

DIRECT-3D: Learning Direct Text-to-3D Generation on Massive Noisy 3D Data

Qihao Liu¹ Yi Zhang¹ Song Bai² Adam Kortylewski^{3,4} Alan Yuille¹

¹Johns Hopkins University ² ByteDance ³Max Planck Institute for Informatics ⁴University of Freiburg

<https://direct-3d.github.io/>

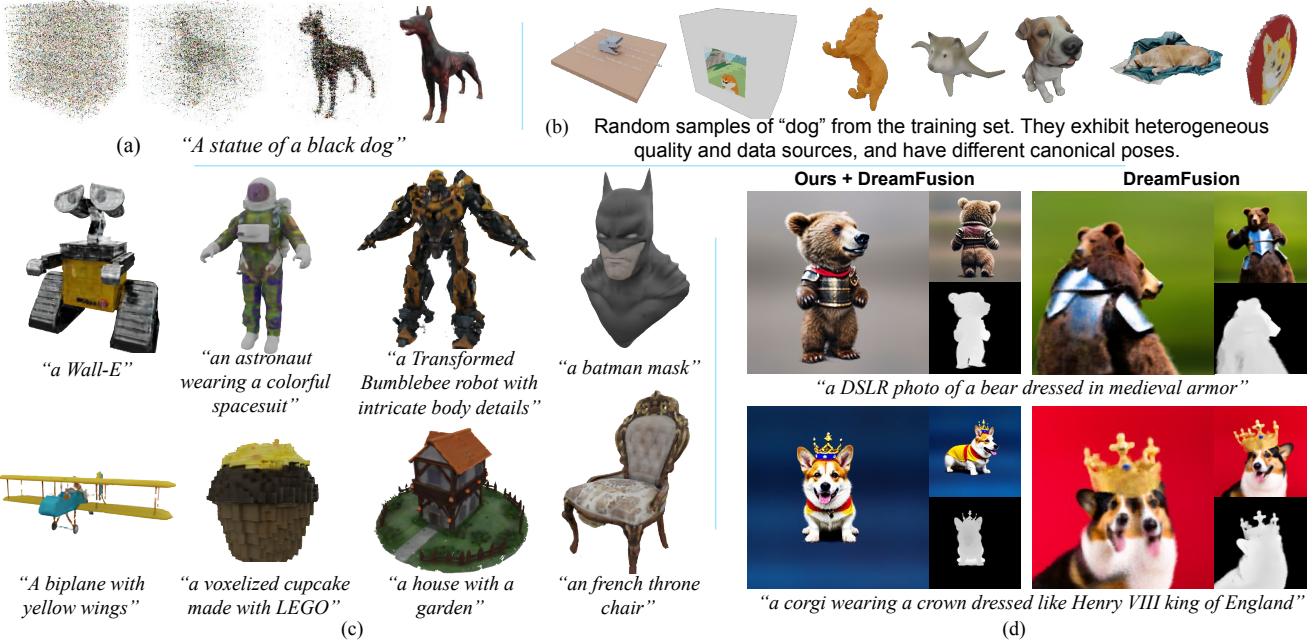


Figure 1. Different from optimization-based 2D-lifting methods such as DreamFusion [49], DIRECT-3D directly generates 3D contents in a single forward pass (a). To mitigate the lack of high-quality 3D data, DIRECT-3D enables efficient end-to-end training of 3D generative models on massive noisy and unaligned ‘in-the-wild’ 3D assets (b). Once trained, DIRECT-3D can generate high-quality 3D objects with accurate geometric details and various textures in 12 seconds on a single V100, driven by text prompts (c). DIRECT-3D can also be used as effective 3D geometry prior that significantly alleviates the Janus problem in 2D-lifting methods (d).

Abstract

We present DIRECT-3D, a diffusion-based 3D generative model for creating high-quality 3D assets (represented by Neural Radiance Fields) from text prompts. Unlike recent 3D generative models that rely on clean and well-aligned 3D data, limiting them to single or few-class generation, our model is directly trained on extensive noisy and unaligned ‘in-the-wild’ 3D assets, mitigating the key challenge (i.e., data scarcity) in large-scale 3D generation. In particular, DIRECT-3D is a tri-plane diffusion model that integrates two innovations: 1) A novel learning framework where noisy data are filtered and aligned automatically during the training process. Specifically, after an initial warm-up phase using a small set of clean data, an iterative optimization is introduced in the diffusion process to explicitly

estimate the 3D pose of objects and select beneficial data based on conditional density. 2) An efficient 3D representation that is achieved by disentangling object geometry and color features with two separate conditional diffusion models that are optimized hierarchically. Given a prompt input, our model generates high-quality, high-resolution, realistic, and complex 3D objects with accurate geometric details in seconds. We achieve state-of-the-art performance in both single-class generation and text-to-3D generation. We also demonstrate that DIRECT-3D can serve as a useful 3D geometric prior of objects, for example to alleviate the well-known Janus problem in 2D-lifting methods such as DreamFusion. The code and models are available for research purposes at: <https://github.com/qihao067/direct3d>.

1. Introduction

Diffusion models [27, 57] have achieved significant success in 2D image synthesis [4, 52, 53], owing to the large amount of image-text pairs and scaleable framework. However, applying diffusion models to the 3D domain is challenging, mostly due to the lack of 3D data: Current 3D datasets are orders of magnitude smaller than their 2D counterparts, and also exhibit significant disparities in quality and complexity. Specifically, the most widely-used dataset (*i.e.* ShapeNet [10]) comprises only 51K 3D models and focuses on individual objects. Larger datasets like Objaverse [18] and Objaverse-XL [17], despite containing over 10M objects from Sketchfab, are noisy in quality and lack alignment (*i.e.*, objects in varying poses). As clean and well-aligned data continue to be very important for current methods [11, 45, 55], people have to rely on high-quality yet small datasets like ShapeNet for training, and no previous 3D generative model can be directly trained on larger ‘in-the-wild’ 3D data such as Objaverse. As a result, these models are constrained to single-class generation, and can only generate objects with limited diversity and complexity, such as cars and tables. In addition, the lack of efficient network design poses additional challenges, as there is no consensus on 3D data representation or network architecture that can efficiently handle high-dimensional 3D data.

To circumvent the shortage of 3D data and efficient architectures, one line of work [37, 49] leverages image priors from 2D diffusion models to optimize a Neural Radiance Field (NeRF) [43]. However, they are time-consuming and fragile, and often lack of geometric consistency, leading to the Janus problem (*e.g.*, multiple faces on an animal). Recently, one important step was made by Shap-E [31] that directly models the distribution of large-scale 3D objects for implicit 3D representation generation. However, they do not address the aforementioned strict requirement for training data. Instead, they rely on vast amounts of proprietary data, which is time-consuming and costly to obtain, and they still need to invest considerable efforts to further enhance data quality [46]. In addition, Shap-E necessitates multi-stage training with a complex recipe, requiring point clouds and RGBA images with per-pixel 3D coordinates as input.

In this work, we present DIRECT-3D, a **D**iffusion model with **I**te**R**at**E** optimization for **C**onditional **T**ext-to-**3D** generation (Fig. 1). It enables direct training on massive noisy and unaligned ‘in-the-wild’ 3D data in an end-to-end manner, with multi-view images as supervision. Given a text prompt, it generates a variety of high-quality 3D objects (NeRFs) with precise geometric details and diverse textures within seconds. Our model consists of a 2D diffusion module to generate tri-plane features [9] and a NeRF decoder to extract NeRF parameters from the generated tri-plane. Tri-plane features facilitate an efficient 3D representation in well-established 2D networks, and NeRF offers an

effective and compact way to model intricate details of 3D objects. To tackle the aforementioned challenges, we made the following important technical innovations:

Firstly, we incorporate an iterative optimization process into the diffusion step to explicitly estimate the pose and quality of the 3D data based on the conditional density of the diffusion model, enabling automatic cleaning and alignment of the data during training. It considerably reduces the need for high-quality and precisely aligned 3D data and opens up a novel method to efficiently train 3D generative models on large amounts of ‘in-the-wild’ 3D assets. **Secondly**, we disentangle 3D geometry and 2D color of the object, modeling them hierarchically with two separate diffusion models. The geometry tri-plane is generated first, and the color is generated conditioned on geometry and the text prompt. This disentanglement enhances the efficiency and capabilities for modeling 3D data. It also allows for more flexible usage of our model. For example, our geometry diffusion module can be seamlessly integrated in existing Score Distillation Sampling [49] based approaches, and provide additional 3D geometry priors, which significantly improve the geometry consistency while preserving the high-fidelity texture from the 2D image diffusion models. **Finally**, we propose an automated method to generate multiple descriptive prompts for each object, spanning from coarse to fine-grained levels, which enhances the alignment between prompt features and the generated 3D objects.

We evaluate DIRECT-3D on both single-class generation and text-to-3D generation. For single-class generation, our method outperforms all previous methods on all tested categories by a large margin when trained on exactly the same data (*e.g.*, from 14.27 to 7.26 in FID), proving our effectiveness in modeling 3D data. For text-to-3D generation, we achieve superior performances compared to previous work (Shap-E [31]), excelling in quality, detail, complexity, and realism. User studies show that 73.9% of raters prefer our approach over Shap-E. In addition, when used as geometry prior, our method significantly improves the 3D consistency of previous 2D-lifting models (*e.g.* DreamFusion [49]), and raises the generation success rate from 12% to 84%.

In summary, we make the following contributions:

- We propose DIRECT-3D, which enables end-to-end training of 3D generative models on extensive noisy and unaligned ‘in-the-wild’ 3D data. It achieves state-of-the-art performance on both single-class and large-scale text-guided 3D generation.
- Given text prompts, DIRECT-3D is able to generate high-quality, high-resolution, realistic, and complex 3D objects (NeRFs) with precise geometric details in seconds.
- DIRECT-3D provides important and easy-to-use 3D geometry prior of arbitrary objects, complementing 2D priors provided by image diffusion models.

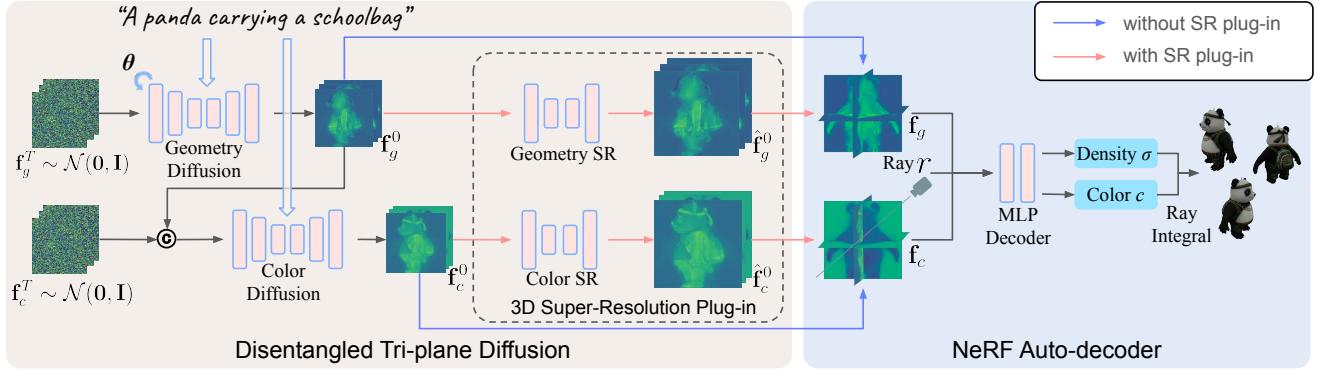


Figure 2. **Method overview.** Given a prompt, we generate a NeRF with two modules: The disentangled tri-plane diffusion module uses 2 (or 4 if the super-resolution plug-in is used) diffusion models to generate geometry (f_g) and color (f_c) tri-plane separately. Then both tri-planes are reshaped and fed into a NeRF auto-decoder to get the final outputs. During training, an iterative optimization process is introduced in the geometry diffusion to explicitly model the pose θ of objects and select beneficial ones, enabling efficient training on noisy ‘in-the-wild’ data. The whole model is end-to-end trainable (with or without SR plug-in), with only multi-view 2D images as supervision.

2. Related Work

Direct 3D generation. Early work relies on either GAN [23] or VAE [34] to model the distribution of 3D objects, represented by voxel grids [6, 22, 63], point clouds [1, 44, 66, 70], or implicit representations [13, 47, 56]. Recently, diffusion models [27, 57] have been utilized to create objects with appearance [2, 11, 24, 32, 33, 45] or pure geometric shapes [14, 21, 30, 36, 39, 55, 67–70]. However, these methods are constrained by their reliance on clean and well-aligned 3D datasets such as ShapeNet [10]. Hence, they can only focus on a single category or a few categories.

Recently, Cao *et al.* [7] train a class-conditional 3D diffusion model on OmniObject3D [64], which contains 216 object categories, enabling large-vocabulary 3D generation. However, their need for well-aligned 3D data limits their training set to just 5.9K objects, averaging only 27 objects per category, which severely restricts the quality and diversity. To enable large-scale 3D generation, Point-E [46] and Shap-E [31] train text-conditional diffusion models on massive proprietary data. However, acquiring such data is costly and time-consuming, and large efforts are still required to further enhance the data quality [46]. In contrast, we directly tackle this key constraint on training data by enabling direct training on extensive ‘in-the-wild’ 3D data, which is cost-effective and easy to scale up.

Text-to-3D generation with 2D diffusion. To circumvent the constraints imposed by limited 3D data and enable large-scale generation, another line of work [12, 29, 37, 42, 49, 59, 60, 62] leverages pre-trained 2D image diffusion priors for 3D generation. However, they are known for suffering from the Janus problem, in which radially asymmetric objects exhibit unintended symmetries, due to the lack of 3D consistency in 2D diffusion models. MV-Dream [54] mitigates this issue by fine-tuning a pre-trained image diffusion model to produce multi-view images, highlighting the

importance of 3D knowledge. In contrast, we directly generate objects in 3D space with accurate geometry information. Moreover, our method provides accurate 3d geometry priors to these 2D-based methods, complementing the 2D priors from image diffusion models, and hence effectively alleviating the Janus problem. In addition, these methods require tens of minutes to hours for optimizing a single object, whereas our method generates NeRFs in seconds.

3. Method

Our model consists of a tri-plane diffusion module to generate tri-planes of a 3D object, and a NeRF auto-decoder [47] to decode the tri-planes into final radiance field. In Sec. 3.1, we introduce our architecture design. Sec. 3.2 describes how we can train our model on noisy and unaligned 3D data. In Sec. 3.3, we introduce the 3D super-resolution plug-in for high-resolution generation. Sec. 3.4 describes an automated way to generate descriptive captions in different granularities. Training and implementation details are available in the Supp. An overall illustration is provided in Fig. 2.

3.1. Tri-plane Diffusion for NeRF Generation

NeRF generation from disentangled tri-plane representation. Given a set of 2D multi-view images of a subject, one can learn its 3D representation with a NeRF, which models the subject using volume density $\sigma \in \mathbb{R}_+$ and RGB color $c \in \mathbb{R}_+^3$. For a more efficient representation, we follow previous work [11, 61] that uses the tri-plane representation to model the NeRFs. Specifically, it factorizes a 3D volume into three axis-aligned orthogonal 2D feature planes $f_{xy}, f_{xz}, f_{yz} \in \mathbb{R}^{N \times N \times C}$. Then, one can query the feature f of any 3D point $p \in \mathbb{R}^3$ by projecting it onto each of the three planes and aggregating the retrieved features.

However, we find it necessary to disentangle the geometry and color features into two separate tri-planes, denoted

by \mathbf{f}_g and \mathbf{f}_c respectively, which improves model capability and provides important geometry prior (see Sec. 4.4.2). Then, with the tri-planes \mathbf{f}_g and \mathbf{f}_c , and a set of rays $\{r_i\}$, we can get the integral radiance y of this subject with an auto-decoder: $y_i = \mathcal{R}(\mathcal{D}_\omega(\mathbf{f}_g, \mathbf{f}_c, r_i))$, where \mathcal{D}_ω is a multi-layer perceptron decoder with parameters ω , \mathcal{R} denotes volume rendering [41], and i is the ray index. Our decoder processes the tri-planes \mathbf{f}_g and \mathbf{f}_c separately to generate density and color, thereby ensuring that \mathbf{f}_g only encapsulates the geometry information and \mathbf{f}_c only contains the corresponding color features (see Supp. for details). Given the ground-truth pixel RGB \hat{y} , the tri-planes \mathbf{f}_g , \mathbf{f}_c and parameters ω can be optimized by minimizing the rendering loss:

$$\mathcal{L}_{rad}(\mathbf{f}_g, \mathbf{f}_c, \omega) = \sum_i \|\hat{y}_i - \mathcal{R}(\mathcal{D}_\omega(\mathbf{f}_g, \mathbf{f}_c, r_i))\|_2^2 \quad (1)$$

Disentangled tri-plane generation. For conditional generation of tri-plane $\mathbf{f}_{(.)}$ from prompt p , we adopt a 2D latent diffusion model [27, 52]. In our framework, the diffusion model denoises tri-plane features $\mathbf{f}_g, \mathbf{f}_c \in \mathbb{R}^{N \times N \times 3C}$ that stack the channels of all three axes into a single image.

Given an input tri-plane \mathbf{f}_g^0 (or \mathbf{f}_c^0), the diffusion model progressively adds noise to it and produces a noisy output $\mathbf{f}_g^t := \alpha^t \mathbf{f}_g^0 + \sigma^t \epsilon$ at timestep t , where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the added Gaussian noise, α^t and σ^t are noise schedule functions. During each training step, we first train a geometry denoising network $\epsilon_\phi(\mathbf{f}_g^t, t, \tau(p))$ via

$$\mathcal{L}_{geo}(\phi) = \mathbb{E}_{\mathbf{f}_g^0, \epsilon, p, t} [\|\epsilon - \epsilon_\phi(\mathbf{f}_g^t, t, \tau(p))\|_2^2] \quad (2)$$

where τ denotes a pre-trained CLIP text encoder [50]. Then, a color denoising network $\epsilon_\psi(\mathbf{f}_c^t, t, \tau(p), \mathbf{f}_g)$ conditioned on both prompt p and geometry \mathbf{f}_g is optimized by

$$\mathcal{L}_{col}(\psi) = \mathbb{E}_{\mathbf{f}_c^0, \epsilon, p, \mathbf{f}_g, t} [\|\epsilon - \epsilon_\psi(\mathbf{f}_c^t, t, \tau(p), \mathbf{f}_g)\|_2^2] \quad (3)$$

Prompt condition is added by a cross-attention mechanism [52] with classifier-free guidance [26], and geometry condition for color diffusion is added via concatenation.

During inference, the geometry tri-plane \mathbf{f}_g^0 is sampled starting from the Gaussian noise $\mathbf{f}_g^T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ conditioned on prompt p , then the color tri-plane \mathbf{f}_c^0 is sampled similarly but conditioned on prompt p and geometry \mathbf{f}_g^0 .

3.2. Training with Noisy and Unaligned Data

Beyond our disentangled architecture and the introduced training objective, large-scale text-to-3D synthesis requires a substantial amount of 3D data for training. Recent efforts [17, 18] have gathered over 10M ‘in-the-wild’ 3D objects from Sketchfab. However, these datasets are difficult to use due to the heterogeneous quality and data sources, and the lack of alignment, leading to poor performance or even non-convergence during training (see

Sec. 4.4.1). Manual cleaning and alignment of 10M data is time-consuming and impractical to scale up. To this end, we introduce an iterative optimization process within the diffusion training step to autonomously identify noisy 3D data and automatically align clean data samples during training.

To achieve this goal, for each object, we explicitly model its 3D rotation angle as $\theta = \{\theta_\mu, \theta_\sigma\}$, where $\theta_\mu, \theta_\sigma \in \mathbb{R}^3$ denote the estimated mean and variance of its 3D rotation angle. Once estimated, the rotation angle can be sampled from $\mathcal{N}(\theta_\mu, \theta_\sigma)$. Note that the geometry tri-plane \mathbf{f}_g is now conditioned on the rotation θ , so Eqn. 2 becomes

$$\mathcal{L}_{geo}(\phi, \theta) = \mathbb{E}_{\mathbf{f}_g^0; \theta, \epsilon, p, t} [\|\epsilon - \epsilon_\phi(\mathbf{f}_g^t; \theta, t, \tau(p))\|_2^2] \quad (4)$$

Then, we can estimate the rotation parameter θ by also minimizing the diffusion loss $\mathcal{L}_{geo}(\phi, \theta)$. However, directly minimizing it w.r.t θ is challenging, since our model only uses multi-view images as supervision, and the tri-plane reconstruction already requires hundreds of optimization iterations per object (although effectively). Note that we do not need an accurate estimate of θ ; instead, a rough pose with good axis disentanglement in tri-plane suffices (see Fig. 5).

To perform this estimation, we consider θ as a hidden variable and propose an iterative optimization process. We first initialize the model with a very short warm-up phase on a small aligned dataset (details in Supp.). Subsequently, during each training iteration on the entire noisy dataset, we sample m different θ following $\mathcal{N}(\theta_\mu, \theta_\sigma)$ and estimate the corresponding tri-planes \mathbf{f}_g^0 . Then with a frozen geometry diffusion model, we compute the loss in Eqn. 4 with fixed parameter ϕ at a fixed time step t , which gives us a loss distribution w.r.t the sampled rotations θ . After that, we can update the rotation parameter by $\theta_\mu \leftarrow (1 - \lambda_\mu) \theta_\mu + \lambda_\mu \theta_{min}$ and $\theta_\sigma \leftarrow \lambda_\sigma |\theta_\mu - \theta_{min}|$, where θ_{min} is the sampled rotation with the smallest loss and $\lambda_{(.)}$ are momentum parameters. Finally, given threshold T , we can use θ_{min} to update the geometry denoising network ϵ_ϕ if $\mathcal{L}_{geo}(\phi, \theta_{min}) \leq T$.

We initialize θ_μ and θ_σ with all elements equal to 0 and π , respectively. Then we set $m = \text{ceil}(36/\pi \cdot \theta_\sigma)$, which is updated every iteration. In practice, it converges fast, often requiring just 5-10 iterations. We filter out the objects that do not converge after 10 iterations. This step does not require back-propagation through the diffusion model when optimizing θ , which also speeds up the process.

3.3. 3D Super Resolution

Directly training a high-resolution diffusion model is slow and inefficient. Therefore, we train our base module at a resolution of 128^2 , and rely upon a 3D Super-Resolution (SR) plug-in with the tri-plane diffusion structure to increase the resolution from 128^2 to 512^2 . Given a low-resolution tri-plane $\mathbf{f}_{(.)}$, we first apply a roll-out operation [61] that concatenates the tri-plane features horizontally, followed by a bilinear interpolation to get an intermediate tri-plane $\mathbf{f}'_{(.)}$ at

a resolution of 512^2 . Then, a parameterized diffusion model is used to directly predict the high-resolution tri-plane $\hat{f}_{(\cdot)}$. Alongside the L2 loss on tri-plane, we apply an entropy loss to the generated NeRF to encourage full transparent or opaque points, ensuring a smoother SR generation. It's worth noting that **our model can directly generate high-quality objects without the SR plug-in**. In fact, except for results in Fig. 1 (c), all experiments/results in this paper are conducted **without** the SR module to ensure fair comparisons with baselines, as they are all evaluated at 128^2 . More details are provided in the Supp.

3.4. Coarse to Fine-gained Caption Generation

Text prompts play a crucial role in large-scale generation, but datasets like Objaverse only contain paired metadata that do not serve as informative captions. To solve this problem, Cap3D [40] proposed to use LLM to consolidate captions generated from multiple views of a 3D object. We follow their pipeline to generate captions for all training examples. However, we found that these captions may be overly detailed and contain irrelevant objects, making it difficult to train a model from scratch. In addition, considering the limited availability of 3D data, we find that caption enrichment with different granularities is an effective and cost-efficient manner to ‘scale up’ the training set.

To generate more accurate captions with multiple granularities, we begin by rendering 8 images at 512^2 from different camera angles for each object. Next, a pretrained DeiT [58] on ImageNet-1K [19] is used to classify the object in each image and output object proposals based on the top-5 confidence scores. After that, we use BLIP2 [35] and LLaVA [38] for captioning through a two-stage question-answering process. In the first stage, they are tasked to identify the object in the image. Then we compare the identified object with the object proposals using the CLIP similarity, and eliminate irrelevant objects. In the second stage, for each image, the top-ranked matched answer is passed to the vision-language models for (1) assigning a title to this object, and providing descriptions of the object’s (2) color and texture, and (3) structure and geometry. 5 answers are generated for each question. Then we adopt the caption selection and consolidation from Cap3D [40] to get the final captions. We retain four captions per object, which correspond to (1) the object category, (2) the generated title, and the descriptions focusing on (3) texture and (4) geometry. Finally, we use the category and title information to further eliminate the irrelevant objects in descriptions (3) and (4). These captions are selected randomly during training.

4. Experiments

In this section, we first evaluate the performance of our method on single-class generation (Sec. 4.1) and large-scale text-to-3D generation (Sec. 4.2). Then, we show that our

Method	Car		Chair		Table	
	FID (\downarrow)	KID (\downarrow)	FID (\downarrow)	KID (\downarrow)	FID (\downarrow)	KID (\downarrow)
π -GAN [8]	36.7	-	52.71	13.64	41.67	13.82
EG3D [9]	10.46	4.90	16.54	8.41	31.18	11.67
DiffRF [45]	-	-	15.95	7.94	27.06	10.3
SSDNeRF [11]	11.08	3.47	-	-	14.27	4.08
Ours	6.90	1.84	7.01	2.12	7.26	1.89

Table 1. **Single-class generation on SRN Cars, PS Chairs, and ABO Tables.** Baseline results are reported by DiffRF and SSDNeRF. We train our model from scratch using exactly the same rendered images as the baselines. KID is multiplied by 10^3 .

method can function as a critical object-level 3D geometry prior, significantly improving previous optimization-based text-to-3D models (Sec. 4.3). Finally, we prove the effectiveness of our main ingredients in ablation (Sec. 4.4). Additional experimental results are provided in the Supp.

Datasets. We warm up our model on OmniObject3D [64] and a split of ShapeNet [10], which contain 6342 objects spanning 216 categories. Then we train our full model on Objaverse [18] that contains 800K+ objects.¹ For *single-class generation*, we strictly follow the previous methods [11, 20, 45] and conduct experiment on ShapeNet SRN Cars [10], Amazon Berkeley Objects (ABO) Tables [15], and PhotoShape (PS) Chairs [48]. For Chairs, we generate images following the render pipeline in DiffRF [45]. For Cars and Tables, we directly use the rendered images in SSDNeRF [11] for both training and testing.

4.1. Single-class 3D Generation

We compare against four leading methods: π -GAN [8], EG3D [9], DiffRF [45], SSDNeRF [11]. Following the latest SOTA method (SSDNeRF), we evaluate the generation quality using the Fréchet Inception Distance (FID) [25] and Kernel Inception Distance (KID) [5]. All metrics are evaluated at a resolution of 128^2 . Results are reported in Tab. 1. We reduce our model size to 135M parameters for a fair comparison with SSDNeRF (122M). We also remove the prompt condition and train a separate model on each category following the baselines. **Even when trained from scratch on the same data with a similar model size, our approach significantly outperforms all previous methods.** It underscores the high quality of our generated objects and the effectiveness of our method in modeling 3D data. Qualitative comparisons are provided in the Supp.

4.2. Direct Text-to-3D Generation

We compare our method with the current SOTA method Shap-E [31]. For a fair and comprehensive comparison, we evaluate both methods on 475 prompts, including all

¹We did not use Objaverse-XL [17] since the data were not public available when this project was conducted.

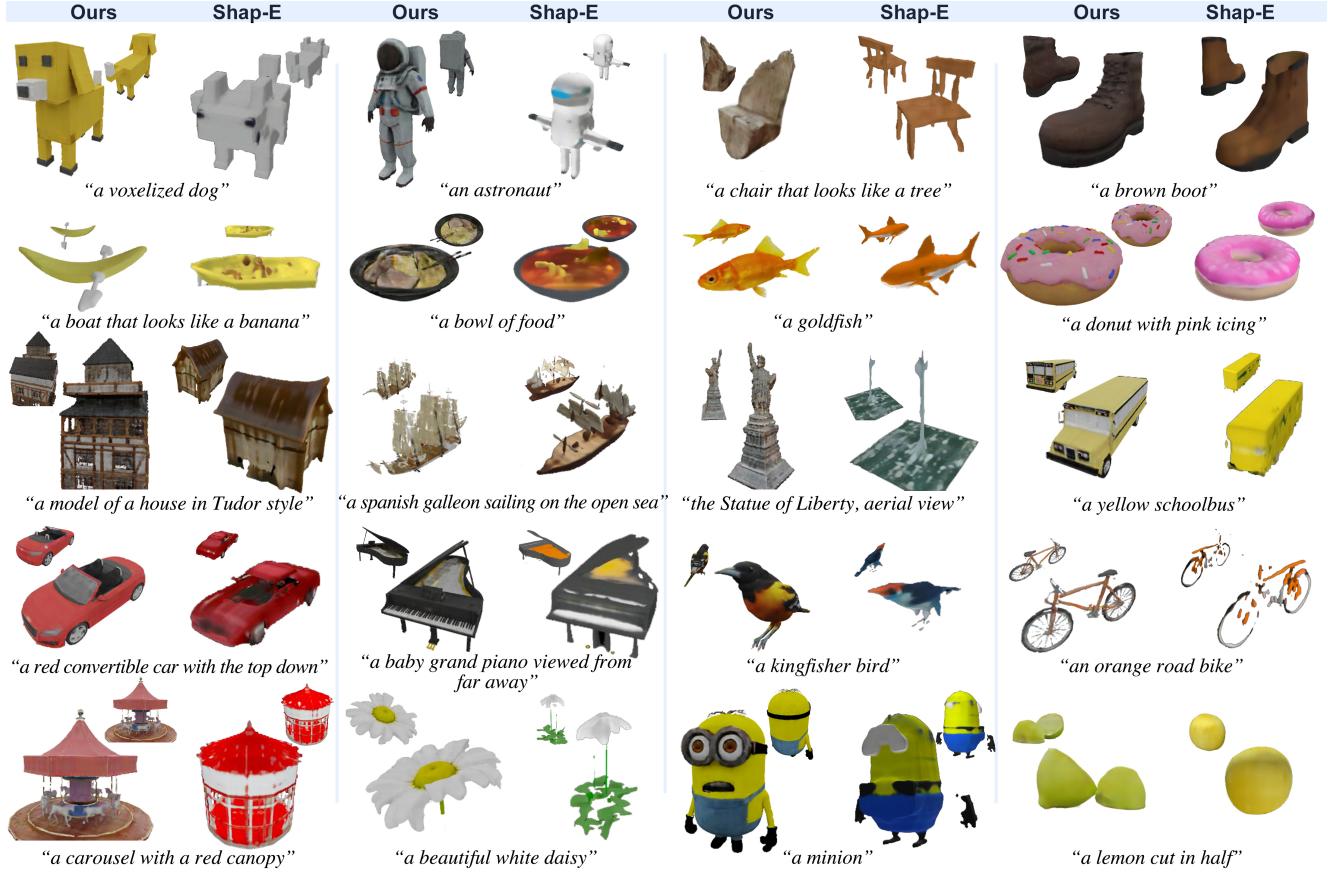


Figure 3. **Qualitative comparison with Shap-E [31]**. We use the same text prompt as in Shap-E (top 2 rows) and DreamFusion (middle 2 rows), we also compare the performance on complex objects (last row). For Shap-E, we use the official code and model. For our method, we generate objects in 128^3 without the super-resolution plug-in. All images of both methods are rendered at 256^2 . Our DIRECT-3D generates 3D objects with enhanced quality in both geometry and texture. We also generate more various and complex objects.

	More realistic	More detailed	Overall preference
Shap-E [31]	28.4%	22.9%	26.1%
Ours	71.6%	77.1%	73.9%

Table 2. **User preference studies.** We conduct user studies on 475 prompts, including all prompts from Shap-E and 162 prompts from DreamFusion. 73.9% of users prefer ours over Shape-E.

prompts in the official paper and website of Shap-E and 162 prompts from DreamFusion gallery.² Qualitative results are provided in Fig. 3. **Our model is able to generate more various and complex objects with much higher quality in both geometric details and textures.** More results can be found in the Supp.

Following Magic3D [37], we also conduct user studies to evaluate different methods based on user preferences on Amazon MTurk. For each generated object, we render a video recording its rotation along the z-axis, covering a full 360-degree view. Then we show users two side-by-side

²<https://dreamfusion3d.github.io/gallery.html>

videos generated by two algorithms, both using the same input prompt. We randomly switch the order of these two videos for different prompts. Users are instructed to evaluate which video is (1) more realistic, (2) more detailed, and (3) which one they prefer overall. Each prompt is evaluated by 3 different users, yielding a total of 1425 comparison results. As shown in Tab. 2, **we generate more realistic and detailed objects, leading to higher user preference.**

4.3. Improving 2D-lifting Methods with 3D Prior

Recent 2D-lifting text-to-3D methods [37, 49] have demonstrated impressive visual quality and compositionality using pretrained 2D text-to-image diffusion models as image prior. However, they suffer from the multi-face (Janus) problem. Here we show that plugging DIRECT-3D into the 2D-lifting framework as a 3D prior greatly alleviates the Janus problem and improves the geometry consistency.

We use an open-source implementation of DreamFusion [49] using StableDiffusion v2.1 [51] (DreamFusion-SD) or DeepFloyd [16] (DreamFusion-IF) as the 2D image

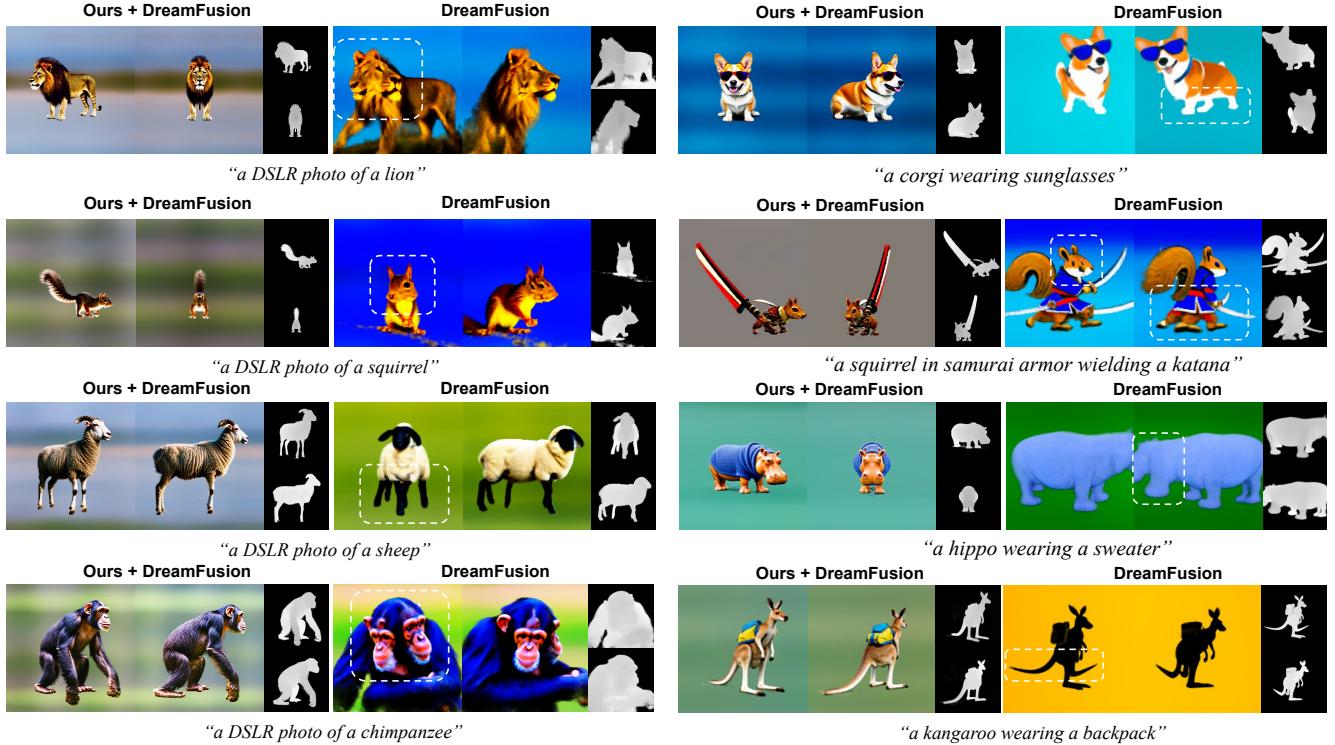


Figure 4. **DIRECT-3D provides a useful 3D prior for 2D-lifting methods [49].** Our 3D prior alleviates issues such as multiple faces and missing/extral limbs, while also improving texture quality. Please check the video results in Supp. for a better comparison.

	Succ. Rate	Geo. Consist.	Tex. Consist.
DreamFusion-SD [49]	12%	16%	30%
DreamFusion-IF [49]	10%	10%	72%
Ours + DreamFusion	84%	84%	98%

Table 3. **Improving 2D-lifting text-to-3D generation.** DIRECT-3D provides a useful 3D geometry prior, enhancing the geometry consistency and increasing the generation success rate.

prior. Our 3D prior is implemented as a Score Distillation Sampling (SDS) [49] loss added to the original text-to-3D loss. As the Janus problem only happens on radially asymmetric objects like animals, we concentrate our quantitative experiments on animals. We conducted 50 trials using the prompt ‘A DSLR photo of a [animal]’, with [animal] randomly sampled from a list of 14 animal types. The prompt for DIRECT-3D is set to ‘A [animal]’. Only generations with both correct geometry and texture are counted as success. The detailed criterion is described in the Supp. As shown in Tab. 3, **adding DIRECT-3D as 3D prior greatly improves the success rate of text-to-3D generation, alleviating the multi-face problem.**

We also show qualitative comparisons in Fig 4. Our method provides important geometry prior that greatly improves the generation success rate and the geometry consistency of the baseline method. In addition, we find that

	Car (R)		Chair (R)		Car + Chair + Table (R)	
	FID (\downarrow)	KID (\downarrow)	FID (\downarrow)	KID (\downarrow)	FID (\downarrow)	KID (\downarrow)
w/o AAC	46.77	34.35	45.57	27.17	39.06	27.24
w/ AAC	8.69	2.82	10.53	5.35	13.62	5.03

Table 4. **Automatic Alignment and Cleaning (AAC) improves performance on unaligned data.** To simulate unaligned data, all objects are rotated by a random degree, with a maximum of 360° along z-axis and $\pm 30^\circ$ along x/y axes (denoted as R). C+C+T means a same model is trained on all 3 datasets for multi-class generation, with ‘A 3D mesh of a [Class]’ as prompt condition.

with better geometry information, the texture consistency and quality are also improved.

4.4. Ablation Studies

4.4.1 Ablation of Automatic Alignment and Cleaning

We show the effectiveness of the Automatic Alignment and Cleaning (AAC) in Tab. 4, Fig. 5, and Fig. 5. For *quantitative evaluation*, we randomly rotated the aligned objects in SRN Cars, ABO Tables, and PS Chairs, and evaluate the models on their test set. Results are provided in Tab. 4. For *visualization*, we select cars and chairs from the Objaverse dataset based on their assigned category title, and directly train our model on them. We visualize the learned tri-plane

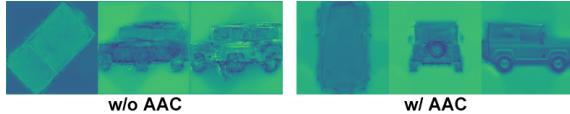


Figure 5. **Tri-plane feature learned with/without Automatic Alignment and Cleaning (AAC) on Objaverse.** It roughly aligns the objects to get clear tri-plane features. Unaligned objects can be captured by tri-plane representation, but the inadequate axis disentanglement makes it challenging for the diffusion model to learn.



Figure 6. **Model learned with/without AAC on Objaverse.** AAC enables direct and more efficient training on noisy, unaligned data.

	Car		Table		Car + Chair + Table	
	FID (↓)	KID (↓)	FID (↓)	KID (↓)	FID (↓)	KID (↓)
Not Disentangled	9.98	2.96	12.86	3.87	17.74	8.15
Disentangled	6.90	1.84	7.26	1.89	10.06	3.44

Table 5. **Improvement of Disentanglement.**



Figure 7. **Disentangling geometry and color provides a proper 3D geometrical prior, while improving the high-fidelity texture from 2D image diffusion models.**

and the generated NeRFs in Fig. 5 and Fig 6. AAC learns reasonable alignments of 3D objects while effectively filtering out toxic data. It enables direct and more efficient training on noisy and unaligned ‘in-the-wild’ data.

4.4.2 Ablation of Disentanglement

Tab. 5 highlights the enhancements achieved through disentanglement. For models without disentanglement, we double the number of layers to maintain similar model parameters. Disentanglement greatly improves model capabilities, establishing the foundation for large-scale generation.

More importantly, it provides pure geometry priors for various tasks. Considering 2D-lifting text-to-3D generation, Fig. 7 shows that when geometry and color are *not* disentangled, using our model as a geometry prior also affects the texture (*i.e.*, harms the image feature prior learned from 2D diffusion models). However, with disentanglement, we are able to provide critical geometry priors while preserving the high-fidelity texture from 2D image diffusion models. In addition, with better geometry consistency, the textures

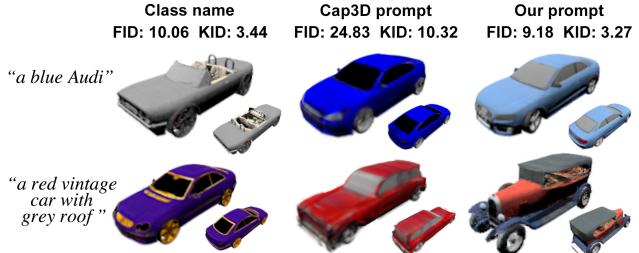


Figure 8. **Prompt Enrichment.** FID and KID are computed on the entire test set. We provide captions with varying granularities: Coarse captions enhance object-category connections, simplifying the training, while fine-gained captions enable a better understanding of detailed features such as color and part-level information.

learned from 2D diffusion models are also improved.

4.4.3 Ablation of Prompt Enrichment

Fig. 8 compares the performance variance when the model is trained with different prompts. ‘Class name’ means caption with template ‘A 3D mesh of a [Class]’.

Class name gives a better performance on FID and KID scores (reported in the figure). It simplifies the problem into a class-conditional multi-class generation task, ensuring higher quality in the generated object. However, training only with class names leads to a lack of basic understanding regarding detailed attributes.

Cap3D prompt contains finer details, yet can be overly intricate and occasionally contains irrelevant objects or even incorrect captions due to the failure of BLIP2 on synthetic objects. Directly training on them is more challenging, resulting in reduced quality and lower FID/KID scores.

Our prompt enrichment provides 4 different prompts for each object under different granularities. It ensures high-quality generation while offering better control over details.

5. Conclusion

We have presented DIRECT-3D, a diffusion-based text-to-3D generation model that is directly trained on extensive noisy and unaligned ‘in-the-wild’ 3D assets. Given text prompts, DIRECT-3D can generate high-quality 3D objects with precise geometric details in seconds. It also provides important and easy-to-use 3D geometry priors, complementing 2D priors provided by image diffusion models.

Acknowledgement

This work was done in part during an internship at ByteDance. AY acknowledges support from the ONR N00014-21-1-2812 and Army Research Laboratory award W911NF2320008. AK acknowledges support via his Emmy Noether Research Group funded by the German Science Foundation (DFG) under Grant No. 468670075.

DIRECT-3D: Learning Direct Text-to-3D Generation on Massive Noisy 3D Data

Supplementary Material

In this supplementary document, we provide details and extended experimental results omitted from the main paper for brevity. Specifically, Sec. 6.1 provides details of the NeRF Auto-decoder. Sec. 6.2 provides details of the 3D Super-Resolution module. Then, we cover the training details in Sec. 6.3, including loss functions and warm-up training on clean data. Sec. 7 presents experiment details and hyperparameters. Sec. 8 gives additional ablation studies and more qualitative results. Finally, the limitations of our method are discussed in Sec. 9.

In addition, we provide [video results for all visualizations in the supplementary file](#).

6. Model Details

6.1. NeRF Auto-decoder

We employ a NeRF Auto-decoder to extract features from the generated tri-planes and get NeRF parameters. This auto-decoder consists of several multi-layer perceptrons to process the tri-plane features \mathbf{f}_g and \mathbf{f}_c separately. Fig. 9 illustrates its architecture, which contains several fully connected layers with non-linear activation functions. The decoding process involves two distinct branches to handle the tri-plane features separately, ensuring that \mathbf{f}_g encapsulates only the geometry information and \mathbf{f}_c contains only the corresponding color features.

6.2. 3D Super Resolution

Similar to the structure of the base tri-plane diffusion model, the 3D super-resolution (SR) module also employs a U-Net model as its backbone. However, we apply only one up-sampling layer that directly scales the tri-plane feature from 128^2 to 512^2 . To enable efficient training with a larger batch size, we train the SR module separately. Therefore, we can directly use the saved tri-plane features during the training of the base model to train the SR module. Following cascaded image generation [28], we add Gaussian blurring and Gaussian noises to the intermediate tri-plane feature $\mathbf{f}'_{(\cdot)}$.

For training, alongside the L2 loss $\mathcal{L}_{geo}(\psi^{SR})$ and $\mathcal{L}_{col}(\phi^{SR})$ on tri-plane, we apply an entropy loss

$$\mathcal{L}_{entropy} = \rho \cdot \log_2(\rho) - (1 - \rho) \cdot \log_2(1 - \rho)$$

to the generated NeRF to encourage full transparent or opaque points, ensuring a smoother SR generation. Here ρ denotes the cumulative sum of density weights computed when computing NeRF parameters from tri-plane features.

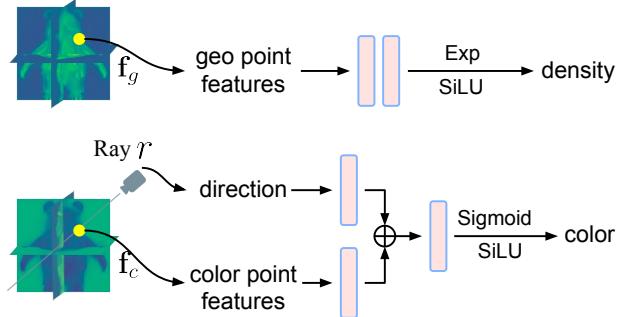


Figure 9. Architecture of the NeRF auto-decoder.

6.3. Training and Implementation Details

Loss function. To enable larger batch size and expedite training, we first exclude the 3D super-resolution (SR) module and train the base model end-to-end at 128^2 , by minimizing the following objective:

$$\begin{aligned} \mathcal{L}_{base} = & \lambda_{geo} \mathcal{L}_{geo}(\phi, \theta) + \lambda_{col} \mathcal{L}_{col}(\psi) \\ & + \lambda_{rad} \mathcal{L}_{rad}(\mathbf{f}_g, \mathbf{f}_c, \omega) \end{aligned}$$

To speed up the convergence of tri-planes learned from multi-view images (*i.e.*, $\mathcal{L}_{rad}(\mathbf{f}_g, \mathbf{f}_c, \omega)$), we adopt prior gradient caching [11] and save the diffusion gradients $\nabla_{\mathbf{f}_g} \mathcal{L}_{geo}$ and $\nabla_{\mathbf{f}_c} \mathcal{L}_{col}$ for re-using to update the tri-plane. It enables us to update $\mathcal{L}_{rad}(\mathbf{f}_g, \mathbf{f}_c, \omega)$ multiple times in one training iteration.

Then we freeze the base tri-plane diffusion module and only train the SR module to get high-resolution generations at 512^2 , with the following objective:

$$\begin{aligned} \mathcal{L}_{SR} = & \lambda_{geo} \mathcal{L}_{geo}(\phi^{SR}) + \lambda_{col} \mathcal{L}_{col}(\psi^{SR}) \\ & + \lambda_{rad} \mathcal{L}_{rad}(\mathbf{f}_g, \mathbf{f}_c, \omega) + \lambda_{entropy} \mathcal{L}_{entropy} \end{aligned}$$

In this step, we load, resize, and fine-tune the tri-plane features saved during the training of the base diffusion module. We use bilinear interpolation to scale the saved tri-planes from 128^2 to 512^2 .

Warm-up training. Training the entire system is challenging due to the intricate interdependencies between different modules. Specifically, optimizing diffusion model is less effective when tri-plane $\mathbf{f}_{(\cdot)}$ in Eqn. 1 is far from convergence, but learning \mathbf{f}_g with rotation θ needs a reasonably functioning diffusion model. Therefore, we warm up the model on clean and well-aligned data for the first 1/50 of the total iterations. It also defines a universal canonical pose for all objects. After that, we continue the training on all datasets with a learnable rotation parameter θ using the algorithm described in Sec. 3.2.

7. Experiment Details

7.1. Direct Text-to-3D Generation

We warm up our model on OmniObject3D [64] and a split of ShapeNet [10], which contain 6342 objects spanning 216 categories. Then we train our full model on Objaverse [18] that contains 800K+ objects.

Hyperparameters. We first train our base model for 2M iterations with a batch size of 256. Then the SR module is trained for 500K iterations with a batch size of 32. Both module are trained on 32 A100 GPUs. We set the number of channels for tri-plane features $C = 6$, and train a diffusion model with 1000 diffusion steps with linear noise schedule to generate the tri-plane features. During inference we sample 50 diffusion steps. The latent base learning rate is $1e^{-2}$ for all experiments. The learning rates for both geometry and color diffusion models are set to $1e^{-4}$, and the learning rate for NeRF auto-decoder is set to $1e^{-3}$. $\lambda_{geo} = \lambda_{col} = 5$, $\lambda_{rad} = 20$, and $\lambda_{entropy} = 0.1$. We update the tri-plane reconstructions from multi-view images 16 times per iteration for the initial 200K training iterations, and once per iteration for the subsequent training iterations. The latent base learning rate is reduced by a factor of 0.5 after 500K iterations and by a factor of 0.1 after 1M iterations.

7.2. Single-class 3D Generation

We reduce our model size to 135M parameters for a fair comparison with SSDNeRF [11] (122M). We also remove the prompt condition and train a separate model on each category following the baselines.

Hyperparameters. All models are trained for 500K iterations on 8 A100 GPUs, utilizing a batch size of 64. No SR plug-in is trained during these experiments. For cars and tables, the latent base learning rate is set to $4e^{-2}$. In the case of chairs, the latent base learning rate is set to $5e^{-3}$. The remaining hyperparameters align with those specified in direct text-to-3D generation.

7.3. Improving DreamFusion with 3D Prior

In our experiments, we sample [animal] from 14 animal types: *bear, corgi, dog, bird, cat, pig, elephant, horse, sheep, zebra, squirrel, chimpanzee, tiger, lion*.

Criterion for successful generation. We consider a text-to-3D generation successful when both the generated geometry and texture are consistent. Consistent geometry implies the correct number of parts is generated without missing or extra ones. Consistent texture implies the generated texture contains a consistent and plausible pattern that may appear on an actual animal of that type, regardless of the geometry.

Hyperparameters. For DreamFusion and DIRECT-3D, we run 10K iterations of optimization using the Adam optimizer [65] with a learning rate of 5×10^{-3} . Perp-Neg [3] is enabled for the 2D diffusion guidance with $w_{\text{neg}} = -4$



Figure 10. **Comparison of generated objects with and without the 3D super-resolution plug-in.** Please zoom in for better visualization.

for all methods, which we found useful to reduce incorrect textures such as multiple head textures. We set the weight of the 3D prior SDS loss provided by DIRECT-3D to 0.01. The classifier-free guidance is set to 100 as suggested in DreamFusion [49]. We use a coarse-to-fine training process for all methods, starting from a spatial resolution of 64^2 for the first 5K iterations and increasing to 128^2 afterward. The remaining hyperparameters are set to the default values.

8. Additional Experiments

8.1. Ablation on the Super-resolution Module

We employ an additional 3D super-resolution plug-in to enhance the resolution from 128^3 to 256^3 . Fig. 10 compares the generated objects with and without the SR plug-in, demonstrating its effectiveness in producing high-resolution objects with reduced computational resources. However, it's worth noting that the SR plug-in may slightly alter the generated low-resolution objects and introduce additional noise.

8.2. Ablation on 3D Prior Loss Weight

We also study the impact of different 3D prior loss weights. Ablation in Fig. 11 shows that utilizing only DIRECT-3D as initialization can alleviate the Janus problem, but also results in many artifacts, while large weights could compromise the quality of the generated geometry (*e.g.*, missing rear feet in this case).

8.3. Additional Qualitative Examples

We provide additional qualitative comparisons here. Specifically, Fig. 12 provides qualitative comparison with EG3D [9] and SSDNeRF [11] on single-class 3D generation. Fig. 13 provides additional comparisons with Shap-E [31] on direct text-to-3D generation, using the same text prompts as in Shap-E. Fig. 14 provides additional qualita-



Figure 11. Ablation of 3D prior loss weight.

tive results on using DIRECT-3D as a 3D prior to improve 2D-lifting text-to-3D methods such as DreamFusion [49].

9. Limitations

While DIRECT-3D consistently produces high-quality results and surpasses previous methods in single-class 3D generation and direct text-to-3D synthesis, it does exhibit certain limitations. First of all, despite the abundant geometry information provided by large-scale 3D datasets, a significant proportion of them lacks realistic textures. Additionally, the synthetic-to-real gap still persists, even for objects with nice and detailed textures. Therefore, training a 3D generative model, such as DIRECT-3D, solely on these extensive 3D datasets may result in a lack of appearance information for specific objects. One potential solution is to further fine-tune our color diffusion model on real images, which we leave for future exploration.

Secondly, the current model demonstrates limitations in compositionality. Although DIRECT-3D can generate multiple objects with close relations, such as “a house with a garden”, it struggles to generate novel combinations like “an astronaut riding a horse”. This issue is also observed in previous methods such as Shap-E [31]. We attribute this limitation to two main factors: (1) The scarcity of multiple objects in a single CAD model contributes to the difficulty of generating diverse objects within one tri-plane. Unlike 2D images, where multiple objects are commonly present, most 3D CAD models consist of either a single object or two or three highly related objects. (2) Current 3D datasets are still orders of magnitude smaller than their 2D counterparts, resulting in insufficient training data to effectively learn novel compositionality.

References

- [1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. [3](#)
- [2] Titas Auciukevičius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J Mitra, and Paul Guerrero. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12608–12618, 2023. [3](#)
- [3] 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. [2](#)
- [4] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. [2](#)
- [5] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International Conference on Learning Representations*, 2018. [5](#)
- [6] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*, 2016. [3](#)
- [7] Ziang Cao, Fangzhou Hong, Tong Wu, Liang Pan, and Ziwei Liu. Large-vocabulary 3d diffusion model with transformer. *arXiv preprint arXiv:2309.07920*, 2023. [3](#)
- [8] 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](#)
- [9] 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. [2, 5](#)
- [10] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. [2, 3, 5](#)
- [11] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In *International Conference on Computer Vision*, 2023. [2, 3, 5, 1, 4](#)
- [12] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *arXiv preprint arXiv:2303.13873*, 2023. [3](#)
- [13] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. [3](#)



(a) EG3D



(b) SSDNeRF



(c) Ours

Figure 12. **Qualitative comparison on ShapeNet SRN Cars.** Baseline results come from the original paper of SSDNeRF [11]. Following the baseline methods, we generate and render images at 128^2 .



Figure 13. **Qualitative comparison with Shap-E [31].** All text prompts are sourced from the original paper of Shap-E. For Shap-E, we use the official code and model with the default random seed. For our method, we generate objects in 128^2 without the super-resolution plug-in. All images of both methods are rendered at 256^2 . Our DIRECT-3D generates 3D objects with enhanced quality in both geometry and texture.



Figure 14. **More qualitative results on using DIRECT-3D as a 3D prior for 2D-lifting methods.** Our 3D prior alleviates issues such as multiple faces and missing/extral limbs, while also improving texture quality. Please also check the video demos for a better visualization.

- [14] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4456–4465, 2023. [3](#)
- [15] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21126–21136, 2022. [5](#)
- [16] DeepFloyd-Team. Deepfloyd-if, 2023. [6](#)
- [17] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Chris-tian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023. [2, 4, 5](#)
- [18] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. [2, 4, 5](#)
- [19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. [5](#)
- [20] Emilien Dupont, Hyunjik Kim, SM Ali Eslami, Danilo Jimenez Rezende, and Dan Rosenbaum. From

- data to functa: Your data point is a function and you can treat it like one. In *International Conference on Machine Learning*, pages 5694–5725. PMLR, 2022. 5
- [21] Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. *arXiv preprint arXiv:2303.17015*, 2023. 3
- [22] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. In *2017 International Conference on 3D Vision*, pages 402–411. IEEE, 2017. 3
- [23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 3
- [24] Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oğuz. 3dgen: Triplane latent diffusion for textured mesh generation. *arXiv preprint arXiv:2303.05371*, 2023. 3
- [25] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5
- [26] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4
- [27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 3, 4
- [28] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022. 1
- [29] Yukun Huang, Jianan Wang, Yukai Shi, Xianbiao Qi, Zheng-Jun Zha, and Lei Zhang. Dreamtime: An improved optimization strategy for text-to-3d content creation. *arXiv preprint arXiv:2306.12422*, 2023. 3
- [30] Ka-Hei Hui, Ruihui Li, Jingyu Hu, and Chi-Wing Fu. Neural wavelet-domain diffusion for 3d shape generation. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 3
- [31] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 2, 3, 5, 6
- [32] Animesh Karnewar, Niloy J Mitra, Andrea Vedaldi, and David Novotny. Holofusion: Towards photo-realistic 3d generative modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22976–22985, 2023. 3
- [33] Animesh Karnewar, Andrea Vedaldi, David Novotny, and Niloy J Mitra. Holodiffusion: Training a 3d diffusion model using 2d images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18423–18433, 2023. 3
- [34] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3
- [35] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 5
- [36] Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12642–12651, 2023. 3
- [37] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 2, 3, 6
- [38] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 5
- [39] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021. 3
- [40] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. *arXiv preprint arXiv:2306.07279*, 2023. 5
- [41] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. 4
- [42] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12663–12673, 2023. 3
- [43] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2
- [44] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy J Mitra, and Leonidas J Guibas. Structurenet: hierarchical graph networks for 3d shape generation. *ACM Transactions on Graphics (TOG)*, 38(6):1–19, 2019. 3
- [45] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulo, Peter Kortschieder, and Matthias Nießner. Diffrr: Rendering-guided 3d radiance field diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4328–4338, 2023. 2, 3, 5
- [46] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 2, 3
- [47] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 3
- [48] Keunhong Park, Konstantinos Rematas, Ali Farhadi, and Steven M Seitz. Photoshape: Photorealistic materials for large-scale shape collections. *arXiv preprint arXiv:1809.09761*, 2018. 5
- [49] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The*

- [50] 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. 4
- [51] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 6
- [52] 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. 2, 4
- [53] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 2
- [54] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 3
- [55] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20875–20886, 2023. 2, 3
- [56] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020. 3
- [57] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2, 3
- [58] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 5
- [59] Christina Tsalicoglou, Fabian Manhardt, Alessio Tonioni, Michael Niemeyer, and Federico Tombari. Textmesh: Generation of realistic 3d meshes from text prompts. *arXiv preprint arXiv:2304.12439*, 2023. 3
- [60] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12619–12629, 2023. 3
- [61] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4563–4573, 2023. 3, 4
- [62] 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*, 2023. 3
- [63] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016. 3
- [64] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 803–814, 2023. 3, 5, 2
- [65] Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, and Shuicheng Yan. Adam: Adaptive nesterov momentum algorithm for faster optimizing deep models. *arXiv preprint arXiv:2208.06677*, 2022. 2
- [66] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4541–4550, 2019. 3
- [67] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. *arXiv preprint arXiv:2210.06978*, 2022. 3
- [68] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *arXiv preprint arXiv:2301.11445*, 2023.
- [69] Xin-Yang Zheng, Hao Pan, Peng-Shuai Wang, Xin Tong, Yang Liu, and Heung-Yeung Shum. Locally attentional sdf diffusion for controllable 3d shape generation. *arXiv preprint arXiv:2305.04461*, 2023.
- [70] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5826–5835, 2021. 3