Theorem 1 shows that our SCTD loss achieves an arbitrarily small distillation error when the ODE solver's step size $\Delta_t$ and the segment length $s_{m+1}-s_m$ are sufficiently small. By choosing an appropriate segment number $N_s$, we obtain a tighter error upper bound, which improves upon the bound $\mathcal{O}(\Delta t)(T)$ in CDS [48] and $\mathcal{O}(\Delta t)(T-e)$ in GCS [22]. This refinement enables more accurate gradients for optimizing the 3D representation $\boldsymbol{\theta}$, leading to higher-quality generative results.

## 5. Experiments

### 5.1. Implementation Details

Our algorithm is implemented in PyTorch and trained on an A100 GPU for 5,000 iterations using the Adam optimizer [19]. We utilize Stable Diffusion 2.1 [34] as our foundational generative model and employ Point-E [30] to initialize 3D Gaussians. Hyperparameters related to 3D Gaussian Splatting (3DGS), such as learning rates and camera poses, are set following the guidelines in previous works [22, 23]. We have $t\sim\mathcal{U}(20,500+t_{warm})$, where $t_{warm}$ decreases linearly from 480 to 0 during the first 1,500 epochs.

### 5.2. Text-to-3D Generation

**Qualitative Comparison.** We present a collection of 3D assets generated by SegmentDreamer from different camera views in Fig. 1. Our algorithm effectively produces high-quality, text-aligned 3D assets. To highlight the superior generative capability of SegmentDreamer, we conduct qualitative experiments comparing it with state-of-the-art methods based on Consistent Distillation (CD)—namely, Consistent3D [48] and Connect3D [22]—as well as Score Distillation (SD)-based methods, including DreamFusion [31] and LucidDreamer [23], as illustrated in Fig. 4. For a fair comparison, all methods use Stable Diffusion 2.1 as the foundational generative model. As shown in Fig. 4, both DreamFusion and Consistent3D suffer from severe Janus artifacts and fail to generate accurate conditional outputs. LucidDreamer encounters issues with overexposure and ineffective conditional generation. Connect3D performs well in both conditional generation and color rendering; however, its results still contain undesired artifacts, particularly in the tiger and beagle models (see Fig. 4). Moreover, as a 3DGS-based method, it requires over an hour to generate a single 3D asset, limiting its practical usability. In contrast, SegmentDreamer can generate high-quality results within about 32 minutes using CFG [9] and 38 minutes using Perp-Neg [1], proving its efficiency and superiority. More comparison results are shown in our supplementary material.

We also compare our SCTD loss with existing CD-based losses, namely CDS [48] and GCS [22]. For a fair comparison, we use Stable Diffusion 2.1 as the generative model and keep the random seed, guidance scale, and 3DGS-

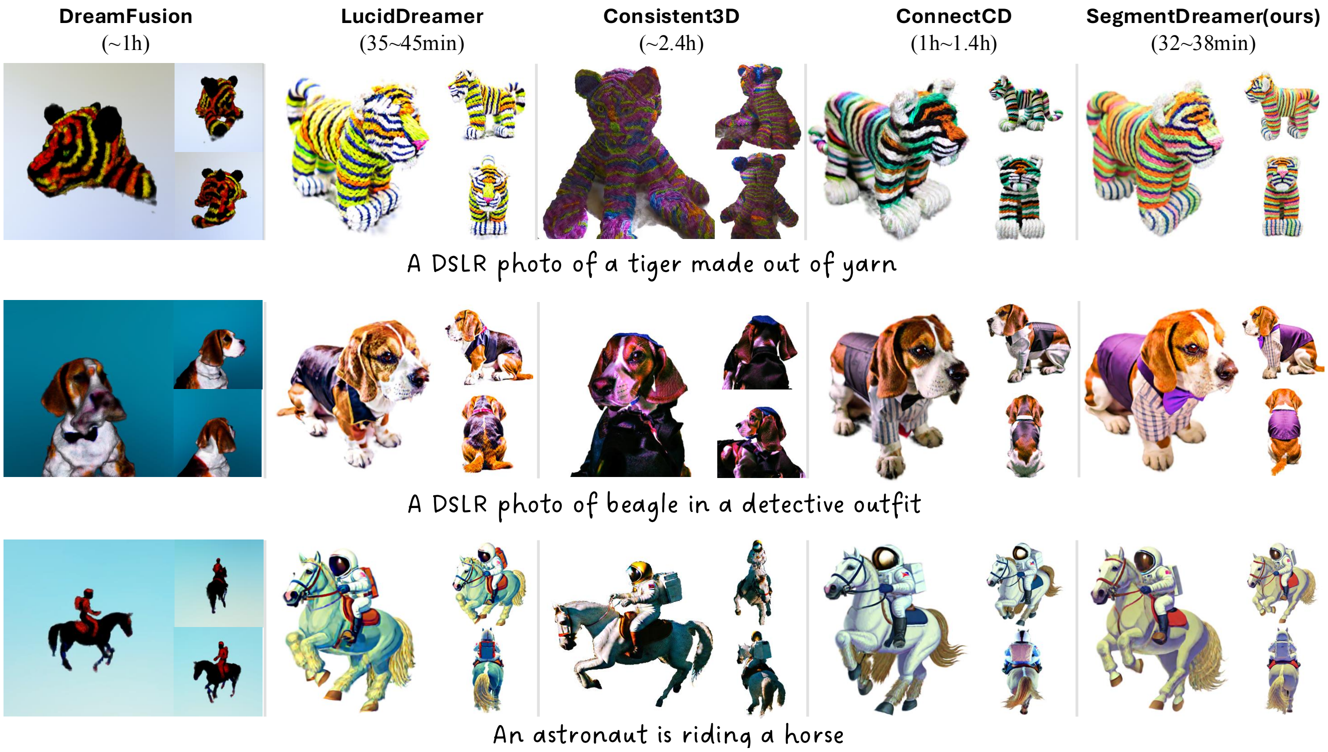


Figure 4. Qualitative comparison results of two SD-based metheds: DreamFusion [31] and LucidDreamer [23], and two CD-based methods: Consistent3D [48] and ConnectCD [22]. CFG scales are set to 100, 7.5, 20∼40, 7.5, 7.5, respectively. Our approach yields high-quality results within the shortest time. Please zoom in for details.

related parameters consistent across all methods. As shown in Fig. 5, CDS struggles to provide effective conditional guidance at a normal CFG scale, as also noted in [22]. Even at a higher CFG scale, its results remain unsatisfactory due to the lack of cross-consistency in its distillation loss. GCS, meanwhile, suffers from overexposure and artifacts due to excessive conditional guidance. While it attempts to address this with Brightness-Equalized Generation (BEG), artifacts persist (see Fig. 5; zoom in for details). In contrast, our SCTD mitigates ineffective conditional generation, overexposure, and artifacts, striking a better balance between self-consistency and cross-consistency.

**Quantitative Comparison.** We evaluate the text alignment and human preference of each text-to-3D method as follows: we generate 3D models from 40 prompts selected from the DreamFusion Gallery [1], and calculate the average CLIP Score and ImageReward (IR) score across 120 camera views for each model. Following Connect3D [22], we compute the FID score between 40 objects (with 120 rendered views for each) and 50,000 images generated by Stable Diffusion 2.1. As shown in the results, our method achieves the best performance in text alignment, human preference, and fidelity, while requiring the shortest generation time, demonstrating its superior overall performance. The 40 prompts used are provided in the supplementary material.

[1]https://dreamfusion3d.github.io/gallery.html

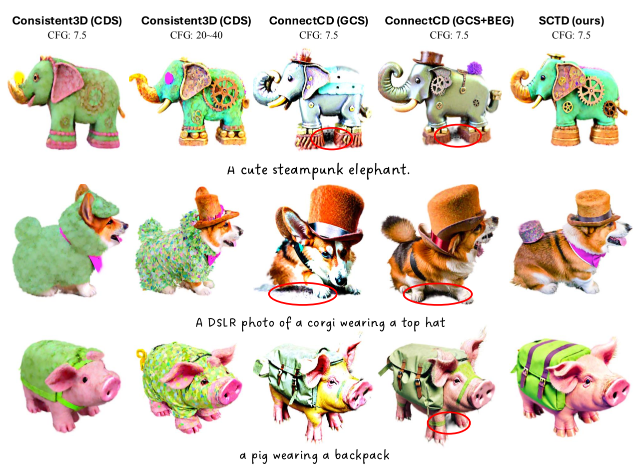


Figure 5. Visual comparison results of CDS [48], GCS [22], and GCS+BEG [22]. Artifacts are marked with red circles.

**User Study.** We conduct a user study involving 40 volunteers to evaluate the generation quality of each algorithm. Each volunteer is asked to review eight groups of five 360-degree rendered videos and to rank them based on three


Figure 6. Ablation study of the trajectory segmentation strategy with different segment numer $N_s$ and time theshold $t_\tau$. Prompt: *a DSLR photo of the Imperial State Crown of England.*

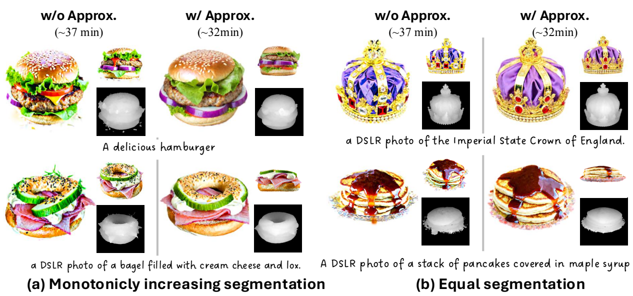


Figure 7. Visual comparison results before and after performing the approximation strategy proposed in Sec. 4.3. "*w/ Approx.*" and "*w/o Approx.*" indicate whether the approximation strategy is used or not. The GFG scale is set to 7.5 for all generations.

Table 2. Quantitative comparison using CLIP score, IR score, training time (min), FID score, and three user study questions given by 40 volunteers. **Bold** and <u>underline</u> texts denote the best and second-best performance, respectively.

| | CLIP-L↑ | IR ↑ | FID ↓ | Time ↓ | **Q1**↓ | **Q2**↓ | **Q3**↓ |
|---|---|---|---|---|---|---|---|
| DreamFusion[31] | 28.47 | -0.004 | 140.84 | 60 | 4.73 | 4.87 | 4.90 |
| LucidDreamer[23] | 29.99 | 0.006 | 121.80 | <u>45</u> | 2.93 | 2.98 | 2.93 |
| Consistent3D[48] | 30.60 | 0.004 | 113.61 | 140 | 4.14 | 3.88 | 3.93 |
| ConnectCD [22] | <u>30.73</u> | <u>0.018</u> | <u>112.61</u> | 80 | <u>1.63</u> | <u>1.80</u> | <u>1.93</u> |
| **SegmentDreamer** | **30.88** | **0.020** | **110.45** | **38** | **1.57** | **1.47** | **1.33** |

criteria: text alignment (**Q1**), object realism (**Q2**), and object details (**Q3**; e.g., artifacts, brightness, and the Janus problem). To minimize bias, the order of the videos is randomized. As shown in Tab. 2, users consistently prefer our method across all criteria.

## 6. Ablation Study

**Effect of Trajectory Segmentation Strategy** We visualize the effect of different trajectory segmentation strategies applied to our algorithm in Fig. 6. When the segment number $N_s$ and the time threshold $t_\tau$ are fixed, we observe that the segmentation strategy has minimal impact on the generated results. This is primarily because the segmentation strategy essentially determines $\sup |t - e|$. However, since $t$ and $s$ are randomly selected, $|t - s_m|$, $|t - s|$, and $|s - s_m|$ vary randomly during each training iteration, and their values can be either large or small.

**Effect of Segment Number** $N_s$. We visualize the effect of different segment numbers $N_s$ in Fig. 6. As observed, a small $N_s$, regardless of the segmentation strategy, consistently results in overly smoothed outputs. As $N_s$ increases, the generation quality improves. However, when $N_s$ becomes too large (*e.g.* 10), while more details emerge, the overall model representation becomes vague—a phenomenon also noted in LucidDreamer [23]. We find that $N_s = 5$ performs well in most cases.

**Effect of time threshold** $t_\tau$. As shown in Fig. 6, a large $t_\tau$ leads to overly smoothed results for both segmentation strategies. This is because a limited number of sampling steps in the diffusion process fail to effectively preserve the information of the initial point $\mathbf{z}_0$, highlighting the necessity of incorporating multi-step deterministic sampling.

**Validation of Approximation Strategy**. We analyze the impact of replacing $\boldsymbol{G_\theta}(\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}}, t, \mathbf{y})$ with $\mathbf{z}_{s_m}$ in Fig. 7. As shown, $\boldsymbol{G_\theta}(\tilde{\mathbf{z}}_t^{\boldsymbol{\Phi}}, t, \mathbf{y})$ introduces slight overexposure and geometric distortion. In contrast, $\mathbf{z}_{s_m}$ not only produces better generation results but also reduces the training time. This suggests a gap between theory and practice, similar to the omission of U-Net Jacobian in DreamFusion [31].

## 7. Conclusion

We analyze improper conditional guidance in previous Consistency Distillation methods and identify its root cause as an imbalance between self-consistency and cross-consistency in denoising trajectories. Additionally, we find that large distillation error upper bounds limit generative capability. To address this, we propose the Segmented Consistency Trajectory Distillation loss, which improves conditional guidance and tightens the distillation bound. Experiments show that our SegmentDreamer achieves high-quality generation efficiently, advancing distillation-based text-to-3D methods toward practical use.

**Limitation.** SegmentDreamer mainly focuses on single-instance generation and performs suboptimally in multi-instance generation.

**Potential Negative Impact.** SegmentDreamer focuses on content synthesis, which may be used to create misleading content.

## Acknowledgements

This project is supported by the National Natural Science Foundation of China (12326618, U22A2095), the National Key Research and Development Program of China (2024YFA1011900), the Major Key Project of PCL (PCL2024A06), and the Project of Guangdong Provincial Key Laboratory of Information Security Technology (2023B1212060026).

## References

[1] Mohammadreza Armandpour, Ali Sadeghian, Huangjie Zheng, Amir Sadeghian, and Mingyuan Zhou. Re-imagine the negative prompt algorithm: Transform 2d diffusion into 3d, alleviate janus problem and beyond. *arXiv preprint arXiv:2304.04968*, 2023. 1, 5, 6

[2] 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 *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 5855–5864, 2021. 3

[3] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 22246–22256, 2023. 2, 3

[4] Zixuan Chen, Lingxiao Yang, Jian-Huang Lai, and Xiaohua Xie. Cunerf: Cube-based neural radiance field for zero-shot medical image arbitrary-scale super resolution. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 21185–21195, 2023. 3

[5] Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3d using gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21401–21412, 2024. 2, 3

[6] Zixuan Chen, Guangcong Wang, Jiahao Zhu, Jian-Huang Lai, and Xiaohua Xie. Guardsplat: Efficient and robust watermarking for 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3

[7] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 31841–31854, 2022. 3

[8] Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oğuz. 3dgen: Triplane latent diffusion for textured mesh generation. *arXiv preprint arXiv:2303.05371*, 2023. 3

[9] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *Proceedings of the International Conference on Neural Information Processing Systems Workshop (NeurIPS Workshop)*, 2021. 1, 4, 6

[10] Zhipeng Hu, Minda Zhao, Chaoyi Zhao, Xinyue Liang, Lincheng Li, Zeng Zhao, Changjie Fan, Xiaowei Zhou, and Xin Yu. Efficientdreamer: High-fidelity and robust 3d creation via orthogonal-view diffusion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4949–4958, 2024. 2, 3

[11] Yukun Huang, Jianan Wang, Yukai Shi, Boshi Tang, Xianbiao Qi, and Lei Zhang. Dreamtime: An improved optimization strategy for diffusion-guided 3d generation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. 2, 3

[12] Chenhan Jiang, Yihan Zeng, Tianyang Hu, Songcun Xu, Wei Zhang, Hang Xu, and Dit-Yan Yeung. Jointdreamer: Ensuring geometry consistency and text congruence in text-to-3d generation via joint score distillation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 439–456. Springer, 2025. 2, 3

[13] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 3

[14] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. In *ACM SIGGRAPH*, pages 165–174. ACM New York, NY, USA, 1984. 3

[15] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 26565–26577, 2022. 3

[16] Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. Noise-free score distillation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 2, 3

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ACM TOG)*, 42(4):139–1, 2023. 2, 3

[18] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 2, 3, 4

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[20] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. *Computer Graphics Forum (CGF)*, 40 (4), 2021. 3

[21] Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12642–12651, 2023. 3

[22] Zongrui Li, Minghui Hu, Qian Zheng, and Xudong Jiang. Connecting consistency distillation to score distillation for text-to-3d generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 274–291. Springer, 2025. 2, 3, 4, 6, 7, 8

[23] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6526, 2024. 6, 7, 8

[24] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 300–309, 2023. 2, 3

[25] Zexiang Liu, Yangguang Li, Youtian Lin, Xin Yu, Sida Peng, Yan-Pei Cao, Xiaojuan Qi, Xiaoshui Huang, Ding Liang, and Wanli Ouyang. Unidream: Unifying diffusion priors for relightable text-to-3d generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 74–91. Springer, 2025. 2, 3

[26] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 5775–5787, 2022. 3, 4

[27] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 3, 4, 6

[28] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18039–18048, 2024. 3

[29] 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*, 65(1):99–106, 2021. 2, 3

[30] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 3, 5, 6

[31] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–10, 2022. 2, 3, 6, 7, 8

[32] Thomas Porter and Tom Duff. Compositing digital images. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, pages 253–259, 1984. 3

[33] Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 2, 4

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 2, 3, 6

[35] Junyoung Seo, Wooseok Jang, Min-Seop Kwak, Hyeonsu Kim, Jaehoon Ko, Junho Kim, Jin-Hwa Kim, Jiyoung Lee, and Seungryong Kim. Let 2d diffusion model know 3d-consistency for robust text-to-3d generation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 2, 3

[36] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1606–1616, 2024. 3

[37] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 2, 3

[38] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 3

[39] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 3

[40] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the International Conference on Machine Learning (ICLR)*, 2023. 2, 3, 4

[41] Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10208–10217, 2024. 3

[42] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 3

[43] Uy Dieu Tran, Minh Luu, Phong Ha Nguyen, Khoi Nguyen, and Binh-Son Hua. Diverse text-to-3d synthesis with augmented text embedding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 217–235. Springer, 2025. 2, 3

[44] Fu-Yun Wang, Zhaoyang Huang, Alexander William Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, et al. Phased consistency model. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 2, 4, 6

[45] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 9065–9076, 2023. 3

[46] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. 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)*, pages 12619–12629, 2023. 2

[47] 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. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 8406–8441, 2023. 2, 3

[48] Zike Wu, Pan Zhou, Xuanyu Yi, Xiaoding Yuan, and Hanwang Zhang. Consistent3d: Towards consistent high-fidelity text-to-3d generation with deterministic sampling prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9892–9902, 2024. 2, 3, 4, 6, 7, 8

[49] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024. 3

[50] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19595–19604, 2024. 3

[51] Junliang Ye, Fangfu Liu, Qixiu Li, Zhengyi Wang, Yikai Wang, Xinzhou Wang, Yueqi Duan, and Jun Zhu. Dreamreward: Text-to-3d generation with human preference. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 259–276. Springer, 2025. 2, 3

[52] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6796–6807, 2024. 3

[53] Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. Text-to-3d with classifier score distillation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 2, 3, 4, 5

[54] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19447–19456, 2024. 3

[55] Ye Yuan, Xueting Li, Yangyi Huang, Shalini De Mello, Koki Nagano, Jan Kautz, and Umar Iqbal. Gavatar: Animatable 3d gaussian avatars with implicit mesh learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 896–905, 2024. 3

[56] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 3836–3847, 2023. 5

[57] Junzhe Zhu, Peiye Zhuang, and Sanmi Koyejo. Hifa: High-fidelity text-to-3d generation with advanced diffusion guidance. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 2, 3

[58] Wenjie Zhuo, Fan Ma, Hehe Fan, and Yi Yang. Vividdreamer: Invariant score distillation for hyper-realistic text-to-3d generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 122–139. Springer, 2025. 2, 3