GAUSSIANANYTHING: INTERACTIVE POINT CLOUD LATENT DIFFUSION FOR 3D GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

While 3D content generation has advanced significantly, existing methods still face challenges with input formats, latent space design, and output representations. This paper introduces a novel 3D generation framework that addresses these challenges, offering scalable, high-quality 3D generation with an interactive *Point Cloud-structured Latent* space. Our framework employs a Variational Autoencoder (VAE) with multi-view posed RGB-D(epth)-N(ormal) renderings as input, using a unique latent space design that preserves 3D shape information, and incorporates a cascaded latent diffusion model for improved shape-texture disentanglement. The proposed method, GAUSSIANANYTHING, supports multi-modal conditional 3D generation, allowing for point cloud, caption, and single image inputs. Notably, the newly proposed latent space naturally enables geometry-texture disentanglement, thus allowing 3D-aware editing. Experimental results demonstrate the effectiveness of our approach on multiple datasets, outperforming existing native 3D methods in both text- and image-conditioned 3D generation.

1 INTRODUCTION

3D content generation holds great potential for transforming the virtual reality, film, and gaming industries. Current approaches typically follow one of two paths: either a 2D-lifting method or the design of native 3D diffusion models. While the 2D-lifting approach (Shi et al., 2023b; Liu et al., 2023b) benefits from leveraging 2D diffusion model priors, it is often hindered by expensive optimization, the Janus problem, and inconsistencies between views. In contrast, native 3D diffusion models (Jun & Nichol, 2023; Lan et al., 2024; Zhang et al., 2024) are trained from scratch for 3D generation, offering improved generality, efficiency, and control.

Despite the progress in native 3D diffusion models, several design challenges still persist: **(1)** *Input format to the 3D VAE.* Most methods (Zhang et al., 2024; Li et al., 2024) directly adopt point cloud as input. However, it fails to encode the high-frequency details from textures. Besides, this limits the available training dataset to artist-created 3D assets, which are challenging to collect on a large scale. LN3Diff (Lan et al., 2024) adopt multi-view images as input. Though straightforward, it lacks direct 3D information input and cannot comprehensively encode the given object. **(2)** *3D latent space structure.* Since 3D objects are diverse in geometry, color, and size, most 3D VAE models adopt the permutation-invariant *set latent* (Zhang et al., 2023a; Sajjadi et al., 2022; Zhang et al., 2024) to encode incoming 3D objects. Though flexible, this design lacks the image-latent correspondence as in Stable Diffusion VAE (Rombach et al., 2022), where the VAE latent code can directly serve as the proxy for editing input image (Mou et al., 2023b;a). Other methods adopt latent tri-plane (Wu et al., 2024; Lan et al., 2024) as the 3D latent representation. However, the latent tri-plane is still unsuitable for interactive 3D editing as changes in one plane may not map to the exact part of the objects that need editing. **(3)** *Choice of 3D output representations.* Existing solutions either output texture-less SDF (Wu et al., 2024; Zhang et al., 2024), which requires additional shading model for post-processing; or volumetric tri-plane (Lan et al., 2024), which struggles with high-resolution rendering due to extensive memory required by volumetric rendering (Mildenhall et al., 2020).

In this study, we propose a novel 3D generation framework that resolves the problems above and enables scalable, high-quality 3D generation with an interactive *Point Cloud-structured Latent* space. The resulting method, dubbed GAUSSIANANYTHING, supports multi-modal conditional 3D generation, including point cloud, caption, and image. Specifically, we propose a 3D VAE that adopts multi-view posed RGB-D(epth)-N(ormal) renderings as the input, which are easy to render and contain comprehensive 3D attributes corresponding to the input 3D object. The information of each input view is channel-wise concatenated and efficiently encoded with the scene representation transformer (Sajjadi et al., 2022), yielding a *set latent* that compactly encodes the given 3D input. Instead of directly applying it for diffusion learning (Zhang et al., 2024; Li et al., 2024), our novel design concretizes the unordered tokens into the shape of the 3D input. Specifically, this is achieved by cross-attending (Huang et al., 2024b) the *set latent* via a sparse point cloud sampled from the input 3D shape, as visualized in Fig. 1. The resulting point-cloud structured latent space significantly facilitate shape-texture disentanglement and 3D editing. Afterward, a DiT-based 3D decoder (Peebles & Xie, 2023; Lan et al., 2024) gradually decodes and upsamples the latent point cloud into a set of dense surfel Gaussians (Huang et al., 2024a), which are rasterized to high-resolution renderings to supervise 3D VAE training.

After the 3D VAE is trained, we conduct cascaded latent diffusion modeling on the latent space through flow matching (Albergo et al., 2023; Lipman et al., 2023; Liu et al., 2023c) using the DiT (Peebles & Xie, 2023) framework. To encourage better shape-texture disentanglement, a point cloud diffusion model is first trained to carve the overall layout of the input shape. Then, a point-cloud feature diffusion model is cascaded to output the corresponding feature conditioned on the generated point cloud. The generated featured point cloud is then decoded into surfel Gaussians via pre-trained VAE for downstream applications.

In summary, we contribute a comprehensive 3D generation framework with a point cloud-structured 3D latent space. The redesigned 3D VAE efficiently encodes the 3D input into an interactive latent space, which is further decoded into high-quality surfel Gaussians. The diffusion models trained on the compressed latent space have shown superior performance in text-conditioned 3D generation and editing, as well as impressive image-conditioned 3D generation on general real world data.

2 Related Work

3D Generation via 2D Diffusion Models. The success of 2D diffusion models (Song et al., 2021; Ho et al., 2020) has inspired their application to 3D generation. Score distillation sampling (Poole et al., 2022; Wang et al., 2023) distills 3D from a 2D diffusion model, but faces challenges like expensive optimization, mode collapse, and the Janus problem. More recent methods propose learning the 3D via a two-stage pipeline: multi-view images generation (Shi et al., 2023b; Long et al., 2024; Shi et al., 2023a) and feed-forward 3D reconstruction (Hong et al., 2024b; Xu et al., 2024a; Tang et al., 2024). Though promising results have been achieved, their performance is bounded by the multi-view generation results, which usually violate view consistency (Liu et al., 2023b) and fails to scale up to higher resolution (Shi et al., 2023a). Moreover, this two-stage pipeline limits the 3D editing capability due to the lack of a 3D-aware latent space.

Native 3D Diffusion Models. Native 3D diffusion models (Zhang et al., 2023a; Zeng et al., 2022; Zhang et al., 2024; Lan et al., 2024; Li et al., 2024) are recently proposed to achieve high-quality, efficient and scalable 3D generation. A native 3D diffusion pipeline involves a two-stage training process: encoding 3D objects into the VAE latent space (Kingma & Welling, 2013; Kosiorek et al., 2021), and latent diffusion model on the corresponding latent codes. Though straightforward, existing methods differ in VAE input formats, latent space structure and output 3D representations. While most methods adopt point alone as the VAE input (Zhang et al., 2023a; 2024; Li et al., 2024), our proposed method encodes a hybrid 3D information through convolutional encoder. Moreover, comparing to the latent set (Zhang et al., 2023a; Sajjadi et al., 2022) representation, our proposed method adopts a point cloud-structured latent space, which can be directly used for interactive 3D editing. Besides, rather than producing textureless SDF, our method directly decodes the 3D latent codes into high-quality surfel Gaussians (Huang et al., 2024a), which can be directly used for efficient rendering.

Point-based Shape Representation and Rendering. The proliferation of 3D scanners and RGB-D cameras makes the capture and processing of 3D point clouds commonplace (Gross & Pfister, 2011). In the era of deep learning, learning-based methods are emerging for point set processing (Qi et al., 2016; Zhao et al., 2021), up-sampling (Yu et al., 2018), shape representation (Genova et al., 2020; Lan et al., 2023b), and rendering (Pfister et al., 2000; Yifan et al., 2019; Lassner & Zollhöfer, 2021; Xu et al., 2022; Kerbl et al., 2023). Moreover, given its affinity for modern network architectures (Huang et al., 2024b; Zhao et al., 2021), more explicit nature against other 3D representations (Chan et al., 2022; Mildenhall et al., 2020; Müller et al., 2022), efficient rendering (Kerbl et al., 2023), and even high-quality surface modeling (Huang et al., 2024a), point-based 3D representations are rapidly developing towards the canonical 3D representation for learning 3D shapes. Thus, we choose (featured) point cloud as the representation for the 3D VAE latent space, and 2D Gaussians (Huang et al., 2024a) as the output 3D representations.

Feed-forward 3D Reconstruction and View Synthesis. To bypass the per-scene optimization of NeRF, researchers have proposed learning a prior model through image-based rendering (Wang et al., 2021; Yu et al., 2021). However, these methods are primarily designed for view synthesis and lack explicit 3D representations. Sajjadi et al. (2022; 2023) propose Scene representation transformer (SRT) to process RGB images with Vision Transformer (Dosovitskiy et al., 2021) and infers a "set-latent scene representation". Though benefiting from the flexible design, its geometry-free paradigm also fails to generate explicit 3D outputs. Recently, LRM-line of work (Hong et al., 2024b; Tang et al., 2024; Wang et al., 2024) have proposed a feed-forward framework for generalized monocular reconstruction. However, they are still regression-based models and lack the latent space designed for generative modeling and 3D editing. Besides, they are limited to 3D reconstruction only and fail to support other modalities.

3 GAUSSIANANYTHING

This section introduces our native 3D diffusion model, which learns 3D-aware diffusion prior over the novel point-cloud structured latent space through a dedicated 3D VAE. The goal of training is to learn

1. An encoder $\mathcal{E}_{\boldsymbol{\phi}}$ that maps a set of posed RGB-D-N images $\mathcal{R} = \{R_i, ..., R_V\}$, corresponding to the given 3D object to a point-cloud structured latent $\mathbf{z} = [\mathbf{z}_x \oplus \mathbf{z}_h]$;
2. A conditional cascaded transformer denoiser $\boldsymbol{\epsilon}^h_\Theta(\mathbf{z}_{h,t}, \mathbf{z}_{x,0}, t, c) \circ \boldsymbol{\epsilon}^x_\Theta(\mathbf{z}_{x,t}, t, c)$ to denoise the noisy latent code $\mathbf{z}_t$ given diffusion time step t and condition prompt c;
3. A decoder $\mathcal{D}_{\boldsymbol{\psi}}$ (including a Transformer $\mathcal{D}_T$ and a cascaded attention-base Upsampler $\mathcal{D}_U$) to map $\mathbf{z}_0$ to the surfel Gaussian $\widetilde{\mathcal{G}}$ corresponding to the input object. Moreover, our attention-based decoding of dense surfel Gaussian also provides a novel way for efficient Gaussian prediction

Beyond the advantages shared by existing 3D LDM (Zhang et al., 2024; Lan et al., 2024), our design offers a flexible point-cloud structured latent space and enables interactive 3D editing.

In the following subsections, we first discuss the proposed 3D VAE with a detailed framework design in Sec 3.1. Based on that, we introduce the cascaded conditional 3D diffusion stage in Sec. 3.2. The method overview is shown in Fig. 1.

3.1 POINT-CLOUD STRUCTURED 3D VAE

Unlike image and video, the 3D domain is un-uniform and represented differently for different purposes. Thus, how to encode 3D objects into the latent space for diffusion learning plays a crucial role in the 3D generation performance. This challenge is two-fold: what 3D representations to encode, and what network architecture to process the input.

Versatile 3D Input. Instead of using dense point cloud (Zhang et al., 2024; Li et al., 2024), we adopt multi-view posed RGB-D(epth)-N(ormal) images as input, which encode the 3D input more comprehensively and can be efficiently processed by well-established network architectures (Sajjadi et al., 2022; Wu et al., 2023a) in a flexible manner. Specifically, the input is a set of multi-view renderings $\mathcal{R}$ of a 3D object, where each rendering within the set $R = (I, \Delta, N, \pi)$ contains thorough 3D attributes that depict the underlying 3D object from the given viewpoint: the rendered RGB image $I \in \mathbb{R}^{H\times W\times 3}$, depth map $\Delta \in \mathbb{R}^{H\times W}$, normal map $N \in \mathbb{R}^{H\times W\times 3}$, and the corresponding camera pose π.

To unify these 3D attributes in the same format, we further process the camera π into Plücker coordinates (Sitzmann et al., 2021) $\mathbf{p}_i = (\mathbf{o} \times \mathbf{d}_{u,v}, \mathbf{d}_{u,v}) \in \mathbb{R}^6$, where $\mathbf{o}_i \in \mathbb{R}^3$ is the camera

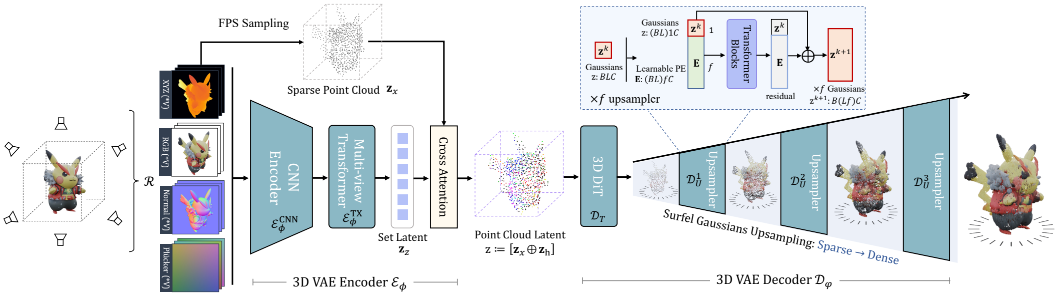

FPS Sampling
Sparse Point Cloud $\mathbf{z}_x$
$\mathcal{R}$
XYZ (*V)
RGB (*V)
Normal (*V)
Plücker (*V)
CNN Encoder $\mathcal{E}_\phi^{\text{CNN}}$
Multi-view Transformer $\mathcal{E}_\phi^{\text{TX}}$
Set Latent $\mathbf{z}_z$
Cross Attention
Point Cloud Latent $\mathrm{z} := [\mathbf{z}_x \oplus \mathbf{z}_\mathrm{h}]$
3D VAE Encoder $\mathcal{E}_\phi$
3D DiT $\mathcal{D}_T$
$\mathbf{z}^k$
Gaussians z: BLC
Gaussians z: $(BL)1C$
Learnable PE E: $(BL)fC$
E
1
f
Transformer Blocks
residual
$\mathbf{z}^{k+1}$
$\times f$ Gaussians z^{k+1}: $B(Lf)C$
$\times f$ upsampler
$\mathcal{D}_U^1$ Upsampler
$\mathcal{D}_U^2$ Upsampler
$\mathcal{D}_U^3$ Upsampler
Surfel Gaussians Upsampling: Sparse → Dense
3D VAE Decoder $\mathcal{D}_\varphi$

Figure 1: **Pipeline of the 3D VAE of GaussianAnything.** In the 3D latent space learning stage, our proposed 3D VAE $\mathcal{E}_\phi$ encodes $V-$views of posed RGB-D(epth)-N(ormal) renderings $\mathcal{R}$ into a point-cloud structured latent space. This is achieved by first processing the multi-view inputs into the un-structured *set latent*, which is further projected onto the 3D manifold through a cross attention block, yielding the point-cloud structured latent code $\mathbf{z}$. The structured 3D latent is further decoded by a 3D-aware DiT transformer, giving the coarse Gaussian prediction. For high-quality rendering, the base Gaussian is further up-sampled by a series of cascaded upsampler $\mathcal{D}_U^k$ towards a dense Gaussian for high-resolution rasterization. The 3D VAE training objective is detailed in Eq. (5).

origin, $\mathbf{d}_{u,v} \in \mathbb{R}^3$ is the normalized ray direction, and $\times$ denotes the cross product. Thus, the Plücker embedding of a given camera π can be expressed as $\mathbf{P} \in \mathbb{R}^{H \times W \times 6}$. Besides, following MCC (Wu et al., 2023a), we use π to unproject the depth map into their 3D positions $X \in \mathbb{R}^{H \times W \times 3}$. The resulting information is channel-wise concatenated, giving $\tilde{R} = [I \oplus X \oplus N \oplus \mathbf{P}] \in \mathbb{R}^{H \times W \times (3+3+3+6=15)}$.

Transformer-based 3D Encoding. Given the 3D renderings $\mathcal{R}$, encoding them into a 3D latent space remains a significant challenge. Independently processing each input rendering $\tilde{R}$ with existing network architecture (Wu et al., 2023a; Dosovitskiy et al., 2021) overlooks the information from other views, leading to 3D inconsistency and content drift across views (Liu et al., 2023b).

Existing multi-view generation alleviates this issue by injecting 3D attention (Shi et al., 2023b; Tang et al., 2024; Shi et al., 2023a) into the U-Net architecture. Inspired by its effectiveness, here we directly adopt Scene Representation Transformer (SRT)-like encoder (Sajjadi et al., 2022; 2023) to process the multi-view inputs, which fully adopts 3D attention transformer block for the 3D representation learning. Specifically, the encoder first down-samples the multi-view inputs via a shared CNN backbone, and then processes the aggregated multi-view tokens through the transformer encoder (Dosovitskiy et al., 2021):

$$\mathbf{z}_z = \mathcal{E}_\phi^{\text{TX}}(\mathcal{E}_\phi^{\text{CNN}}(\{\tilde{R}\})), \tag{1}$$

where $\mathbf{z}_z$ is the *set latent* corresponding to the 3D input. This can be seen as the full-attention version of the existing 3D attention-augmented architecture. The resulting latent codes $\mathbf{z}_z$ fully capture the intact 3D information corresponding to the input. Compared to existing work that adopts point clouds only as input (Zhang et al., 2024; Li et al., 2024), our proposed solution supports more 3D properties as input in a flexible way. In addition, attention operations can be well optimized in modern GPU architecture (Dao et al., 2022; Dao, 2024).

Point Cloud-structured Latent Space. Though $\mathbf{z}_z$ fully captures the given 3D input, it is not ideal to serve as a latent space for our task due to the following limitations: 1) The latent space is cumbersome to perform diffusion learning. Specifically, $\mathbf{z}_z$ has a shape of $V \times (H/f) \times (W/f) \times C$, where V is the number of input views, H, W is the input resolution and f is the down-sampling factor of the CNN backbone. Given $V = 8$, $f = 8$, and $H = W = 512$, the resulting latent codes will have the shape of $32768 \times C$. This latent space incurs a high computation cost for multi-view attention (Shi et al., 2023b). 2) The multi-view features $\mathbf{z}_z$ are not native 3D representations and naturally suffer from view inconsistency (Liu et al., 2023b) even with enough compute available (Shi et al., 2023a). 3) Since $\mathbf{z}_z$ is an un-structured *set* (Lee et al., 2019) 3D latent space (Zhang et al., 2023a; 2024), it also sacrifices an explicit, editable latent space (Mou et al., 2023a) for flexibility.

Here, we resolve these issues by proposing a point cloud-structured latent space. Specifically, we project the un-structured features $\mathbf{z}_z$ onto the sparse 3D point cloud of the input 3D shape through the cross attention layer:

$$\mathbf{z}_h := \text{CrossAttn}(\text{PE}(\mathbf{z}_x), \mathbf{z}_z, \mathbf{z}_z), \tag{2}$$

where CrossAttn(Q, K, V) denotes a cross attention block with query Q, key K, and value V. $\mathbf{z}_x \in \mathbb{R}^{3\times N}$ is a sparse point cloud sampled from the surface of the 3D input with Farthest Point Sampling (FPS) (Qi et al., 2017), and PE denotes positional embedding (Tancik et al., 2020). Intuitively, we define a *read* cross attention block (Huang et al., 2024b) that cross attends information from unstructured representation $\mathbf{z}_z$ into the point-cloud structured feature $\mathbf{z}_h \in \mathbb{R}^{C_h\times N}$, with $C_h \ll C$. In this way, we obtain the point-cloud structured latent code $\mathbf{z} = [\mathbf{z}_x \oplus \mathbf{z}_h] \in \mathbb{R}^{(3+C_h)\times N}$ for diffusion learning.

High-quality 3D Gaussian Decoding. Given the point cloud-structured latent codes, how to decode them into high-quality 3D representation for supervision remains challenging. Though dense point cloud (Huang et al., 2024b) is a straightforward solution, it fails to depict high-quality 3D structure with limited point quantity. Here, we resort to surfel Gaussian (Huang et al., 2024a), an augmented point-based 3D representation that supports high-fidelity 3D surface modeling and efficient rendering. Specifically, our decoder first decodes the input through the 3D-DiT blocks (Peebles & Xie, 2023; Lan et al., 2024), which has shown superior performance against traditional transformer layer:

$$\tilde{\mathbf{z}} \coloneqq \mathcal{D}_T(\text{MLP}(\mathbf{z})), \tag{3}$$

where an MLP layer first projects the input latent to the corresponding dimension, and $\mathcal{D}_T$ is the DiT transformer. Since dense Gaussians are preferred for high-quality splatting (Kerbl et al., 2023), we gradually upsample the latent features through transformer blocks. Specifically, given a learnable embedding $\mathbf{z}_u \in \mathbb{R}^{f_u\times C}$ where f_u is the up-sampling ratio, we prepend it to each token in the latent sequence. Then, H layers of transformer blocks are used to model the upsampling process:

$$\mathbf{z}_i^{(k+1)} \coloneqq \mathcal{D}_U^k([\mathbf{z}_u \oplus \tilde{\mathbf{z}}_i]), \tag{4}$$

where $\mathcal{D}_U^k$ is a transformer block for predicting the $k-$th levels of details (LoD) Gaussian as shown in Fig. 1, and $\mathbf{z}_i^{(k+1)} \in \mathbb{R}^{f_u\times C}$ are the upsampled set of tokens. The overall tokens $\mathbf{z}^{(k+1)} \in \mathbb{R}^{(f_u\times N)\times C}$ after up-sampling are used to predict the 13-dim attributes of surfel Gaussians.

To achieve denser Gaussians prediction, we cascade the upsampling transformer defined in Eq. (4) for K times, giving the final Upsampler $\mathcal{D}_U$ for high-quality Gaussian output. Note that our solution outputs a set of Gaussians that are uniformly distributed on the 3D object surface with near 100% Gaussian utilization ratio. Existing pixel-aligned Gaussian prediction models (Tang et al., 2024; Yinghao et al., 2024; Szymanowicz et al., 2023), however, usually waste 50% Gaussians due to view overlaps and empty background color. Besides, our intermediate Gaussians output naturally serves as K LoD (Takikawa et al., 2021), which can be used in different scenarios to balance the rendering speed and quality.

Training. Our 3D VAE model is end-to-end optimized across both input views and randomly chosen views, minimizing image reconstruction objectives between the splatting renderings and ground-truth renderings. Besides image reconstruction loss, we also impose loss over geometry regularizations, KL constraints, and adversarial loss:

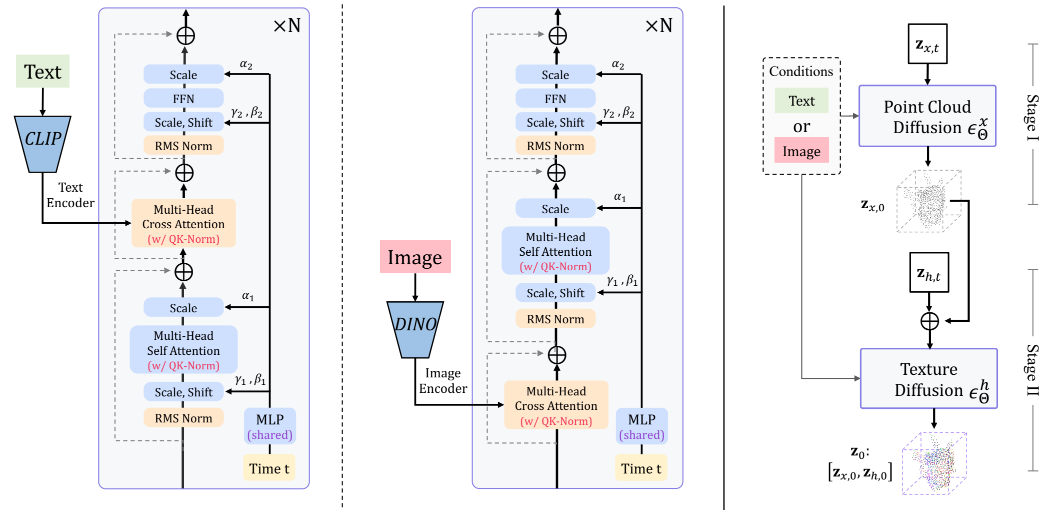
$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\psi}) = \mathcal{L}_{\text{render}} + \mathcal{L}_{\text{geo}} + \lambda_{\text{kl}}\mathcal{L}_{\text{KL}} + \lambda_{\text{GAN}}\mathcal{L}_{\text{GAN}}, \tag{5}$$

where $\mathcal{L}_{\text{render}}$ is a mixture of the L_1 and VGG loss (Zhang et al., 2018), $\mathcal{L}_{\text{geo}}$ improves geometry reconstruction (Huang et al., 2024a), $\mathcal{L}_{\text{KL}}$ is the *KL-reg* loss (Kingma & Welling, 2013; Rombach et al., 2022) to regularize a structured latent space, and $\mathcal{L}_{\text{GAN}}$ improves perceptual fidelity. All loss terms except $\mathcal{L}_{\text{KL}}$ are applied over a randomly chosen LoD in each iteration, and the $\mathcal{L}_{\text{render}}$ is applied to both input-view and randomly sampled novel-view images. For details of geometry loss $\mathcal{L}_{\text{geo}}$, please refer to Sec. A.1.

3.2 Cascaded 3D Generation with Flow Matching

After training the point-cloud structured 3D VAE, we get a dataset of D shapes paired with condition vectors (*e.g.*, caption or images), $\{(\mathbf{z}_i, c_i)\}_{i\in[D]}$, where the shape is represented by latent code $\mathbf{z}$ through the 3D VAE aforementioned. Our goal is to train a flow-matching generative model to learn a diffusion prior on top of it. Below we present how we adapt flow-based models to our case.

Cascaded Flow Matching over Symmetric Data. As detailed in Sec. A.3, flow matching involves training a neural network $\boldsymbol{\epsilon}_\Theta$ to predict the velocity v of the noisy input $\mathbf{z}_t$ with the straight-line trajectory. After training, $\boldsymbol{\epsilon}_\Theta$ can sample from a standard Normal prior $\mathcal{N}(0, I)$ by solving the reverse



(a) DiT Block (Text condition)

(b) DiT Block (Image condition)

(c) Two-stage Cascaded Diffusion

Figure 2: **Diffusion training of GaussianAnything.** Based on the point-cloud structure 3D VAE, we perform cascaded 3D diffusion learning given text (a) and image (b) conditions. We adopt DiT architecture with AdaLN-single (Chen et al., 2023) and QK-Norm (Dehghani et al., 2023; Esser et al., 2021). For both condition modality, we send in the conditional feature with cross attention block, but at different positions. The 3D generation is achieved in two stages (c), where a point cloud diffusion model first generates the 3D layout $\mathbf{z}_{x,0}$, and a texture diffusion model further generates the corresponding point-cloud features $\mathbf{z}_{h,0}$. The generated latent code $\mathbf{z}_0$ is decoded into the final 3D object with the pre-trained VAE decoder.

ODE/SDE (Karras et al., 2022). In our case, the training data point is the point-cloud structured latent code $\mathbf{z} = [\mathbf{z}_x \oplus \mathbf{z}_h] \in \mathbb{R}^{(3+C_h)\times N}$, which is symmetric and *permutation invariant* (Zeng et al., 2022; Nichol et al., 2022). Based on this property, we opt for diffusion transformer (Peebles & Xie, 2023) without positional encoding as the $\boldsymbol{\epsilon}_\Theta$ parameterization.

Here, rather than modeling $\mathbf{z}_x$ and $\mathbf{z}_h$ jointly, we empirically found that a cascaded framework (Ho et al., 2021; Lyu et al., 2024; 2023) leads to better performance. Specifically, a conditioned sparse point cloud generative model $\boldsymbol{\epsilon}^x_\Theta$ is first trained to generate the overall structure of the given object:

$$\mathcal{L}^x_w(x_0) = -\frac{1}{2}\mathbb{E}_{t\sim\mathcal{U}(t),\epsilon\sim\mathcal{N}(0,I)}\left[w_t^{\mathrm{FM}}\lambda_t'\|\boldsymbol{\epsilon}^x_\Theta(\mathbf{z}_{x,t},t,c)-\epsilon\|^2\right] \;, \tag{6}$$

and a point cloud feature generative model $\boldsymbol{\epsilon}^h_\Theta$ is cascaded to learn the corresponding KL-regularized feature conditioned on the sparse point cloud:

$$\mathcal{L}^h_w(x_0) = -\frac{1}{2}\mathbb{E}_{t\sim\mathcal{U}(t),\epsilon\sim\mathcal{N}(0,I)}\left[w_t^{\mathrm{FM}}\lambda_t'\|\boldsymbol{\epsilon}^h_\Theta(\mathbf{z}_{h,t},\mathbf{z}_x,t,c)-\epsilon\|^2\right] \;. \tag{7}$$

The detailed cascading process is detailed Fig. 2 (c). Our proposed design enables better geometry-texture disentanglement and facilitates 3D editing over specific shape properties. For derivations of the flow matching training objective, please refer to Sec. A.3 for more details.

Conditioning Mechanism. Compared to LRM (Hong et al., 2024b; Tang et al., 2024) line of work which intrinsically relies on image(s) as the input, our native diffusion-based method enables more flexible 3D generation from diverse conditions. As shown in Fig. 2 (a-b), for the text-conditioned model, we adopt CLIP (Radford et al., 2021) to extract *penultimate* tokens as the condition embeddings; and for the image conditioned model, we use DINOv2 (Oquab et al., 2023) to extract global and patch features. All conditions are injected into the DiT architecture through a pre-norm (Xiong et al., 2020) cross-attention block following the common practice Zhang et al. (2024). All the models are trained with Classifier-free Guidance (CFG) (Ho, 2021) by randomly dropping the conditions with a probability of 10%.

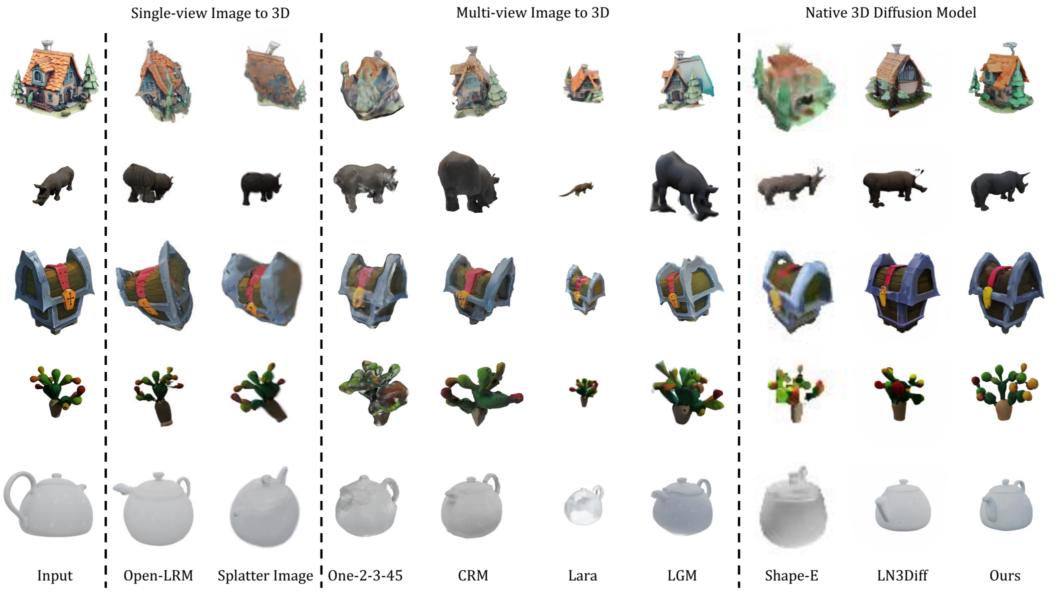


Figure 3: **Qualitative Comparison of Image-to-3D**. We showcase the novel view 3D reconstruction of all methods given a single image from unseen GSO dataset. Our proposed method achieves consistently stable performance across all cases. Note that though feed-forward 3D reconstruction methods achieve sharper texture reconstruction, these method fail to yield intact 3D predictions under challenging cases (e.g., the rhino in row 2). In contrast, our proposed native 3D diffusion model achieve consistently better performance. Better zoom in.

To cascade two diffusion models, we encode the output of stage-1 model $\boldsymbol{\epsilon}_\Theta^x$ with $\text{PE}(\mathbf{z}_x)$ as in Eq. (2), and add it to the first-layer features of $\boldsymbol{\epsilon}_\Theta^h$. This guarantees that generated features are paired with the input sparse point cloud structure.

4 Experiments

Datasets. To train our 3D VAE, we use the renderings provided by G-Objaverse (Qiu et al., 2023; Deitke et al., 2023b) and choose a high-quality subset with around $176K$ 3D instances, where each consists of 40 random views with RGB, normal, depth map and camera pose. For text-conditioned diffusion training, we use the caption provided by Cap3D (Luo et al., 2023; 2024) and 3DTopia Hong et al. (2024a). For image-conditioned training, we randomly select an image in the dataset of the corresponding 3D instance as the condition.

Implementation Details. For 3D VAE, we choose $V = 8$ views of RGB-D-N renderings as input to guarantee a thorough coverage of the 3D object. The CNN Encoder is implemented with a similar architecture as LDM VAE (Rombach et al., 2022) with a down-sampling factor of $f = 8$, and the multi-view transformer has five layers as in RUST (Sajjadi et al., 2023). The sparse point cloud $\mathbf{z}_x$ has a size of $N \times 3$ where $N = 768$, and the corresponding featured point cloud $\mathbf{z}_h$ has a dimension of $N \times 10$. For upsampling blocks, we employ $K = 3$ blocks with $f_u^1 = 8$, $f_u^1 = 4$, and $f_u^1 = 3$, giving $73,768$ Gaussians in total. All transformer blocks follow a pre-norm design (Xiong et al., 2020). During 3D VAE training, the model is supervised by randomly chosen LoD renderings, with $\lambda_{\text{kl}} = 2e-6$, $\lambda_{\text{d}} = 1000$, $\lambda_{\text{n}} = 0.2$, and $\lambda_{\text{GAN}} = 0.1$. We adopt batch size 64 with both input and random novel views for training. During the conditional diffusion training stage, we adopt batch size 256. All models are efficiently and stably trained with $\text{lr} = 1e-4$ on 8×A100 GPUs for $1M$ iterations with BF16 and FlashAttention (Dao, 2024) enabled. We use CFG=4 and 250 ODE steps for all sampling results.

4.1 Metrics and Baselines

Evaluating Image-to-3D Generation. We evaluate GaussianAnything on both image and text conditioned generation. Regarding image-conditioned 3D generation methods, we compare

Table 2: **Quantitative evaluation of image-conditioned 3D generation.** Here, quality of both 2D rendering and 3D shapes is evaluated. As shown below, the proposed method demonstrates strong performance across all metrics. Although multi-view images-to-3D approaches like LGM achieves better performance on the FID/KID metrics, they fall short on more advanced image quality assessment metrics such as CLIP-I, MUSIQ, and performs significantly worse in 3D shape quality. For multi-view to 3D methods, we also include the number of input views (V=#).

Method	CLIP-I↑	FID↓	KID(%)↓	MUSIQ↑	P-FID↓	P-KID(%)↓	COV(%)↑	MMD(‰)↓
OpenLRM	86.37	38.41	1.87	45.46	35.74	12.60	39.33	29.08
Splatter-Image	84.10	48.80	3.65	30.33	19.72	7.03	37.66	30.69
One-2-3-45 (V=12)	80.72	88.39	6.34	59.02	72.40	30.83	33.33	35.09
CRM (V=6)	85.76	45.53	1.93	64.10	35.21	13.19	38.83	28.91
Lara (V=4)	84.64	43.74	1.95	39.37	32.37	12.44	39.33	28.84
LGM (V=4)	87.99	19.93	0.55	54.78	40.17	19.45	50.83	22.06
LN3Diff	87.24	29.08	0.89	50.39	27.17	10.02	55.17	19.94
Ours	89.06	24.21	0.76	65.17	8.72	3.22	59.50	15.48

the proposed method with three lines of methods: *single-image to 3D methods*: OpenLRM (He & Wang, 2023; Hong et al., 2024b), Splatter Image (Szymanowicz et al., 2023), *multi-view images to 3D methods*: One-2-3-45 Liu et al. (2023a), CRM (Wang et al., 2024), Lara (Chen et al., 2024a), LGM (Tang et al., 2024), and *native 3D diffusion models*: LN3Diff-image (Lan et al., 2024).

Quantitatively, we benchmark rendering metrics with CLIP-I Radford et al. (2021), FID (Heusel et al., 2017), KID (Bińkowski et al., 2018), and MUSIQ-koniq (Ke et al., 2021; Zhou et al., 2022). For 3D quality metrics, we adopt Point cloud FID (P-FID), Point cloud KID (P-KID), Coverage Score (COV), and Minimum Matching Distance (MMD) as the metrics. Following previous works Nichol et al. (2022); Zhang et al. (2023a); Yariv et al. (2024), we adopt the pre-trained PointNet++ provided by Point-E (Nichol et al., 2022) for calculating P-FID and K-FID. Qualitatively, GSO (Downs et al., 2022; Zheng & Vedaldi, 2023) dataset is used for visually inspecting image-conditioned generation.

Evaluating Text-to-3D Generation. Regarding text-conditioned 3D generation methods, we compare against Point-E (Nichol et al., 2022), Shape-E (Jun & Nichol, 2023), 3DTopia (Hong et al., 2024a), and LN3Diff-text (Lan et al., 2024). CLIP score (Radford et al., 2021) is reported following the previous works (Lan et al., 2024; Hong et al., 2024a), with aesthetic scores MUSIQ-AVA (Ke et al., 2021) and Q-Align (Wu et al., 2023b) also included.

Table 1: **Quantitative Evaluation on Text-to-3D.** The proposed method outperforms existing methods on both CLIP scores and aesthetic scores over competitive alternatives.

Method	ViT-B/32↑	ViT-L/14↑	MUSIQ-AVA ↑	Q-Align ↑
Point-E	26.35	21.40	4.08	1.21
Shape-E	27.84	25.84	3.69	1.56
LN3Diff	29.12	27.80	4.16	2.22
3DTopia	30.10	28.11	3.31	1.42
Ours	**31.80**	**29.38**	**4.99**	**3.13**

4.2 Evaluation

Image-to-3D Generation. Our proposed framework enables 3D generation given single-view image conditions, leveraging the architecture detailed in Fig. 2 (b). Following current method (Chen et al., 2024a; Tang et al., 2024), we qualitatively benchmark our method in Fig. 3 over the single-view 3D reconstruction task on the unseen images from the GSO dataset. Our proposed framework is robust to inputs with complicated structures (row 1,3,4) and self-occlusion (row 2,5), yielding consistently intact 3D reconstruction. Besides, our generative-based method shows a more natural back-view reconstruction, as opposed to regression-based methods that are commonly blurry on uncertain areas.

Quantitatively, we showcase the evaluation in Tab. 2. As can be seen, our proposed method achieves state-of-the-art performance over CLIP-I and all 3D metrics, with competitive results over conventional 2D rendering metrics FID/KID. Note that LGM leverages pre-trained MVDream (Shi et al., 2023b) as the first-stage generation, and then maps the generated 4 views to pixel-aligned 3D Gaussians. This cascaded pipeline achieves better visual quality, but prone to yield distorted 3D geometry, as visualized in Fig. 3.

Text-to-3D Generation. We demonstrate the text-to-3D generation performance in Fig. 4 and Tab. 1. The diffusion model trained on GAUSSIANANYTHING's latent space has demonstrated high-quality text-to-3D generation of generic 3D objects, yielding superior performance in terms of object structure, textures, and surface normals. Quantitatively, our proposed method achieves better text-3D alignment against competitive baselines.

3D-aware Editing. Compared to existing methods that use unstructured tokens for 3D diffusion learning (Jun & Nichol, 2023), our proposed point-cloud structured latent space naturally facilitates geometry-texture disentanglement and allows for interactive 3D editing. As visualized in Fig. 5, given the text-conditioned generated point cloud $\mathbf{z}_0$ by $\boldsymbol{\epsilon}_\Theta^x$, we sample the final 3D objects with $\boldsymbol{\epsilon}_\Theta^h$ with a different random seed. As can be seen, the generated 3D objects maintain a consistent structure layout while yielding diverse textures. Besides, by directly manipulating the conditioned point cloud $\mathbf{z}_{x,0}$, our proposed method enables interactive 3D editing, as in 2D models (Pan et al., 2023; Mou et al., 2023b). This functionality greatly facilitates the 3D content creation process for artists and opens up new possibilities for 3D editing with diffusion models.

4.3 Ablation Study and Analysis

Table 3: **Ablation of 3D VAE Design.** We ablate the design of our 3D VAE. Input-side, leveraging multi-view RGB-D-N renderings shows superior performance against dense point cloud. Besides, adding Gaussian up-sampling modules leads to consistent performance gain.

Design	LPIPS@100K
Dense PCD as Input	0.174
Multi-view RGB-D as Input	0.163
+ Normal Map	0.157
+ Gaussian SR Module	0.095
+ 3 × Gaussian SR Module	**0.067**

Table 4: **Gaussian Utilization Ratio.** We compare the effective Gaussians (opacity > 0.005) used during splatting here. Pixel-aligned Gaussian prediction methods waste a large portion of Gaussians when representing 3D object due to white background and multi-view overlap, while our proposed Gaussian predictions yields more compact reconstruction results.

Method	High-opacity Gaussians (%)
Splatter Image	17.14
LGM	52.63
Ours	**96.84**

3D VAE Design. In Tab. 3, we benchmark each component of our 3D VAE architecture over a subset of Objaverse with $50K$ instances and record the LPIPS at $100K$ iterations. As shown in Tab. 3, our input design performs better against dense $(16,384)$ colored point cloud (Zhang et al., 2024), and the reconstruction quality consistently improves by including normal map as input and cascading more Gaussian upsampling blocks.

Gaussian Utilization Ratio. Besides, we showcase a high Gaussian utilization ratio of our proposed method. Specifically, we calculate the ratio of Gaussians with an opacity greater than 0.005 as *effective* Gaussians, as they contribute well to the final rendering. We calculate the statistics over

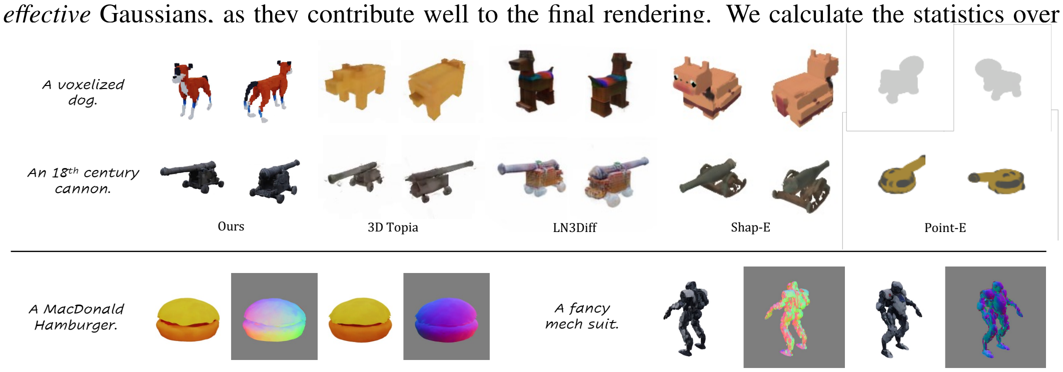


Figure 4: **Qualitative Comparison of Text-to-3D**. We present text-conditioned 3D objects generated by GAUSSIANANYTHING, displaying two views of each sample. The top section compares our results with baseline methods, while the bottom shows additional samples from our method along with their geometry maps. Our approach consistently yields better quality in terms of geometry, texture, and text-3D alignment.

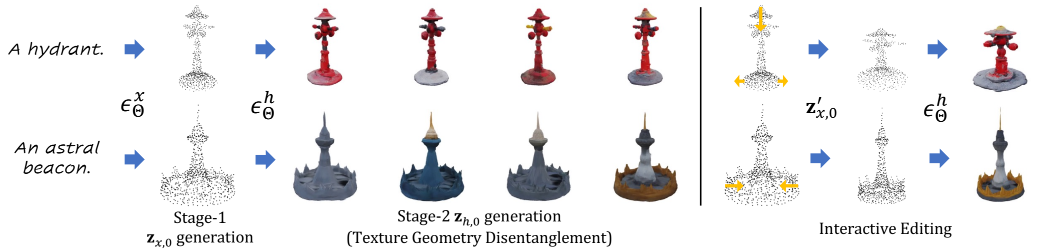


Figure 5: **3D editing.** Given two text prompts, we generate the corresponding point cloud $\mathbf{z}_{0,x}$ with stage-1 diffusion model with $\boldsymbol{\epsilon}_\Theta^x$, and the corresponding point cloud features $\mathbf{z}_{0,h}$ can be further generated with $\boldsymbol{\epsilon}_\Theta^h$. As can be seen, the samples from stage-2 are consistent in overall 3D structures but with diverse textures. Thanks to the proposed Point Cloud-structured Latent space, our method supports interactive 3D structure editing. This is achieved by first modifying the stage-1 point cloud $\mathbf{z}_{0,x} \rightarrow \mathbf{z}'_{0,x}$, and then regenerate the 3D object with the same Gaussian noise.

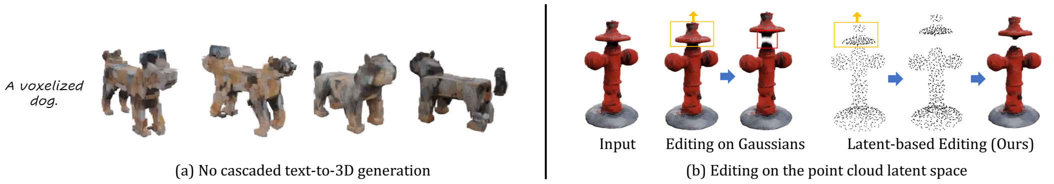

A voxelized dog.
(a) No cascaded text-to-3D generation
Input
Editing on Gaussians
Latent-based Editing (Ours)
(b) Editing on the point cloud latent space

Figure 6: **Qualitative ablation of Cascaded diffusion and latent space editing.** We first show the effectiveness of our two-stage cascaded diffusion framework in (a). Compared to Fig. 4, the single-stage 3D diffusion yields worse texture details and 3D structure intactness. In (b), we disjoint the hydrant cover to demonstrate that our latent point cloud editing yields less 3D artifacts compared to direct 3D Gaussians editing.

$50K$ 3D instances. As shown in Tab. 4, our proposed Gaussian prediction framework achieves a much higher utilization ratio. On the contrary, pixel-aligned Gaussian prediction models waste a noticeable portion of Gaussians on the overlapping views and white backgrounds.

Effectiveness of Cascaded 3D Diffusion. We qualitatively ablate the cascaded 3D diffusion choice in Fig. 6 (a), where a single text-conditioned DiT is trained to synthesize the 3D point cloud and features jointly. Clearly, the jointly trained model has a worse texture with 3D shape artifacts to our cascaded design. Besides bringing better editing capability as shown in Fig. 5, our cascaded design enables more flexible training, where the models of two stages can be trained in parallel.

3D Editing on the 3D Latent Space. Finally, we ablate the 3D editing performance in Fig. 6 (b). As can be seen, direct editing on the final Gaussians leads to 3D artifacts, while editing on our 3D latent space yields more holistic and cleaner results since suitable features are re-generated after editing. Besides, our method enables easy editing on the sparse point cloud, compared to directly manipulating dense 3D Gaussians (Dong et al., 2024).

5 Conclusion and Discussions

In this work, we present a new paradigm of 3D generative model by learning the diffusion model over a interactive 3D latent space. A dedicated 3D variational autoencoder encodes multi-view 3D attributes renderings into a point-cloud structured latent space, where multi-modal diffusion learning can be efficiently performed. Our framework achieves superior performance over both text- and image-conditioned 3D generation, and potentially facilitates numerous downstream applications in 3D vision and graphics tasks.

Limitations and Future Work. Our method comes with some limitations to be resolved. 3D VAE side, we observe that the reconstruction textures are sometimes blurry on 3D objects with intricate textures. A potential solution is to leverage pixel-aligned features (Saito et al., 2019; Lan et al., 2023a; Melas-Kyriazi et al., 2023) to alleviate this issue. Moreover, incorporating rendering loss (Anciukevičius et al., 2023) during diffusion training may further boost the generation quality. Besides, adding more real-world data such as MVImageNet Yu et al. (2023) and more control conditions Zhang et al. (2023b) is also worth exploring.

REFERENCES

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Titas Anciukevičius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J Mitra, and Paul Guerrero. RenderDiffusion: Image diffusion for 3D reconstruction, inpainting and generation. In *CVPR*, 2023.

Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022.

Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *ICLR*, 2018.

Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022.

Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. Lara: Efficient large-baseline radiance fields. In *ECCV*, 2024a.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023.

Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, Guosheng Lin, and Chi Zhang. Meshanything: Artist-created mesh generation with autoregressive transformers, 2024b.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *ICLR*, 2024.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *NeurIPS*, 2022.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *ICML*, 2023.

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. *arXiv preprint arXiv:2307.05663*, 2023a.

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*, 2023b.

Shaocong Dong, Lihe Ding, Zhanpeng Huang, Zibin Wang, Tianfan Xue, and Dan Xu. Interactive3d: Create what you want by interactive 3d generation. *arXiv preprint arXiv:2404.16510*, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

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.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3D shape. In *CVPR*, 2020.

Markus Gross and Hanspeter Pfister. Point-based graphics. *Elsevier*, 2011.

Zexin He and Tengfei Wang. OpenLRM: Open-source large reconstruction models. `https://github.com/3DTopia/OpenLRM`, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017.

Jonathan Ho. Classifier-free diffusion guidance. In *NeurIPS*, 2021.

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

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *arXiv preprint arXiv:2106.15282*, 2021.

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*, 2024a.

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*, 2024b.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024a. doi: 10.1145/3641519.3657428.

Zixuan Huang, Justin Johnson, Shoubhik Debnath, James M Rehg, and Chao-Yuan Wu. Pointinfinity: Resolution-invariant point diffusion models. In *CVPR*, 2024b.

Heewoo Jun and Alex Nichol. Shap-E: Generating conditional 3D implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NIPS*, 2022.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *ICCV*, pp. 5148–5157, 2021.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

Diederik P Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. In *Neurips*, 2023.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *arXiv*, 2013.

Adam R. Kosiorek, Heiko Strathmann, Daniel Zoran, Pol Moreno, Rosalia Schneider, Sovna Mokr'a, and Danilo Jimenez Rezende. NeRF-VAE: A geometry aware 3D scene generative model. *ICML*, 2021.

Yushi Lan, Xuyi Meng, Shuai Yang, Chen Change Loy, and Bo Dai. E3dge: Self-supervised geometry-aware encoder for style-based 3D gan inversion. In *CVPR*, 2023a.

Yushi Lan, Feitong Tan, Di Qiu, Qiangeng Xu, Kyle Genova, Zeng Huang, Sean Fanello, Rohit Pandey, Thomas Funkhouser, Chen Change Loy, and Yinda Zhang. Gaussian3Diff: 3D gaussian diffusion for 3D full head synthesis and editing. *arXiv*, 2023b.

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. In *ECCV*, 2024.

Christoph Lassner and Michael Zollhöfer. Pulsar: Efficient sphere-based neural rendering. In *CVPR*, 2021.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, 2019.

Weiyu Li, Jiarui Liu, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Crafts-Man: High-fidelity mesh generation with 3D native generation and interactive geometry refiner, 2024.

Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. *WACV*, 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Zexiang Xu, Hao Su, et al. One-2-3-45: Any single image to 3D mesh in 45 seconds without per-shape optimization. *arXiv preprint arXiv:2306.16928*, 2023a.

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, 2023b.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023c.

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. In *CVPR*, 2024.

Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3D captioning with pre-trained models. *arXiv preprint arXiv:2306.07279*, 2023.

Tiange Luo, Justin Johnson, and Honglak Lee. View selection for 3D captioning via diffusion ranking. *arXiv preprint arXiv:2404.07984*, 2024.

Zhaoyang Lyu, Jinyi Wang, Yuwei An, Ya Zhang, Dahua Lin, and Bo Dai. Controllable mesh generation through sparse latent point diffusion models. In *CVPR*, 2023.

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, 2024.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. 2024.

Luke Melas-Kyriazi, Christian Rupprecht, and Andrea Vedaldi. PC^2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction, 2023. URL `https://arxiv.org/abs/2302.10668`.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Diffeditor: Boosting accuracy and flexibility on diffusion-based image editing. *arXiv preprint arXiv:2402.02583*, 2023a.

Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Dragondiffusion: Enabling drag-style manipulation on diffusion models. *arXiv preprint arXiv:2307.02421*, 2023b.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *TOG*, 41(4):102:1–102:15, July 2022.

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-E: A system for generating 3D point clouds from complex prompts, 2022.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision, 2023.

Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag Your GAN: Interactive point-based manipulation on the generative image manifold. In *SIGGRAPH*, 2023.

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

Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *PACMCGIT*, 2000.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. *ICLR*, 2022.

Charles Qi, Hao Su, Kaichun Mo, and Leonidas Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. *arXiv*, 2016.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017.

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.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

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

Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, October 2019.

Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene Representation Transformer: Geometry-free novel view synthesis through set-latent scene representations. *CVPR*, 2022.

Mehdi S. M. Sajjadi, Aravindh Mahendran, Thomas Kipf, Etienne Pot, Daniel Duckworth, Mario Lučić, and Klaus Greff. RUST: Latent Neural Scene Representations from Unposed Imagery. *CVPR*, 2023.

Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. In *arXiv*, 2023a.

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3D generation. *arXiv:2308.16512*, 2023b.

Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. *arXiv preprint arXiv:2311.15475*, 2023.

Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *NeurIPS*, 2021.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3D reconstruction. In *arXiv*, 2023.

Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3D shapes. In *CVPR*, 2021.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *NeurIPS*, 2020.

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. In *ECCV*, 2024.

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas A. Funkhouser. IBRNet: Learning Multi-View Image-Based Rendering. In *CVPR*, 2021.

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.

Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. CRM: Single image to 3D textured mesh with convolutional reconstruction model. In *ECCV*, 2024.

Chao-Yuan Wu, Justin Johnson, Jitendra Malik, Christoph Feichtenhofer, and Georgia Gkioxari. Multiview compressive coding for 3D reconstruction. *arXiv preprint arXiv:2301.08247*, 2023a.

Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Chunyi Li, Liang Liao, Annan Wang, Erli Zhang, Wenxiu Sun, Qiong Yan, Xiongkuo Min, Guangtai Zhai, and Weisi Lin. Q-align: Teaching lmms for visual scoring via discrete text-defined levels. *arXiv preprint arXiv:2312.17090*, 2023b. Equal Contribution by Wu, Haoning and Zhang, Zicheng. Project Lead by Wu, Haoning. Corresponding Authors: Zhai, Guangtai and Lin, Weisi.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3D: Scalable image-to-3d generation via 3D latent diffusion transformer, 2024.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture, 2020.

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*, 2024a.

Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-NeRF: Point-based neural radiance fields. In *CVPR*, 2022.

Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, and Kai Zhang. DMV3D: Denoising multi-view diffusion using 3D large reconstruction model. In *ICLR*, 2024b.

Lior Yariv, Omri Puny, Natalia Neverova, Oran Gafni, and Yaron Lipman. Mosaic-sdf for 3d generative models. In *CVPR*, 2024.

Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA)*, 38(6), 2019.

Xu Yinghao, Shi Zifan, Yifan Wang, Chen Hansheng, Yang Ceyuan, Peng Sida, Shen Yujun, and Wetzstein Gordon. GRM: Large gaussian reconstruction model for efficient 3d reconstruction and generation, 2024.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. PixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021.

Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *CVPR*, 2018.

Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Tianyou Liang, Guanying Chen, Shuguang Cui, and Xiaoguang Han. MVImgNet: A large-scale dataset of multi-view images. In *CVPR*, 2023.

Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3D shape generation. In *NeurIPS*, 2022.

Biao Zhang, Jiapeng Tang, Matthias Nießner, and Peter Wonka. 3DShape2VecSet: A 3d shape representation for neural fields and generative diffusion models. *ACM Trans. Graph.*, 42(4), jul 2023a. ISSN 0730-0301. doi: 10.1145/3592442.

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. CLAY: A controllable large-scale generative model for creating high-quality 3D assets. *ACM Transactions on Graphics*, 2024.

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

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *CVPR*, 2021.

Chuanxia Zheng and Andrea Vedaldi. Free3d: Consistent novel view synthesis without 3d representation. *arXiv*, 2023.

Shangchen Zhou, Kelvin C.K. Chan, Chongyi Li, and Chen Change Loy. Towards robust blind face restoration with codebook lookup transformer. In *NeurIPS*, 2022.

APPENDIX

A IMPLEMENTATION DETAILS

A.1 TRAINING DETAILS

VAE Architecture. For the convolutional encoder $\mathcal{E}_{\phi}$, we adopt a lighter version of LDM Rombach et al. (2022) encoder with channel 64 and 1 residual blocks for efficiency. When training on Objaverse with $V = 8$, we incorporate 3D-aware attention Shi et al. (2023b) in the middle layer of the convolutional encoder. The multi-view transformer architecture is similar to RUST (Sajjadi et al., 2023; 2022). For each upsampler $\mathcal{D}_U^k$, we have 2 transformer blocks in the middle. All hyperparameters remain at their default settings. Regarding the transformer decoder $\mathcal{D}_T$, we employ the DiT-B/2 architecture due to VRAM constraints. Compared to LN3Diff (Lan et al., 2024), we do not adopt cross-plane attention in the transformer decoder.

Diffusion Model. We mainly adopt the diffusion training pipeline implementation from SiT Ma et al. (2024), with pred-v objective, GVP schedule, and uniform t sampling. ODE solver with 250 steps is used for all the results shown in the paper. For the DiT architecture with cross attention and single-adaLN-zero design, we mainly refer to PixArt Chen et al. (2023). The diffusion transformer is built with 24 layers with 16 heads and 1024 hidden dimension, which result in 458M parameters. For all the diffusion models, we further add the global token to t features as part of the condition input.

Details of Geometry Loss in VAE Training. The geometry loss $\mathcal{L}_{\text{geo}}$ is composed of two regularization terms, including the depth distortion loss to concentrate the weight distribution along rays, inspired by Mip-NeRF (Barron et al., 2021; 2022). Given a ray of pixel, the distortion loss is defined as

$$\mathcal{L}_d = \sum_{i,j} \omega_i \omega_j |d_i - d_j|, \tag{8}$$

where $\omega_i = \alpha_i \, \hat{\mathcal{G}}_i(\mathbf{u}(\mathbf{x})) \prod_{j=1}^{i-1}(1 - \alpha_j \, \hat{\mathcal{G}}_j(\mathbf{u}(\mathbf{x})))$ is the blending weight of the $i-$th intersection and d_i is the depth of the intersection points. Besides, as surfel Gaussians explicitly model the primitive normals, we encourage the splats' normal to locally approximate the actual object surface:

$$\mathcal{L}_n = \sum_{i} \omega_i (1 - \hat{N}_i^{\mathrm{T}} N), \tag{9}$$

where $\hat{N}$ is the predicted normal maps. The final geometry loss is given by $\mathcal{L}_{\text{geo}} = \lambda_{\text{d}} \mathcal{L}_d + \lambda_{\text{n}} \mathcal{L}_n$.

A.2 DATA AND BASELINE COMPARISON

Training Data. For Objaverse, we use a high-quality subset from the pre-processed rendering from G-buffer Objaverse Qiu et al. (2023) for experiments. Since G-buffer Objaverse splits the subset into 10 general categories, we use all the 3D instances except from "Poor-quality": Human-Shape, Animals, Daily-Used, Furniture, Buildings&Outdoor, Transportations, Plants, Food and Electronics. The ground truth camera pose, rendered multi-view images, normal, depth maps, and camera poses are used for stage-1 VAE training.

Details about Baselines. We use the official released code and checkpoint for all the comparisons shown in the paper. For the evaluation on the GSO dataset, we use the rendering provided by Free3D (Zheng & Vedaldi, 2023).

Evaluation details. For quantitative benchmark in Tab. 2, we use 600 instances from Objaverse with ground truth 3D mesh for evaluation. To calculate the visual metrics (FID/KID/MUSIQ), we use the first rendered instance as the image condition and render 24 images with fixed elevation (+15 degrees) with uniform azimuths trajectory (24×15 degrees) with radius$= 1.8$. For 3D metrics, we export the extracted 3D mesh and sample 4096 points using FPS sampling on the mesh surface. The ground truth surface point cloud is processed in the same way. The pre-trained PointNet++ model from Point-E is used for P-FID and P-KID evaluation. All generated 3D models are aligned into the same canonical space before 3D metrics calculation. All intermediate results of the baselines for evaluation will be released.

A.3 More Preliminaries

2D Gaussian Splatting (2DGS). Since 3DGS (Kerbl et al., 2023) models the entire angular radiance in a blob, it fails to reconstruct high-quality object surfaces. To resolve this issue, Huang et al. (2024a) proposed 2DGS (surfel-based GS) that simplifies the 3-dimensional modeling by adopting "flat" 2D Gaussians embedded in 3D space, which enables better alignment with thin surfaces.

Notation-wise, the 2D splat is characterized by its central point $\mathbf{p}_k$, two principal tangential vectors $\mathbf{t}_u$ and $\mathbf{t}_v$, and a scaling vector $\mathbf{S} = (s_u, s_v)$ that controls the variances of the 2D Gaussian. Notice that the primitive normal is defined by two orthogonal tangential vectors $\mathbf{t}_w = \mathbf{t}_u \times \mathbf{t}_v$. Thus, the 2D Gaussian is parameterized with

$$P(u,v) = \mathbf{p}_k + s_u \mathbf{t}_u u + s_v \mathbf{t}_v v = \mathbf{H}(u,v,1,1)^{\mathrm{T}} \tag{10}$$

$$\text{where } \mathbf{H} = \begin{bmatrix} s_u\mathbf{t}_u & s_v\mathbf{t}_v & \mathbf{0} & \mathbf{p}_k \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{RS} & \mathbf{p}_k \\ \mathbf{0} & 1 \end{bmatrix} \tag{11}$$

Where $\mathbf{H}$ parameterizes the local 2D Gaussian geometry. For the point $\mathbf{u} = (u, v)$ in uv space, its 2D Gaussian value can then be evaluated by standard Gaussian $\mathcal{G}(\mathbf{u}) = \exp\left(-\frac{u^2+v^2}{2}\right)$, and the center $\mathbf{p}_k$, scaling (s_u, s_v), and the rotation $(\mathbf{t}_u, \mathbf{t}_v)$ are all learnable parameters. Following 3DGS Kerbl et al. (2023), each 2D Gaussian primitive has opacity α and view-dependent appearance $\mathbf{c}$, and can be rasterized via volumetric alpha blending:

$$\mathbf{c}(\mathbf{x}) = \sum_{i=1} \mathbf{c}_i\, \alpha_i\, \hat{\mathcal{G}}_i(\mathbf{u}(\mathbf{x})) \prod_{j=1}^{i-1} (1 - \alpha_j\, \hat{\mathcal{G}}_j(\mathbf{u}(\mathbf{x}))), \tag{12}$$

where the integration process is terminated when the accumulated opacity reaches saturation. During optimization, pruning and densification operations are iteratively applied.

Flow Matching and Diffusion Model. Diffusion models create data from noise (Song et al., 2021) and are trained to invert forward paths of data towards random noise. The forward path is constructed as $z_t = a_t x_0 + b_t \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$, a_t and b_t are hyper parameters. The choice of forward process has proven to have important implications for the backward process of data sampling (Lin et al., 2023).

Recently, flow matching (Liu et al., 2023c; Albergo et al., 2023; Lipman et al., 2023) has introduced a particular choice for the forward path, which has better theoretical properties and has been verified on the large-scale study (Esser et al., 2024). Given a unified diffusion objective (Karras et al., 2022):

$$\mathcal{L}_w(x_0) = -\frac{1}{2}\mathbb{E}_{t\sim\mathcal{U}(t),\epsilon\sim\mathcal{N}(0,I)}\left[w_t \lambda_t' \|\boldsymbol{\epsilon}_\Theta(z_t, t) - \epsilon\|^2\right], \tag{13}$$

where $\lambda_t := \log \frac{a_t^2}{b_t^2}$ denotes *signal-to-noise ratio*, and λ_t' denotes its derivative. By setting $w_t = \frac{t}{1-t}$ with $z_t = (1-t)x_0 + t\epsilon$, flow matching defines the forward process as a straight path between the data distribution and the Normal distribution. The network $\boldsymbol{\epsilon}_\Theta$ directly predicts the *velocity* v_Θ, and please check the following section for more detailed derivation.

Derivation of the Training Objective of Flow Matching. Since three works (Liu et al., 2023c; Albergo et al., 2023; Lipman et al., 2023) proposed the flow matching idea simultaneously, we adopt the unified formulation defined in Esser et al. (2024) in Eq. 6 and Eq. 7. Here we brief the background of conditional flow matching, and please read the Sec.2 of Esser et al. (2024) for in-depth analysis.

Specifically, consider the forward diffusion process (Ho et al., 2020)

$$z_t = a_t x_0 + b_t \epsilon, \quad \text{where } \epsilon \sim \mathcal{N}(0, I). \tag{14}$$

To express the relationship between z_t, x_0, and ϵ, we define the mappings ψ_t and u_t as:

$$\psi_t(\cdot \mid \epsilon) : x_0 \mapsto a_t x_0 + b_t \epsilon, \tag{15}$$

$$u_t(z \mid \epsilon) := \psi_t'(\psi_t^{-1}(z \mid \epsilon) \mid \epsilon), \tag{16}$$

where ψ_t^{-1} and ψ_t' are the inverse and derivative of ψ_t, respectively.

Since z_t can be viewed as a solution to the ODE

$$z_t' = u_t(z_t \mid \epsilon), \quad \text{with initial condition } z_0 = x_0, \tag{17}$$

the conditional vector field $u_t(\cdot \mid \epsilon)$ generates the conditional probability path $p_t(\cdot \mid \epsilon)$.

Remarkably, one can construct a marginal vector field u_t that generates the marginal probability paths p_t (Lipman et al., 2023), using the conditional vector fields $u_t(\cdot \mid \epsilon)$:

$$u_t(z) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)} \left[u_t(z \mid \epsilon) \frac{p_t(z \mid \epsilon)}{p_t(z)} \right]. \tag{18}$$

The marginal vector field u_t can be learned by minimizing the *Conditional Flow Matching* objective:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t, p_t(z|\epsilon), p(\epsilon)} \left\| v_\Theta(z,t) - u_t(z \mid \epsilon) \right\|_2^2. \tag{19}$$

To make this objective explicit, we substitute:

$$\psi_t'(x_0 \mid \epsilon) = a_t' x_0 + b_t' \epsilon, \tag{20}$$

$$\psi_t^{-1}(z \mid \epsilon) = \frac{z - b_t \epsilon}{a_t}, \tag{21}$$

into the expression for $u_t(z \mid \epsilon)$:

$$z_t' = u_t(z_t \mid \epsilon) = \frac{a_t'}{a_t} z_t - \epsilon b_t \left(\frac{a_t'}{a_t} - \frac{b_t'}{b_t} \right). \tag{22}$$

Next, consider the *signal-to-noise ratio* $\lambda_t := \log \frac{a_t^2}{b_t^2}$. With $\lambda_t' = 2\left(\frac{a_t'}{a_t} - \frac{b_t'}{b_t}\right)$, the expression for $u_t(z_t \mid \epsilon)$ can be rewritten as:

$$u_t(z_t \mid \epsilon) = \frac{a_t'}{a_t} z_t - \frac{b_t}{2} \lambda_t' \epsilon. \tag{23}$$

Using this reparameterization, the $\mathcal{L}_{\text{CFM}}$ objective can be reformulated as a noise-prediction objective:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t, p_t(z|\epsilon), p(\epsilon)} \left\| v_\Theta(z,t) - \frac{a_t'}{a_t} z + \frac{b_t}{2} \lambda_t' \epsilon \right\|_2^2 \tag{24}$$

$$= \mathbb{E}_{t, p_t(z|\epsilon), p(\epsilon)} \left(-\frac{b_t}{2} \lambda_t' \right)^2 \left\| \epsilon_\Theta(z,t) - \epsilon \right\|_2^2, \tag{25}$$

where we define:

$$\epsilon_\Theta := \frac{-2}{\lambda_t' b_t} \big(v_\Theta - \frac{a_t'}{a_t} z \big). \tag{26}$$

Since the optimal solution remains invariant to time-dependent weighting, one can derive various weighted loss functions that guide optimization towards the desired solution. For a unified analysis of different approaches, including classic diffusion formulations, we express the objective as Kingma & Gao (2023):

$$\mathcal{L}_w(x_0) = -\frac{1}{2} \mathbb{E}_{t \sim \mathcal{U}(t), \epsilon \sim \mathcal{N}(0,I)} \left[w_t \lambda_t' \left\| \epsilon_\Theta(z_t,t) - \epsilon \right\|^2 \right], \tag{27}$$

where $w_t = -\frac{1}{2} \lambda_t' b_t^2$ corresponds to w_t^{FM} used in Eq. 6 and Eq. 7.

B Discussions of limitations

We acknowledge that the texture quality of our proposed method is still inferior to the state-of-the-art multi-view based 3D generative models, i.e., LGM. Besides, the visual quality of the text-conditioned 3D generation of native 3D generation methods is worse compared to SDS-based alternatives, despite being much faster and shows better diversity. We believe our method has made a step forward towards bridging the gap. To further improve the performance, we list some potential directions in the following:

1. **Enhancing the 3D VAE Quality.**. The performance of the 3D VAE could be improved by increasing the number of latent points and incorporating a pixel-aligned 3D reconstruction paradigm, such as PiFU (Saito et al., 2019), to achieve finer-grained geometry and texture alignment.
2. **Incorporating Additional Losses in Diffusion Training.**. Currently, the diffusion training relies solely on latent-space flow matching. Prior work, such as DMV3D (Xu et al., 2024b), demonstrates that incorporating a rendering loss can significantly enhance the synthesis of high-quality 3D textures. Adding reconstruction supervision during diffusion training is another promising avenue to improve output fidelity.
3. **Leveraging 2D Pre-training Priors.**. At present, the models are trained exclusively on 3D datasets and do not utilize 2D pre-training priors as effectively as multi-view (MV)-based 3D generative models. A potential improvement is to incorporate 2D priors more effectively, for instance, by using multi-view synthesized images as conditioning during training instead of single-view images.
4. **Expanding Dataset Diversity.**. Utilizing more diverse and extensive 3D datasets, such as Objaverse-XL (Deitke et al., 2023a) and MVImageNet (Yu et al., 2023), could further enhance the quality and generalizability of 3D generation.

By addressing these aspects, the proposed method could achieve significant advancements in both the quality and versatility of its 3D generation capabilities.

C More visual results and videos

Please also check our supplementary *video demo* and *attached folders* for video results.

Overview of the Qualitative Performance. In Fig. 7, we include an overview of the qualitative performance of the proposed method, GaussianAnything. Here, we show the single-view conditioned 3D generation, text-conditioned 3D generation, and 3D-aware editing capabilities.

3D VAE Reconstruction. In Fig. 8, we include the 3D VAE reconstruction results of our model at 3 level of details (LoD). Thanks to the versatile multi-view 3D attributes input and transformer design, our 3D VAE enables high-quality 3D reconstruction with visually attractive textures and smooth surface. The encoded point cloud-structured latent codes, $\mathbf{z}$, serves as a compact proxy for efficient 3D diffusion training. Besides, the 2D Gaussians Upsampler naturally facilitates LoD and facilitates speed / quality trade off in practice.

More Text-to-3D results. In Fig. 9, we present additional qualitative comparisons of text-to-3D generation with GaussianAnything. For this evaluation, we use relatively complex captions as input conditions and display two random samples generated by our model. As shown, GaussianAnything produces visually appealing results that characterized rich textures, smooth surface, and notable diversity. To further demonstrate the generality of our proposed method, in Fig. 10 we include the *uncurated* text-to-3D results over DF-415 (Poole et al., 2022) prompts with captions and more detailed descriptions.

Point-to-3D Generation with Cascaded Point-E. Moreover, the cascaded design of our stage-2 diffusion model, ϵ_Θ^h, enables flexible 3D generation given point clouds from diverse sources. To demonstrate this capability, we integrate the output of a state-of-the-art 3D point cloud generative model, such as Point-E (Nichol et al., 2022), into the GaussianAnything generation pipeline. Specifically, we first generate the point cloud using Point-E based on a caption condition c. This generated point cloud is then used as input $\mathbf{z}_x$ to our stage-2 point cloud/text-conditioned diffusion model ϵ_Θ^h. As shown in Fig. 11, the generated surfel Gaussians exhibit significantly improved texture quality and geometry fidelity compared to the Point-E point cloud outputs. This capability broadens the applicability of our method, enabling it to benefit from recent advances in point cloud generation (Huang et al., 2024b) and mesh generation (Siddiqui et al., 2023; Chen et al., 2024b) for producing high-quality object-level surfel Gaussians.

Broader Social Impact. In this paper, we introduce a new latent 3D diffusion model designed to produce high-quality surfel Gaussians using a single model. As a result, our approach has the potential to be applied to generate DeepFakes or deceptive 3D assets, facilitating the creation of

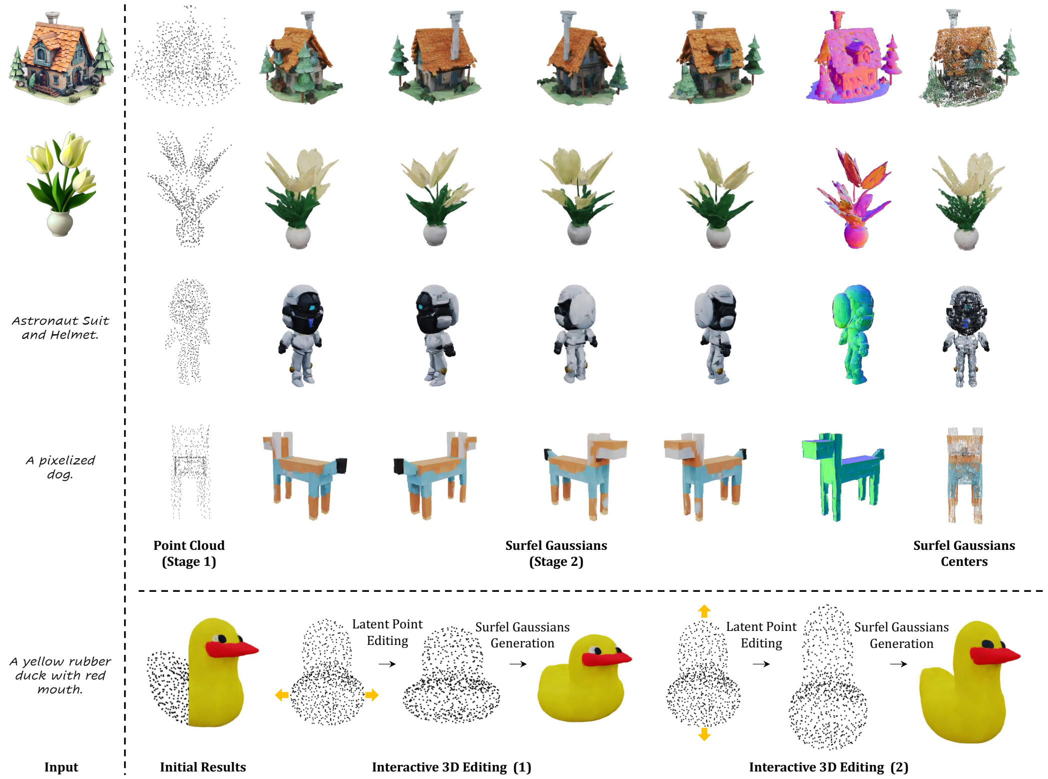

Astronaut Suit and Helmet.
A pixelized dog.
Point Cloud (Stage 1)
Surfel Gaussians (Stage 2)
Surfel Gaussians Centers
A yellow rubber duck with red mouth.
Latent Point Editing
Surfel Gaussians Generation
Latent Point Editing
Surfel Gaussians Generation
Input
Initial Results
Interactive 3D Editing (1)
Interactive 3D Editing (2)

Figure 7: Our method generates *high-quality* and *editable* surfel Gaussians through a cascaded 3D diffusion pipeline, given single-view images or texts as the conditions.

falsified images or videos. This raises concerns, as individuals could exploit such technology with malicious intent, aiming to spread misinformation or tarnish reputations.

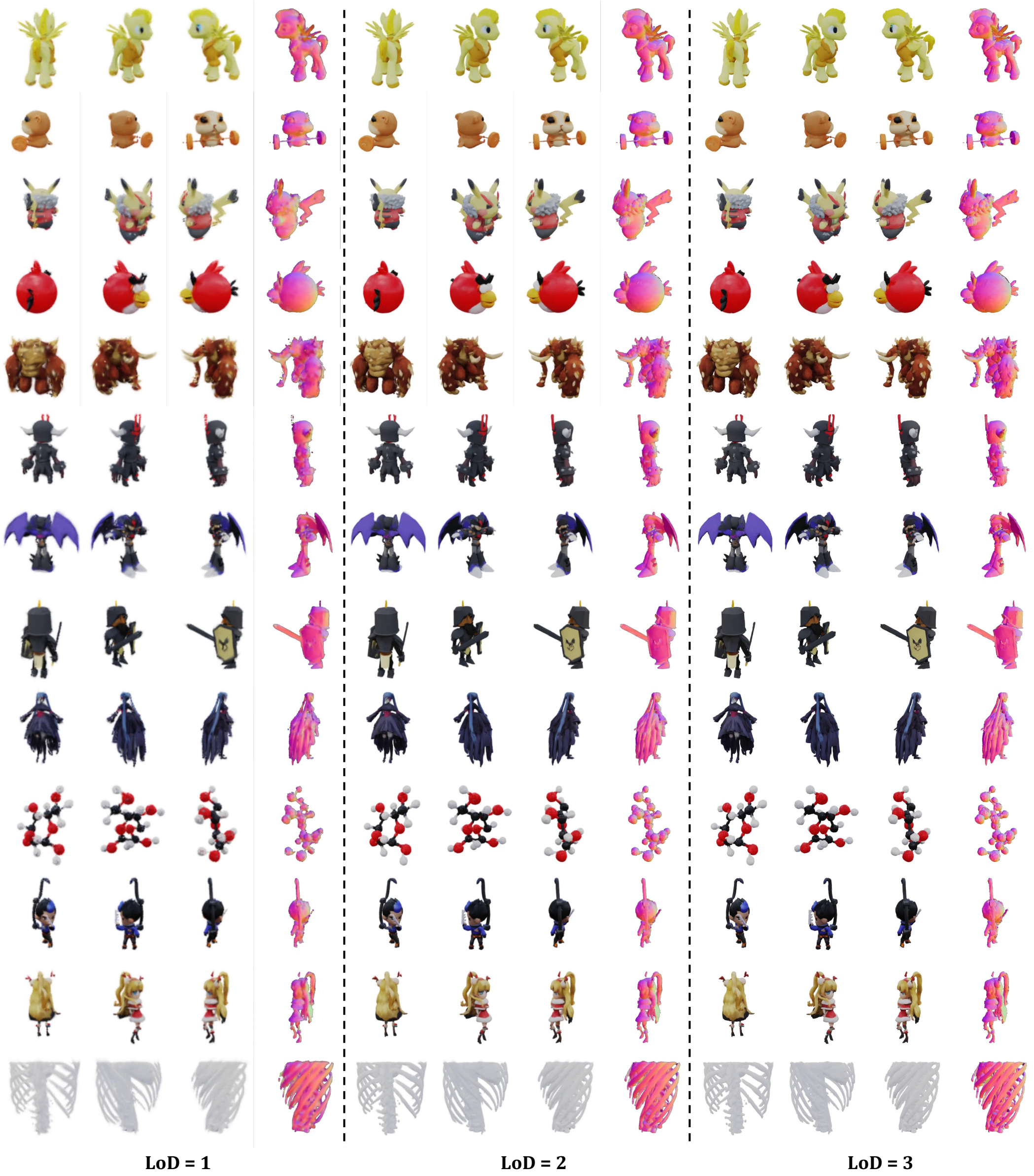


Figure 8: **3D VAE Reconstruction**. Here, we visualize the 3D VAE reconstruction performance across different level of details (LoD). As shown, higher LoD results in sharper textures and smoother surface. Better zoom in.

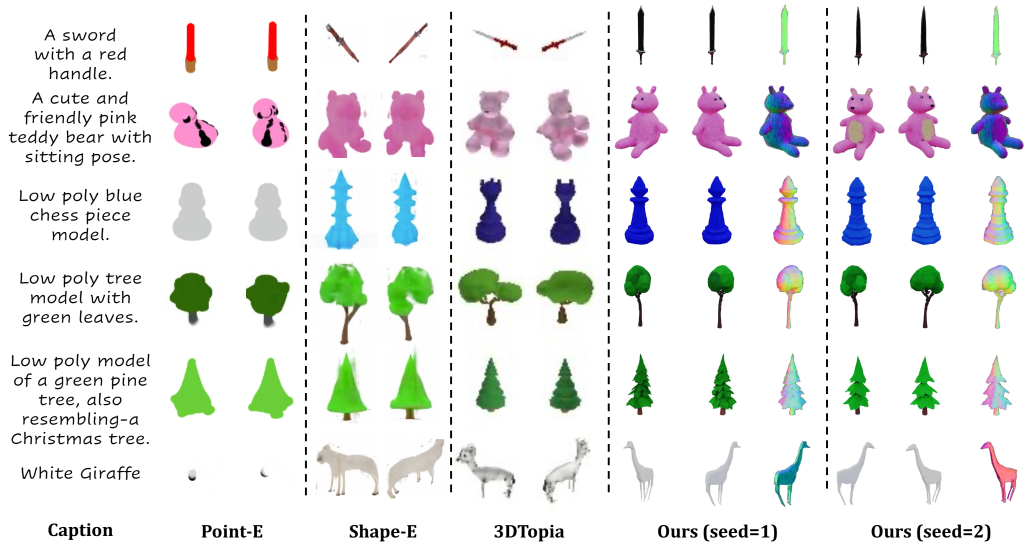

A sword with a red handle.
A cute and friendly pink teddy bear with sitting pose.
Low poly blue chess piece model.
Low poly tree model with green leaves.
Low poly model of a green pine tree, also resembling-a Christmas tree.
White Giraffe
Caption
Point-E
Shape-E
3DTopia
Ours (seed=1)
Ours (seed=2)

Figure 9: **More Qualitative Comparison of Text-to-3D**. We present more text-conditioned 3D objects generated by GaussianAnything, alongside comparisons with competitive alternatives, including Point-E, Shape-E, and 3DTopia. As demonstrated, our approach consistently achieves superior quality in geometry, texture, and alignment between text and 3D content.

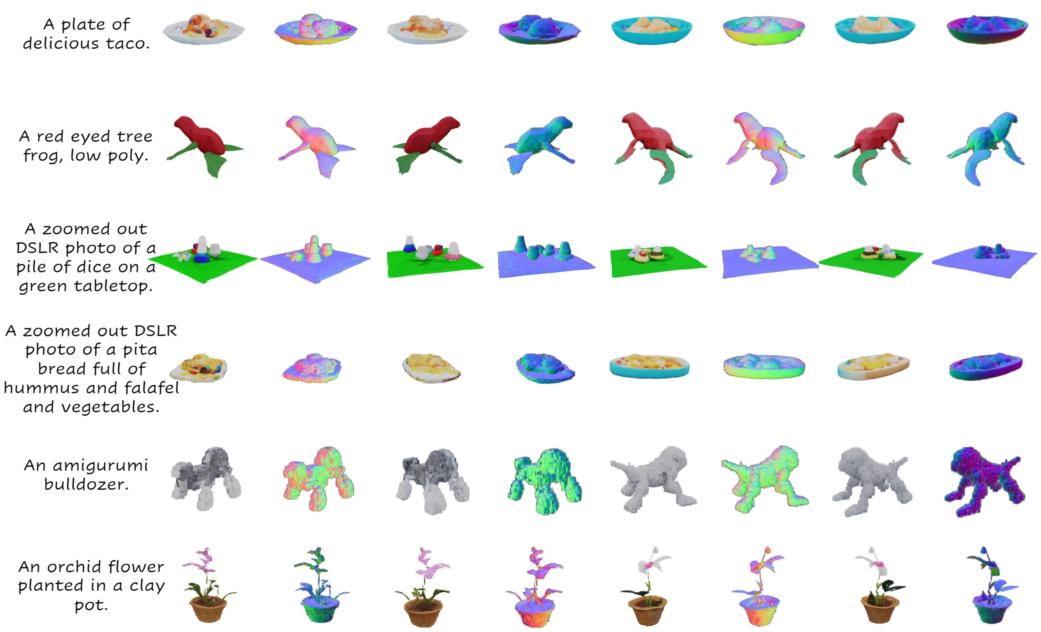

A plate of delicious taco.
A red eyed tree frog, low poly.
A zoomed out DSLR photo of a pile of dice on a green tabletop.
A zoomed out DSLR photo of a pita bread full of hummus and falafel and vegetables.
An amigurumi bulldozer.
An orchid flower planted in a clay pot.

Figure 10: **More Qualitative Results of Text-to-3D over DF-415 Captions**. Our proposed method generalizes to long captions with detailed descriptions. All results are uncurated.

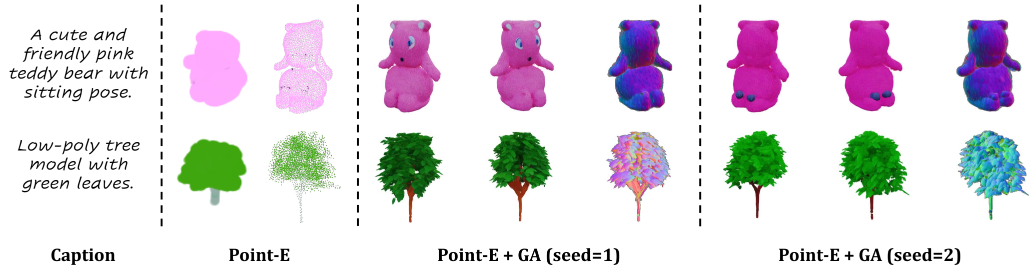

A cute and friendly pink teddy bear with sitting pose.
Low-poly tree model with green leaves.
Caption
Point-E
Point-E + GA (seed=1)
Point-E + GA (seed=2)

Figure 11: **Cascaded Text-to-3D with Point-E**. Thanks to our cascaded 3D generation design, the stage-2 diffusion model in GAUSSIANANYTHING seamlessly integrates with other point cloud generative models. To illustrate this capability, we leverage the state-of-the-art Point-E model. As demonstrated, our stage-2 diffusion model effectively transforms the point clouds generated by Point-E into diverse surfel Gaussians with more visually appealing features and enhanced geometric details.