# TEXT-TO-3D USING GAUSSIAN SPLATTING

**Anonymous authors**
Paper under double-blind review

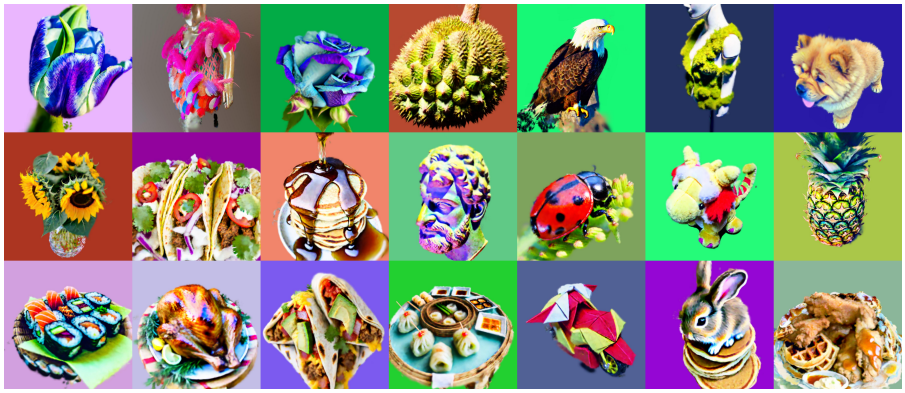


Figure 1: Delicate 3D assets generated using the proposed GSGEN. See our project page gsgen3d.github.io for videos of these images.

## ABSTRACT

In this paper, we present Gaussian Splatting based text-to-3D generation (GSGEN), a novel approach for generating high-quality 3D objects. Previous methods suffer from inaccurate geometry and limited fidelity due to the absence of 3D prior and proper representation. We leverage 3D Gaussian Splatting, a recent state-of-the-art representation, to address existing shortcomings by exploiting the explicit nature that enables the incorporation of 3D prior. Specifically, our method adopts a progressive optimization strategy, which includes a geometry optimization stage and an appearance refinement stage. In geometry optimization, a coarse representation is established under a 3D geometry prior along with the ordinary 2D SDS loss, ensuring a sensible and 3D-consistent rough shape. Subsequently, the obtained Gaussians undergo an iterative refinement to enrich details. In this stage, we increase the number of Gaussians by compactness-based densification to enhance continuity and improve fidelity. With these designs, our approach can generate 3D content with delicate details and more accurate geometry. Extensive evaluations demonstrate the effectiveness of our method, especially for capturing high-frequency components. Video results are provided in gsgen3d.github.io.

## 1 INTRODUCTION

Diffusion model based text-to-image generation (Saharia et al., 2022; Rombach et al., 2022; Ramesh et al., 2022; Alex et al., 2023) has achieved remarkable success in synthesizing photo-realistic images from textual prompts. Nevertheless, for high-quality text-to-3D content generation, the advancements lag behind that of image generation due to the inherent complexity of real-world 3D scenes. Recently, DreamFusion (Poole et al., 2023) has made great progress in generating delicate assets by utilizing score distillation sampling with a pre-trained text-to-image diffusion prior. Its follow-up works further improve this paradigm in quality (Wang et al., 2023c; Chen et al., 2023), training speed (Lin et al., 2023; Metzer et al., 2022), and generating more reasonable geometry (Armandpour et al., 2023; Zhu & Zhuang, 2023; Seo et al., 2023). However, most existing text-to-3D methods still suffer greatly from collapsed geometry and limited fidelity, and are difficult to incorporate 3D priors due to the implicit nature of NeRF (Mildenhall et al., 2020) and DMTET (Shen et al., 2021).

Stable DreamFusion (Tang, 2022; Poole et al., 2023) GSGEN (Ours)

*A DSLR photo of a panda*

Fantasia3D (Chen et al., 2023) GSGEN (Ours)

*A high quality photo of a furry corgi*

threestudio Magic3D (Guo et al., 2023; Lin et al., 2023) GSGEN (Ours)

*A zoomed out DSLR photo of an amigurumi bulldozer*

Figure 2: Compared to previous methods, GSGEN alleviates the Janus problem by representing the 3D scene using 3D Gaussian Splatting, which is capable of applying direct 3D geometry guidance and expressing content with delicate details. Note that the results of DreamFusion and Magic3D are obtained using Stable DreamFusion (Tang, 2022) and threestudio (Guo et al., 2023) since the official implementations are not publicly available due to the utilization of private diffusion models. All the results are obtained using StableDiffusion (Rombach et al., 2022) on checkpoint *runwayml/stable-diffusion-v1-5* for a fair comparison. Videos of these images are provided in the supplemental video.

Recently, 3D Gaussian Splatting (Kerbl et al., 2023) has garnered significant attention in the field of 3D reconstruction, primarily due to its remarkable ability to represent intricate scenes and capability of real-time rendering. By modeling a scene using a set of 3D Gaussians, Kerbl et al. (2023) adopt an explicit and object-centric approach that fundamentally diverges from implicit representations like NeRF and DMTET. This distinctive approach paves the way for the integration of explicit 3D priors into text-to-3D generation. Building upon this insight, instead of a straightforward replacement of NeRFs with Gaussians, we propose to guide the generation with an additional 3D point cloud diffusion prior to enhancing geometrical coherence. By adopting this strategy, we can better harness the inherent advantages of 3D Gaussians in the creation of complex and 3D-consistent assets.

Specifically, we propose to represent the generated 3D content with a set of Gaussians and optimize them progressively in two stages, namely geometry optimization and appearance refinement. In the geometry optimization stage, we optimize the Gaussians under the guidance of a 3D point cloud diffusion prior along with the ordinary 2D image prior. The incorporation of this extra 3D SDS loss ensures a 3D-consistent rough geometry. In the subsequent refinement stage, the Gaussians undergo an iterative enhancement to enrich the delicate details. Due to the sub-optimal performance of the original adaptive control under SDS loss, we introduce an additional compactness-based densification technique to enhance appearance and fidelity. Besides, to prevent potential degeneration and break the symmetry in the early stage, the Gaussians are initialized with a coarse point cloud generated by a text-to-point-cloud diffusion model. As a result of these techniques, our approach can generate

3D assets with accurate geometry and exceptional fidelity. Fig.2 illustrates a comparison between GSGEN and previous state-of-the-art methods on generating assets with asymmetric geometry.

In summary, our contributions are:

- We propose GSGEN, the first text-to-3D generation method using 3D Gaussians as representation. By incorporating geometric priors, we highlight the distinctive advantages of Gaussian Splatting in text-to-3D generation.
- We introduce a two-stage optimization strategy that first exploits joint guidance of 2D and 3D diffusion prior to shaping a coherent rough structure in geometry optimization; then enriches the details with a compactness-based densification in appearance refinement.
- We validate GSGEN on various textual prompts. Experiments show that our method can generate 3D assets with more accurate geometry and enhanced fidelity than previous methods. Especially, GSGEN demonstrates superior performance in capturing *high-frequency components* in objects, such as feathers, surfaces with intricate textures, animal fur, etc.

## 2 Related Work

### 2.1 3D Scene Representations

Representing 3D scenes in a differentiable way has achieved remarkable success in recent years. NeRFs (Mildenhall et al., 2020) demonstrates outstanding performance in novel view synthesis by representing 3D scenes with a coordinate-based neural network. After works have emerged to improve NeRF in reconstruction quality (Barron et al., 2021; 2023; Wang et al., 2022c), handling large-scale (Tancik et al., 2022; Zhang et al., 2020; Martin-Brualla et al., 2021; Chen et al., 2022b) and dynamic scenes (Park et al., 2021; Attal et al., 2023; Wang et al., 2022b; Sara Fridovich-Keil and Giacomo Meanti et al., 2023; Pumarola et al., 2021), improving training (Yu et al., 2021a; Chen et al., 2022a; Sun et al., 2022; Müller et al., 2022) and rendering (Reiser et al., 2023; Hedman et al., 2021; Yu et al., 2021b) speed. Although great progress has been made, NeRF-based methods still suffer from low rendering speed and high training-time memory usage due to their implicit nature. To tackle these challenges, Kerbl et al. (2023) propose to represent the 3D scene as a set of anisotropic Gaussians and render the novel views using a GPU-optimized tile-based rasterization technique. 3D Gaussian Splatting could achieve comparing reconstruction results while being capable of real-time rendering. Our research highlights the distinctive advantages of Gaussian Splatting within text-to-3D by incorporating explicit 3D prior, generating 3D consistent and highly detailed assets.

### 2.2 Diffusion Models

Diffusion models have arisen as a promising paradigm for learning and sampling from a complex distribution. Inspired by the diffusion process in physics, these models involve a forward process to gradually add noise and an inverse process to denoise a noisy sample with a trained neural network. After DDPM (Ho et al., 2020; Song et al., 2021b) highlighted the effectiveness of diffusion models in capturing real-world image data, a plethora of research has emerged to improve the inherent challenges, including fast sampling (Lu et al., 2022; Bao et al., 2022; Song et al., 2021a) and backbone architectural enhancement (Bao et al., 2023; Podell et al., 2023; Liu et al., 2023b; Dhariwal & Nichol, 2021; Hoogeboom et al., 2023; Peebles & Xie, 2022). One of the most successful applications of diffusion models lies in text-to-image generation, where they have shown remarkable progress in generating realistic images from text prompts (Ho & Salimans, 2022; Ramesh et al., 2022; Alex et al., 2023). To generate high-resolution images, current solutions either adopt a cascaded structure that consists of a low-resolution diffusion model and several super-resolution models (Saharia et al., 2022; Balaji et al., 2022; Alex et al., 2023) or trains the diffusion model in latent space with an auto-encoder (Rombach et al., 2022; Gu et al., 2022). Our proposed GSGEN is built upon StableDiffusion (Rombach et al., 2022), an open-source latent diffusion model that provides fine-grained guidance for high-quality 3D content generation.

### 2.3 TEXT-TO-3D GENERATION

Early efforts in text-to-3D generation, including CLIP-forge (Sanghi et al., 2021), Dream Fields (Jain et al., 2022), Text2Mesh (Michel et al., 2022), TANGO (Chen et al., 2022c), CLIPNeRF (Wang et al., 2022a), and CLIP-Mesh (Khalid et al., 2022), harness CLIP (Radford et al., 2021) guidance to create 3D assets. To leverage the stronger diffusion prior, DreamFusion (Poole et al., 2023) introduces the score distillation sampling loss that optimizes the 3D content by minimizing the difference between rendered images and the diffusion prior. This development sparked a surge of interest in text-to-3D generation through image diffusion prior (Wang et al., 2023a; Raj et al., 2023; Lorraine et al., 2023; Zhu & Zhuang, 2023). Magic3D (Lin et al., 2023) employs a coarse-to-fine strategy, optimizing a NeRF with a low-resolution diffusion prior and then enhancing texture under a latent diffusion model with a DMTET initialized using the coarse NeRF. Latent-NeRF (Metzer et al., 2022) trains a NeRF within the latent space of StableDiffusion and introduces the Sketch-Shape method to guide the generation process. Fantasia3D (Chen et al., 2023) disentangles the learning of geometry and material, harnessing physics-based rendering techniques to achieve high-fidelity mesh generation. ProlificDreamer (Wang et al., 2023c) introduces variational score distillation to improve SDS and facilitate the generation of high-quality and diverse 3D assets, whose contribution is orthogonal to ours since we focus on incorporating 3D prior with more advanced representation. Another line of work lies in generating 3D assets directly through a 3D diffusion model based on NeRF or other differentiable representations (Wang et al., 2023b; Jun & Nichol, 2023; Liu et al., 2023a; Cheng et al., 2023). Our approach builds upon Point-E (Nichol et al., 2022), a text-to-point-cloud diffusion model trained on millions of 3D models, which offers valuable 3D guidance and coarse initialization.

## 3 PRELIMINARY

### 3.1 SCORE DISTILLATION SAMPLING

Instead of directly generating 3D models, recent studies have achieved notable success by optimizing 3D representation with a 2D pre-trained image diffusion prior based on score distillation sampling, as proposed by Poole et al. (2023). In this paradigm, the scene is represented as a differentiable image parameterization (DIP) denoted as $\theta$, where the image can be differentiably rendered based on the given camera parameters through a transformation function $g$. The DIP $\theta$ is iteratively refined to ensure that, for any given camera pose, the rendered image $\mathbf{x} = g(\theta)$ closely resembles a plausible sample derived from the guidance diffusion model. DreamFusion achieves this by leveraging Imagen (Saharia et al., 2022) to provide a score estimation function denoted as $\epsilon_\phi(x_t; y, t)$, where $x_t$, $y$, and $t$ represent the noisy image, text embedding, and timestep, respectively. This estimated score plays a pivotal role in guiding the gradient update, as expressed by the following equation:

$$\nabla_\theta \mathcal{L}_{\text{SDS}} = \mathbb{E}_{\epsilon,t}\left[w(t)(\epsilon_\phi(x_t; y, t) - \epsilon)\frac{\partial \mathbf{x}}{\partial \theta}\right] \tag{1}$$

where $\epsilon$ is a Gaussian noise and $w(t)$ is a weighting function. Our approach combines score distillation sampling with 3D Gaussian Splatting at both 2D and 3D levels with different diffusion models to generate 3D assets with both detailed appearance and 3D-consistent geometry.

### 3.2 3D GAUSSIAN SPLATTING

Gaussian Splatting, as introduced in Kerbl et al. (2023), presents a pioneering method for novel view synthesis and 3D reconstruction from multi-view images. Unlike NeRF, 3D Gaussian Splatting adopts a distinctive approach, where the underlying scene is represented through a set of anisotropic 3D Gaussians parameterized by their positions, covariances, colors, and opacities. When rendering, the 3D Gaussians are projected onto the camera's imaging plane (Zwicker et al., 2001). Subsequently, the projected 2D Gaussians are assigned to individual tiles. The color of $\mathbf{p}$ on the image plane is rendered sequentially with point-based volume rendering technique (Zwicker et al., 2001):

$$C(\mathbf{p}) = \sum_{i\in\mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j) \quad \text{where, } \alpha_i = o_i e^{-\frac{1}{2}(\mathbf{p}-\mu_i)^T \Sigma_i^{-1}(\mathbf{p}-\mu_i)}, \tag{2}$$

where $c_i$, $o_i$, $\mu_i$, and $\Sigma_i$ represent the color, opacity, position, and covariance of the $i$-th Gaussian respectively, and $\mathcal{N}$ denotes the Gaussians in this tile. To maximize the utilization of shared memory,

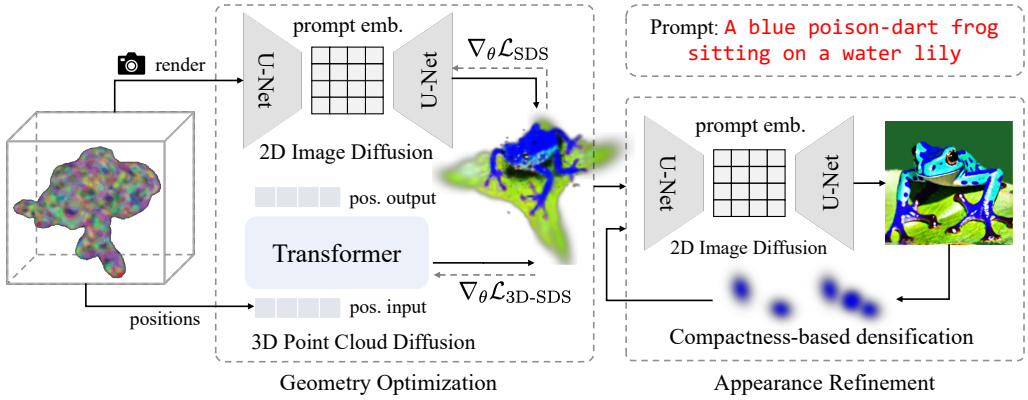



Figure 3: **Overview of the proposed GSGEN**. Our approach aims at generating 3D assets with accurate geometry and delicate appearance. GSGEN starts by utilizing Point-E to initialize the positions of the Gaussians (Sec 4.3). The optimization is grouped into geometry optimization (Sec 4.1) and appearance refinement (Sec 4.2) to meet a balance between coherent geometry structure and highly detailed texture.

Gaussian Splatting further designs a GPU-friendly rasterization process where each thread block is assigned to render an image tile. These advancements enable Gaussian Splatting to achieve more detailed scene reconstruction, significantly faster rendering speed, and reduction of memory usage during training compared to NeRF-based methods. In this study, we expand the application of Gaussian Splatting into text-to-3D generation and introduce a novel approach that leverages the explicit nature of Gaussian Splatting by integrating 3D diffusion priors, highlighting the potential of 3D Gaussians as a fundamental representation for generative tasks.

## 4 APPROACH

Our goal is to generate 3D content with accurate geometry and delicate detail. To accomplish this, GSGEN exploits the 3D Gaussians as representation due to its flexibility to incorporate geometry priors and capability to represent high-frequency details. Based on the observation that a point cloud can be seen as a set of isotropic Gaussians, we propose to integrate a 3D SDS loss with a pre-trained point cloud diffusion model to shape a 3D-consistent geometry. With this additional geometry prior, our approach could mitigate the Janus problem and generate more sensible geometry. Subsequently, in appearance refinement, the Gaussians undergo an iterative optimization to gradually improve fine-grained details with a compactness-based densification strategy, while preserving the fundamental geometric information. The detailed GSGEN methodology is presented as follows.

### 4.1 GEOMETRY OPTIMIZATION

Many text-to-3D methods encounter the significant challenge of overfitting to several views, resulting in assets with multiple faces and collapsed geometry (Poole et al., 2023; Lin et al., 2023; Chen et al., 2023). This issue, known as the Janus problem (Armandpour et al., 2023; Seo et al., 2023), has posed a persistent hurdle in the development of such methodologies. In our early experiments, we faced a similar challenge that relying solely on 2D guidance frequently led to collapsed results. However, we noticed that the geometry of 3D Gaussians can be directly rectified with a point cloud prior, which is not feasible for previous text-to-3D methods using NeRFs and DMTET. Recognizing this distinctive advantage, we introduce a geometry optimization process to shape a reasonable geometry. Concretely, in addition to the ordinary 2D image diffusion prior, we further optimize the positions of Gaussians using Point-E (Nichol et al., 2022), a pre-trained text-to-point-cloud diffusion model. Instead of directly aligning the Gaussians with a Point-E generated point cloud, we apply a 3D SDS loss to lead the positions inspired by image diffusion SDS, which avoids challenges including registration, scaling, and potential degeneration. Notably, we only apply the Point-E SDS gradients to positions, as empirical observations suggest that Point-E may generate relatively simple color patterns. We summarize the loss in the geometry optimization stage as the following equation:

$$\nabla_\theta \mathcal{L}_{\text{geometry}} = \mathbb{E}_{\epsilon_I, t}\left[ w_I(t)(\epsilon_\phi(x_t; y, t) - \epsilon_I)\frac{\partial \mathbf{x}}{\partial \theta} \right] + \lambda_{\text{3D}} \cdot \mathbb{E}_{\epsilon_P, t}\left[ w_P(t)(\epsilon_\psi(p_t; y, t) - \epsilon_P) \right], \quad (3)$$

where $p_t$ and $x_t$ represent the noisy Gaussian positions and the rendered image, $w_*$ and $\epsilon_*$ refer to the corresponding weighting function and Gaussian noise.

### 4.2 APPEARANCE REFINEMENT

While the introduction of 3D prior does help in learning a more reasonable geometry, we experimentally find it would also disturb the learning of appearance, resulting in insufficiently detailed assets. Based on this observation, GSGEN employs another appearance refinement stage that iteratively optimizes and densifies the Gaussians utilizing only the 2D image prior. To densify the Gaussians, Kerbl et al. (2023) propose to split Gaussians with a large view-space spatial gradient. However, we encountered challenges in determining the appropriate threshold for this spatial gradient under score distillation sampling. Due to the stochastic nature of SDS loss, employing a small threshold is prone to be misled by some stochastic large gradient thus generating an excessive number of Gaussians, whereas a large threshold will lead to a blurry appearance, as illustrated in Fig.8. To tackle this, we propose compactness-based densification as a supplement to positional gradient-based split. Specifically, for each Gaussian, we first obtain its K nearest neighbors with a KD-Tree. Then, for each of the neighbors, if the distance between the Gaussian and its neighbor is smaller than the sum of their radius, a Gaussian will be added between them with a radius equal to the residual. As illustrated in Fig.4, compactness-based densification could "fill the holes", resulting in a more complete geometry. To prune unnecessary Gaussians, we add an extra loss to regularize opacity with a weight proportional to its distance to the center and remove Gaussians with opacity smaller than a threshold $\alpha_{min}$ periodically. Furthermore, we recognize the importance of ensuring the geometry consistency of the Gaussians throughout the refinement phase. With this concern, we penalize Gaussians which deviates significantly from their positions obtained during the preceding geometry optimization. The loss in the appearance refinement stage is summarized as the following:

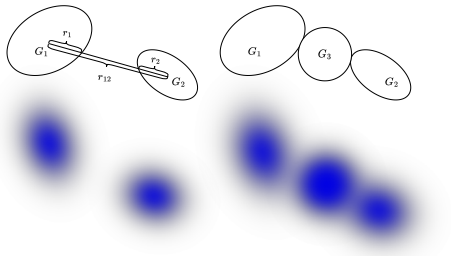


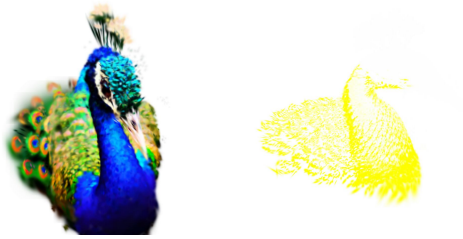

Figure 4: An illustration of the proposed compactness-based densification. For two Gaussians, if the distance between them ($r_{12}$) is larger than the sum of their radius ($r_1 + r_2$), a Gaussian will be augmented to achieve a more complete geometry.

$$\nabla_\theta \mathcal{L}_{\text{refine}} = \lambda_{\text{SDS}} \mathbb{E}_{\epsilon_I, t} \left[ w_I(t)(\epsilon_\phi(x_t; y, t) - \epsilon_I) \frac{\partial \mathbf{x}}{\partial \theta} \right] + \lambda_{\text{p}} \nabla_\theta \sum_i ||\mathbf{p}_i - \mathbf{p}_i^{(g)}|| + \lambda_{\text{o}} \nabla_\theta \sum_i \texttt{sg}(||\mathbf{p}_i||) \cdot o_i, \tag{4}$$

where $\texttt{sg}(\cdot)$ refers to the stop gradient operation, $\mathbf{p}_i$, $\mathbf{p}_i^{(g)}$ and $o_i$ represents the position, the position obtained through geometry optimization and opacity of the $i$-th Gaussian respectively. $\lambda_{\text{SDS}}$, $\lambda_{\text{p}}$ and $\lambda_{\text{o}}$ are loss weights for SDS loss, position regularization, and opacity regularization.

### 4.3 INITIALIZATION WITH GEOMETRY PRIOR

Previous studies (Chen et al., 2023; Lin et al., 2023; Metzer et al., 2022) have demonstrated the critical importance of starting with a reasonable geometry initialization. In our early experiments, we also found that initializing with a simple pattern could potentially lead to a degenerated 3D object. To overcome this, we opt for initializing the positions of the Gaussians either with a generated point cloud or with a 3D shape provided by the users. In the context of general text-to-3D generation, we employ a text-to-point-cloud diffusion model, *Point-E* (Nichol et al., 2022), to generate a rough geometry according to the text prompt. While Point-E can produce colored point clouds, we opt for random color initialization based on empirical observations,



*w/o Point-E color* *w/ Point-E color*

Figure 6: The impact of adopting Point-E generated color.


Figure 5: Qualitative comparison between the proposed GSGEN and previous state-of-the-art text-to-3D generation methods, including Magic3D (Lin et al., 2023) and Fantasia3D (Chen et al., 2023). Our approach achieves better visual quality, especially in high-frequency details, such as the thatched roof, intricate details of sushi and banana, and the feather of the peacock, etc. The prompts are provided under the images. For more qualitative comparison results, please refer to Appendix B.3. Videos of these images are provided in the supplemental video.

as direct utilization of the generated colors has been found to have detrimental effects in early experiments (shown in Fig.6). For user-guided generation, we convert the preferred shape to a point cloud to initialize the positions. To avoid too many vertices in the provided shape, we use farthest point sampling (Eldar et al., 1997) for point clouds and uniform surface sampling for meshes to extract a subset of the original shape instead of directly using all the vertices or points.

## 5 EXPERIMENTS

In this section, we present our experiments on validating the effectiveness of the proposed GSGEN. Specifically, we compare our approach with previous state-of-the-art methods in general text-to-3D generation. Additionally, we conduct several ablation studies to evaluate the importance of initialization, 3D guidance, and densification strategy. The detailed results are shown as follows.


Figure 7: Ablation study results on initialization and 3D prior. *Coarse Model* here refers to the rough assets obtained after geometry optimization. We can observe that the assets generated with random initialization suffer from degeneration severely, resulting in completely inconsistent geometry (in the first column). Although the Point-E initialized assets have a slightly better geometry, they still encounter the Janus problem (in the second column). The proposed GSGEN utilizes Point-E initialization and 3D guidance to generate shapes with better 3D consistency.

### 5.1 Implementation Details

**Guidance model setup.** We implement the guidance model based on the publicly available diffusion model, StableDiffusion (Rombach et al., 2022; von Platen et al., 2022). All the assets demonstrated in this section are obtained with the checkpoint *runwayml/stable-diffusion-v1-5*. For the guidance scale, we adopt 100 for *StableDiffusion* as suggested in DreamFusion and other works. We also exploit the view-dependent prompt technique proposed by DreamFusion.

**3D Gaussian Splatting setup.** We implement the 3D Gaussian Splatting in a pytorch CUDA extension. We split the Gaussians by view-space position gradient every 500 iterations with a threshold of 0.02 and perform compactness-based densification every 1000 iterations. We remove Gaussians with excessively large radii and opacity lower than $\alpha_{min} = 0.05$ every 200 iterations.

**Traning setup.** We use the same focal length, elevation, and azimuth range as those of DreamFusion (Poole et al., 2023). To sample more uniformly in the camera position, we employ a stratified sampling on azimuth. We choose the loss weight hyperparameters $\lambda_{\text{SDS}} = 0.1$ and $\lambda_{\text{3D}} = 0.01$ in geometry optimization, and $\lambda_{\text{SDS}} = 0.1$, $\lambda_{\text{p}} = 1.0$ and $\lambda_{\text{o}} = 100.0$ in appearance refinement.

### 5.2 Text-to-3D Generation

We evaluate the performance of the proposed GSGEN in the context of general text-to-3D generation and present qualitative comparison results against state-of-the-art methods. As illustrated in Fig.2, our approach produces delicate 3D assets with more accurate geometry and intricate details. In contrast, previous state-of-the-art methods (Tang, 2022; Poole et al., 2023; Lin et al., 2023; Guo et al., 2023; Chen et al., 2023) struggle in generating collapsed geometry under the same guidance and prompt, which underscores the effectiveness of our approach. We present more qualitative comparison results in Fig.5, where we compare the 3D assets generated by GSGEN with those generated by Magic3D (Lin et al., 2023) and Fantasia3D (Chen et al., 2023). Our approach showcases notable enhancements in preserving high-frequency details such as the intricate patterns on sushi, the feathers of the peacock, and the thatched roof. In contrast, Magic3D and Fantasia3D yield over-smoothed geometry due to the limitation of mesh-based methods, making the generated assets less realistic. For more one-to-one qualitative comparisons, please refer to the supplemental material for the video results and appendix B.3 for multi-view image comparison.

### 5.3 Ablation Study

**Initialization.** To assess the impact of initialization, we introduce a variant that initiates the positions of the Gaussians with an origin-centered Gaussian distribution which emulates the initialization adopted in DreamFusion (Poole et al., 2023). The qualitative comparisons are shown in Fig.7a. It is evident that assets generated with DreamFusion-like initialization encounter severe degeneration issues, especially for prompts depicting asymmetric scenes, resulting in a completely collapsed geometry. In contrast, Point-E initialization breaks the symmetry by providing an anisotropic geometry prior, leading to the creation of more 3D-consistent objects.

**3D Prior.** We evaluate the necessity of incorporating 3D prior by generating assets without point cloud guidance during geometry optimization. The qualitative comparisons of multi-view images are visualized in Fig.7b. Although achieved better geometry compared to DreamFusion-like initialization, relying solely on image diffusion prior still suffers from the Janus problem, which is particularly evident in cases with asymmetric geometries. In contrast, our approach effectively addresses this issue with the introduction of 3D prior, rectifying potentially collapsed structures in the geometry optimization stage and resulting in a 3D-consistent rough shape.

**Densification Strategy.** To valid the effectiveness of the proposed densification strategy, we propose two variants for comparison: (1) The original densification strategy that split Gaussians with an average view-space gradient larger than $T_{pos} = 0.0002$. (2) With larger $T_{pos} = 0.02$ that avoids too many new Gaussians. While effective in 3D reconstruction, the original densification strategy that relies only on view-space gradient encounters a dilemma in the context of score distillation sampling: within limited times of densification, a large threshold tends to generate an over-smoothed appearance while a small threshold is easily affected by unstable gradients. As shown in Fig.8, the proposed compactness-based densification is an effective supplement to the original densification strategy under SDS guidance, facilitating the generation of highly detailed assets. For more experiments including ablations on the 3D guidance model and 2D diffusion prior, please refer to appendix B.4.


Figure 8: Ablation study on densification strategy. The textual prompt used in this figure is *A mug of hot chocolate with whipped cream and marshmallows*.

## 6 Limitations and Conclusion

**Limitations.** GSGEN tends to generate unsatisfying results when the provided text prompt contains a complex description of the scene or with complicated logic due to the limited language understanding ability of Point-E and the CLIP text encoder used in *StableDiffusion*. Moreover, although incorporating 3D prior mitigates the Janus problem, it is far from eliminating the potential degenerations, especially when the textual prompt is extremely biased in the guidance diffusion models. Concrete failure cases and corresponding analyses are illustrated in appendix C.

**Conclusion.** In this paper, we propose GSGEN, a novel method for generating highly detailed and 3D consistent assets using Gaussian Splatting. In particular, we adopt a two-stage optimization strategy including geometry optimization and appearance refinement. In the geometry optimization stage, a rough shape is established under the joint guidance of a point cloud diffusion prior along with the ordinary image SDS loss. In the subsequent appearance refinement, the Gaussians are further optimized to enrich details and densified to achieve better continuity and fidelity with compactness-based densification. We conduct comprehensive experiments to validate the effectiveness of the proposed method, demonstrating its ability to generate 3D consistent assets and superior performance in capturing high-frequency components. We hope our method can serve as an efficient and effective approach for high-quality text-to-3D generation and could pave the way for more extensive applications of Gaussians Splatting and direct incorporation of 3D prior.

### ETHICS STATEMENT

In our endeavor to advance 3D generative modeling, we remain steadfast in our commitment to ethical principles. It is essential to recognize that unscrupulous individuals could potentially exploit generative models to create deceptive content, particularly when presented in the form of 3D objects, which can be more convincing than 2D images. Our approach, based on StableDiffusion, inherits any biases present in the training data. Therefore, great care must be taken when selecting datasets for text-to-image and image-to-3D models to prevent the perpetuation of harmful content. We firmly believe that with appropriate regulation, the positive impact of generative models far surpasses their negative potential.

## REFERENCES

Alex, Misha Konstantinov, apolinário, Daria Bakshandaeva, Ksenia Ivanova, Sayak Paul, Will Berman, and Emad. deep-floyd/if, 6 2023. URL https://github.com/deep-floyd/IF.

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

Benjamin Attal, Jia-Bin Huang, Christian Richardt, Michael Zollhoefer, Johannes Kopf, Matthew O'Toole, and Changil Kim. Hyperreel: High-fidelity 6-dof video with ray-conditioned sampling. *arXiv preprint arXiv:2301.02238*, 2023.

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *CoRR*, abs/2211.01324, 2022. doi: 10.48550/arXiv.2211.01324. URL https://doi.org/10.48550/arXiv.2211.01324.

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=0xiJLKH-ufZ.

Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *CVPR*, 2023.

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*, pp. 5855–5864, 2021.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. *arXiv preprint arXiv:2203.09517*, 2022a.

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 IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023.

Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Ying Feng, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12943–12952, 2022b.

Yongwei Chen, Rui Chen, Jiabao Lei, Yabin Zhang, and Kui Jia. TANGO: text-driven photorealistic and robust 3d stylization via lighting decomposition. In *NeurIPS*, 2022c. URL http://papers.nips.cc/paper_files/paper/2022/hash/c7b925e600ae4880f5c5d7557f70a72b-Abstract-Conference.html.

Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander Schwing, and Liangyan Gui. SDFusion: Multimodal 3d shape completion, reconstruction, and generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 8780–8794, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y. Zeevi. The farthest point strategy for progressive image sampling. *IEEE Trans. Image Process.*, 6(9):1305–1315, 1997. doi: 10.1109/83.623193. URL https://doi.org/10.1109/83.623193.

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10686–10696. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01043. URL https://doi.org/10.1109/CVPR52688.2022.01043.

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

Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. *ICCV*, 2021.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *CoRR*, abs/2207.12598, 2022. doi: 10.48550/arXiv.2207.12598. URL https://doi.org/10.48550/arXiv.2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 13213–13232. PMLR, 2023. URL https://proceedings.mlr.press/v202/hoogeboom23a.html.

Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 857–866. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00094. URL https://doi.org/10.1109/CVPR52688.2022.00094.

Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *CoRR*, abs/2305.02463, 2023. doi: 10.48550/arXiv.2305.02463. URL https://doi.org/10.48550/arXiv.2305.02463.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Popa Tiberiu. Clip-mesh: Generating textured meshes from text using pretrained image-text models. *SIGGRAPH Asia 2022 Conference Papers*, December 2022.

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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *https://arxiv.org/abs/2303.11328*, 2023a.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. *arXiv preprint arXiv:2309.06380*, 2023b.

Jonathan Lorraine, Kevin Xie, Xiaohui Zeng, Chen-Hsuan Lin, Towaki Takikawa, Nicholas Sharp, Tsung-Yi Lin, Ming-Yu Liu, Sanja Fidler, and James Lucas. Att3d: Amortized text-to-3d object synthesis. In *International Conference on Computer Vision ICCV*, 2023.

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. *arXiv preprint arXiv:2206.00927*, 2022.

Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021.

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.

Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 13482–13492. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01313. URL https://doi.org/10.1109/CVPR52688.2022.01313.

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. In *ECCV*, 2020.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. doi: 10.1145/3528223.3530127. URL https://doi.org/10.1145/3528223.3530127.

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *CoRR*, abs/2212.08751, 2022. doi: 10.48550/arXiv.2212.08751. URL https://doi.org/10.48550/arXiv.2212.08751.

Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: improving latent diffusion models for high-resolution image synthesis. *CoRR*, abs/2307.01952, 2023. doi: 10.48550/arXiv.2307.01952. URL https://doi.org/10.48550/arXiv.2307.01952.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=FjNys5c7VyY.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10318–10327, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL http://proceedings.mlr.press/v139/radford21a.html.

Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Nataniel Ruiz, Ben Mildenhall, Shiran Zada, Kfir Aberman, Michael Rubinstein, Jonathan Barron, Yuanzhen Li, and Varun Jampan. DreamBooth3D: Subject-driven text-to-3d generation. In *International Conference on Computer Vision ICCV*, 2023.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022. doi: 10.48550/arXiv.2204.06125. URL https://doi.org/10.48550/arXiv.2204.06125.

Christian Reiser, Richard Szeliski, Dor Verbin, Pratul P. Srinivasan, Ben Mildenhall, Andreas Geiger, Jonathan T. Barron, and Peter Hedman. Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. *SIGGRAPH*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10674–10685. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01042. URL https://doi.org/10.1109/CVPR52688.2022.01042.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/ec795aeadae0b7d230fa35cbaf04c041-Abstract-Conference.html.

Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, and Marco Fumero. Clip-forge: Towards zero-shot text-to-shape generation. *arXiv preprint arXiv:2110.02624*, 2021.

Sara Fridovich-Keil and Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, 2023.

Junyoung Seo, Wooseok Jang, Min-Seop Kwak, Jaehoon Ko, Hyeonsu Kim, Junho Kim, Jin-Hwa Kim, Jiyoung Lee, and Seungryong Kim. Let 2d diffusion model know 3d-consistency for robust text-to-3d generation. *arXiv preprint arXiv:2303.07937*, 2023.

Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5459–5469, 2022.

Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8248–8258, 2022.

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

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 3825–3834. IEEE, 2022a. doi: 10.1109/CVPR52688.2022.00381. URL `https://doi.org/10.1109/CVPR52688.2022.00381`.

Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, and Huaping Liu. Mixed neural voxels for fast multi-view video synthesis. *arXiv preprint arXiv:2212.00190*, 2022b.

Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 12619–12629. IEEE, 2023a. doi: 10.1109/CVPR52729.2023.01214. URL `https://doi.org/10.1109/CVPR52729.2023.01214`.

Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, and Baining Guo. RODIN: A generative model for sculpting 3d digital avatars using diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 4563–4573. IEEE, 2023b. doi: 10.1109/CVPR52729.2023.00443. URL `https://doi.org/10.1109/CVPR52729.2023.00443`.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023c.

Zhongshu Wang, Lingzhi Li, Zhen Shen, Li Shen, and Liefeng Bo. 4k-nerf: High fidelity neural radiance fields at ultra high resolutions. *arXiv preprint arXiv:2212.04701*, 2022c.

Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning unified representation of language, image and point cloud for 3d understanding. *arXiv preprint arXiv:2212.05171*, 2022.

Le Xue, Ning Yu, Shu Zhang, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip-2: Towards scalable multimodal pre-training for 3d understanding, 2023.

Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *arXiv preprint arXiv:2112.05131*, 2021a.

Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5752–5761, 2021b.

Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

Junzhe Zhu and Peiye Zhuang. Hifa: High-fidelity text-to-3d with advanced diffusion guidance. *CoRR*, abs/2305.18766, 2023. doi: 10.48550/arXiv.2305.18766. URL `https://doi.org/10.48550/arXiv.2305.18766`.

Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus H. Gross. EWA volume splatting. In Thomas Ertl, Kenneth I. Joy, and Amitabh Varshney (eds.), *12th IEEE Visualization Conference, IEEE Vis 2001, San Diego, CA, USA, October 24-26, 2001, Proceedings*, pp. 29–36. IEEE Computer Society, 2001. doi: 10.1109/VISUAL.2001.964490. URL https://doi.org/10.1109/VISUAL.2001.964490.

# A Implementation Details

**3D Gaussian Splatting Details.** Instead of directly using the official 3D Gaussian Splatting code provided by Kerbl et al. (2023), we reimplement this algorithm by ourselves due to the need to support learnable MLP background. The official 3D Gaussian Splatting implementation propagates the gradients of the Gaussians in an inverse order, i.e. the Gaussians rendered last get gradient first. Our implementation follows a plenoxel (Yu et al., 2021a) style back propagation that calculates the gradient in the rendering order, which we found much easier to incorporate a per-pixel background.

The depth maps are rendered using the view-space depth of the centers of the Gaussians, which we claim is accurate enough due to the tiny scale of the Gaussians (Zwicker et al., 2001). Besides, we implement a z-variance renderer to support z-var loss proposed by (Zhu & Zhuang, 2023). However, we found that z-var loss seems to have a limited impact on the generated 3D asset, mainly due to the sparsity of Gaussians naturally enforcing a relatively thin surface.

During rendering and optimizing, we follow the original 3D Gaussian Splatting to clamp the opacity of the Gaussians into $[0.004, 0.99]$ to ensure a stable gradient and prevent potential overflows or underflows.

**Guidance Details.** All the guidance of 2D image diffusion models we used in this paper is provided by huggingface diffusers (von Platen et al., 2022). For StableDiffusion guidance, we opt for the *runwayml/stable-diffusion-v1-5* checkpoint for all the experiments conducted in this paper. We also test the performance of GSGEN under other checkpoints, including *stabilityai/stable-diffusion-2-base* and *stabilityai/stable-diffusion-2-1-base*, but no improvements are observed.

For Point-E diffusion model and its checkpoints, we directly adopted their official implementation.

**Training Details.** All the assets we demonstrate in this paper and the supplemental video are trained on 4 NVIDIA 3090 GPUs with a batch size of 8 and take about 30 min to optimize for a prompt. We have also observed that our approach can be trained on a single GPU with over 11 GB of VRAM in approximately 1 hour and 40 minutes, using a batch size of 8. The 3D contents we showcase in this paper and supplemental video are obtained under the same hyper-parameter setting since we found our parameters robust toward the input prompt. The number of Gaussians after densification is around $[1e^5, 1e^6]$.

**Open-Sourced Resources and Corresponding Licenses.** We summarize open-sourced code and resources with corresponding licenses used in our experiments in the following table.

Table 1: Open-sourced resources used in the experiment in this work.

| Resource | License |
|---|---|
| Stable DreamFusion (Tang, 2022) | Apache License 2.0 |
| Fantasia3D (Chen et al., 2023) | Apache License 2.0 |
| threestudio (Guo et al., 2023) | Apache License 2.0 |
| StableDiffusion (Rombach et al., 2022) | MIT License |
| DeepFloyd IF (Alex et al., 2023) | DeepFloyd IF License Agreement |
| HuggingFace Diffusers | Apache License 2.0 |
| OpenAI Point-E | MIT License |
| ULIP | BSD 3-Clause License |

We use Stable DreamFusion and threestudio to obtain the results of DreamFusion and Magic3D under StableDiffusion and on the prompts that are not included in their papers and project pages since the original implementation has not been open-sourced due to the usage of private diffusion models. The results of Fatansia3D are obtained by running their official implementation with their parameter setting for dog-like shapes.

# B Addtional Results

## B.1 User-Guided Generation

Initialization is straightforward for 3D Gaussian Splatting due to its explicit nature, thereby automatically supporting user-guided generation. We evaluate the proposed GSGEN on user-guided generation with shapes provided in Latent-NeRF (Metzer et al., 2022). In this experiment, the initial points are generated by uniformly sampling points on the mesh surface. To better preserve the user's desired shape, we opt for a relatively small learning rate for positions. We compare the 3D content generated by GSGEN with those of the state-of-the-art user-guided 3D generation method Latent-NeRF (Metzer et al., 2022) and Fantasia-3D (Chen et al., 2023) in Fig.9. Our proposed GSGEN achieves the best results among all alternatives in both geometry and textures and mostly keeps the geometrical prior given by the users.

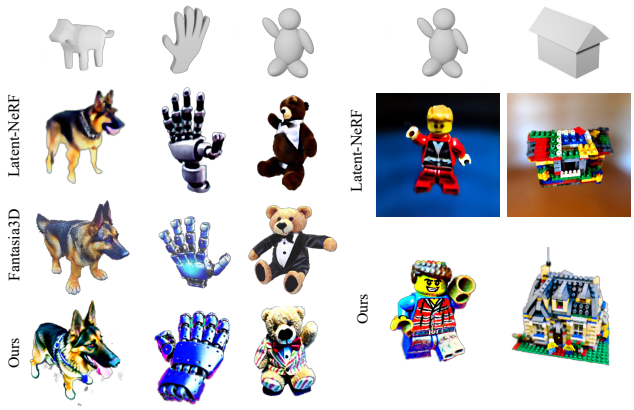



Figure 9: Qualitative comparison results on user-guided generation. The prompts from left to right are *(1)A German Sheperd; (2)A robot hand, realistic; (3) A teddy bear in a tuxedo; (4) a lego man; (5) a house made of lego*.

### B.2 More Text-To-3D Results

We present more general text-to-3D generation results of GSGEN in Fig.14 and Fig.15. Our approach can generate 3D assets with accurate geometry and improved fidelity. For more delicate assets generated with GSGEN and their spiral videos, please visit our project page `gsgen3d.github.io` or watch our supplemental video.

### B.3 More Qualitative Comparisons

In addition to the qualitative comparison in the main text, we provide more comparisons with DreamFusion (Poole et al., 2023) in Fig.16 and Fig.17, Magic3D (Lin et al., 2023) in Fig.18, Fantasia3D (Chen et al., 2023) and LatentNeRF (Metzer et al., 2022) in Fig.19. In order to make a fair comparison, the images of these methods are directly copied from their papers or project pages. Video comparisons are presented in the supplemental video.

### B.4 More Ablations

#### B.4.1 3D Point Cloud Guidance

Except for the Point-E (Nichol et al., 2022) used in our proposed GSGEN, we also test a CLIP-like text-to-point-cloud generation model ULIP (Xue et al., 2022; 2023). While achieving superior performance in zero-shot point cloud classification, ULIP seems ineffective in the context of generation. Fig.10 demonstrates point clouds generated under the guidance of ULIP and Point-E. Under SDS loss, Point-E can guide the point cloud to a consistent rough shape while ULIP leads to a mess. We substitute the 3D prior in GSGEN from Point-E to ULIP in Fig.11, yielding the same results as point cloud optimization.

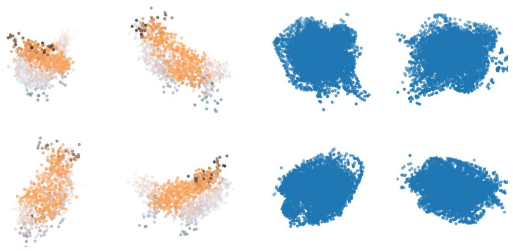
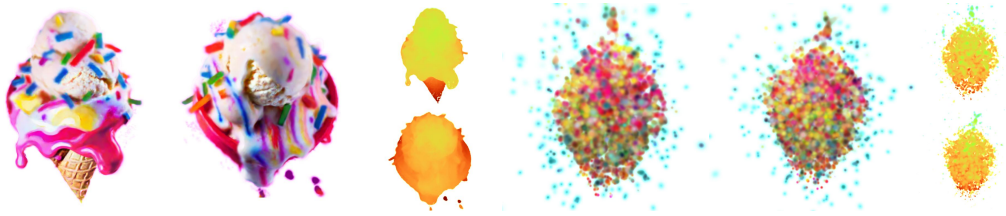



Figure 10: Point clouds optimized under *Point-E* and *ULIP*. Prompt: *A corgi.*



GSGEN under Point-E guidance GSGEN under ULIP guidance

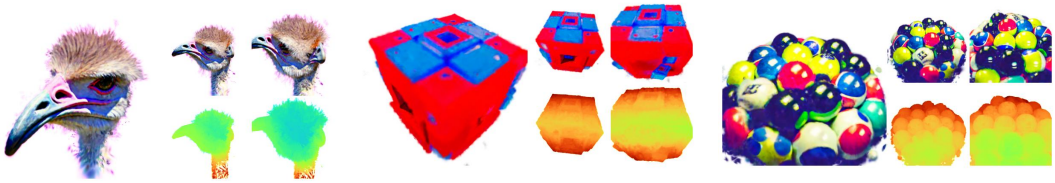
Figure 11: Text-to-3D generation qualitative comparison with 3D prior as Point-E or ULIP. Prompt: *A DSLR photo of an ice cream sundae.*

### B.4.2 2D IMAGE GUIDANCE

Except for StableDiffusion, we also test the performance of GSGEN under the guidance of *DeepFloyd IF*, another open-sourced cutting-edge text-to-image diffusion model. Compared to StableDiffusion, DeepFloyd IF has an Imagen-like architecture and a much more powerful text encoder. We demonstrate the qualitative comparison between GSGEN under different guidance in Fig.20. Obviously, assets generated with *DeepFloyd IF* have a much better text-3D alignment, which is primarily attributed to the stronger text understanding provided by T-5 encoder than that of CLIP text encoder. However, due to the modular cascaded design, the input to *DeepFloyd IF* has to be downsampled to $64 \times 64$, which may result in a blurry appearance compared to those generated under StableDiffusion.

## C FAILURE CASES

Despite the introduction of 3D prior, we could not completely eliminate the Janus problem, due to the ill-posed nature of text-to-3D through 2D prior and the limited capability of the 3D prior we used.



(a) *A high quality photo of an ostrich* (b) *A small red cube is sitting on top of a large blue cube. red on top, blue on bottom* (c) *a zoomed out DSLR photo of a few pool balls sitting on a pool table*

Figure 12: Three typical failure cases of GSGEN.

Fig.12 showcases three typical failure cases we encountered in our experiments. In Fig.12a, the geometrical structure is correctly established, but the Janus problem happens on the appearance (another ostrich head on the back head). Fig.12b demonstrates another failure case caused by the limited language understanding of the guidance model. StableDiffusion also fails to generate reasonable images with these prompts, as illustrated in Fig.13.

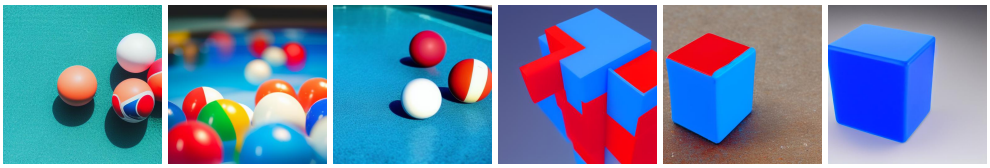


Figure 13: Prompts that StableDiffusion cannot correctly process, which leads to the failure of corresponding text-to-3D generation.

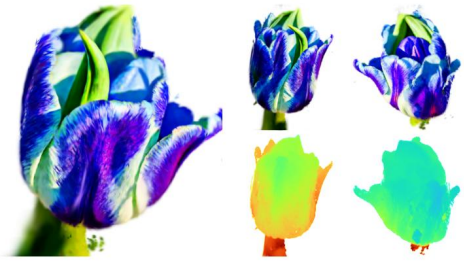
*A bunch of blue rose, highly detailed*

*A high quality photo of a dragon*

*A plush dragon toy*

*A zoomed out DSLR photo of a plate of fried chicken and waffles with maple syrup on them*

*A beautiful dress made of feathers, on a mannequin*

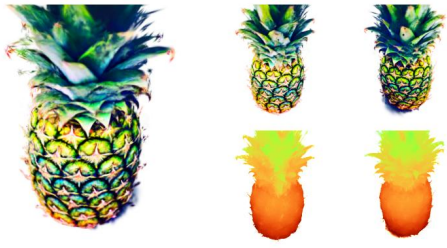


*A high quality photo of a blue tulip*

*A DSLR photo of a plush triceratops toy, studio lighting, hight resolution*

*A DSLR photo of a tray of Sushi containing pugs*

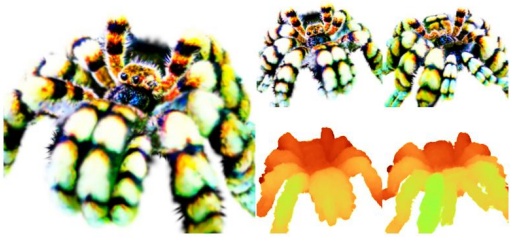
*A DSLR photo of an origami motorcycle*



*A DSLR photo of a pineapple*

Figure 14: More 3D assets generated with GSGEN.

*A high quality photo of a stack of pancakes covered in maple syrup*



*A tarantula, highly detailed*

*A sliced loaf of fresh bread*

*A high quality photo of a pinecone*

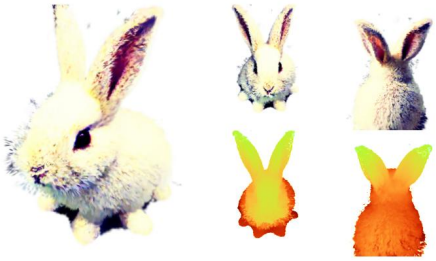
*A high quality photo of a durian*

*A zoomed out DSLR photo of a table with dim sum on it*

*A DSLR photo of a bald eagle*

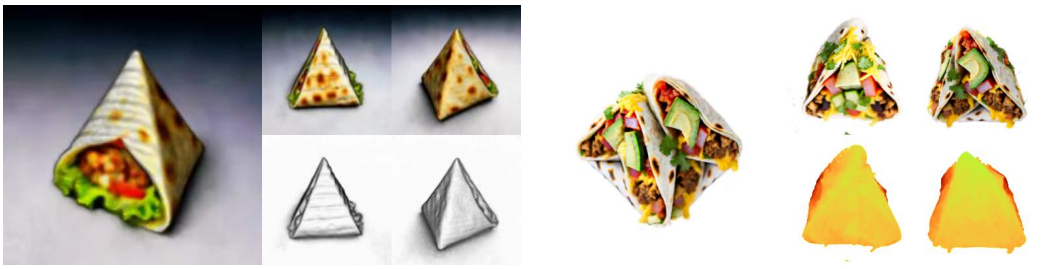
*A high quality photo of a chow chow puppy*

*A high quality photo of a kangaroo*



*A high quality photo of a furry rabbit*

Figure 15: More 3D assets generated with GSGEN.



*DreamFusion* *GSGEN*

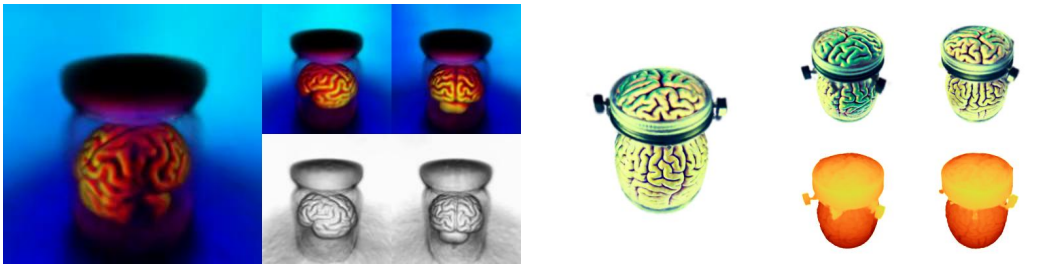
*A DSLR photo of pyramid shaped burrito with a slice cut out of it*

*DreamFusion* *GSGEN*

*A DSLR photo of a roast turkey on a platter*

*DreamFusion* *GSGEN*

*A plate of delicious tacos*



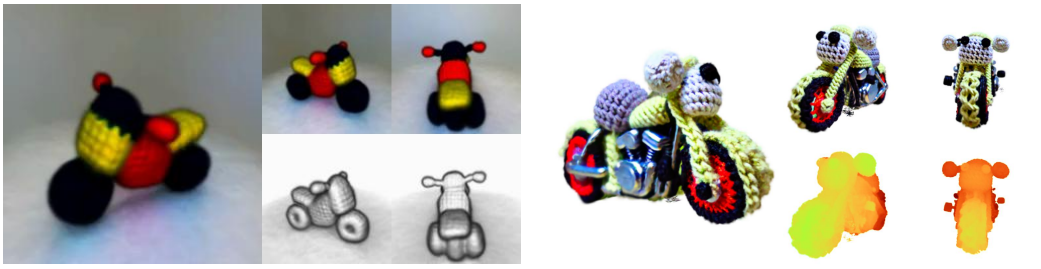
*DreamFusion* *GSGEN*

*A zoomed out DSLR photo of a brain in a jar*

*DreamFusion* *GSGEN*

*A zoomed out DSLR photo of a cake in the shape of a train*

Figure 16: More comparison results with DreamFusion.

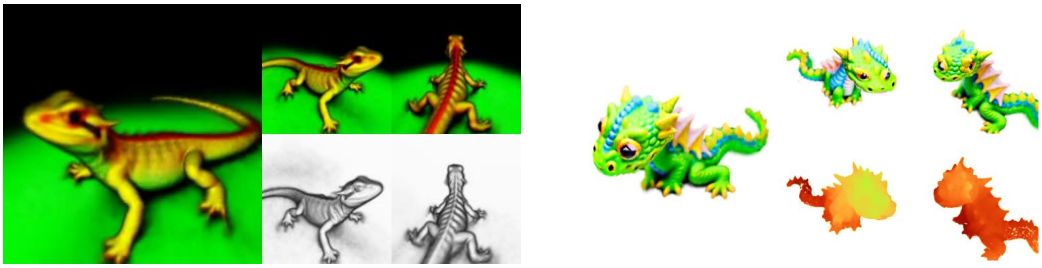


*DreamFusion* *GSGEN*

*A zoomed out DSLR photo of an amigurumi motorcycle*

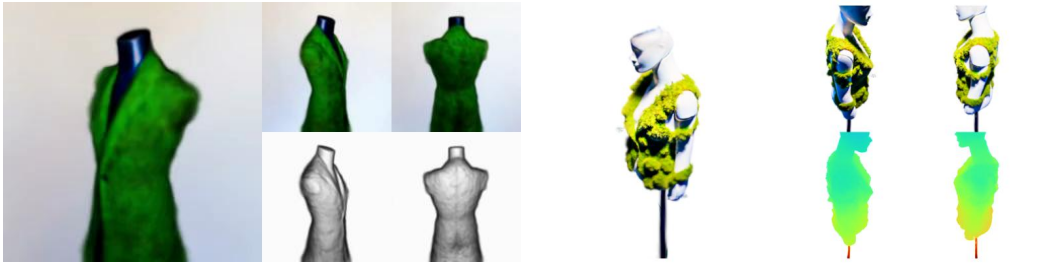
*DreamFusion* *GSGEN*

*A delicious hamburger*



*DreamFusion* *GSGEN*

*A zoomed out DSLR photo of a baby dragon*



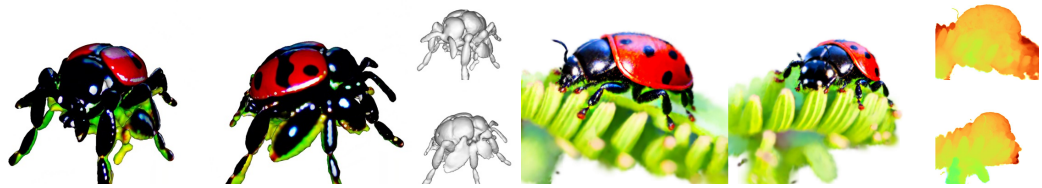
*DreamFusion* *GSGEN*

*A zoomed out DSLR photo of a beautiful suit made out of moss, on a mannequin. Studio lighting, high quality, high resolution*

*DreamFusion* *GSGEN*

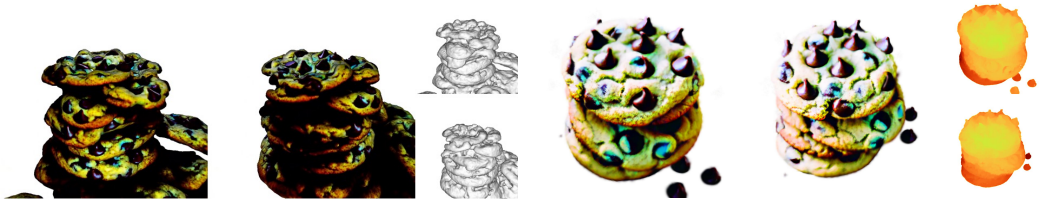
*A zoomed out DSLR photo of a complex movement from an expensive watch with many shiny gears, sitting on a table*

Figure 17: More comparison results with DreamFusion.



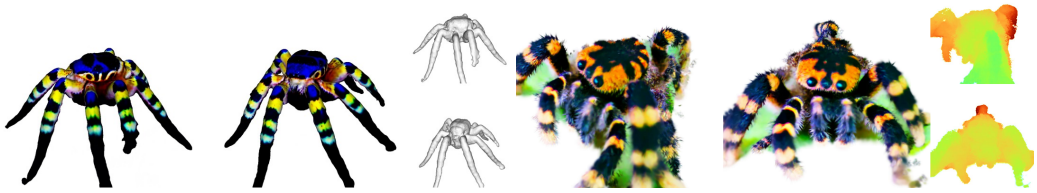
*Magic3D* *GSGEN*

*A zoomed out DSLR photo of a ladybug*



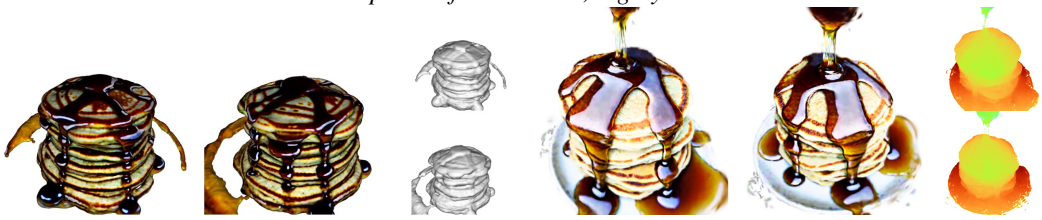
*Magic3D* *GSGEN*

*A zoomed out DSLR photo of a plate piled high with chocolate chip cookies*



*Magic3D* *GSGEN*

*A DSLR photo of a tarantula, highly detailed*



*Magic3D* *GSGEN*

*A DSLR photo of a stack of pancakes covered in maple syrup*

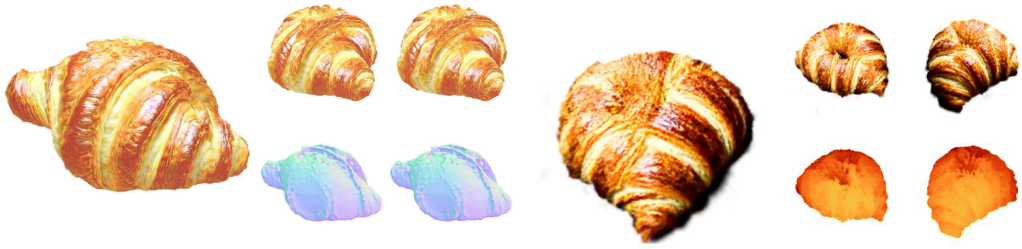
*Magic3D* *GSGEN*

*A zoomed out DSLR photo of a beautifully carved wooden knight chess piece*

Figure 18: More comparison results with Magic3D.

*Fantasia3D* *GSGEN*

*A fresh cinnamon roll covered in glaze, high resolution*

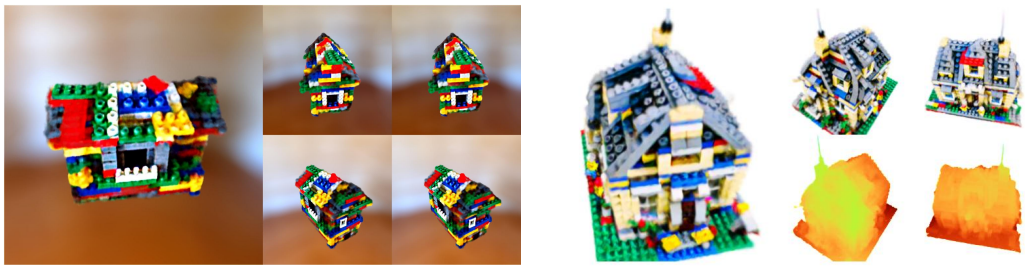


*Fantasia3D* *GSGEN*

*A delicious croissant*

*LatentNeRF* *GSGEN*

*A photo of a vase with sunflowers*



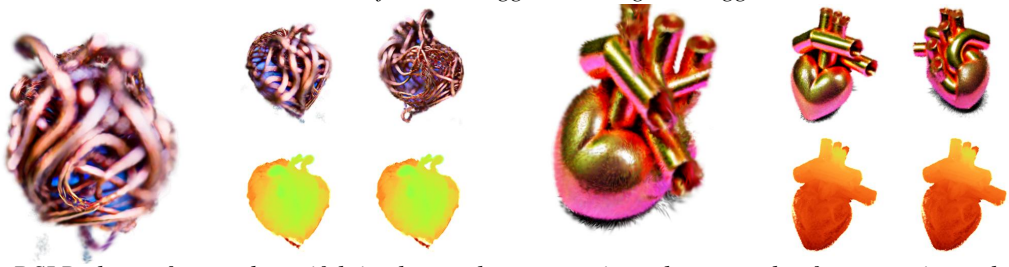
*LatentNeRF* *GSGEN*

*A house made of lego*

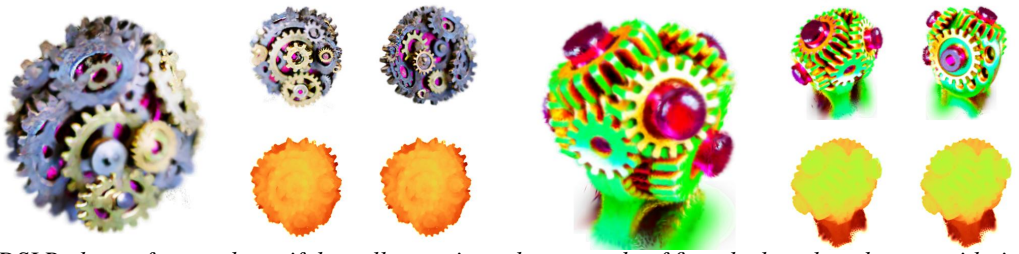
Figure 19: More comparison results with LatentNeRF and Fantasia3D.

GSGEN with *StableDiffusion* GSGEN with *DeepFloyd IF*

*A nest with a few white eggs and one golden egg*



*A DSLR photo of a very beautiful tiny human heart organic sculpture made of copper wire and threaded pipes, very intricate, curved, Studio lighting, high resolution*



*A DSLR photo of a very beautiful small organic sculpture made of fine clockwork and gears with tiny ruby bearings, very intricate, caved, curved. Studio lighting, High resolution*

*An anthropomorphic tomato eating another tomato*

Figure 20: Qualitative comparison of GSGEN under StableDiffusion guidance and DeepFloyd IF guidance.