PI3D: Efficient Text-to-3D Generation with Pseudo-Image Diffusion

Ying-Tian Liu[1]   Yuan-Chen Guo[1,2*]   Guan Luo[1]   Heyi Sun[1]   Wei Yin[4]   Song-Hai Zhang[3,1†]

[1] BNRist, Tsinghua University [2] VAST [3] Qinghai University [4] The University of Adelaide
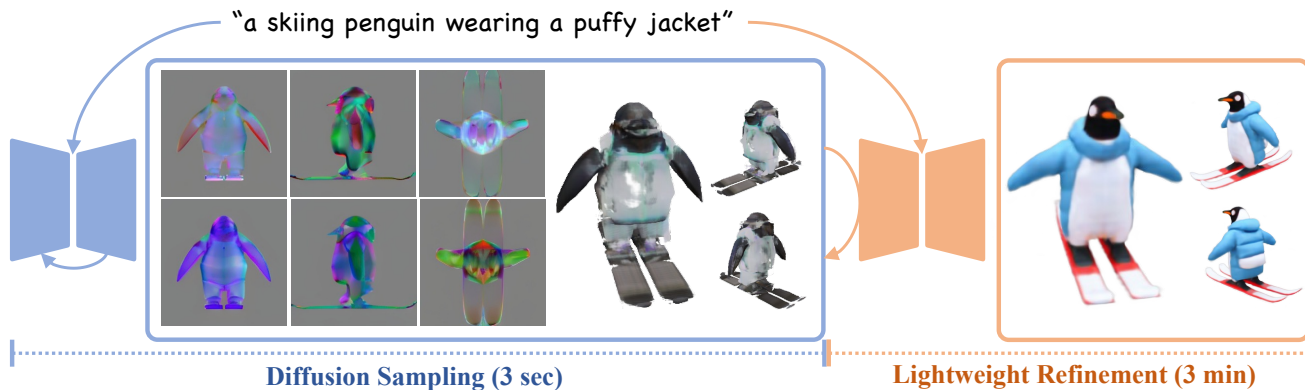

Figure 1. Our method, PI3D, is able to generate the pseudo-images of a 3D shape in seconds from text prompts and further refine it in 3 minutes to achieve better quality.

Abstract

*Diffusion models trained on large-scale text-image datasets have demonstrated a strong capability of controllable high-quality image generation from arbitrary text prompts. However, the generation quality and generalization ability of 3D diffusion models is hindered by the scarcity of high-quality and large-scale 3D datasets. In this paper, we present PI3D, a framework that fully leverages the pre-trained text-to-image diffusion models' ability to generate high-quality 3D shapes from text prompts in minutes. The core idea is to connect the 2D and 3D domains by representing a 3D shape as a set of **P**seudo RGB **I**mages. We fine-tune an existing text-to-image diffusion model to produce such pseudo-images using a small number of text-3D pairs. Surprisingly, we find that it can already generate meaningful and consistent 3D shapes given complex text descriptions. We further take the generated shapes as the starting point for a lightweight iterative refinement using score distillation sampling to achieve high-quality generation under a low budget. PI3D generates a single 3D shape from text in only 3 minutes and the quality is validated to outperform existing 3D generative models by a large margin.*

1. Introduction

With recent advances in generative modeling, various models have emerged capable of capturing complex data distributions across different modalities such as audio, video, and images. Notably, pioneering image generation models like Imagen [40], DALL-E [36, 37], Stable Diffusion [39], and others demonstrate the ability to generate highly realistic and natural-looking images from text prompts, often indistinguishable from those produced by humans.

While algorithmic progress plays a crucial role in these achievements, it's essential to acknowledge the foundational role of large-scale datasets [42] in facilitating such applications. Unlike 2D images, 3D objects face inherent challenges in data acquisition. The labor-intensive and time-consuming processes involved in 3D scanning and creation make gathering 3D data notably challenging. Although datasets like Objaverse [9] and Objaverse-XL [8] have recently been released with millions of 3D assets, they suffer from insufficient filtering and processing, leading to the inclusion of duplicate, low-quality, and texture-less content. The absence of a high-quality, abundant, and semantically rich 3D dataset remains a significant obstacle in directly constructing 3D generative models.

In order to generate 3D assets in scenarios where 3D priors are difficult to obtain, researchers have recently been keen to utilize various 3D representations [24, 31, 43, 49] to lift 2D diffusion models with Score Distillation Sampling (SDS) [34, 48]. However, applying only 2D prior is prone to
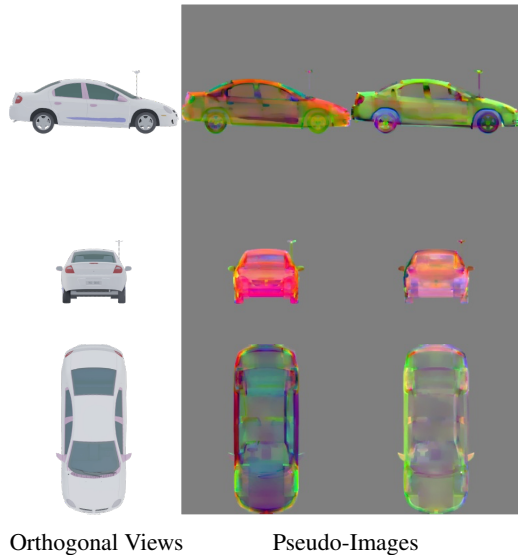
---

*Work done during internship at VAST.
†Corresponding author.

Orthogonal Views   Pseudo-Images

Figure 2. We show the 3 orthogonal rendering results and the tri-plane representation as 6 pseudo-images for the same scene. We can observe semantic congruence between them, such as the contours of different parts.

hinder the 3D nature, inevitably compromising 3D fidelity and consistency. 3D knowledge is a key issue that cannot be bypassed for high-quality 3D generation.

The pivotal question arises: can a 3D diffusion model exhibit robust generalization when 3D data cannot reach the scale and quality of image data? We answer this question by presenting PI3D, a framework that fully utilizes the knowledge in pre-trained text-to-image diffusion models and image datasets to generate high-quality 3D shapes conditioned on text prompts. Given the observation that the triplane representation for 3D scenes [4, 33] shares semantic congruence with orthogonal rendered images, we believe that the architecture of a pre-trained image diffusion model is also capable of generating high-quality triplanes. In order to borrow knowledge from 2D generation, we treat every 3 channels of the triplane representation as one image to form a set of pseudo RGB images (see Fig. 2). To build our 3D diffusion model, we start from a pre-trained text-to-image diffusion model and keep most of its architecture to inherit the generalizability while changing the self-attention layer to global attention performed on all pseudo-images. We inject 3D knowledge into the pre-trained 2D diffusion model by fine-tuning it on the pseudo-images obtained from 20k text-3D pairs from the Objaverse dataset [9]. Further, we find that the simultaneous use of 2D image data in training significantly improves generalization ability on complex concepts while keeping the requirements of the magnitude of 3D data at a low level. Based on the pseudo-image diffusion model, we can create brand-new 3D objects from text prompts by performing diffusion sampling in seconds. We take the sampled object as the starting point for a subsequent refinement process where score distillation sampling

(SDS) [34] is used to further improve the quality. The refinement process converges quickly and maintains good 3D consistency thanks to the strong initialization provided by the pseudo-image diffusion model. PI3D is able to generate high-quality 3D objects from text prompts within 3 minutes, bringing new possibilities for 3D content creation.

The main contributions of this paper are as follows:
- We propose PI3D, a framework that fully leverages prior knowledge from 2D diffusion models for high-quality and text-aligned 3D generation in minutes.
- We propose to represent a 3D shape as a set of pseudo-images and adapt a 2D diffusion model to directly output such a representation, enabling fast sampling of 3D objects from texts while providing strong initialization for further refinement.
- PI3D is demonstrated to be effective in 3D generation with respect to visual quality, 3D consistency, and generation speed.

2. Related Work

In this section, we will primarily review recent text-to-3D research. They can be broadly categorized based on the data they utilize and how they incorporate 3D knowledge: methods using 3D diffusion methods and lifting 2D diffusion models.

2.1. Diffusion-based 3D Generative Model

This line of research focuses on training 3D diffusion models on popular 3D datasets [5, 8, 9], each with distinct representations of 3D objects. As most of these methods have access to 3D raw data, they inherently acquire 3D knowledge inductively from the dataset. Point·E [32] trains two point-cloud diffusion models conditioned on the text prompt or single-view image to generate a low-resolution point cloud and up-sample it respectively. It can further obtain meshes from the point cloud via SDF prediction. Shap·E [20] trains the diffusion model on the 3D implicit function. It is capable of simultaneously producing NeRF and SDF with two heads, but it struggles with semantics when faced with complex prompts. NFD [45], 3DGen [11], and DiffTF [3] adopt diffusion on the triplane representation to achieve highly efficient mesh generation but they constrain the generation to be category-specific on ShapeNet [5] and OmniObject3D [52]. Some other methods [22, 23] build 3D diffusion models with only posed 2D images on smaller datasets [38]. All these methods actually utilize only poorly labeled 3D data or limited image data, which leads to their poor ability to generate reasonable objects when confronted with some complex concepts.

2.2. 3D Knowledge in Lifting 2D Diffusion Models

Several recent works focus on optimizing a 3D representation by leveraging a pre-trained 2D diffusion model. The

foundational technique involved is Score Distillation Sampling (SDS) [34, 48]. SDS aims to align the rendering results of a parameterized renderer with the distribution described by a pre-trained image diffusion model, achieving parameter space sampling. In DreamFusion [34] and SJC [48], 3D knowledge originates solely from spatial representations and the view-dependency prompting, which are relatively weak. This limitation increases the likelihood of 3D inconsistency, such as the Janus problem, since there is no direct informative constraint between different views. ProlificDreamer [50] and HiFA [53] reduce the oversaturation and greatly increase the generation quality by reinventing the optimization target. Perp-Neg [2] mitigates the inconsistency problem by correcting the direction of parameter updates guided by the 2D diffusion model but leaves the lack of a 3D prior unsolved. Zero-1-to-3 [27] proposes to train a 2D viewpoint transformation model conditioned on the input view and the relative camera transformation, injecting implicit 3D knowledge into the 2D diffusion model. Magic123 [35] fuses and balances guidance from pre-trained 2D diffusion model [39] and viewpoint transformation model [27], achieving detailed image-to-3D results. MVDream [44] models the distribution of multi-view images and thus greatly improves the stability of existing 2D-lifting methods. ATT3D [29] amortizes the optimization cost among prompts by introducing a prompt-conditioned parameter prediction hyper-network. DreamGaussian [46] accelerates the generation by utilizing real-time rendered 3D representation. Although these methods are usually capable of generating an attractive appearance, they all require tons of optimizing iterations and are of lower diversity.

Some other methods view 3D generation as multi-view generation combined with reconstruction [25, 26, 28, 47]. The reconstruction process can be optionally replaced by a feed-forward model [18, 26]. These approaches allow for fast 3D generation but are often jointly affected by minor inconsistencies of multi-view images and errors in the reconstruction process.
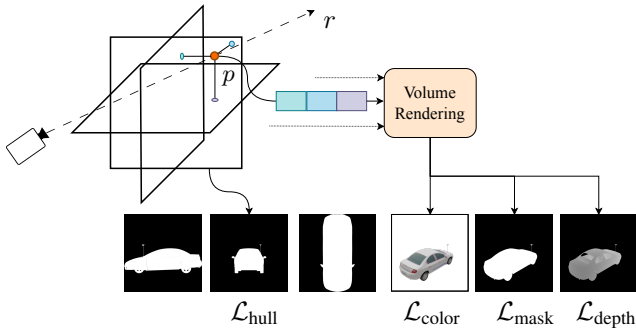

Figure 3. We fit the triplane representation for each object with the supervision of the rendered RGB images, binary mask, and object depth. We also adopt axis-aligned masks on the triplane to encourage the geometry to approach the correct surface faster.

3. Method

In this section, we mainly focus on how PI3D is built. It consists of two parts as a whole, the first part is a pseudo-image diffusion model that is responsible for fast sampling of 3D objects. It follows two stages similar to many other latent diffusion methods [7, 11, 19, 20]. In the first stage (in Sec. 3.1), a 3D representation for each 3D object is fitted and next (in Sec. 3.2), we build a text-conditioned pseudo-image diffusion model adapting from a pre-trained 2D diffusion model. The second part (in Sec. 3.3) is a lightweight refinement using Score Distillation Sampling (SDS) that converges quickly and further improves the quality of the sampled 3D object.

3.1. Depth-Aware Triplane Fitting

We represent each object as three axis-aligned feature maps $F_{xy}, F_{xz}, F_{yz} \in \mathbb{R}^{C \times H \times W}$. Specifically, we set $C = 6$ to form 6 pseudo-images. The spatial feature $F(p)$ for any query point p is constructed by concatenating the features bi-linearly interpolated on the three maps at the projected 2D positions.

$$F(p) = [F_{xy}(p_{xy}), F_{xz}(p_{xz}), F_{yz}(p_{yz})] \quad (1)$$

The color and density at the point p are then decoded by a shared MLP $\mathcal{D}(\cdot; \theta)$.

$$c(p), \sigma(p) = \mathcal{D}(F(p); \theta) \quad (2)$$

The expected color of a camera ray r is calculated in a volume rendering manner [21, 31].

$$\hat{C}(r) = \sum_{i=1}^{N} T_i(1 - \exp(-\delta\sigma(p_i)))c(p_i),$$

$$\text{where } T_i = \exp(-\sum_{j=1}^{i-1} \delta\sigma(p_j)) \quad (3)$$

$\{p_i\}_{i=1}^{N}$ denotes the sampled points along that ray. We do not involve additional positional embedding like in vanilla NeRF [31], since we want the decoder to be position-agnostic, ensuring that what is similar on the rendered images is also similar on the triplane. To encourage a correct underlying geometry, we also supervise the object mask $\hat{M}(r)$ and the expected depth $\hat{D}(r)$ along that ray, which are given by:

$$\hat{M}(r) = \sum_{i=1}^{N} T_i(1 - \exp(-\delta\sigma(p_i))) \quad (4)$$

$$\hat{D}(r) = \sum_{i=1}^{N} T_i(1 - \exp(-\delta\sigma(p_i)))t(p_i) \quad (5)$$

where $t(p)$ is the distance from the ray origin to p.

The losses on rendering views are as follows:

$$\mathcal{L}_{\text{color}} = \sum_{r}(||C(r) - \hat{C}(r)||_2 + \lambda_{\text{L1}}||C(r) - \hat{C}(r)||_1) \quad (6)$$

$$\mathcal{L}_{\text{mask}} = \sum_r \text{BCE}(\hat{M}(r), M(r)) \tag{7}$$

$$\mathcal{L}_{\text{depth}} = \sum_r (M(r) \otimes ||\hat{D}(r) - D(r)||_1) \tag{8}$$

where $\otimes$ is element-wise production. We use both MSE and L1 loss on the rendered color for better convergence.

In addition to encouraging the triplane to fit the object well, we still want it to be friendly to the later diffusion process. So we put several regularizations on the triplane itself. Inspired by [3], we use L2 and TV loss as follows:

$$\begin{aligned} \mathcal{L}_{\text{L2}} &= ||F||_2 \\ \mathcal{L}_{\text{TV}} &= \sum_{1 \le i < H} ||F(i, \cdot) - F(i+1, \cdot)||_1 \\ &+ \sum_{1 \le i < W} ||F(\cdot, i) - F(\cdot, i+1)||_1 \end{aligned} \tag{9}$$

In order for the geometry to converge fast, we render 3 axis-aligned views in orthogonal projection and adopt these masks O_{xy}, O_{yz}, O_{xz} on 3 planes to constrain where it has non-zero features, which is basically equivalent to a 3-view visual hull constraint.

$$\mathcal{L}_{\text{hull}} = ||(1 - O) \otimes F||_1 \tag{10}$$

Above we neglect the subscript xy, xz, yz as they are adopted on all planes.

In general, we adopt an autodecoder architecture to fit all the triplanes. However, a large number of triplanes cannot be put into the GPU memory at the same time. So we first train the shared MLP $\mathcal{D}(\cdot; \theta)$ on a subset and fit all the 3D assets independently by freezing $\mathcal{D}(\cdot; \theta)$. The final optimization objective for either training $\mathcal{D}$ or F is

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{\text{color}} + \lambda_{\text{mask}} \mathcal{L}_{\text{mask}} + \lambda_{\text{depth}} \mathcal{L}_{\text{depth}} \\ &+ \lambda_{\text{L2}} \mathcal{L}_{\text{L2}} + \lambda_{\text{TV}} \mathcal{L}_{\text{TV}} + \lambda_{\text{hull}} \mathcal{L}_{\text{hull}} \end{aligned} \tag{11}$$

3.2. Pseudo-Image Diffusion

We choose Stable Diffusion [39] as our base model and build a pseudo-image diffusion model on top of it without altering its architecture. The two main challenges here are: (1) how to simultaneously output multiple multi-channel feature maps under the architecture, and (2) how to model the cross-plane consistency to form a reasonable underlying 3D geometry. We view each triplane representation as a set of 6 pseudo-images and stack them along the batch dimension as the model input. A global pseudo-image attention block, inspired by video and multi-view diffusion models [16, 17, 44, 51], is adapted from the original self-attention block without changing the parameters to aggregate tokens on 6 pseudo-images. In this way, we enrich the feature interaction across three planes and incorporate the 3D knowledge into the global attention block. So far, all the attention layers are order-invariant. Thus, to identify the input images, we allocate a learnable embedding for each

pseudo-image and add it to the timestep embedding in the cross-attention block.

We directly trained the model on our fitted triplanes but observed a decline in generalization ability on 3D assets over prolonged training, particularly for prompts involving combined concepts and multiple objects, showing less robustness than image diffusion models. We believe the performance degradation is attributed to the quality and scale of 3D data. There are basically all single objects in the Objaverse dataset [9] and still exist many misaligned text descriptions in Cap3D [30]. To enhance robustness, we train the pseudo-image diffusion with additional real images from LAION-5B [42] by disabling global attention blocks and learnable embedding. We observe obvious improvements in the diversity and prompt alignment after training with mixed 2D-3D data which we will discuss later in the ablation study (see Sec. 4.3).

3.3. Lightweight Refinement

Upon obtaining the diffusion model, two intuitive ways to generate 3D models from input texts are considered. One can directly sample new 3D models within seconds from it and further refine it with 2D SDS. Alternatively, we can view the 3D diffusion model as a prior and jointly optimize a triplane-based NeRF with 2D and 3D SDS, like Magic123 [35] to guarantee the 3D consistency. In practice, we find that sampling a 3D model first using our method and then refining it with only 2D diffusion models is a simple yet effective way to produce high-quality objects and guarantee 3D consistency. In the refinement, we continue to optimize the triplane generated by the pseudo-image diffusion along with the shared decoder with SDS loss. To prevent the 3D shape from being over-optimized, we refine the appearance and geometry by sampling only small noise levels in SDS. A good coarse 3D model provided by the pseudo-image diffusion gives 2D SDS a strong appearance prior, making it converge much faster than starting from scratch [6, 34, 44, 50].

4. Experiments

4.1. Implementation Details

Data For the 3D dataset, we use 20k 3D objects from Objaverse [9] LVIS subset along with their corresponding captions from Cap3D [30]. In the triplane fitting stage, we have standardized the size of the 3D objects by scaling them to fit within a unit-length bounding box. Subsequently, we placed them at the origin and generated 100 random views for each object at a resolution of 512×512.

Triplane fitting We define the triplane size as $H = W = 256, C = 6$. The shared decoder used in volume rendering is a 4-layer MLP, with each hidden layer having a width
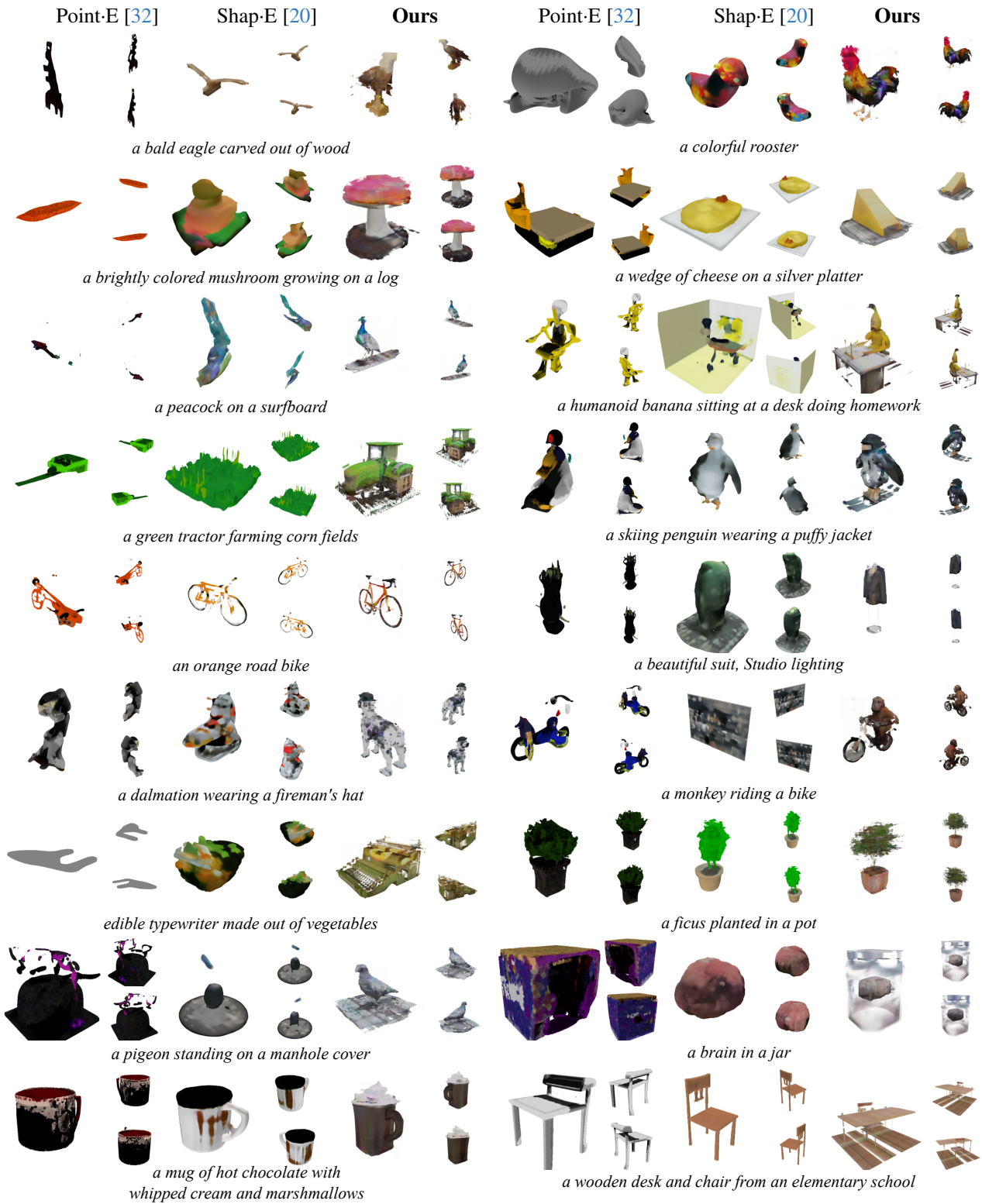
Point·E [32]　　Shap·E [20]　　**Ours**　　Point·E [32]　　Shap·E [20]　　**Ours**

a bald eagle carved out of wood

a colorful rooster

a brightly colored mushroom growing on a log

a wedge of cheese on a silver platter

a peacock on a surfboard

a humanoid banana sitting at a desk doing homework

a green tractor farming corn fields

a skiing penguin wearing a puffy jacket

an orange road bike

a beautiful suit, Studio lighting

a dalmation wearing a fireman's hat

a monkey riding a bike

edible typewriter made out of vegetables

a ficus planted in a pot

a pigeon standing on a manhole cover

a brain in a jar

a mug of hot chocolate with whipped cream and marshmallows

a wooden desk and chair from an elementary school

Figure 4. Examples of 3D models generated by Point·E [32], Shap·E [20] and our method conditioned on various text prompts. 3D models are presented by 3 rendered views. We follow Point·E [32] to extract the surface for the point cloud rendering. The prompts are listed below each sample.

Table 1. Quantitative comparisons between PI3D and state-of-the-art text-to-3D generative models on CLIP Score [13] and CLIP R-Precision metrics. The R-precision is calculated based on top 1 and top 10 recall.

Method	ViT-B-32			ViT-L-14		
	CLIP Score	R-Precision@1	R-Precision@10	CLIP Score	R-Precision@1	R-Precision@10
Point·E	22.3	4.6	15.7	42.6	5.7	26.3
Shap·E	62.0	16.0	41.0	48.3	17.3	43.7
PI3D	**65.9**	**25.2**	**56.7**	**53.8**	**29.3**	**63.3**

of 32. It is trained on a set of 500 objects. Subsequently, we freeze the shared decoder and independently fit the triplane representation for each object, implementing a linearly decayed learning rate. The balancing coefficients for loss items are as follows: $\lambda_{\text{L1}} = 0.2, \lambda_{\text{mask}} = 0.1, \lambda_{\text{depth}} = 0.5, \lambda_{\text{L2}} = 0.05, \lambda_{\text{TV}} = 0.05, \lambda_{\text{hull}} = 1.0$. The triplanes are scaled to $[-1, 1]$ to ensure a similar variance with images for easy diffusion training later. The process of fitting one object on a single A100 GPU takes 2 minutes.

Pseudo-image diffusion We start from *Stable Diffusion v2.1* and train the learnable embedding with a learning rate of 10^{-3} while the remaining parameters are trained with 10^{-4} for 15,000 iterations. Training is conducted using 256×256 real images from LAION-5B [42] and pseudo-images with 50% probability each. The equivalent batch size for triplanes and real images are set to 384 and 2304, respectively, to match their computational overhead. We drop the text embedding with a 5% probability, jointly training an unconditional diffusion model to enable classifier-free guidance [14]. Utilizing the v-objective [41] and Min-SNR strategy [12] enhances convergence during diffusion training. The entire diffusion training process spans about 36 hours, utilizing 8 A100 GPUs.

Text-to-3D generation In pseudo-image sampling, we use DDPM [15] scheduler for 50 sample steps with a classifier free guidance [14] scale of 5.0. It takes 3 seconds to sample a triplane with the 256×256 resolution. We implement the subsequent refinement based on threestudio [10], utilizing DeepFloyd-IF [1] as the 2D guidance model. The triplane-based NeRF is optimized for 2000 steps with a batch size of 1, typically requiring 3 minutes on an A100 GPU. The timestep t is sampled in $\mathcal{U}(0.1, 0.5)$ to introduce small corrections and prevent excessive changes. The classifier-free guidance adopted in SDS is 20.

4.2. Comparisons with Other Text-to-3D Methods

We compare PI3D with general text-to-3D diffusion models, Point·E [32] and Shap·E [20] and SDS-based method DreamFusion [34], Fantasia3D [6], ProlificDreamer [50], MVDream [44]. To evaluate the fidelity, we adhere to the metrics used in prior works [20, 34], namely CLIP R-

Precision and CLIP similarity score [13]. CLIP R-precision measures the accuracy of retrieving text prompts using rendered images based on CLIP similarity. We select 4 views at different camera poses with azimuth angles $(0, 90, 180, 270$ degrees) and the same elevation angle (15 degrees), performing multiple retrievals to average a score. All 415 testing prompts are from the DreamFusion gallery.

We first compare the pseudo-image diffusion model with the baseline text-to-3D diffusion models. The quantitative results are reported in Tab. 1. Our model significantly outperforms the other two methods in all the metrics, revealing that it could generate more text-aligned 3D models. We also demonstrate the generation quality through some qualitative cases (see Fig. 4), each presented with 3 rendered views. The baselines, Point·E [32] and Shap·E [20], can generate simple objects that are part of complex prompts but struggle to capture the semantics of intricate text prompts, for example, depicting "corn fields" without the tractor or a "penguin" without the skiing activity. We believe the limited generalization ability is partly because they are both trained only on 3D datasets with variable quality of text annotation and 3D geometry. However, leveraging the similarity between pseudo-images and 2D real images, our method can learn rich concepts from limited 3D data and achieve reasonable generation quality even with prompts involving concept combination.

We also compare the refinement stage with SDS-based methods on the generation efficiency and quality. Several examples by DreamFusion-IF from threestudio [10] and our method are compared in Fig. 5. The baseline method requires tons of iterations in hours while still performing poorly in 3D consistency and convergence. Our method can generate 3D consistent models and converge much faster with an even smaller batch size in only 3 minutes by leveraging a good 3D initialization sampled from the diffusion model. Also, our method is compatible with other SDS variants. As demonstrated in Fig. 6, we compare our method with other SDS-based methods by switching the 2D guidance to the corresponding variants. We can achieve superior or comparable quality with even much faster convergence. In terms of efficient generation, ATT3D [29] has an inference runtime of 1s, much faster compared to optimization-based methods. But we contend that it is much less generalizable than our method. Limited by a prompt set, it can only
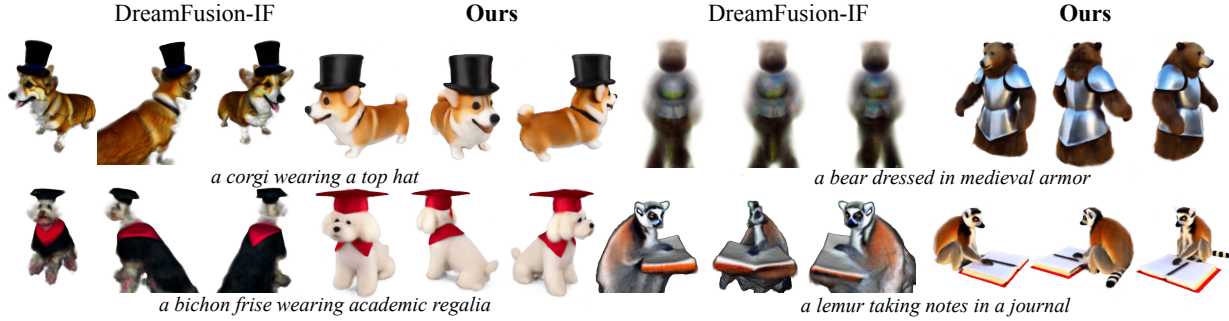
Figure 5. Examples generated by Dreamfusion-IF [10, 34] and our method. PI3D can generate 3D consistent models much faster.
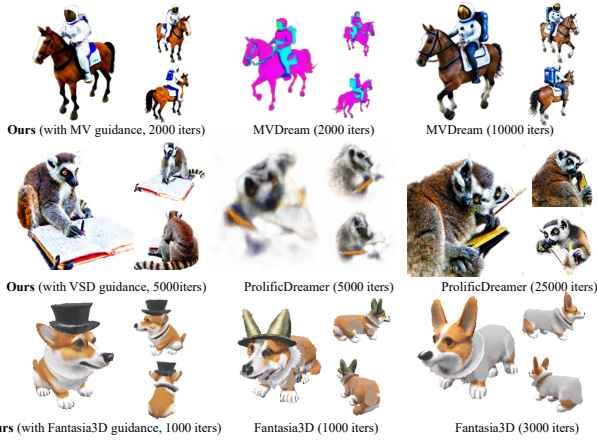

Figure 6. We replace the 2D guidance used in the refinement with those in other methods. We can achieve comparable or better results on significantly fewer iterations.

generate new shapes by combining seen words while PI3D supports complex concepts. Our sampling takes around 3s and ATT3D has a runtime of 1s. As it is not open-sourced, we show the results from their paper in Fig. 7. Furthermore, it employs a deterministic mapping network with no diversity.
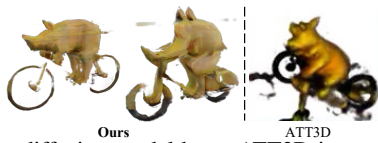

Figure 7. Our diffusion model beats ATT3D in terms of 3D consistency and can produce diverse results.

4.3. Ablation study

Depth loss in triplane fitting To enforce a correct underlying geometry in the triplane fitting stage, we incorporate a depth loss on the expected depth in volume rendering. We compare the outcomes of using depth loss versus fitting without it, as illustrated in Fig. 9. It is evident that the imposition of the depth loss constraint is advantageous for forming the correct geometry surface. In the absence of depth loss, the object's surface tends to exhibit cavities near the spherical area, resulting in hollowness in the corresponding part of the triplane. Therefore, adopting depth
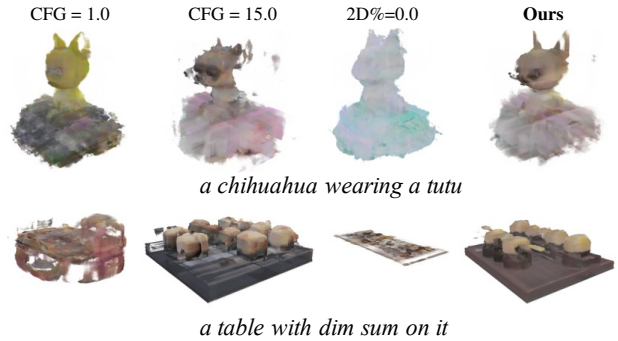

Figure 8. Qualitative results of ablation experiments on different classifier-free guidance scales and the probabilities of training with real images.

loss during the fitting stage is crucial to ensure a plausible triplane distribution. Both examples utilize the same shared decoder and hyper-parameters, except for λ_{depth}.
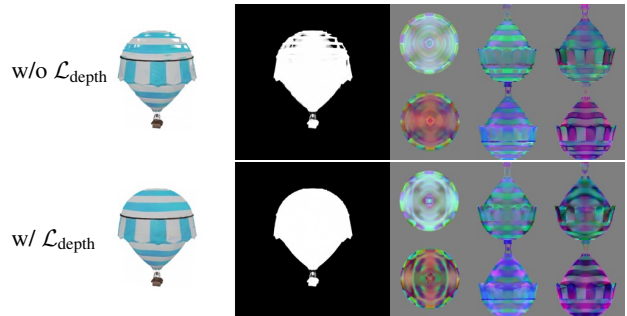

Figure 9. Fitting the object w/ and w/o the depth loss. Rendered RGB images, opacity images, and pseudo-images are shown.

Probability of training with real images In the pseudo-image diffusion training phase, we observe that jointly fine-tuning the model with both pseudo-images and real images is effective in preserving the prior knowledge from the 2D pre-trained diffusion model. Consequently, the generalization ability can be maintained to some degree. However, using different probabilities to train with real images can lead to distinct outcomes, such as an excessive focus on 2D image generation, neglecting triplane quality, or a susceptibility to overfitting to 3D datasets. We evaluate various

Table 2. Quantitative results of the ablation experiments on probability for training with 2D data and classifier free guidance.

CFG	2D%	ViT-B-32			ViT-L-14		
		CLIP Score	R-Precision@1	R-Precision@10	CLIP Score	R-Precision@1	R-Precision@10
1.0	0.5	59.2	8.4	26.9	45.6	10.1	28.0
3.0	0.5	65.9	24.6	55.1	53.2	29.1	61.7
5.0	0.5	65.9	**25.2**	**56.7**	**53.8**	29.3	63.6
7.5	0.5	**66.0**	24.4	**56.7**	**53.8**	29.3	63
10.0	0.5	64.4	23.4	54.2	53.2	28.1	61.6
5.0	0	61.5	15.7	41.8	47.4	16.7	43.1
5.0	0.3	64.7	22.2	53.9	51.5	23.9	57.3
5.0	0.7	65.4	**25.2**	56.3	53.4	**30.8**	**64.0**

training settings using the metrics and prompt datasets used above. The results are presented in Tab. 2, where the probability of utilizing real images is marked as 2D%. When training with only 3D data, the model behaves similarly to Shap·E [20]. They both lose the rich semantic knowledge in the 2D data and are prone to overfitting to low-quality 3D data, resulting in poor performance on more generalized complex prompts. The generalization ability of the model increases significantly as the proportion of 2D data increases, and models trained with real images at probabilities of 50% and 70% perform comparably. We select 50% as our final model. Additionally, we show several qualitative results in Fig. 8. Training without real images makes the model less generalizable and unable to comprehend complex text inputs.

Classifier free guidance We experiment with various classifier-free guidance scales to identify the optimal match between the generated 3D content and the input text prompt. The qualitative results for different scales are presented in Tab. 2. It is evident that PI3D performs best when the scale is in the range of $5.0 \sim 7.5$. We attribute this to the parallel training with real images, aligning the optimal range with sampling the image diffusion model. The qualitative results in Fig. 8 indicate that the 3D model poorly conforms to the text and exhibits rough geometry when the scale is too small, while a large scale introduces noisy geometry although the semantics are more text-aligned.

5. Limitations

Although PI3D demonstrates the ability to efficiently generate 3D consistent shapes from text prompts, there still exist some limitations left for the future to address. First, our training procedure involves a per-object triplane fitting phase, which leads to a linear increase in training costs when scaling up to larger 3D datasets. This might be solved by encoding triplanes from point clouds or multi-view images with an autoencoder architecture like [11, 32]. Additionally, the capacity of our scene representation is limited

by the feature dimensions. When it comes to more detailed 3D generation, higher dimensional features are inevitably required, which raises the number of pseudo-images and poses a challenge in modeling the relationships between them. This may be solved by designing more explicit and effective feature interactions.

6. Conclusion

In this paper, we introduce PI3D, a novel and efficient framework that utilizes the pre-trained text-to-image diffusion models to generate high-quality 3D shapes in minutes. On the one hand, it fine-tunes a pre-trained 2D diffusion model into a 3D diffusion model, enabling both 3D generative capabilities and generalization derived from the 2D model. On the other, it utilizes score distillation sampling of 2D diffusion models to quickly improve the quality of the sampled 3D shapes. PI3D enables the migration of knowledge from image to triplane generation by treating it as a set of pseudo-images. We adapt the modules in the pre-training model to enable hybrid training using pseudo and real images, which has proved to be a well-established strategy for improving generalizability. The efficiency of PI3D is highlighted by its ability to sample diverse 3D models in seconds and refine them in minutes. The experimental results confirm the advantages of PI3D over existing methods based on either 3D diffusion models or lifting 2D diffusion models in terms of fast generation of 3D consistent and high-quality models. The proposed PI3D stands as a promising advancement in the field of text-to-3D generation, and we hope it will inspire more research into 3D generation leveraging the knowledge in both 2D and 3D data.

References

[1] Stability AI. Deepfloyd-if. https://github.com/deep-floyd/IF, 2023. 6

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

[3] Ziang Cao, Fangzhou Hong, Tong Wu, Liang Pan, and Ziwei Liu. Large-vocabulary 3d diffusion model with transformer. *CoRR*, abs/2309.07920, 2023. 2, 4

[4] 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, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, pages 16102–16112. IEEE, 2022. 2

[5] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. 2

[6] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *CoRR*, abs/2303.13873, 2023. 4, 6

[7] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G. Schwing, and Liangyan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *CVPR*, pages 4456–4465. IEEE, 2023. 3

[8] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. *CoRR*, abs/2307.05663, 2023. 1, 2

[9] 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 *CVPR*, pages 13142–13153. IEEE, 2023. 1, 2, 4

[10] 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. 6, 7

[11] Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oguz. 3dgen: Triplane latent diffusion for textured mesh generation. *CoRR*, abs/2303.05371, 2023. 2, 3, 8

[12] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. *CoRR*, abs/2303.09556, 2023. 6

[13] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP (1)*, pages 7514–7528. Association for Computational Linguistics, 2021. 6

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *CoRR*, abs/2207.12598, 2022. 6

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 6

[16] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey A. Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *CoRR*, abs/2210.02303, 2022. 4

[17] Jonathan Ho, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *NeurIPS*, 2022. 4

[18] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: large reconstruction model for single image to 3d. *CoRR*, abs/2311.04400, 2023. 3

[19] Ka-Hei Hui, Ruihui Li, Jingyu Hu, and Chi-Wing Fu. Neural wavelet-domain diffusion for 3d shape generation. In *SIGGRAPH Asia*, pages 24:1–24:9. ACM, 2022. 3

[20] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *CoRR*, abs/2305.02463, 2023. 2, 3, 5, 6, 8

[21] James T. Kajiya and Brian Von Herzen. Ray tracing volume densities. In *SIGGRAPH*, pages 165–174. ACM, 1984. 3

[22] Animesh Karnewar, Niloy J. Mitra, Andrea Vedaldi, and David Novotný. Holofusion: Towards photo-realistic 3d generative modeling. In *ICCV*, pages 22919–22928. IEEE, 2023. 2

[23] Animesh Karnewar, Andrea Vedaldi, David Novotný, and Niloy J. Mitra. HOLODIFFUSION: training a 3d diffusion model using 2d images. In *CVPR*, pages 18423–18433. IEEE, 2023. 2

[24] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14, 2023. 1

[25] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *CoRR*, abs/2311.06214, 2023. 3

[26] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. In *NeurIPS*, 2023. 3

[27] 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. *CoRR*, abs/2303.11328, 2023. 3

[28] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *CoRR*, abs/2309.03453, 2023. 3

[29] 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 *ICCV*, pages 17900–17910. IEEE, 2023. 3, 6

[30] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. *CoRR*, abs/2306.07279, 2023. 4

[31] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV (1)*, pages 405–421. Springer, 2020. 1, 3

[32] 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. 2, 5, 6, 8

[33] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020. 2

[34] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*. OpenReview.net, 2023. 1, 2, 3, 4, 6, 7

[35] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, and Bernard Ghanem. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *CoRR*, abs/2306.17843, 2023. 3, 4

[36] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 1

[37] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 1

[38] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotný. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *ICCV*, pages 10881–10891. IEEE, 2021. 2

[39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10674–10685. IEEE, 2022. 1, 3, 4

[40] 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. 1

[41] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*. OpenReview.net, 2022. 6

[42] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: an open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 1, 4, 6

[43] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *NeurIPS*, 34: 6087–6101, 2021. 1

[44] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *CoRR*, abs/2308.16512, 2023. 3, 4, 6

[45] J. Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *CVPR*, pages 20875–20886. IEEE, 2023. 2

[46] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *CoRR*, abs/2309.16653, 2023. 3

[47] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. In *NeurIPS*, 2023. 3

[48] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *CVPR*, pages 12619–12629. IEEE, 2023. 1, 3

[49] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 1

[50] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *NeurIPS*, 2023. 3, 4, 6

[51] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633, 2023. 4

[52] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *CVPR*, pages 803–814. IEEE, 2023. 2

[53] Junzhe Zhu and Peiye Zhuang. Hifa: High-fidelity text-to-3d with advanced diffusion guidance. *CoRR*, abs/2305.18766, 2023. 3