DreamTime: An Improved Optimization Strategy for Text-to-3D Content Creation

Yukun Huang[1,2*†], Jianan Wang[1*‡], Yukai Shi[1], Xianbiao Qi[1], Zheng-Jun Zha[2], Lei Zhang[1]
[1]International Digital Economy Academy (IDEA)
[2]University of Science and Technology of China
kevinh@mail.ustc.edu.cn, {wangjianan,shiyukai,qixianbiao}@idea.edu.cn,
zhazj@ustc.edu.cn, leizhang@idea.edu.cn

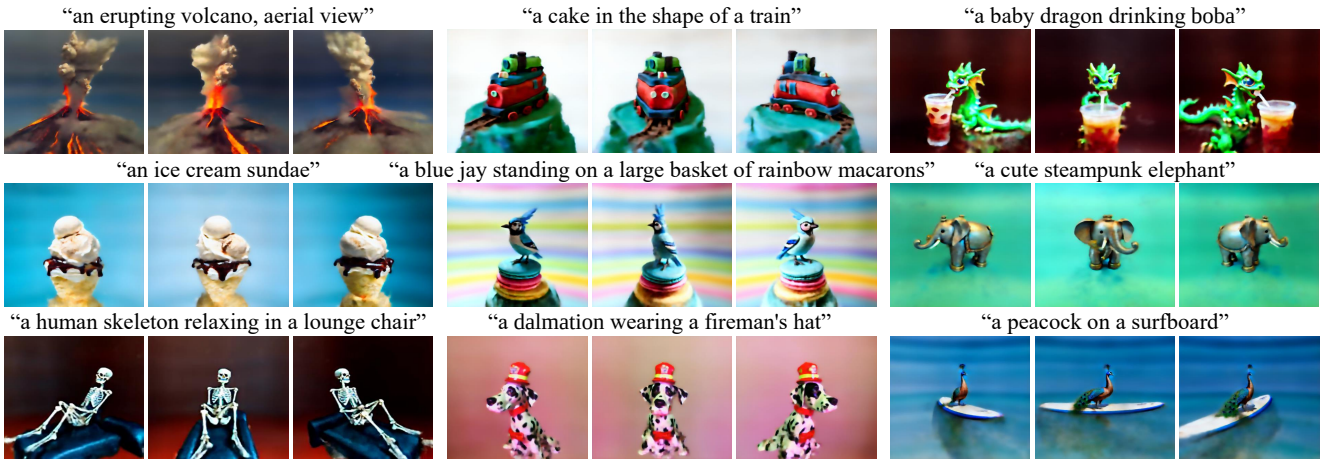


Figure 1: Results for text-driven 3D content generation using DreamTime.

Abstract

Text-to-image diffusion models pre-trained on billions of image-text pairs have recently enabled text-to-3D content creation by optimizing a randomly initialized Neural Radiance Fields (NeRF) with score distillation. However, the resultant 3D models exhibit two limitations: (a) quality concerns such as saturated color and the Janus problem; (b) extremely low diversity comparing to text-guided image synthesis. In this paper, we show that the conflict between NeRF optimization process and uniform timestep sampling in score distillation is the main reason for these limitations. To resolve this conflict, we propose to prioritize timestep sampling with monotonically non-increasing functions, which aligns NeRF optimization with the sampling process of diffusion model. Extensive experiments show that our simple redesign significantly improves text-to-3D content creation with higher quality and diversity.

1. Introduction

Humans are situated in a 3D environment. To simulate this experience for entertainment or research, we require a significant number of 3D object assets to populate virtual environments like games and robotics simulations. Generating such 3D content is both expensive and time-consuming, necessitating skilled artists with extensive aesthetic and 3D modeling knowledge. It's reasonable to inquire whether we can enhance this procedure to make it less arduous and allow beginners to create 3D content that reflects their own experiences and aesthetic preferences.

Recent advancements in text-to-image generation [30–32] have democratized image creation, enabled by large-scale image-text datasets, *e.g.* Laion5B [33], scraped from the internet. However, 3D data is not as easily accessible, making 3D generation with 2D supervision very attractive. Previous works [3, 4, 22, 23, 26, 34] have explored supervising 2D renderings of 3D models using adversarial loss, but they are limited to modeling a single domain. Dream Fields [11] and CLIPmesh [19] replace adversarial supervision with CLIP [28] to discriminate alignment of a 3D

*Equal contribution.
†Work done during an internship at IDEA.
‡Corresponding author.

model's 2D renderings with given text prompt. Since CLIP is pre-trained on large-scale image-text pairs, this approach is able to create general objects but often in low quality with unrealistic appearance. To harness richer supervision other than discriminative guidance from GAN or CLIP, recent works [15, 17, 27, 40] have utilized pre-trained text-to-image diffusion models as a strong image prior to supervise 2D renderings of 3D models, with promising showcases for text-to-3D generation. However, challenges remain for creative content creation, as the resultant 3D models are often of low diversity with quality concerns such as saturated colors and the Janus (multi-face) problem.

As a class of score-based generative models [10, 37, 38], diffusion models contain a data noising and a data denoising process according to a predefined schedule over fixed number of timesteps. They model the denoising score $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$, which is the gradient of the log-density function with respect to the data on a large number of noise-perturbed data distributions. Each timestep (t) corresponds to a fixed noise with the score containing coarse-to-fine information as t decreases. For image synthesis, the sampling process respects the discipline of coarse-to-fine content creation by iteratively refining samples with monotonically decreasing t. However, the recent works leveraging pre-trained data scores for text-to-3D generation [15, 27, 31, 40] randomly sample t during the process of 3D model optimization, which is counter-intuitive.

In this paper, we first investigate what a 3D model learns from pre-trained text-to-image diffusion models at each noise level. Our key intuition is that pre-trained diffusion models provide different levels of visual concepts for different noise levels. At 3D model initialization, it needs coarse high-level information for structure formation. Later optimization steps should instead focus on refining details for better visual quality. These observations motivate us to propose time prioritized score distillation sampling (TP-SDS) for text-to-3D generation, which aims to prioritize information from different diffusion timesteps (t) at different stages of 3D optimization. More concretely, we propose a non-increasing timestep sampling strategy: at the beginning of optimization, we prioritize the sampling of large t for guidance on global structure, and then gracefully decrease t with training iteration to get more information on visual details. To validate the effectiveness of the proposed TP-SDS, we first analyze the score distillation process illustrated on 2D examples. We then evaluate TP-SDS against standard SDS on a wide range of text-to-3D generations in comparison for model quality and diversity.

Our main contributions are as follows:

- We thoroughly reveal the conflict between text-to-3D optimization and uniform timestep sampling of score distillation sampling (SDS) from three perspectives: mathematical formulation, gradient visualization and frequency analysis.
- To resolve this conflict, we introduce DreamTime, an improved optimization strategy for text-to-3D content creation. Concretely, we propose to use a non-increasing time sampling strategy instead of uniform time sampling. The introduced strategy is simple but effective because it aligns text-to-3D optimization with the sampling process of DDPM [10].
- We conduct extensive experiments and show that our simple redesign of the optimization process significantly improves text-to-3D generation with higher quality and diversity.

2. Related Work

Text-to-image generation. Recently, text-to-image models such as GLIDE [24], unCLIP [30], Imagen [32], and Stable Diffusion [31] have demonstrated an impressive capability of generating photorealistic and creative images given textual instructions. The remarkable progress is enabled by advances in modeling such as diffusion models [7, 25, 36], as well as large-scale web data curation exceeding billions of image-text pairs [5, 33, 35]. Such datasets have wide coverage of general objects, likely containing instances with great variety such as color, texture and camera viewpoints. As a result, text-to-image diffusion models pre-trained on those billions of image-text pairs exhibit remarkable understanding of general objects, good enough to synthesize them with high quality and diversity. However, generating different viewpoints of the same object respecting its structure and appearance remains a challenging problem [41].

3D generation with 2D supervision. Compared to easily available images, it is challenging to obtain a large amount of training data in the form of 3D assets. To address this issue and to enable training of 3D generative models using only unstructured 2D images, previous researches such as pi-GAN [4], EG3D [3], GRAF [34], and GIRAFFE [26] have explored supervising 2D renderings of 3D models through adversarial loss against a collection of 2D images. While these methods have shown great promises, they are limited to modelling a single domain, such as human [13] and animal [12] faces, which hampers scalability and creative control for 3D content creation. This paper focuses on using image generative models pre-trained on large-scale image-text pairs for text-to-3D generation.

Text-to-3D generation. The pioneering works of Dream Fields [11] and CLIPmesh [19] utilize CLIP [28] to optimize the underlying 3D representation so that its 2D renderings align well with user-provided text prompt, without

requiring expensive 3D training data. However, this approach tends to produce less realistic 3D models because CLIP only offers discriminative supervision on high-level semantics. In contrast, recent studies such as DreamFusion [27] and Magic3D [15] have demonstrated remarkable text-to-3D generation results by employing powerful text-to-image diffusion models as a robust 2D prior. We build upon this line of work and improve over the design choice of 3D model optimization process to enable significantly higher-fidelity and higher-diversity text-to-3D generation.

3. Method

We first review relevant methods, including NeRF [18], Stable Diffusion [31] and SDS [27] in Section 3.1. Then, we analyze the existing drawbacks of SDS in Section 3.2. Finally, to alleviate the problems in SDS, we introduce an improved optimization strategy in Section 3.3.

3.1. Preliminary

Neural Radiance Fields (NeRF) [2, 18, 20] has shown outstanding performance on novel view synthesis task by modelling a scene with implicit representation parameterized by a trainable MLP. Given the camera position $\mathbf{o}$ and direction $\mathbf{d}$, a batch of rays $\mathbf{r}(k) = \mathbf{o} + k\mathbf{d}$ is sampled to render a pixel. The MLP takes $\mathbf{r}(k)$ as input and predicts the density τ and color c. The final rendered color of one pixel is approximated by the volume rendering integral using numerical quadrature:

$$\hat{C}_c(\mathbf{r}) = \sum_{i=1}^{N_c} \Omega_i (1 - \exp(-\tau_i \delta_i)) c_i,$$

where N_c is the number of sampled points on a ray, the accumulated transmittance $\Omega_i = \exp(-\sum_{j=1}^{i-1} \tau_j \delta_j)$, and δ_i is the distance between adjacent samples.

Diffusion models [10, 25] estimate the denoising score $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$ by adding noise to clean data $\mathbf{x} \sim p(\mathbf{x})$ in T timesteps with pre-defined schedule $\alpha_t \in (0, 1)$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$, according to:

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{x} + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (1)$$

then learns to denoise by minimizing the noise prediction error:

$$\mathcal{L}_t = \mathbb{E}_{\mathbf{x}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\left\| \boldsymbol{\epsilon}_\phi (\mathbf{z}_t, t) - \boldsymbol{\epsilon} \right\|_2^2 \right].$$

In the sampling stage, one can derive $\mathbf{x}$ from noisy input and noise prediction, and subsequently the score of data distribution.

Score Distillation Sampling (SDS) [15, 17, 27] is a widely used method to distill 2D image priors from a pre-trained diffusion model $\boldsymbol{\epsilon}_\phi$ into differentiable 3D representations. Given a differentiable generator g and a NeRF model parameterized by $\boldsymbol{\theta}$, its rendered image $\mathbf{x}$ can be obtained by $\mathbf{x} = g(\boldsymbol{\theta})$. Then, SDS calculates the gradients of NeRF parameters $\boldsymbol{\theta}$ by:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{SDS}}(\phi, \mathbf{x}) = \mathbb{E}_{t, \boldsymbol{\epsilon}} \left[w(t) (\boldsymbol{\epsilon}_\phi(\mathbf{x}_t; y, t) - \boldsymbol{\epsilon}) \frac{\partial \mathbf{z}_t}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\theta}} \right],$$

where $w(t)$ is a weighting function that depends on the timestep t and y is the text embedding of given prompt. SDS optimization is robust to the choice of $w(t)$ as mentioned in [27].

Remark. Our target is to optimize a NeRF model by distilling knowledge from pre-trained Stable Diffusion [31] given a text prompt. In the training process, SDS is used to supervise the distillation process.

3.2. Analysis of Existing Drawbacks in SDS

A diffusion model generates an image by sequentially denoising a noisy image, where the denoising signal provides different granularity of information at different timestep (t), from structure to details [1, 6]. For diffusion-guided 3D content generation, however, SDS [27] samples t from a uniform distribution throughout the NeRF optimization process, which is counter-intuitive because the nature of 3D generation is closer to DDPM sampling (sequential t-sampling) than DDPM training (uniform t-sampling). This motivates us to explore the potential impact of uniform t-sampling on text-to-3D generation.

In this subsection, we analyze the drawbacks of SDS from three perspectives: mathematical formulation, gradient visualization and frequency analysis.

Mathematical formulation. We contrast SDS loss:

$$\mathcal{L}_{\text{SDS}}(\phi, \mathbf{z}_t) = \mathbb{E}_{t \sim \mathcal{U}(1,T)} \left[w(t) \|\boldsymbol{\epsilon}_\phi(\mathbf{z}_t; y, t) - \boldsymbol{\epsilon}\|_2^2 \right] \quad (2)$$

with DDPM sampling process, *i.e.*, for $t = T \rightarrow 1$:

$$\mathbf{z}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{z}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\phi(\mathbf{z}_t; y, t) \right) + \sigma_t \boldsymbol{\epsilon}, \quad (3)$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, α_t is training noise schedule and σ_t is noise variance, *e.g.*, $\sqrt{1 - \alpha_t}$.

Note that for SDS training, t is randomly sampled as shown in Eq. 2 with *red* color, but for DDPM sampling, t is strictly ordered for Eq. 3 as highlighted in *blue*. Since diffusion model is a general denoiser best utilized by iteratively transforming noisy content to less noisy ones, we argue that random timestep sampling in the optimization process of NeRF is ***unaligned*** with the sampling process in DDPM. Such misalignment leads to ineffective and inaccurate supervision from SDS in the training process.

Gradient visualization. For diffusion models, the denoising prediction provides different granularity of information at different timestep t: from coarse structure to fine

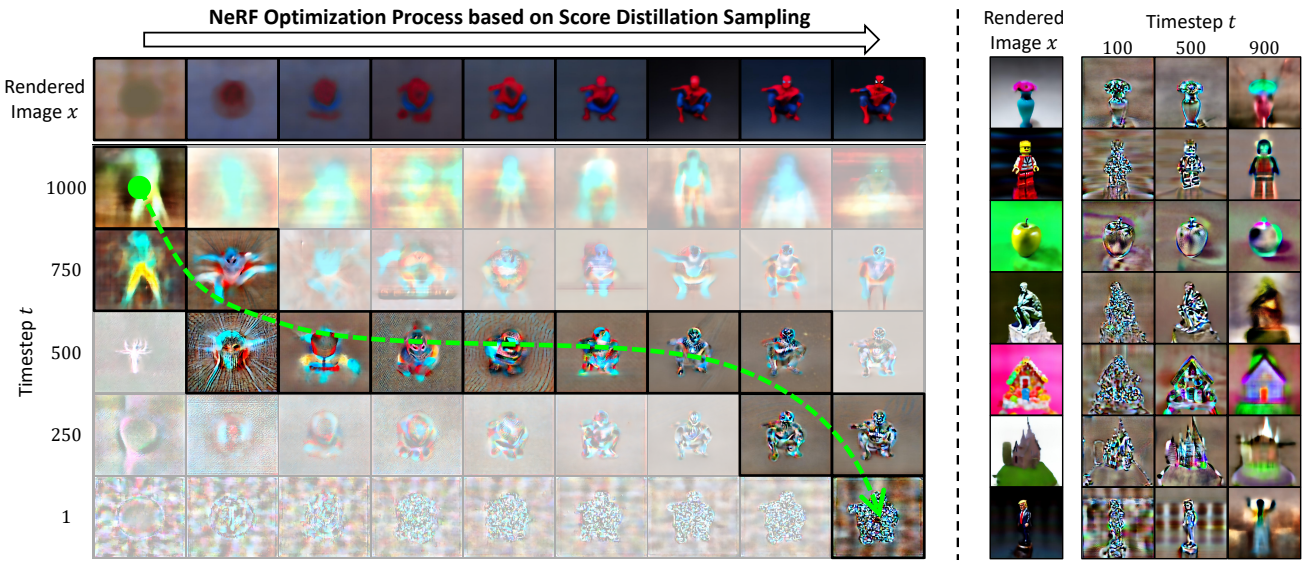


Figure 2: Visualization of SDS gradients under different timesteps t. (Left) Visualization of SDS gradients throughout the NeRF optimization process, where green curved arrow denotes the path for more informative gradient directions as NeRF optimization progresses. It can be observed that a non-increasing timestep t is more suitable for the SDS optimization process. (Right) We provide more examples to illustrate the effects of timestep t on SDS gradients. Small t provides guidance on local details, while large t is responsible for global structure.

details as t decreases. To demonstrate this, we visualize the update gradient $\|\epsilon_\phi(\mathbf{z}_t; y, t) - \epsilon\|$ for NeRF renderings in Figure 2. From the left visualization it is evident that as NeRF optimization progresses, the diffusion timestep that is most informative to NeRF update changes (as highlighted by curved arrow in Figure 2). We provide more examples to reveal the same pattern on the right.

Frequency analysis. Figure 3 illustrates diffusion model's out of domain (OOD) issue on NeRF renderings with randomly sampled noise level. Figure 3 (a) illustrates image sampling process: top in pixels and bottom in frequency after Fourier Transform (FT). To show that randomly adding different levels of noise to NeRF rendered images could pose the OOD issue to pre-trained diffusion model, we provide two extreme examples plausible during SDS optimization: (b) adding large noise to a well-trained NeRF's rendering and (c) adding small noise to a NeRF's rendering at initialization. In both cases the resultant images exhibit evident frequency difference from diffusion model's pre-training: a typical OOD issue hampering effectiveness of guidance from pre-trained diffusion models. With time prioritized SDS (TP-SDS) we effectively avoid such cases and the resultant mismatch in diffusion's training and inference frequency as shown in Figure 3 (d-e).

The reason for the observed frequency discrepancy is that NeRF rendered images tend to exhibit higher correlation in pixel values, therefore of lower frequency than natural images. The discrepancy is largest at initialization: DDPM sampling is initialized with a random Gaussian distribution of high frequency, but NeRF rendered image is of extremely low frequency at initialization as shown in Figure 3 (c). The low frequency bias is persistent even towards the end of SDS optimization: other than correlation in pixel values and blurred object contours, an SDS supervised NeRF usually lacks background details as shown in 3 (b). Note that for the image noising process as formulated in Eq. 1, $\bar{\alpha}_t$ is > 0 even at the largest timestep, meaning that during diffusion training, information of the original input is present even at the largest noise level. As a result, adding large noise to NeRF's lower-frequency images according to Eq. 1 further dilutes image information, resulting in frequencies more resembling random noise than maximally noised natural images as shown in Figure 3 (a,b).

We further show in Figure 4 with 2D examples that the lack of high-frequency signal at early stage of content creation directly contributes to mode collapse (low-diversity 3D models given the same text prompt) as observed in [27]. We circumvent this loss of diversity with TP-SDS by explicitly adding more high-frequency signal (noise) to NeRF renderings at early stage of optimization (bottom).

3.3. Time Prioritized Optimization

Monotonically non-increasing t-sampling. Drawbacks of uniform t-sampling in vanilla SDS motivates us to investigate a more effective t-sampling strategy. Figure 2 shows empirically that non-increasing t (marked by curved arrow)

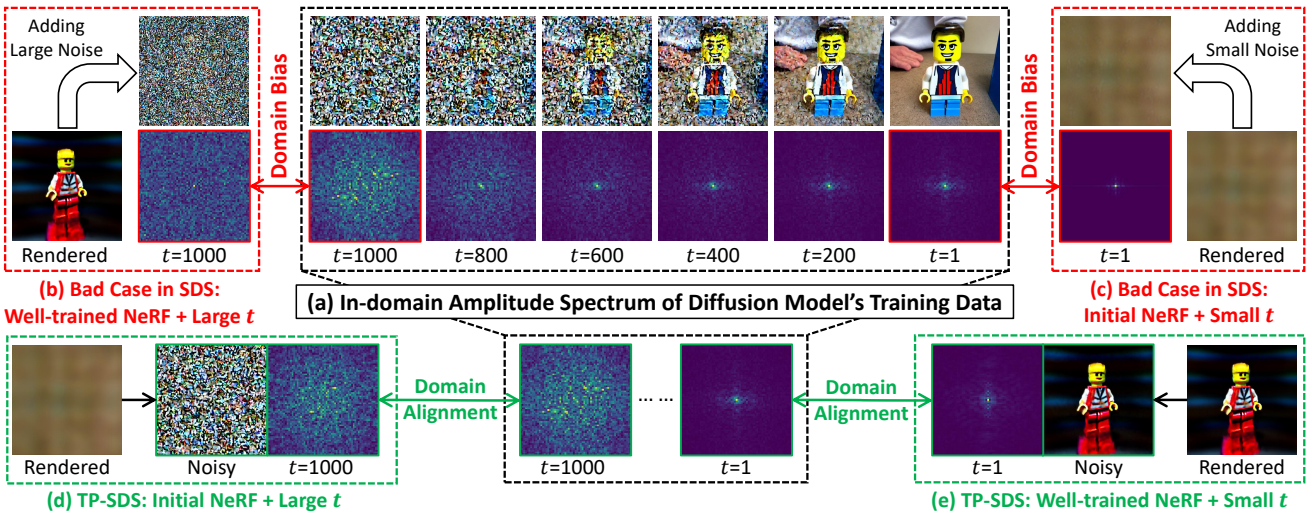


Figure 3: Illustration of OOD issue using web-data pre-trained diffusion model for denoising NeRF rendered images, in frequency domain. We provide two extreme cases to show the frequency domain misalignment: (b) adding large noise to well-trained NeRF's rendering, (c) adding small noise to NeRF's rendering at initialization. (d) and (e) illustrates that TP-SDS avoids such domain gap by choosing the right noise level according to current optimization iteration.

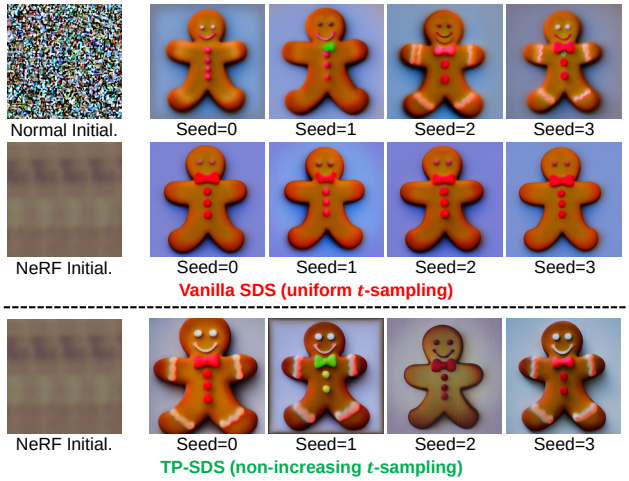


Figure 4: 2D generation results of prompt "gingerbread man" using SDS (Top): Gaussian (high-frequency) initialization generates diverse samples while NeRF rendered initialization suffers mode collapse. (Bottom) TP-SDS brings back diversity by explicitly adding high noise (frequency) at early stage of sampling.

is more informative to the NeRF optimization process.

Based on this observation, we first try a naive strategy that decreases t linearly with optimization iteration, however it fails with severe artifacts in the final rendered image, as shown in Figure 5. We observe that decreasing t works well until later optimization stage when small t dominates. We visualize the SDS gradients (lower-right box within each rendered image) and notice that at small t, variance of the gradients are extremely high, which makes convergence difficult for NeRF. In fact, different denoising t contributes differently [6] to content generation, so it is non-optimal to adopt a uniform decreasing t. As a result, we propose a Time Prioritized (TP) strategy for SDS to better modulate the timestep (t) decreasing given the functionality of their guidance: coarse, content and details.

Figure 5: Visualization of NeRF optimization with SDS using a naive t-sampling strategy, where t decreases linearly with the iteration step i. Severe artifacts appear in the final rendered image due to large gradients variance with small t.

Time prioritized SDS. Existing methods [15, 27] use a term $w(t)$ to weight the strength of noise residual from different t, as shown in Eq. 2. However, they did not observe noticeable influence with different choices of $w(t)$ [27]. In contrast, we propose to explicitly sample different t at different iteration of NeRF optimization, obeying a non-increasing principle for coarse-to-fine generation. To do

this in a principled fashion, we start by introducing a prior weight term $w^*(t)$ to control the decreasing velocity of t, where a large value of $w^*(t)$ corresponds to a flat slope, while a small one corresponds to a steep decline.

To construct $w^*(t)$, we adopt a simple and effective piece-wise function:

$$w^*(t) = \begin{cases} e^{-(t-m_1)^2/2s_1^2} & \text{if } t > m_1 \\ 1.0 & \text{if } m_2 \leq t \leq m_1 \\ e^{-(t-m_2)^2/2s_2^2} & \text{if } t < m_2, \end{cases} \quad (4)$$

where $\{m_1, m_2, s_1, s_2\}$ are hyperparameters that control how t is decreased. As shown in Figure 6, the function $w^*(t)$ consists of three stages: *coarse*, *content* and *detailed*, which is inspired by the observation that denoising with different noise levels focus on the restoration of different visual concepts [6].

Given a prior weight function $w^*(t)$, we get the non-increasing t_i corresponding to an optimization step i by:

$$t_i = \arg\min_{t^*} \left| \sum_{t=t^*}^{T} p(t) - i/N \right|, \quad (5)$$

where T is the number of noise levels of a pre-trained diffusion model, N denotes the number of NeRF optimization iterations, and the normalized time prior $p(t)$ is obtained by:

$$p(t) = w^*(t) / \sum_{t=1}^{T} w^*(t). \quad (6)$$

The detailed algorithm is given in Algorithm 1.

Algorithm 1: Time Prioritized SDS (TP-SDS).

Input: A differentiable generator g with initial parameters $\boldsymbol{\theta}_0$ and number of optimiazation steps N, pre-trained diffusion model ϕ, time prior $p(t)$, learning rate lr and text prompt y.

for $i = 1, ..., N$ **do**

  $t_i = \arg\min_{t^*} \left| \sum_{t=t^*}^{T} p(t) - i/N \right|$;

  $\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i-1} - lr * \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{SDS}}(\phi, g(\boldsymbol{\theta}_{i-1}); y, t_i)$;

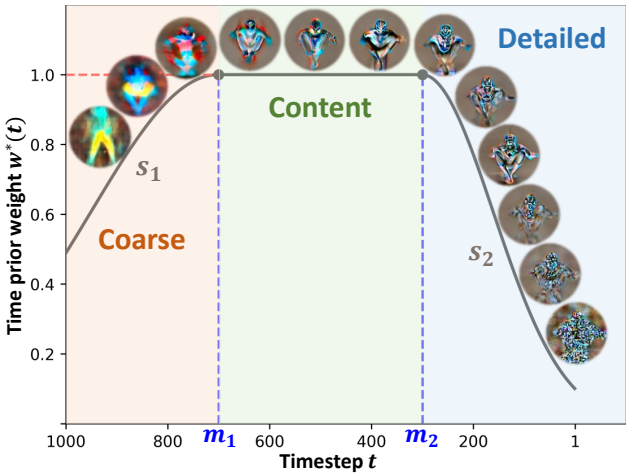
end

Output: θ_N.

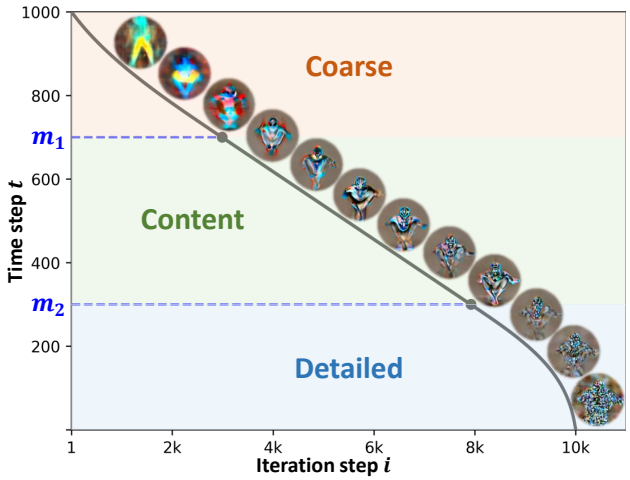
3.4. Implementation Details

Our method is implemented in PyTorch and can be trained and evaluated on a single NVIDIA 3090 GPU. Our code-base mainly refers to the open source Stable-DreamFusion [39] and Latent-NeRF [17]. Here we provide comprehensive implementation details.

Diffusion guidance. Stable Diffusion v1.4 [31] is used as guidance to provide strong image priors. The classifier-free guidance scale is set to 100. The weight term $w(t)$ of SDS loss is set to 1.0, and we normalize the SDS gradients to stabilize the optimization process. The max timestep T is 1,000. For the SDS baseline, we sample timestep $t \sim \mathcal{U}(20, 980)$ following DreamFusion [27]. For the proposed TP-SDS, timestep t decrements from 1000 to 1, and we use a prior weight configuration of $\{m_1 = 800, m_2 = 500, s_1 = 300, s_2 = 100\}$ to control the decreasing velocity, which is obtained by a small-scale grid search.



(a) Time prior weight function $w^*(t)$.



(b) Timestep t decreases with iteration step i.

Figure 6: The proposed time prior weights $w^*(t)$ to modulate the decreasing velocity of t for score distillation sampling (SDS). A large value of $w^*(t)$ corresponds to a flat slope, while a small value corresponds to a steep decline.

NeRF rendering. We employ Instant-NGP [20] as NeRF representation. Our method renders "latent images" in the latent space $\mathbb{R}^{64\times64\times4}$ of Stable Diffusion following Latent-NeRF [17], where the rendered latent images can be decoded into RGB images by the VAE decoder of Stable Diffusion. At each iteration of NeRF optimization,

a camera position is randomly sampled in spherical coordinates, with elevation angle $\phi_{\text{cam}} \in [0°, 120°]$, azimuth angle $\theta_{\text{cam}} \in [0°, 360°]$, and spherical radius $R_{\text{cam}} \in [1.0, 1.5]$. A focal length multiplier $\lambda_{\text{focal}} \in [0.71, 1.37]$ is also randomly sampled at each iteration to obtain focal length $\lambda_{\text{focal}} \times H$, where $H = 64$ is the latent height.

Optimization. The number of iterations N and the batch size are set to 10,000 and 1 respectively, which takes about 15 minutes on a single NVIDIA 3090 GPU. We use an Adam optimizer [14] with learning rate of 1e-3, betas of $(0.9, 0.999)$ without weight decay. The learning rate is constant throughout the optimization process. A sparsity loss as suggested in [17] is also adopted to facilitate 3D shape coherence and to prevent floating "radiance clouds".

Prompt augmentation. Unlike existing methods [15, 27], we found that some prompt prefixes such as "a DSLR photo of..." did not lead to quality improvements in our method, and therefore did not use any such augmentations. In addition, view-dependent prompt augmentations are critical for 3D consistency, and we found that the results are sensitive to the choice of such augmentations. We adopt the following view-dependent prompt augmentations:

$$\begin{cases} \text{"overhead view of..."} & \text{if } \phi_{\text{cam}} \leq 30° \\ \text{"front view of..."} & \text{if } \phi_{\text{cam}} > 30°, \theta_{\text{cam}} \in [0°, 90°] \\ \text{"backside view of..."} & \text{if } \phi_{\text{cam}} > 30°, \theta_{\text{cam}} \in [180°, 270°] \\ \text{"side view of..."} & \text{otherwise.} \end{cases}$$

4. Experiment

We conduct experiments on generation of 2D points, 2D images, and 3D assets for a comprehensive evaluation of the proposed time prioritized score distillation sampling (TP-SDS). For 2D experiments, generator g is an identity mapping where parameters are point or image representations. For 3D experiments, generator g is a differentiable volume renderer that transforms NeRF parameters into an image.

4.1. Start with a Toy Example

We first demonstrate in Figure 7 with toy example that comparing to SDS, TP-SDS generates more accurate samples with better mode coverage. The toy distribution is of 2D (x, y) points with modes clustered as black dots, resembling a dinosaur as a whole [16]. We train a DDPM on the points distribution till convergence so that we could sample faithful points with DDPM sampling. In (b) we simulate properties of random initialization: in high-dimensional space randomly sampled point is likely to be far away from known data distribution [8, 21] (top); and rendered image at NeRF initialization exhibit high correlation in pixel values [29] (bottom). It is evident from (c) that TP-SDS yields samples with higher quality and diversity than SDS.

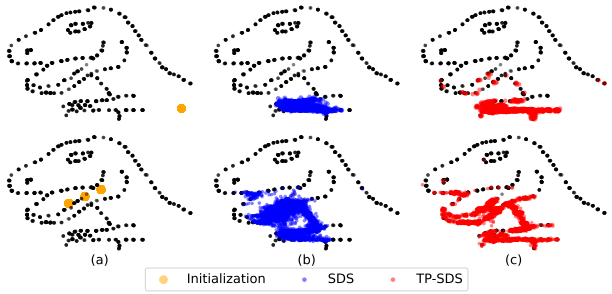


Figure 7: Denoising 2D points with SDS and TP-SDS. We train DDPM on Dino points dataset (black). (a) Simulated sampling initialization: a randomly sampled point is likely to be far away from real data distribution in high-dimensional space (top), NeRF yields highly correlated renderings at initialization (bottom). We optimize 2000 samples with SDS (b) and TP-SDS (c). It is evident that TP-SDS yields more accurate samples (higher quality) with broader mode coverage (higher diversity).

4.2. Visual Quality Evaluation

In this subsection, we demonstrate the effectiveness of the proposed TP-SDS in improving visual quality. We argue that some challenging problems in text-to-3D generation, such as unrealistic appearance, multiple faces, and failure to capture text semantics, can be effectively alleviated by simply changing the sampling strategy of timestep t.

User studies. We conduct user studies to evaluate effectiveness of TP-SDS based on user preferences. We show users two videos side by side using the same text prompt and ask the users to select the one that they consider to be of higher quality: more realistic, more detailed and align better with given prompt. Each prompt is evaluated by 5 different users, resulting in 6,225 pairwise comparisons in total. As shown in Table 1, the raters favor 3D models generated with TP-SD over Latent-NeRF (75.8%), SJC (66.6%) and SDS baseline (80.2%).

Table 1: User study. We measure preference for 3D models generated using 415 prompts released by DreamFusion [27]. 3D models generated with TP-SDS are favoured by significantly more raters over Latent-NeRF, SJC and SDS baseline by 75.8%, 66.6% and 80.2% respectively.

Comparison	Preference (%)
Ours vs. Latent-NeRF [17]	75.8%
Ours vs. SJC [40]	66.6%
Ours vs. SDS (Baseline)	80.2%

R-Precision. We follow DreamFusion [27] to assess the consistency of rendered images with input text prompt by CLIP R-Precision. R-Precision measures how accurately

(a) Avoiding Semantic Misalignment and Multiple Faces. (b) Better Shapes, Colors and Details.

Figure 8: Qualitative comparisons with Latent-NeRF [17] SJC [40], and Baseline [39]. All methods use Stable Diffusion v1.4 for fair comparison. (a) The prompt is "a chimpanzee dressed like Henry VIII king of England". Our results do not have the Janus (multi-face) problem (marked with circles) and align better with the given text (marked with boxes). (b) The text prompt is "Trump figure". The proposed TP-SDS is able to produce more realistic shape and appearance details.

Table 2: Evaluating generations with TP-SDS for alignment of their renderings with corresponding captions, using different CLIP models. We compare to ground-truth images, Latent-NeRF [17], SJC [40] and SDS baseline evaluated on object-centric COCO as in [27].

Method	R-Precision (%) ↑		
	CLIP B/32	CLIP B/16	CLIP L/14
GT Images	77.1	79.1	–
Latent-NeRF	48.4	52.9	59.5
SJC	55.6	58.2	66.0
SDS (Baseline)	58.8	62.7	63.4
TP-SDS (Ours)	**63.4**	**67.3**	**71.2**

CLIP [28] retrieves the correct caption from a group of distractor prompts when presented with a rendered scene. We use the 153 prompts from the object-centric COCO validation subset of Dream Fields [11]. We report in Table 2 that for models utilizing open-sourced Stable Diffusion, our proposed TP-SDS attains the highest R-Precision score.

Qualitative comparisons. We compare our method with existing text-to-3D generation methods utilizing publicly-accessible Stable Diffusion (v1.4) [31], including Latent-NeRF [17], SJC [40] and SDS Baseline [39].

- Figure 8 (a) shows that our generation results *do not* have the multi-face problem and align better with given text semantics. For example, competing methods fail to generate the king's attire described by the text prompt, while our method can. Figure 8 (b) shows that our method is able to produce realistic appearances, avoiding distorted shapes, colors and blurry details. For example, more reasonable body proportions and colors of human figures are generated.
- It has been well-observed that DreamFusion [27] tends to generate 3D models with unrealistic appearance exhibiting saturated colors. Figure 9 shows that in general, DreamTime's generations are of more natural-looking colors. Our intuition is that decaying timesteps avoids fluctuating and possibly conflicting gradient updates, especially towards completion of the optimization, therefore alleviates extreme colors.
- The proposed TP-SDS is orthogonal to the choice of NeRF, diffusion model and implementation details. For confirmation, we further apply the proposed method to recently released DreamFusion implementation threestudio [9]. Figure 10 shows that comparing to the baseline, our method could produce significantly better 3D generations, even superior to DreamFusion.

Figure 9: DreamTime produces more natural-looking colors compared to DreamFusion's saturated colors.

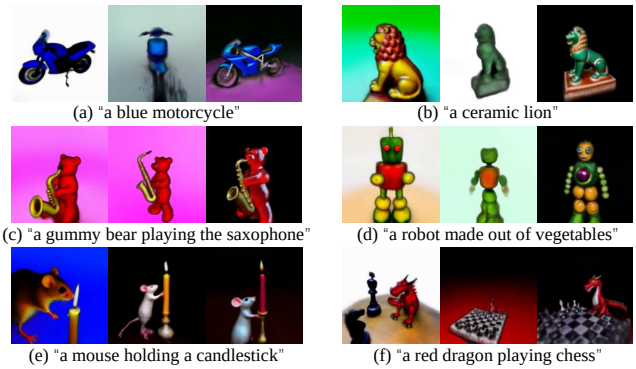


Figure 10: Qualitative comparisons with DreamFusion [27] (left), threestudio baseline [9] (middle) and ours (right). Our method is able to bring significant quality improvement over the strong threestudio baseline, with 3D generations even superior to DreamFusion.

4.3. Ablation Study

Although the relationship between t_i and i in TP-SDS is mostly linear, we encourage fewer large-step sampling for stable structure formation, and fewer small-step sampling to avoid color distortion. Figure 5 shows that a naive linearly decreasing timestep sampling is prone to artifacts and color distortion. In Figure 11, we further evaluate on more truncated linear schedules showing that our timestep schedule produces 3D generations with more realistic shape and appearance, effectively avoiding artifacts and color shifts.

4.4. Faster Convergence

By rendering images in the latent space of Stable Diffusion, our method enables a fast sampling of 15 minutes per prompt on a single 3090 GPU, about three times faster than DreamFusion [27] and Magic3D [15]. Moreover, we empirically find that the proposed non-increasing t-sampling strategy leads to a faster convergence, requiring $\sim$ 35% fewer optimization steps than the uniform t-sampling. This is likely due to more efficient utilization of information, *e.g.*

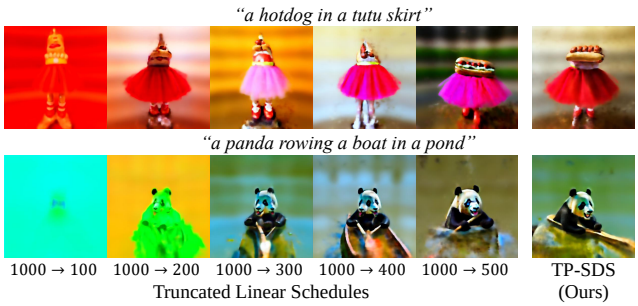


Figure 11: Qualitative comparisons of the truncated linear schedules and our proposed TP-SDS.

it is wasteful to seek structure information at later stage of optimization when the 3D model is already in good shape. To demonstrate the fast convergence of TP-SDS, we conduct experiments on 2D image generation with 153 text prompts from the object-centric COCO validation set, obtaining quantitative and qualitative results.

Quantitative evaluation. We show in Figure 12 the R-Precision scores at different optimization iterations using the vanilla SDS and TP-SDS. The growth rate of TP-SDS curves is consistently higher across various CLIP models, which implies a faster convergence requiring significantly fewer optimization steps to reach the same R-Precision score. This leads to the production of superior text-aligned generations at a quicker pace with fewer resources.

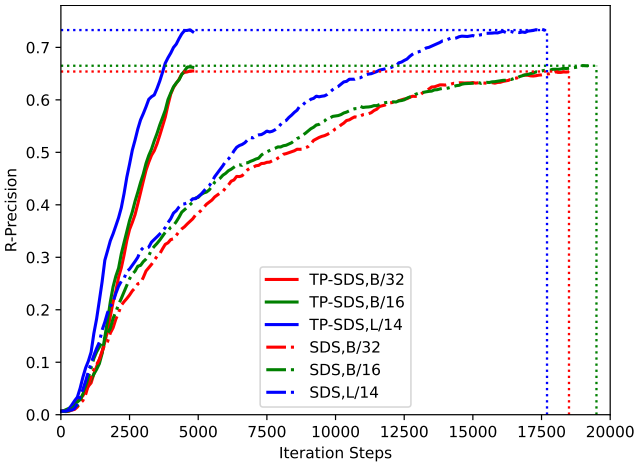


Figure 12: R-Precision curves of the SDS baseline and the proposed TP-SDS, using three CLIP models: B/32, B/16, and L/14. Given 153 text prompts from the object-centric COCO validation set, we employ SDS and TP-SDS to generate corresponding 2D images for evaluation of R-Precision. The R-Precision curves for TP-SDS have a steeper growth rate compared to those of SDS, signifying faster convergence of our method.

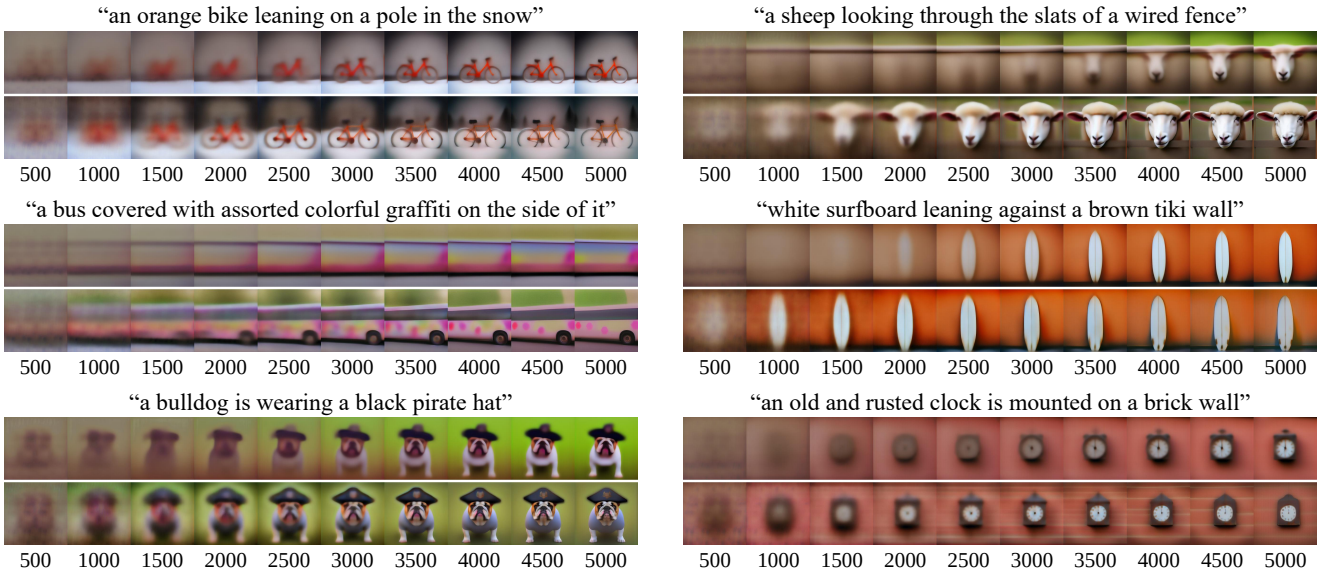


Figure 13: Qualitative comparisons of the SDS baseline (first row in each example) and the proposed TP-SDS (second row in each example) under different iteration steps (from 500 to 5000). The proposed TP-SDS leads to faster content generation than the SDS baseline.

Qualitative comparison. Figure 13 shows the 2D image generation process at different iterations using the vanilla SDS and our TP-SDS. It is clear that with TP-SDS, the emergence of content (*e.g.*, object structures) is faster with better appearance and details.

4.5. Diversity Evaluation

One of the key ingredients for creative content creation with AI is diversity: given a text prompt, Stable Diffusion is able to generate countless number of diverse samples while respecting the given text. However, pioneering work of DreamFusion [27] has already observed the mode collapse problem for text-to-3D generation: one text prompt would always yield highly similar 3D models. In Figure 4, we demonstrate with 2D generation results that mode collapse is largely caused by the low-frequency nature of NeRF initialization, and the proposed TP-SDS is able to circumvent it by adding large noise (*i.e.*, using a large timestep t) early in training. Figure 14 further demonstrates that 3D samples generated with TP-SDS are much more diverse in appearance than that with Latent-NeRF [17] and SJC [40].

4.6. Hyper-Parameter Analysis

The proposed TP-SDS improves generation quality, efficiency, and diversity compared to the SDS baseline. However, the adopted prior weight function $w^*(t)$ parameterized by $\{m_1, m_2, s_1, s_2\}$ introduces extra hyper-parameters. We explore the influence of these hyper-parameters on text-to-3D generation, which can serve as a guide for tuning in practice. Specifically, we explore the impact of different hyper-parameter settings on the generated results in a wide range of search spaces, as shown in Figure 15. s_1 controls the *coarse* stage, a small value of s_1 results in fewer steps of large t sampling, which is likely to hinder formation of 3D structure. s_2 controls the *detailed* stage, a large value of s_2 results in more steps of small t sampling, likely to cause color shifts and artifacts. m_1 and m_2 affects s_1 and s_2 simultaneously, and a general rule of thumb is to make the average of m_1 and m_2 close to $0.5T$, while the difference between m_1 and m_2 is within $[0, 0.4T]$, where T is the max diffusion timestep.

5. Conclusion

We propose DreamTime, an improved optimization strategy for text-to-3D content generation. We thoroughly investigate how the 3D formation process harnesses supervision from pre-trained text-to-image diffusion models at different noise levels and analyze the drawbacks of commonly used score distillation sampling (SDS). We then propose a non-increasing time sampling strategy (TP-SDS) which effectively aligns the training process of NeRF and the sampling process of DDPM. With extensive user studies, qualitative comparisons and quantitative evaluations we show that TP-SDS significantly improves the quality and diversity of text-to-3D generation, and considerably more preferable compared to accessible 3D generators Latent-NeRF [17] (75.8%) and SJC [40] (66.6%). We hope that with DreamTime, 3D content creation can be more accessible for creativity and aesthetics expression.

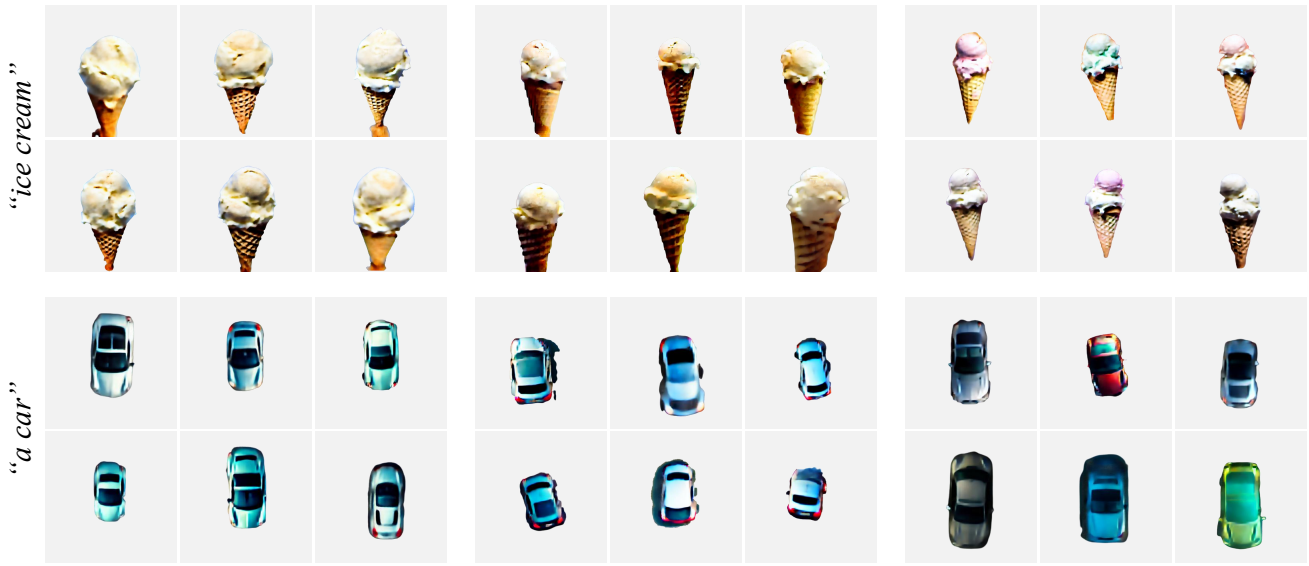


Figure 14: Diversity comparisons with Latent-NeRF [17] and SJC [40]. The text prompts are "ice cream" and "a car" with no additional description. Given different random seeds, our method is able to generate objects with diverse appearance, while competing methods suffer mode collapse, repetitively generating similar-looking results. Backgrounds are removed for readers to concentrate on object diversity.

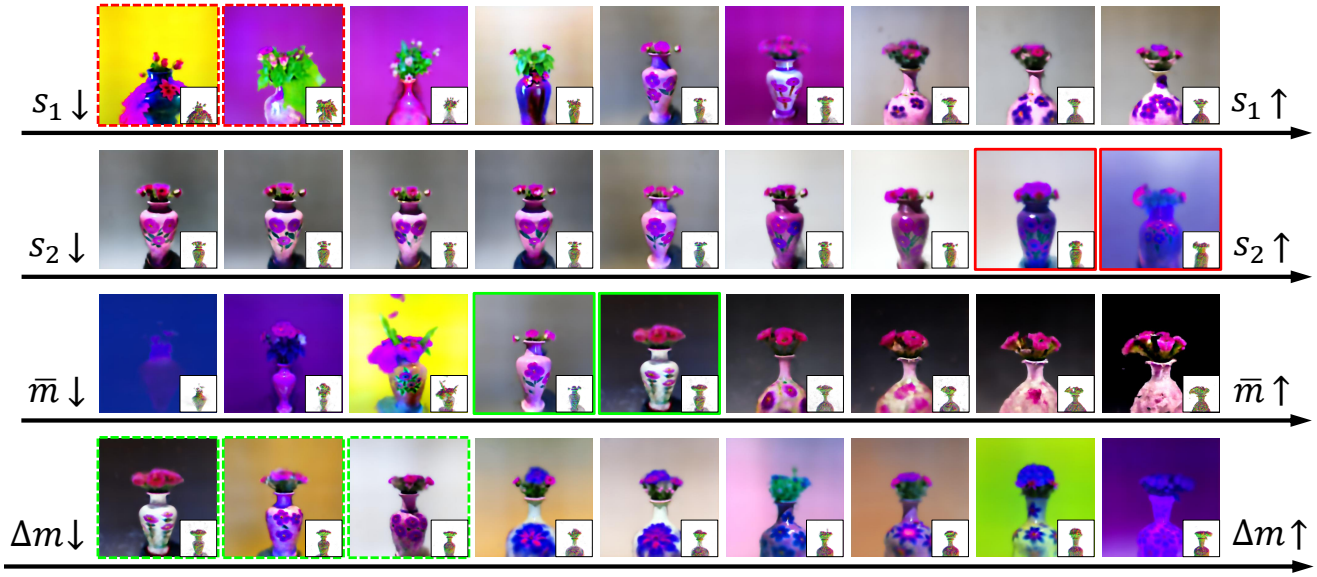


Figure 15: Influence of the time prior configuration $\{m_1, m_2, s_1, s_2\}$ on generated results. The text prompt is "a vase with pink flowers". We define $\bar{m} = (m_1 + m_2)/2$ and $\Delta m = m_1 - m_2$, then perform analysis in the search spaces of $s_1 \in [10, 800]$, $s_2 \in [5, 150]$, $\bar{m} \in [100, 900]$ and $\Delta m \in [0, 800]$. s_1 controls the *coarse* stage, a small value of s_1 results in fewer steps of large t sampling, which is likely to hinder formation of 3D structure (marked in "---"). s_2 controls the *detailed* stage, a large value of s_2 results in more steps of small t sampling, likely to cause color shifts and artifacts (marked in "——"). The adjustment of $\bar{m}$ or Δm affects s_1 and s_2 simultaneously, a general rule of thumb is to place $\bar{m}$ close to $\frac{T}{2}$ (marked in "——") while Δm can be freely chosen within $[0, \frac{2T}{5}]$ (marked in "---"), where T is the max diffusion timestep.

References

[1] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. eDiff-I: Text-to-Image Diffusion Models with an Ensemble of Expert Denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[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 IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[3] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient Geometry-Aware 3D Generative Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022.

[4] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. Pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5799–5809, 2021.

[5] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021.

[6] Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception Prioritized Training of Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11472–11481, 2022.

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion Models Beat GANs on Image Synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[8] David L Donoho et al. High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality. *AMS Math Challenges Lecture*, 1(2000):32, 2000.

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

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[11] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-Shot Text-Guided Object Generation With Dream Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–876, 2022.

[12] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training Generative Adversarial Networks with Limited Data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020.

[13] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[14] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

[15] 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. *arXiv preprint arXiv:2211.10440*, 2022.

[16] Justin Matejka and George Fitzmaurice. Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics through Simulated Annealing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1290–1294, 2017.

[17] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-NeRF for Shape-Guided Generation of 3D Shapes and Textures. *arXiv preprint arXiv:2211.07600*, 2022.

[18] 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.

[19] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. CLIP-Mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–8, 2022.

[20] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022.

[21] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do Deep Generative Models Know What They Don't Know? In *International Conference on Learning Representations*, 2019.

[22] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. HoloGAN: Unsupervised Learning of 3D Representations From Natural Images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7588–7597, 2019.

[23] Thu H Nguyen-Phuoc, Christian Richardt, Long Mai, Yongliang Yang, and Niloy Mitra. BlockGAN: Learning 3D Object-aware Scene Representations from Unlabelled Images. *Advances in Neural Information Processing Systems*, 33:6767–6778, 2020.

[24] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. *arXiv preprint arXiv:2112.10741*, 2021.

[25] Alexander Quinn Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[26] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing Scenes As Compositional Generative Neural Feature Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021.

[27] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D Diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[29] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the Spectral Bias of Neural Networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

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

[31] 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*, pages 10684–10695, 2022.

[32] 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, et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[33] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. LAION-5B: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

[34] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020.

[35] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

[36] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*, 2021.

[37] Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

[38] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021.

[39] Jiaxiang Tang. Stable-dreamfusion: Text-to-3d with stable-diffusion, 2022. https://github.com/ashawkey/stable-dreamfusion.

[40] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score Jacobian Chaining: Lifting Pretrained 2D Diffusion Models for 3D Generation. *arXiv preprint arXiv:2212.00774*, 2022.

[41] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel View Synthesis with Diffusion Models. *arXiv preprint arXiv:2210.04628*, 2022.