Direct3D: Scalable Image-to-3D Generation via 3D Latent Diffusion Transformer

Shuang Wu[1,2*] **Youtian Lin**[2*] **Feihu Zhang**[1] **Yifei Zeng**[1,2] **Jingxi Xu**[1]

Philip Torr[3†] **Xun Cao**[2] **Yao Yao**[2‡]

[1]DreamTech [2]Nanjing University [3]University of Oxford

{wushuang,linyoutian}@smail.nju.edu.cn yaoyao@nju.edu.cn

Abstract

Generating high-quality 3D assets from text and images has long been challenging, primarily due to the absence of scalable 3D representations capable of capturing intricate geometry distributions. In this work, we introduce Direct3D, a native 3D generative model scalable to in-the-wild input images, without requiring a multi-view diffusion model or SDS optimization. Our approach comprises two primary components: a Direct 3D Variational Auto-Encoder (**D3D-VAE**) and a Direct 3D Diffusion Transformer (**D3D-DiT**). D3D-VAE efficiently encodes high-resolution 3D shapes into a compact and continuous latent triplane space. Notably, our method directly supervises the decoded geometry using a semi-continuous surface sampling strategy, diverging from previous methods that rely on rendered images as supervision signals. D3D-DiT models the distribution of encoded 3D latents and is specifically designed to fuse positional information from the three feature maps of the triplane latent, enabling a native 3D generative model scalable to large-scale 3D datasets. Additionally, we introduce an innovative image-to-3D generation pipeline incorporating semantic-level and pixel-level image conditions, allowing the model to produce 3D shapes consistent with the provided conditional image input. Extensive experiments demonstrate the superiority of our large-scale pre-trained Direct3D over previous image-to-3D approaches, achieving significantly better generation quality and generalization ability, thus establishing a new state-of-the-art for 3D content creation. Project page: https://www.neural4d.com/research/direct3d.

1 Introduction

In recent years, substantial advancements have been made in 3D shape generation through the utilization of diffusion models [13, 51]. Inspired by the efficacy demonstrated in text-to-2D image generation, these methods seek to extend the capabilities of diffusion models to the realm of 3D shape generation through extensive training on diverse 3D datasets. Various approaches have explored diverse 3D representations, including point clouds [37, 38], voxels [45], and SDFs [35], aiming not only to faithfully capture object appearance but also to preserve intricate geometric details. However, existing large-scale 3D datasets, such as ObjverseXL [6], are constrained both in the quantity and

[*]Equal contribution.

[†]Chief scientific advisor of DreamTech, all work was done at DreamTech.

[‡]Corresponding author.

This research was supported by DreamTech, and the IP belongs to DreamTech.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3873

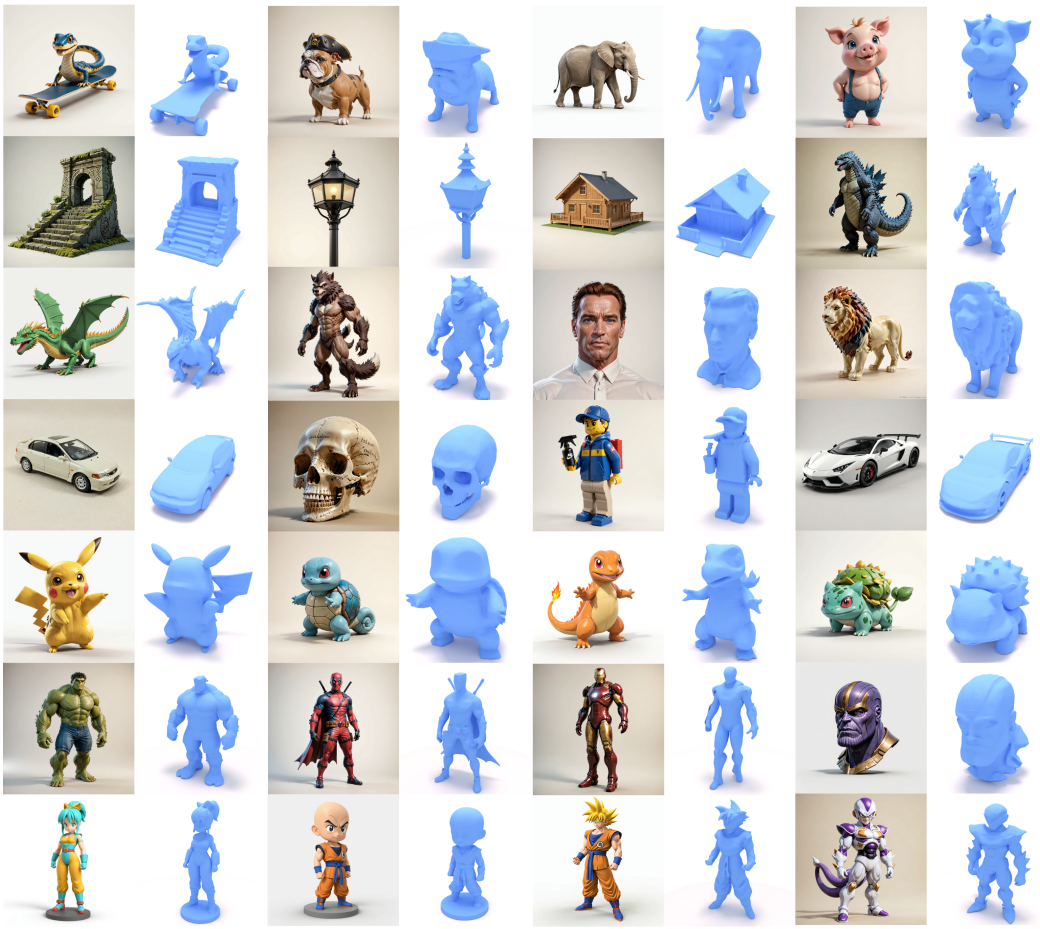

Figure 1: Direct3D is a novel image-to-3D generation method that directly trains on larger-scale 3D datasets and performs state-of-the-art generation quality and generalizability. We achieve this by designing a novel 3D latent diffusion model to take an image as the prompt and generate high-quality 3D shapes that highly consistent with input images. As shown above, our method can generate 3D shapes from existing text-to-image diffusion models, which indicates that our method generalizes to in-the-wild images, while it only trains on 3D data.

diversity of shapes compared to their 2D counterparts like Laion5B [47], which contains 5 billion images, while ObjverseXL only comprises 10 million 3D shapes.

To address this limitation, many existing methods [5, 4, 9, 26, 27, 28, 29, 31, 34, 43, 65] employ a pipeline where multi-view images of an object are initially generated from a single image using a multi-view diffusion model. Subsequently, techniques such as sparse view reconstruction methods [23, 32, 55, 62] or score distillation sampling (SDS) optimization [42, 43, 49, 54] are applied to fuse these multi-view images into 3D shapes. While this pipeline can result in high-quality 3D shape creation, the indirect generation from multi-view images raises efficiency concerns. Additionally, the quality of the resulting shape is heavily dependent on the fidelity of the multi-view images, often leading to detail loss or reconstruction failures.

In this paper, we eschew the conventional approach of indirectly generating multi-view images and instead advocate the direct generation of 3D shapes from single-view images, leveraging a native 3D diffusion model. Inspired by the success of latent diffusion models in 2D image generation, we propose the utilization of a 3D variational auto-encoder (VAE) [20] to encode 3D shapes into a latent space, followed by a diffusion transformer model (DiT) [40] to generate 3D shapes from this latent space, conditioned on an image input. However, efficiently encoding a 3D shape into a latent space conducive to diffusion model training is challenging, as is decoding the latent representation back into 3D geometry. Previous approaches have employed multi-view images as indirect supervision [19, 22, 35, 61] through differentiable rendering, but still encounter accuracy and efficiency issues. To

address these challenges, we employ a transformer model to encode high-resolution point clouds into an explicit triplane latent, which has been widely used in 3D reconstruction methods [2] for its efficiency. While the latent triplane is intentionally set with a low resolution, we introduce a convolutional neural network to upsample the latent resolution and decode it into a high-resolution 3D occupancy grid. Furthermore, to ensure precise supervision of the 3D occupancy grid, we adopt a semi-continuous surface sampling strategy, enabling the sampling and supervision of surface points in both continuous and discrete manners. This approach facilitates the encoding and reconstruction of 3D shapes within a compact and continuous explicit latent space.

For image-to-3D generation, we further leverage an image input as a condition to the 3D diffusion transformer. This involves arranging the 3D latent space as a combination of three orthogonal views of a 3D shape and incorporating pixel-level image information into each DiT block to enhance conditional consistency. Furthermore, we introduce cross-attention layers into each DiT block to incorporate semantic-level image information, thereby facilitating the generation of high-quality 3D shapes consistent with input images.

We demonstrate the high-quality 3D generation and strong generalization abilities of the proposed Direct3D approach through extensive experiments. Figure 1 illustrates the 3D generation results of our method on the in-the-wild images generated from text-to-image model. To summarize, the major contributions of this work include:

- We introduce Direct3D, to our best knowledge, the first native 3D generative model scalable to in-the-wild input images (e.g., from Flux [21], Hunyuan-DiT [24] or SDXL [41]). This enables high-fidelity image-to-3D generation without the need for multi-view diffusion models or SDS optimization.
- We propose D3D-VAE, a novel 3D variational auto-encoder effectively encoding a 3D point cloud into a triplane latent. Instead of using rendered images as supervision signals, we supervise the decoded geometry directly using a semi-continuous surface sampling strategy to preserve detailed 3D information in the latent triplane.
- We present D3D-DiT, a scalable image-conditioned 3D diffusion transformer capable of generating 3D asserts consistent with input images. The D3D-DiT is specially designed to better fuse the positional information from the latent triplane and effectively integrates pixel-level and semantic-level information from the input image.
- We demonstrate through extensive experiments that our large-scale pre-trained Direct3D model surpasses previous image-to-3D approaches in terms of generation quality and generalization ability, setting a new state-of-the-art for the task of 3D content creation.

2 Related Work

2.1 Neural 3D Representations for 3D Generation

Neural 3D representations are essential for 3D generation tasks. The introduction of Neural Radiance Fields (NeRF) [36] has significantly advanced 3D generation. Building on NeRF, DreamFusion [42] introduced a Score Distillation Sampling (SDS) method to generate 3D shapes using an off-the-shelf 2D diffusion model from any text prompt. Many subsequent methods have explored various representations to enhance the speed and quality of 3D generation. For instance, Magic3D [25] improves generation quality by introducing a second stage using the DMtet [48] representation, which combines Signed Distance Function (SDF) with a tetrahedral grid to represent the 3D shape.

Beyond SDS-based methods, some approaches use directly trained networks to generate different representations [16, 17, 59]. For example, LRM [16] uses triplane NeRF representations as network outputs, significantly speeding up the generation process, albeit with some loss in quality. Another approach, One-2-3-45++ [26], proposes to use a 3D occupancy grid as the output representation to enhance geometric quality.

2.2 Multi-view Diffusion

Following the success of novel view prediction methods using diffusion models, such as Zero123 [28], which generates different unknown views of an object from a single image and text guidance.

https://doi.org/10.52202/079017-3873

MVDream [49] extends novel view diffusion to generate multiple views of an object at once, improving consistency across views. Imagedream [54] further enhances generation quality by introducing a novel image conditional module. Some methods adopt this approach to first generate multi-view images of an object and then reconstruct the 3D shape from these views using sparse reconstruction [23, 31, 55, 62]. Instant3D [23] proposes a reconstruction model that takes four multi-view images as input and reconstructs a NeRF representation of the 3D shape. Many subsequent methods have improved on this by enhancing multi-view or reconstruction models [53, 57, 58].

2.3 Direct 3D Diffusion

Despite the challenges of directly training a 3D diffusion model, such as the lack of a diffusible 3D representation, various strategies have been explored. One line of work fits multiple NeRFs to obtain a neural representation of 3D datasets and then applies a diffusion model to generate NeRFs from this learned representation [50]. However, separate training of NeRFs can hinder the diffusion model's ability to generalize to more diverse 3D shapes. 3DGenNeural [50] proposes joint training of triplane fitting of the 3D shape with occupancy as direct supervision to train the triplane reconstruction model.

Another line of work leverages VAEs to encode 3D shapes into a latent space and trains a diffusion model on this latent space to generate 3D shapes [15, 19, 22, 61]. For instance, Shap-E [19] uses a pure transformer VAE to encode a point cloud and image of a 3D shape into an implicit latent space, which is then recovered into a NeRF and SDF field. 3DGen [10] encodes only the point cloud of a 3D shape into an explicit triplane latent space, enhancing generation efficiency. Similar to previous works that fit multiple NeRFs, 3DTopia [15] fits multiple triplane NeRFs and encodes the triplane into a latent space for which a diffusion model is trained to generate 3D shapes. 3DShape2VecSet [60] and Michelangelo [64] employ 3D occupancy as the output representation for the VAE but use multiple 1D vectors as implicit latent space instead of a triplane.

However, these methods often rely on rendering loss to supervise the VAE reconstruction, resulting in suboptimal reconstruction and generation quality. Additionally, using implicit latent representations not designed for efficient encoding and lacking compact explicit 3D representations for diffusion further limits their performance. Our 3D VAE combines the advantages of explicit 3D latent representation and direct 3D supervision to achieve high-quality VAE reconstruction, ensuring robust 3D shape generation. Furthermore, our design for the diffusion architecture specifically addresses conditional 3D latent generation. Our 3D DiT facilitates pixel-level and semantic-level 3D-specific image conditioning, allowing the diffusion process to generate highly detailed 3D shapes consistent with the condition images.

3 Methods

Inspired by LDM [46], we train a latent diffusion model for 3D generation within a 3D latent space. Unlike previous methods [19, 64] that typically rely on a 1D implicit latent space for generative models, our approach addresses two crucial limitations: 1) the struggle of the implicit latent repr esentation to capture structured information inherent in 3D space, leading to sub-optimal quality of decoded 3D shapes; 2) the challenge of training and sampling from the latent distribution, given that the implicit latent space is unstructured and under-constrained.

To mitigate these issues, we adopt an explicit triplane latent representation, utilizing a triplane of three feature maps to represent the 3D geometry latent. The design draws inspiration from LDM, which applies feature maps to represent the 2D image latent. Figure 2 illustrates the overall framework of our proposed method, which comprises a two-step training process: 1) the D3D-VAE is first trained to convert 3D shapes into 3D latents, which is described in Sec. 3.1; 2) the image-conditioned D3D-DiT is then trained to generate high-quality 3D assets, which is detailed in Sec. 3.2.

3.1 Direct 3D Variational Auto-Encoder

The proposed D3D-VAE consists of three components: a point-to-latent encoder, a latent-to-triplane decoder, and a geometry mapping network. Meanwhile, we design a semi-continuous surface sampling strategy that utilizes both continuous and discrete supervision to ensure the high-frequency geometric details of the decoded 3D shape.

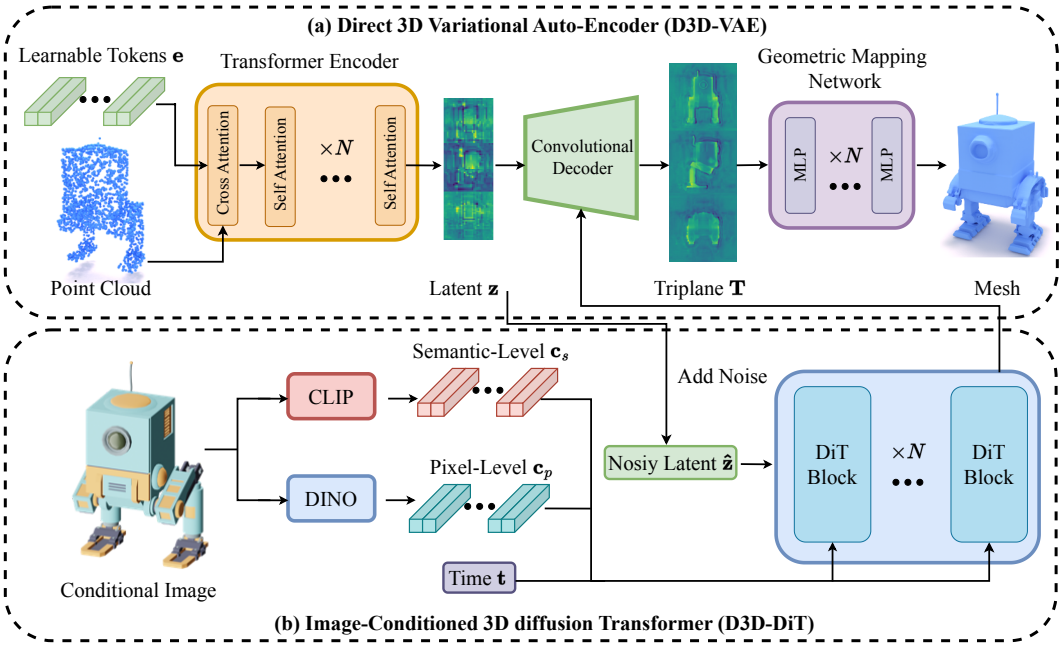

Figure 2: The framework of our Direct3D. (a) We utilize transformer to encode point cloud sampled from 3D model, along with a set of learnable tokens, into an explicit triplane latent space. Subsequently, a CNN-based decoder is employed to upsample these latent representations into high-resolution triplane feature maps. The occupancy values of queried points can be decoded through a geometric mapping network. (b) Then we train the image conditional latent diffusion transformer in the 3D latent space obtained by VAE. Pixel-level information and semantic-level information from images are extracted using DINO-v2 and CLIP, respectively, and then injected into each DiT block.

Point-to-latent encoder. In order to obtain robust representations in the latent space that can effectively capture intricate geometry, we uniformly sample high-resolution point clouds from the surface of 3D objects, which is then encoded to an explicit latent representation $\mathbf{z} \in \mathbb{R}^{(3\times r\times r)\times d_{\mathbf{z}}}$, where r and $\mathbf{d_z}$ denotes the resolution and channel dimensional of the latent representation, respectively. To be specific, given a set of point clouds $P \in \mathbb{R}^{N_P\times(3+3)}$ sampled from 3D models, where N_P denotes the number of points, the channel dimension $(3+3)$ comprises of the normalized position and normal of each point, we first use Fourier features [18] to represent the position structure of point clouds. Then we introduce a series of learnable tokens $\mathbf{e} \in \mathbb{R}^{(3\times r\times r)\times d_{\mathbf{e}}}$ to query the point cloud features using a cross-attention layer, where $d_{\mathbf{e}}$ denotes the channel dimensional of $\mathbf{e}$. This enables the injection of 3D information from the point clouds into the latent tokens. Subsequently, multiple self-attention layers are employed to enhance the representation of these tokens, ultimately yielding the latent representation $\mathbf{z} \in \mathbb{R}^{(3\times r\times r)\times d_{\mathbf{z}}}$, where $d_{\mathbf{z}}$ represents the channel dimensional of $\mathbf{z}$.

Latent-to-triplane decoder. After obtaining the latent representation $\mathbf{z}$, we reshape it to the triplane representation. Inspired by RODIN [56], we concatenate the three planes vertically along the height dimension, yielding $\mathbf{z} \in \mathbb{R}^{r\times(3\times r)\times d_{\mathbf{z}}}$, to prevent incorrect blend of the planes across the channel dimension. Afterwards, the latent-triplane decoder upsamples $\mathbf{z}$ to high-resolution triplane feature maps with upsampling factors f. In contrast to the transformer architecture used in the encoder, our decoder model employs convolutional networks to progressively upsample the explicit latent representation and obtain the final triplane $\mathbf{T} = (\mathbf{T_{XY}}, \mathbf{T_{YZ}}, \mathbf{T_{XZ}})$.

Semi-continuous surface sampling. We employ a Multi-Layer Perceptron (MLP) as the geometric mapping network to predict the occupancy of queried points via features interpolated from the triplane. The MLP contains multiple linear layers with ReLU activation. Typical occupancy is represented by a discrete binary value of 0 and 1 to indicate whether a point is inside an object. However, when the query point is very close to the object surface, it can result in abrupt gradient changes that affect model optimization. In this work, we adopt semi-continuous occupancy, using both continuous and discrete supervision to ensure smooth gradient. Specifically, given a query point $\mathbf{x}$ in 3D space, when its distance to the surface is greater than a small threshold value $s = \frac{1}{512}$, the occupancy value

https://doi.org/10.52202/079017-3873

remains either 0 or 1. When the distance is less than s, a continuous value ranging from 0 to 1 is assigned to it. The formula for the semi-continuous occupancy $o(\mathbf{x})$ is as follows:

$$o(\mathbf{x}) = \begin{cases} 1, & \text{if } sdf(\mathbf{x}) < -s \\ 0.5 - \frac{0.5 \cdot sdf(\mathbf{x})}{s}, & \text{if } -s \leq sdf(\mathbf{x}) \leq s \\ 0, & \text{if } sdf(\mathbf{x}) > s \end{cases}, \tag{1}$$

where $sdf(\mathbf{x})$ denotes the Signed Distance Function (SDF) value of $\mathbf{x}$.

End-to-end optimization. During the training process, we uniformly sample points from the 3D space and sample points proximate to the object surface to predict their semi-continuous occupancy. We utilize Binary Cross-Entropy (BCE) loss L_{BCE} to supervise the predictions. Additionally, we employ KL loss L_{KL} to prevent excessive variance in the latent space. Thus, our D3D-VAE is optimized by minimizing:

$$L_{\text{D3D-VAE}} = L_{\text{BCE}} + \lambda_{\text{KL}} L_{\text{KL}}, \tag{2}$$

where λ_{KL} denotes the weight of KL regularization.

3.2 Image-conditioned Direct3D Diffusion Transformer

After training the D3D-VAE, we have access to a continuous and compact latent space, upon which we train the latent diffusion model. Since the obtained latent embedding is an explicit triplane representation, a naive approach would be to directly use a well-designed 2D U-Net as the diffusion model. However, this would result in a lack of communication between the three planes, thus failing to capture the structured and intrinsic properties required for 3D generation. Therefore, we build the generation model based on the architecture of the Diffusion Transformer (DiT), utilizing the transformer to better extract spatial positional information among the planes. Meanwhile, we propose to incorporate pixel-level and semantic-level information of the image in each DiT block, thereby aligning the image feature space and latent space to generate 3D assets consistent with the conditional image content. The framework of our latent diffusion model is shown in Figure 2 **(b)** and the architecture of each DiT block is illustrated in Figure 3.

Pixel-level alignment module. To ensure the high-frequency details of 3D assets generated by the diffusion model are aligned with the conditional images, we design a pixel-level alignment module to inject pixel-level information from the images into the latent space. We employ the pre-trained DINO-v2 [39] (ViT-L/14) as the pixel-level image encoder, which has been revealed in previous work [1] to outperform other pre-trained vision models in extracting structural information beneficial for 3D tasks. Specifically, we first use two linear layers with GeLU [12] activation to project the image tokens $\mathbf{c}_{\text{p}}$ extracted by DINO-v2 to match the channel dimension of the noisy latent tokens $\mathbf{z}_t$. Then in each DiT block, we concatenate them with the flattened $\mathbf{z}_t$ and feed them into a self-attention layer to model the intrinsic relationship between $\mathbf{c}_{\text{p}}$ and $\mathbf{z}_t$. Subsequently, we eliminate the part of image tokens and only reserve the part of noisy tokens for input to the next module.

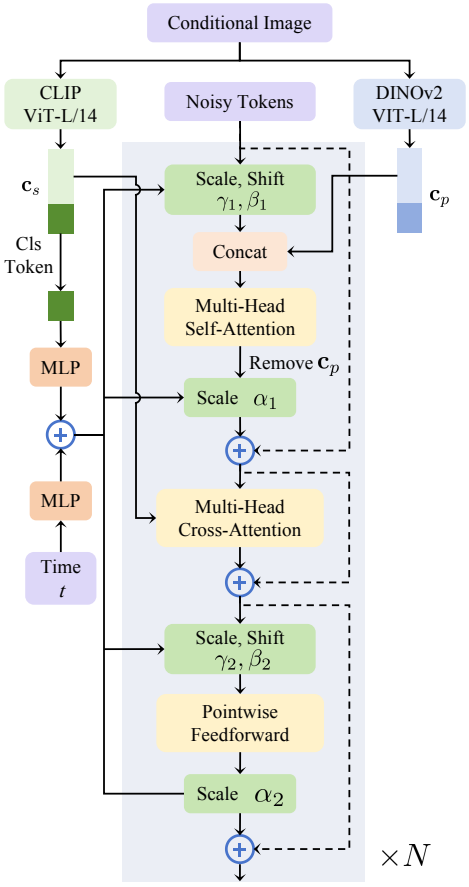

Figure 3: The architecture of our 3D DiT. We employ the pre-trained DINO-v2 and CLIP vision model to extract tokens from conditional images respectively, then incorporate the pixel-level and semantic-level information into each DiT block.

Table 1: Quantitative results on Google Scanned Objects dataset.

Methods	Chamfer Distance ↓	Volume IoU ↑	F-Score ↑
Shap-E [19]	0.0585	0.2347	0.3474
Michelangelo [64]	0.0441	0.1260	0.4371
One-2-3-45 [27]	0.0513	0.2868	0.3149
InstantMesh [57]	0.0327	0.4105	0.5058
Ours (trained on Objaverse)	0.0296	0.4307	0.5356
Ours (trained on Objaverse + internal data)	**0.0271**	**0.4323**	**0.5624**

Semantic-level alignment module. We devise a semantic-level alignment module to ensure semantic consistency between the generated 3D models and the conditional images. We employ the pre-trained CLIP [44] visual model (ViT-L/14) to extract semantic image tokens $\mathbf{c}_{\mathrm{s}}$ from the conditional images, and then utilize a cross-attention layer within each DiT block to facilitate the interaction between $\mathbf{c}_{\mathrm{s}}$ and noisy latent token $\mathbf{z}_t$. Meanwhile, unlike the original class conditional DiT, our image-conditioned diffusion model no longer utilizes class embedding. Instead, we use the classification token from the semantic image tokens $\mathbf{c}_{\mathrm{s}}$ after projection and add it to the time embedding to enhance semantic features. In addition, to reduce the number of parameters and computational cost, we employ adaLN-single, as proposed in PixArt [3], which predicts a set of global shift and scale parameters $P = [\gamma_1, \beta_1, \alpha_1, \gamma_2, \beta_2, \alpha_2]$ using time embeddings, then sets a trainable embedding and adds it to P for adjustment in each block.

Training. Following LDM [46], our 3D latent diffusion transformer model predicts the noise ϵ of the noisy latent representation $\mathbf{z}_{\mathrm{t}}$ at time t, conditioned on image C. When training the diffusion model, we randomly zero the conditional input $\mathbf{c}_{\mathrm{p}}$ and $\mathbf{c}_{\mathrm{s}}$ with a probability of 10% to use classifier-free guidance [14] during inference, thereby improving the quality of conditional generation.

4 Experiments

4.1 Dataset

Our Direct3D is trained on a filtered subset of the Objaverse [7] dataset which consists of 160K high-quality 3D assets. To evaluate the scalability of our Direct3D, we also employ additional internal data for training. Each 3D model is normalized to a unit sphere centered at the world origin. To construct conditional images for training the 3D latent diffusion transformer, we randomly render 24 views at a resolution of 512×512 using Blender for each 3D model. Additionally, we employ depth-conditioned ControlNet [63] to generate 16 diverse images to ensure the generalization of the diffusion model. To evaluate the performance of our Direct3D, we randomly select 30 3D models from the Google Scanned Objects (GSO) [8] dataset for image-to-3D experiments. For the text-to-3D task, we utilize existing text-to-image models like Hunyuan-DiT [24] to generate images with several classic text prompts as conditional inputs for qualitative comparisons with other methods. The ablation studies for each component are presented in the Appendix.

4.2 Implementation Details

D3D-VAE. Our D3D-VAE takes as input 81,920 point clouds with normal uniformly sampled from the 3D model, along with a learnable latent token of a resolution $r = 32$ and a channel dimension $d_{\mathbf{e}} = 768$. The encoder network consists of 1 cross-attention layer and 8 self-attention layers, with each attention layer comprising 12 heads of a dimension 64. The channel dimension of the latent representation is $d_{\mathbf{z}} = 16$. The decoder network comprises of 1 self-attention layer and 5 ResNet [11] blocks to upsample the latent representation into triplane feature maps with resolution of 256×256 and channel dimension of 32. The geometric mapping network consists of 5 linear layers with hidden dimension 64. During training, we sample 20,480 uniform points and 20,480 near-surface points for supervision. The KL regularization weight is set to $\lambda_{\mathrm{KL}} = 1e-6$. We use the AdamW [33] optimizer with a learning rate $1e-4$ and a batch size of 16 per GPU.

D3D-DiT. Our diffusion model adopts the network configuration of DiT-XL/2 [40], which consists of 28 layers of DiT blocks. Each attention layer includes 16 heads with a dimension of 72. We train the

https://doi.org/10.52202/079017-3873

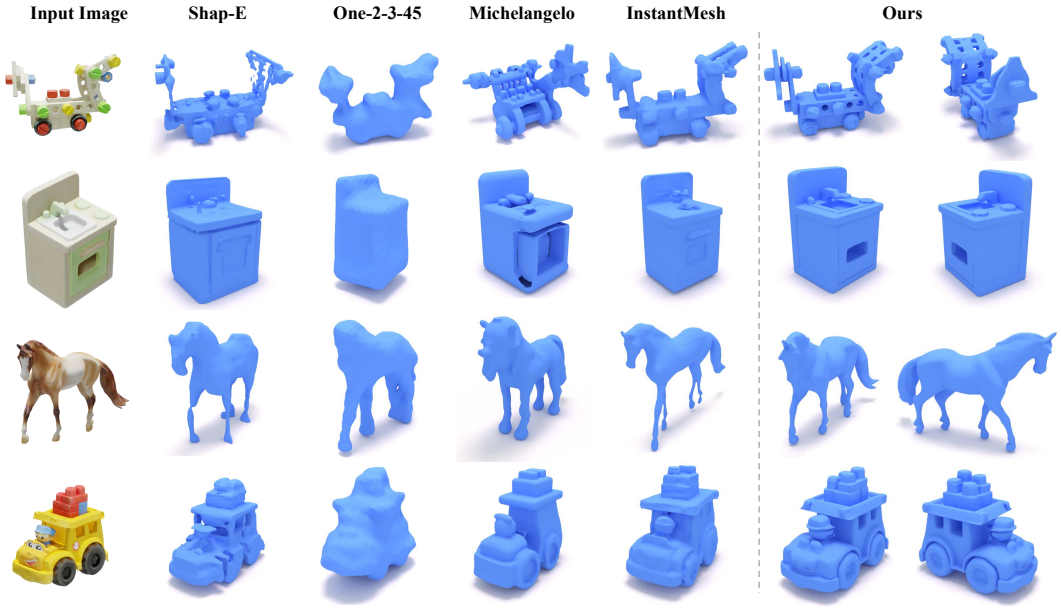

Figure 4: Qualitative comparisons with different baseline methods on GSO dataset.

Table 2: User study on the quality of meshes. The higher the score, ranging from 1 to 5, the better.

	Shap-E [19]	One-2-3-45 [27]	Michelangelo [64]	InstantMesh [57]	Ours
Quality	1.18	1.24	2.51	2.53	**4.41**
Consistency	1.19	1.28	2.32	2.66	**4.35**

diffusion model with 1000 denoising steps using a linear variance scheduler ranging from $1e-4$ to $2e-2$. We employ the AdamW optimizer with a batch size of 32 per GPU and train for 800K steps. During inference, we apply 50 steps of DDIM [52] with the guidance scale set to 7.5.

4.3 Image and Text to 3D Generation

Image-to-3D. We conduct qualitative and quantitative comparisons of our Direct3D with other baseline methods on the GSO dataset for the image-to-3D task, as illustrated in Figure 4 and Table 1, respectively. Shap-E [19], a 3D diffusion model trained on millions of 3D assets, is capable of producing plausible geometry, but it suffers from artifacts and holes in the meshes. Michelangelo [64] performs diffusion process on a 1D implicit latent space, and fails to align the generated mesh with the semantic content of the conditional images. Multi-view based approaches such as One-2-3-45 [27] and InstantMesh [57] heavily rely on the performance of multi-view 2D diffusion model. One-2-3-45 directly employs SparseNeuS [32] for reconstruction, resulting in coarse geometry. The meshes generated by InstantMesh perform decent quality, but lack consistency with the input images in certain details like the water spout on the sink and the windows of the school bus. It also produces some failure cases such as merging the hind legs of the horse together, due to the limitation of multi-view diffusion model. In contrast, our Direct3D consistently generates high-quality meshes that align with the conditional images in most cases. In Table 1, we report the Chamfer Distance, Volume IoU and F-Score to compare the quality of the generated meshes with other methods. It can be observed that our Direct3D achieves state-of-the-art performance across all metrics when trained on Objaverse dataset. Integrating our internal data for training further enhances the model's performance, validating the scalability of our approach.

Text-to-3D. Our Direct3D can produce 3D assets from text prompts by incorporating text-to-image models, such as Flux [21] and Hunyuan-DiT [24]. Figure 5 illustrates the qualitative comparisons of our Direct3D and other baseline methods on the text-to-3D task. To ensure a fair comparison, all methods utilize the same generated image as input. It can be observed that these baseline methods fail in almost all cases, while our Direct3D is still able to generate high-quality meshes, demonstrating

Figure 5: Qualitative comparisons of the meshes generated from text. We employ the existing text-to-image models (e.g. Hunyuan-DiT) to produce highly detailed images as the inputs of each method.

the generalizability of our approach. We also conducted a user study to quantitatively compare our D3D-DiT with other methods. We render videos of meshes generated by each method rotating 360 degrees, and ask 46 volunteers to rate each mesh based on its quality and consistency with the input images. The results in Table 2 indicate that our D3D-DiT perform superior mesh quality and consistency compared to other baseline methods.

Generation of textured mesh. Benefited from the smooth and detailed geometry produced by our Direct3D, we can easily dress up the mesh using existing texture synthesis methods. As shown in Figure 6, we utilize SyncMVD [30] to obtain exquisite textured meshes.

5 Conclusion

In conclusion, our paper introduces a novel approach for direct 3D shape generation from a single image, bypassing the need for multi-view reconstruction. Leveraging a hybrid architecture, our proposed D3D-VAE efficiently encode 3D shapes into a compact latent space, enhancing the fidelity of the generated shapes. Our image-conditioned 3D diffusion transformer (D3D-DiT) further improves the generation quality by integrating image information at both pixel and semantic levels, ensuring

https://doi.org/10.52202/079017-3873

Figure 6: Visualizations of the textured meshes. We employ SyncMVD [30] to generate texture for the meshes produced by our Direct3D.

high consistency between generated 3D shapes and conditional images. Extensive experiments on the image-to-3D and text-to-3D tasks demonstrate the superior performance of our Direct3D in 3D generation, surpassing existing methods in quality and generalizability.

Limitations. Despite the capability of our Direct3D to produce high-fidelity 3D assets, it is currently limited to the generation of individual or multiple objects and cannot generate large-scale scenes. We will focus on it in future research.

Acknowledgment. This work was mainly supported by DreamTech, and in part by the National Natural Science Foundation of China (62441204, 62472213).

References

[1] Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. *arXiv preprint arXiv:2404.08636*, 2024.

[2] 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 *CVPR*, 2022.

[3] Junsong Chen, Jincheng YU, Chongjian GE, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *ICLR*, 2024.

[4] Luxi Chen, Zhengyi Wang, Chongxuan Li, Tingting Gao, Hang Su, and Jun Zhu. Microdreamer: Zero-shot 3d generation in 20 seconds by score-based iterative reconstruction. *arXiv preprint arXiv:2404.19525*, 2024.

[5] Yabo Chen, Jiemin Fang, Yuyang Huang, Taoran Yi, Xiaopeng Zhang, Lingxi Xie, Xinggang Wang, Wenrui Dai, Hongkai Xiong, and Qi Tian. Cascade-zero123: One image to highly consistent 3d with self-prompted nearby views. *arXiv preprint arXiv:2312.04424*, 2023.

[6] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. In *NeurIPS*, 2024.

[7] 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*, 2023.

[8] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *ICRA*, 2022.

[9] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024.

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

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[12] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

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

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[15] Fangzhou Hong, Jiaxiang Tang, Ziang Cao, Min Shi, Tong Wu, Zhaoxi Chen, Tengfei Wang, Liang Pan, Dahua Lin, and Ziwei Liu. 3dtopia: Large text-to-3d generation model with hybrid diffusion priors. *arXiv preprint arXiv:2403.02234*, 2024.

[16] 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. In *ICLR*, 2024.

[17] Ka-Hei Hui, Aditya Sanghi, Arianna Rampini, Kamal Rahimi Malekshan, Zhengzhe Liu, Hooman Shayani, and Chi-Wing Fu. Make-a-shape: a ten-million-scale 3d shape model. *arXiv preprint arXiv:2401.11067*, 2024.

[18] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *ICML*, 2021.

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

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[21] Black Forest Labs. Flux, 2024. https://blackforestlabs.ai/.

[22] Yushi Lan, Fangzhou Hong, Shuai Yang, Shangchen Zhou, Xuyi Meng, Bo Dai, Xingang Pan, and Chen Change Loy. Ln3diff: Scalable latent neural fields diffusion for speedy 3d generation. *arXiv preprint arXiv:2403.12019*, 2024.

[23] 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. In *ICLR*, 2024.

[24] Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, et al. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding. *arXiv e-prints*, pages arXiv–2405, 2024.

[25] 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 *CVPR*, 2023.

[26] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. *arXiv preprint arXiv:2311.07885*, 2023.

[27] 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*, volume 36, 2024.

[28] 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. In *ICCV*, pages 9298–9309, 2023.

[29] 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. In *ICLR*, 2023.

[30] Yuxin Liu, Minshan Xie, Hanyuan Liu, and Tien-Tsin Wong. Text-guided texturing by synchronized multi-view diffusion. *arXiv preprint arXiv:2311.12891*, 2023.

[31] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008*, 2023.

[32] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *ECCV*, 2022.

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[34] Yuanxun Lu, Jingyang Zhang, Shiwei Li, Tian Fang, David McKinnon, Yanghai Tsin, Long Quan, Xun Cao, and Yao Yao. Direct2. 5: Diverse text-to-3d generation via multi-view 2.5 d diffusion. *arXiv preprint arXiv:2311.15980*, 2023.

[35] Zhaoyang Lyu, Ben Fei, Jinyi Wang, Xudong Xu, Ya Zhang, Weidong Yang, and Bo Dai. Getmesh: A controllable model for high-quality mesh generation and manipulation. *arXiv preprint arXiv:2403.11990*, 2024.

[36] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[37] Shentong Mo, Enze Xie, Ruihang Chu, Lanqing Hong, Matthias Niessner, and Zhenguo Li. Dit-3d: Exploring plain diffusion transformers for 3d shape generation. In *NeurIPS*, 2023.

[38] 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.

 https://doi.org/10.52202/079017-3873

[39] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2023.

[40] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

[41] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[42] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2022.

[43] Lingteng Qiu, Guanying Chen, Xiaodong Gu, Qi Zuo, Mutian Xu, Yushuang Wu, Weihao Yuan, Zilong Dong, Liefeng Bo, and Xiaoguang Han. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. *arXiv preprint arXiv:2311.16918*, 2023.

[44] 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 *ICML*, 2021.

[45] Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. *arXiv preprint arXiv:2312.03806*, 2023.

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[47] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R 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.

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

[49] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *ICLR*, 2023.

[50] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *CVPR*, 2023.

[51] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

[52] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2020.

[53] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024.

[54] Peng Wang and Yichun Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023.

[55] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. In *ICLR*, 2024.

[56] 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 *CVPR*, 2023.

[57] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024.

[58] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. *arXiv preprint arXiv:2403.14621*, 2024.

[59] Lior Yariv, Omri Puny, Natalia Neverova, Oran Gafni, and Yaron Lipman. Mosaic-sdf for 3d generative models. *arXiv preprint arXiv:2312.09222*, 2023.

[60] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics (TOG)*, 2023.

[61] Bowen Zhang, Tianyu Yang, Yu Li, Lei Zhang, and Xi Zhao. Compress3d: a compressed latent space for 3d generation from a single image. *arXiv preprint arXiv:2403.13524*, 2024.

[62] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. *arXiv preprint arXiv:2404.19702*, 2024.

[63] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

[64] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. In *NeurIPS*, 2023.

[65] Xin-Yang Zheng, Hao Pan, Yu-Xiao Guo, Xin Tong, and Yang Liu. Mvd: Efficient multiview 3d reconstruction for multiview diffusion. *arXiv preprint arXiv:2402.14253*, 2024.

 https://doi.org/10.52202/079017-3873

A Appendix

A.1 Ablation Studies

Explicit triplane latent. Unlike typical approaches like Michelangelo [64] which employs VAE to encode inputs into a 1D implicit latent space, our D3D-VAE compresses high-resolution point clouds into an explicit triplane latent representation. We conduct comparative experiments on the Objaverse [7] dataset, and the 3D models for validation are not included in the training set. Figure 7 and Table 3 illustrate the comparison of the reconstruction results of VAE between these two approaches, demonstrating that our adopted explicit triplane representation is more capable of recovering high-frequency geometric details.

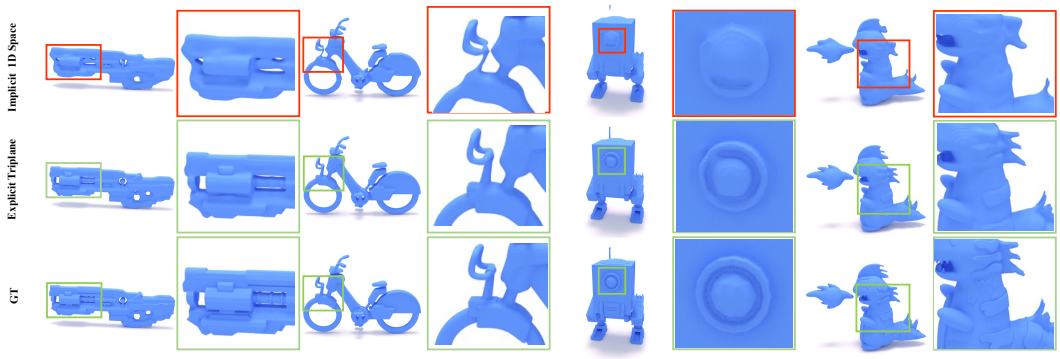

Figure 7: Qualitative comparisons of reconstruction with different latent representation.

Table 3: Quantitative comparisons of reconstruction with different latent representations on Objaverse evaluation set.

Methods	Chamfer Distance ↓	Volume IoU ↑	F-Score ↑
Implicit 1D Space (Shape2VecSet)	0.0057	0.8794	0.9416
Explicit Triplane (Ours)	**0.0042**	**0.9409**	**0.9835**

Semi-continuous surface sampling strategy. We adopt a semi-continuous surface sampling strategy during the training of D3D-VAE to alleviate the optimization difficulty caused by abrupt changes in occupancy near the object surface. To evaluate the effectiveness of this strategy, we train D3D-VAE with and without this sampling strategy separately and compare the reconstructed results. As shown in Figure 8 and Table 4, it can be observed that the reconstruction performance is unsatisfactory when directly training with the original occupancy in some thin structures, but is improved when the semi-continuous sampling strategy is utilized.

Table 4: Quantitative results of semi-continuous surface sampling strategy.

Methods	Chamfer Distance ↓	Volume IoU ↑	F-Score ↑
w/o semi-continuous sampling	0.0060	0.8723	0.9192
w/ semi-continuous sampling	**0.0057**	**0.8794**	**0.9416**

2D U-Net vs D3D-DiT. To demonstrate the superiority of our D3D-DiT network architecture, we conduct experiments to compare it with 2D U-Net. We train diffusion models on the roll-out triplane latent representation using network architectures of SD 1.5 [46] and SD 2.1, respectively. Figure 9 illustrates the qualitative comparisons using the conditional images generated by Hunyuan-DiT [24]. It can be observed that neither SD 1.5 or SD 2.1 is able to produce satisfactory meshes, while our D3D-DiT, due to its powerful scalability and generalization, is capable of generating high-quality 3D shapes that align with the content of the conditional images.

https://doi.org/10.52202/079017-3873

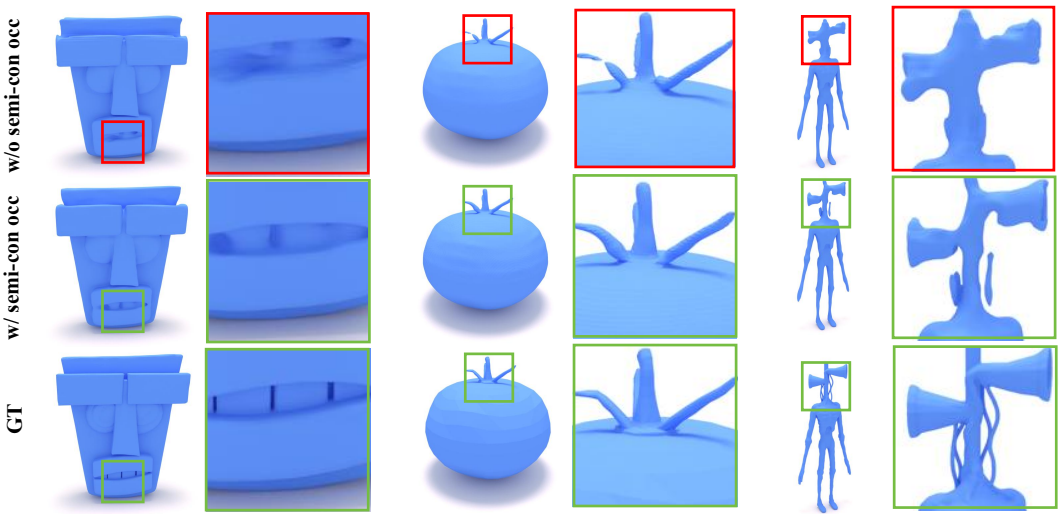

Figure 8: Ablation study for the semi-continuous surface sampling strategy.

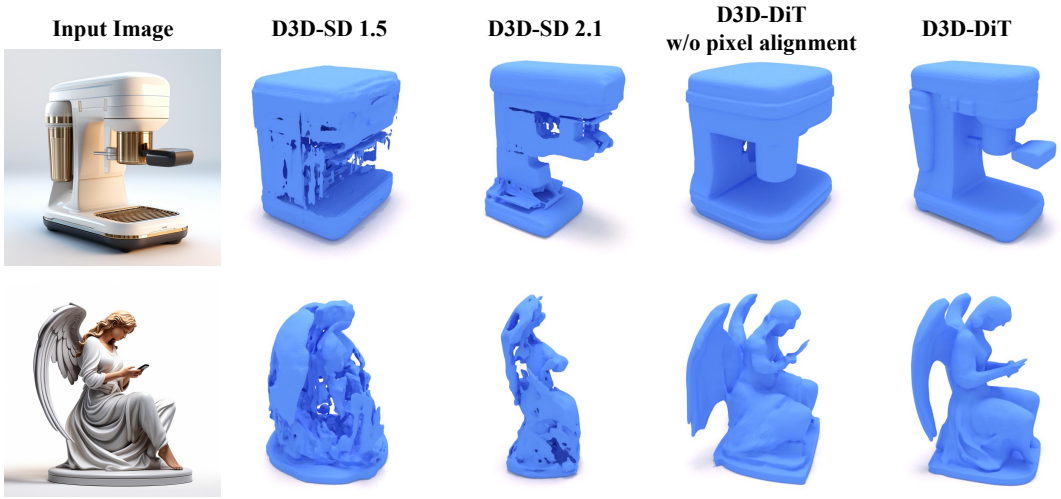

Figure 9: Qualitative comparisons of diffusion models with different network architectures.

Effectiveness of the pixel-level alignment module. We perform ablation experiments to validate the effectiveness of the pixel-level alignment module used in our D3D-DiT. As illustrated in Figure 9, D3D-DiT can still generate meshes of relatively high quality without this module. However, it does not align well with the conditional images, such as the external structure of the coffee machine and the wings of the statue. By injecting the pixel-level information into each DiT block through the pixel-level alignment module, the produced meshes can also maintain consistency with the conditional images in terms of details.

A.2 More Visualizations

We present more visualizations in Figure 10.

A.3 Broader Impacts

Like other creative tools, our project is susceptible to misuse, such as improper or harmful use of the generated characters. To address these risks, it is crucial to establish and implement ethical guidelines and content moderation policies.

https://doi.org/10.52202/079017-3873

Figure 10: More Visualizations.

https://doi.org/10.52202/079017-3873

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We explicitly delineate the contributions of our approach in both the abstract and the introduction in Section. 1 of our paper. The experimental results, which substantiate our claims, are comprehensively detailed and demonstrate the efficacy of our methods. For an in-depth analysis of the experimental outcomes, refer to Section. 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We mention in Section 5 that the paper discusses the limitations of the work performed by the authors.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

https://doi.org/10.52202/079017-3873

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We do not contain the theoretical result.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides detailed descriptions of the experimental setup, including the algorithms used, parameter settings, datasets, and evaluation metrics. Additionally, it outlines the procedures followed to obtain the results, ensuring that other researchers can replicate the experiments and verify the findings. We mention this in Section. 4

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: While a portion of the data used in this research comprises private assets with significant commercial value, releasing this information would also violate the research contract signed by the authors.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the training and testing details in Section. 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Repeated training runs of large transformer models strain available GPU resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

https://doi.org/10.52202/079017-3873

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have carefully reviewed the details provided in the paper, and it includes comprehensive information on the type of computing workers, memory, and time of execution required for each experiment. We mention this in Section. 4

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have carefully reviewed the NeurIPS Code of Ethics, and we confirm that our research adheres to all outlined ethical guidelines, including considerations for fairness, transparency, and respect for all participants involved.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses both potential positive societal impacts, such as advancements in technology and potential benefits to various industries, and negative societal impacts, including ethical considerations and the environmental impact of increased computational resource usage. We mention this in Section A.3.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This question is not applicable because the paper does not involve the release of data or models that pose a high risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The creators or original owners of all assets used in the paper are properly credited. The licenses and terms of use for these assets are explicitly mentioned and have been respected in accordance with their respective requirements.

https://doi.org/10.52202/079017-3873

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper includes comprehensive documentation for all new assets introduced. This documentation is provided alongside the assets, ensuring that users have access to detailed descriptions, usage guidelines, and any necessary supporting information.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: The paper does not include the full text of instructions given to participants, screenshots, or details about compensation.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [No]

Justification: We request several volunteers to participate in a simple anonymous questionnaire for user study, with no potential risks involved for the volunteers.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-3873