

# ULTRA3D: Efficient and High-Fidelity 3D Generation with Part Attention

Yiwen Chen<sup>1,2\*</sup>, Zhihao Li<sup>1</sup>, Yikai Wang<sup>3</sup>,  
Hu Zhang<sup>2</sup>, Qin Li<sup>2,4</sup>, Chi Zhang<sup>5†</sup>, Guosheng Lin<sup>1†</sup>

<sup>1</sup>Nanyang Technological University <sup>2</sup>Math Magic

<sup>3</sup>Tsinghua University <sup>4</sup>Beijing Normal University <sup>5</sup>Westlake University

<https://buaacyw.github.io/ultra3d/>

**Abstract.** Recent advances in sparse voxel representations have significantly improved the quality of 3D content generation, enabling high-resolution modeling with fine-grained geometry. However, existing frameworks suffer from severe computational inefficiencies due to the quadratic complexity of attention mechanisms in their two-stage diffusion pipelines. In this work, we propose ULTRA3D, an efficient 3D generation framework that significantly accelerates sparse voxel modeling without compromising quality. Our method leverages the compact VecSet representation to efficiently generate a coarse object layout in the first stage, reducing token count and accelerating voxel coordinate prediction. To refine per-voxel latent features in the second stage, we introduce Part Attention, a geometry-aware localized attention mechanism that restricts attention computation within semantically consistent part regions. This design preserves structural continuity while avoiding unnecessary global attention, achieving up to a  $6.7\times$  speed-up in latent generation. To support this mechanism, we construct a scalable part annotation pipeline that converts raw meshes into part-labeled sparse voxels. Extensive experiments demonstrate that ULTRA3D supports high-resolution 3D generation at 1024 resolution and achieves state-of-the-art performance in both visual fidelity and user preference.

**Keywords:** 3D generation · sparse voxels · efficient attention · diffusion model

## 1 Introduction

Recent advances in generative modeling [17, 33, 39, 59, 68] have significantly expanded the frontier of 3D content creation. These models enable the generation of high-resolution and structurally consistent 3D assets, supporting a wide array of applications in gaming, augmented and virtual reality (AR/VR), digital

---

\* Work done during a research internship at Math Magic.

† Corresponding Authors.

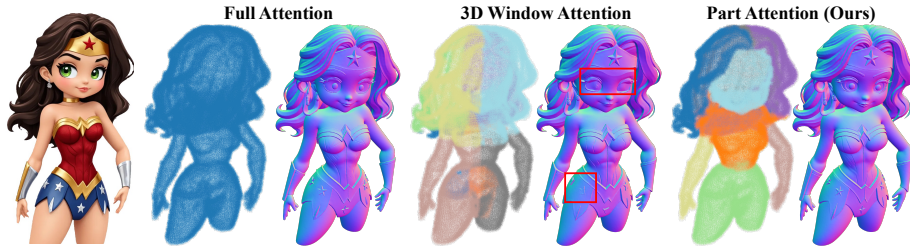


**Fig. 1: Image-to-3D Generation Results of ULTRA3D.** ULTRA3D delivers high-quality 3D meshes with fine-grained geometric details while maintaining efficient generation. Please zoom in to view detailed geometry.

content creation, and robotics. The growing demand for scalable and controllable 3D generation pipelines in these domains has spurred extensive research into expressive representations and generative models [59, 68] that can produce high-fidelity 3D content.

A notable advancement in this direction is the introduction of sparse voxel-based representations [40, 59]. Sparse voxel-based representations have emerged as powerful 3D representations due to their ability to capture fine-grained geometry [13, 26, 40, 58, 59, 65]. In this paradigm, a 3D object is encoded as a sparse voxel grid, where each active voxel is associated with a latent feature vector. This design enables two complementary benefits: (1) the voxel grid provides a coarse yet globally consistent structural layout; and (2) the per-voxel latent features support localized surface modeling with fine granularity. Together, these attributes facilitate high-quality 3D generation with fine-grained geometry quality [59]. As a result, sparse voxels have been adopted by a series of state-of-the-art 3D generation frameworks [13, 26, 58, 59, 65], becoming a mainstream solution for 3D modeling.

However, the benefits of this expressive representation come at the cost of computational efficiency. Most existing frameworks adopt a two-stage pipeline introduced by [59]: first, predicting the coordinates of active voxels, followed by generating per-voxel latent vectors. Both stages are typically implemented using Diffusion Transformer (DiT) [38]. As resolution increases, the number of tokens processed in each stage grows significantly, leading to substantial memory and computation overhead due to the quadratic complexity of attention com-



**Fig. 2: Experiments on different attention mechanisms.** Each color denotes an attention group, within which attention is computed independently. All other settings remain unchanged, with only the attention mechanism being replaced. 3D Window Attention partitions the object space into 8 fixed regions by splitting at the center along each axis. This fixed partitioning often misaligns with semantic boundaries, leading to degraded quality and style inconsistencies.

putation [48]. Consequently, current approaches are often constrained to low resolutions and limited output quality.

Built upon the success of sparse voxel generation frameworks [59], we aim to develop an efficient 3D generation pipeline that overcomes their efficiency limitations and supports high-resolution modeling. We observe that the two stages in the pipeline serve distinct purposes: the first stage constructs a coarse object layout, while the second stage refines fine surface details. In the first stage, prior methods typically compress sparse voxel coordinates into a dense feature grid at one-fourth the resolution for DiT modeling. Although this reduces the number of tokens, the computation remains costly at high resolutions. This is because directly predicting tens of thousands of coordinates is inherently complex and computationally intensive.

Therefore, we depart from direct prediction and instead propose using more efficient methods to first generate a coarse mesh, which is then voxelized to produce the sparse voxel coordinates. Notably, since the coarse mesh is only expected to convey the overall structure rather than fine surface details, it can be efficiently represented using a compact 3D representation. Specifically, we adopt VecSet [68], an efficient 3D representation that encodes each 3D asset into a small set of latent tokens, for generating sparse voxel coordinates. While VecSet is less expressive for fine-grained geometry than sparse voxels, its compactness enables highly efficient generation, which is sufficient for producing the coarse object layout without compromising final generation quality. In practice, this reduces generation time from several minutes to just a few seconds for producing 128-resolution sparse voxels.

To refine the coarse outputs from the first stage with rich and precise latent vectors, previous methods typically adopt a DiT architecture with expensive full attention [26, 59, 65]. However, we observe that the sparse voxels predicted from the first stage already captures the overall object structure, making global attention often redundant and inefficient for local detail refinement. Motivated by this, we introduce **Part Attention**, a geometry-aligned, localized attention mechanism tailored for sparse voxels. It leverages voxel-level part annotations to

restrict attention computation within each part group, thereby significantly enhancing efficiency by avoiding unnecessary interactions across unrelated regions.

Compared to the widely used window attention in LLMs [1, 67], Part Attention is better suited for 3D sparse voxels. As shown in Fig. 2, fixed window attention struggles with the irregular nature of sparse voxels, often leading to inconsistent styles and degraded performance. In contrast, Part Attention partitions tokens according to the object’s parts and geometric layout, respecting semantic boundaries and object structure, thereby preserving consistency and high-quality output. Our experiments demonstrate that Part Attention achieves up to a  $6.7\times$  speed-up without compromising generation quality.

To support Part Attention, we construct an efficient, large-scale part annotation pipeline. Given the limited availability of part-labeled 3D datasets [64], we adopt PartField [31] as our base segmentation model to convert raw mesh data into sparse voxels with part annotations. To ensure data quality, we further apply a series of lightweight filtering strategies. Our pipeline takes only a few seconds to process each mesh with high annotation quality, making it practical for large-scale dataset annotation.

Our experiments show that our design accelerates the generation pipeline by  $3.3\times$  without compromising quality. Furthermore, ULTRA3D achieves state-of-the-art performance in both visual fidelity and user preference.

Our main contributions are summarized as follows:

- We present **ULTRA3D**, an efficient high-resolution 3D generation framework that first produces sparse voxels via the compact VecSet representation [68] and then refines them through per-voxel latent generation [59], enabling both speed and fidelity.
- We introduce **Part Attention**, an efficient localized attention mechanism tailored for sparse voxels that performs attention computation independently within each part group. By preserving geometric continuity, it achieves up to  $6.7\times$  speed-up without compromising quality.
- We develop a scalable part annotation pipeline that efficiently converts raw meshes into part-labeled sparse voxels, enabling high-quality annotations at scale.
- Our experiments demonstrate that ULTRA3D achieves state-of-the-art performance in both visual quality and user preference, while achieving up to  $3.3\times$  speed-up over baseline methods without compromising generation quality.

## 2 Related Work

The field of 3D object generation has witnessed rapid progress in recent years [2, 7, 16, 17, 22, 23, 27, 28, 32–35, 39, 41, 43, 46, 47, 49–51, 53, 54, 56, 61, 62, 66, 69, 74], with several promising directions emerging for scalable 3D object generation [5, 36, 42, 59, 60, 68, 70]. We broadly categorize current methods into three main categories.



## 2.1 Vector Set-based Object Generation

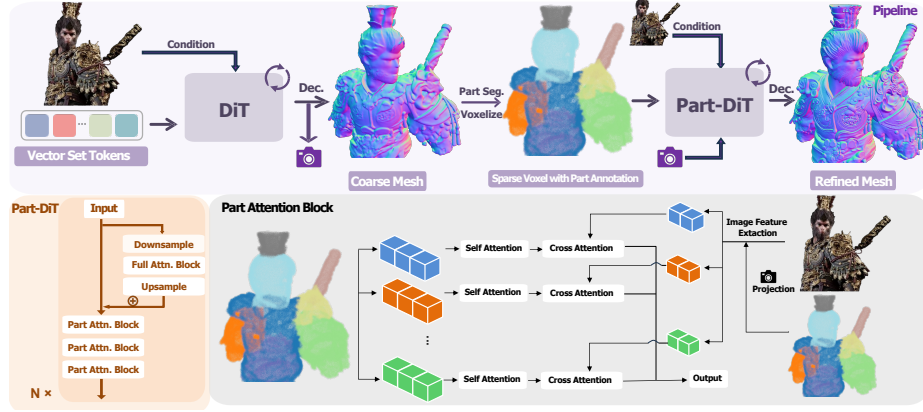
3DShape2Vecset [68] introduces a pipeline that uses a VAE to compress 3D shapes into a compact latent space named Vector Set (VecSet), and subsequently trains a diffusion model in this latent space. Follow-up works [8, 16, 18, 20, 21, 24, 25, 29, 45, 57, 63, 70, 72, 73] have demonstrated that this pipeline is highly scalable and capable of producing high-resolution meshes from large-scale datasets. Due to the compact nature of the VecSet representation (typically only a few thousand tokens), both training and inference are computationally efficient. [29, 45, 63] have shown the flexibility of this representation in integrating part information, enabling simultaneous generation of 3D object and part-label. These methods can be combined with our work, as they can provide high-quality part labels for our Part Attention mechanism. While VecSet is flexible and efficient, it falls short in modeling fine-grained 3D surface details when compared to sparse voxel-based methods [13, 26, 59].

## 2.2 Sparse Voxel-based Object Generation

Trellis [59] introduces a novel 3D representation known as the structured latent, which encodes 3D assets as 3D sparse voxels augmented with latent vectors to capture fine-grained surface details. The generation of this representation typically follows a two-stage pipeline. In the first stage, the voxel coordinates are compressed into a low-resolution continuous feature grid using a lightweight 3D convolutional VAE, and then a Diffusion Transformer (DiT) [38] is trained to generate this grid from noise. In the second stage, another DiT generates the corresponding latent features, conditioned on these voxel coordinates. Both stages are trained independently using the conditional flow matching objective [30]. Subsequent studies [14, 26, 58, 65] demonstrate that sparse voxel excels at modeling extremely fine-grained 3D geometry, significantly outperforming prior representations in terms of geometric fidelity. While sparse voxel provides superior fidelity, it suffers from significant computational overhead. As the resolution increases, the number of sparse voxel tokens can exceed 20K, leading to expensive attention computation.

## 2.3 Autoregressive Mesh Generation

MeshGPT [42] introduced a vertex-by-vertex autoregressive generation approach for meshes, producing outputs that closely resemble those created by human artists, making them particularly valuable in applications such as gaming and digital content creation. Building on this direction, MeshAnything [5] proposed a scalable shape-to-mesh training setting focused on topology generation. Subsequent developments [4, 6, 12, 44, 52, 55, 71], demonstrated that with large-scale training data, such models can generate highly detailed meshes with thousands of faces. This line of work stands out for its ability to produce artist meshes, though it also faces challenges related to high computational cost.



**Fig. 3: Pipeline Overview.** We introduce ULTRA3D, an efficient and high-quality 3D generation framework that first generates sparse voxel layout via VecSet and then refines it by generating per-voxel latent. The core of ULTRA3D is Part Attention, an efficient localized attention mechanism that performs attention computation independently within each part group. Besides, when the input condition is an image, each part group performs cross attention only with the image tokens onto which its voxel tokens are projected.

### 3 Preliminaries

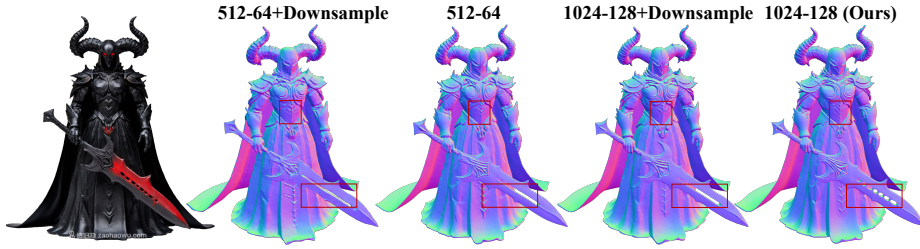
#### 3.1 Sparse Voxel-based Representations

Introduced by [59], sparse voxel-based representations [13, 26, 59, 65] compress a 3D object into a set of latent features distributed over a sparse voxel grid. Formally, a shape is encoded as a collection of tuples  $\{(z_i, \mathbf{p}_i)\}_{i=1}^L$ , where  $\mathbf{p}_i \in \{0, \dots, N-1\}^3$  denotes the coordinate of an active voxel in a cubic grid of resolution  $N$ , and  $\mathbf{z}_i \in \mathbb{R}^C$  is the latent vector associated with that voxel. This formulation preserves both geometric fidelity and spatial locality, making it well-suited for high-resolution surface modeling.

The generation of this representation typically follows a two-stage pipeline [59]. In the first stage, the coordinates  $\{\mathbf{p}_i\}_{i=1}^L$  are compressed into a low-resolution continuous feature grid  $\mathbf{S} \in \mathbb{R}^{D \times D \times D \times C}$  using a lightweight 3D convolutional VAE. A DiT [38] is trained to generate  $\mathbf{S}$  from noise. In the second stage, another DiT conditioned on  $\{\mathbf{p}_i\}_{i=1}^L$  generates the corresponding latent features  $\{\mathbf{z}_i\}_{i=1}^L$ . Both stages are trained independently using the conditional flow matching (CFM) objective [30].

#### 3.2 Vector Set Representation

The Vector Set (VecSet) [68] representation is a compact and permutation-invariant representation for 3D object modeling, designed to facilitate scalable generative pipelines. VecSet encodes a 3D object as an unordered set of



**Fig. 4: Impact of Resolution on Generation Quality.** We compare results under different configurations, where “512-64” denotes a mesh resolution of 512 and a sparse voxel resolution of 64. In previous works, to reduce computational cost in the second stage, the sparse voxels are typically downsampled by half before attention computation in the DiT, then upsampled afterward, annotated as “Downsample” in the figure. As shown, both the mesh resolution and the sparse voxel resolution used during attention computation significantly impact the final quality. However, due to efficiency constraints, prior methods were limited to lower resolutions. In contrast, our efficient framework supports higher sparse voxel resolutions, making high-quality generation feasible.

latent vectors, where each latent vector encodes localized 3D shape and semantic attributes. With only a few thousand vectors per shape, VecSet supports fast diffusion-based generation, scaling well to high-resolution 3D synthesis [68, 70, 72].

While sparse voxels offer strong expressiveness and state-of-the-art surface fidelity [13, 26, 59], their large token count at high resolutions poses significant challenges for efficient training and inference. In contrast, the compactness of VecSet [68] enables fast generation with only a few thousand latent tokens per shape, but its lack of explicit spatial structure limits its ability to capture fine-grained geometry, making it less suitable for high-resolution surface modeling.

## 4 Method

We present ULTRA3D, an efficient 3D generation framework that supports high-resolution and high-fidelity modeling. It first produces sparse voxels using the compact VecSet representation [68], and then refines them through per-voxel latent generation [59], achieving both speed and fidelity.

At the core of ULTRA3D is **Part Attention**, an efficient attention mechanism designed to accelerate sparse voxel generation by performing attention computation independently within each part group (see Sec. 4.1). We describe the overall generation pipeline in Sec. 4.2, and introduce our scalable part annotation pipeline in Sec. 4.3.

#### 4.1 Part Attention

In the widely used sparse voxel generation pipeline [59], the primary computational bottleneck lies in generating the local latents  $\{z_i\}_{i=1}^L$ . For instance, generating a 1024-resolution mesh typically requires performing attention over sparse voxel at resolution 64 or 128. In these cases, the average number of active voxels reaches approximately 20K and 60K, respectively. This makes attention computation prohibitively expensive due to its quadratic scaling with respect to token count. Moreover, as the mesh resolution increases, the corresponding sparse voxel becomes more detailed and effectively captures the global shape and topology of the object. Consequently, the role of sparse voxel latent generation shifts toward refining local surface geometry rather than modeling the overall structure. These observations suggest that full global attention across all tokens is both unnecessary and inefficient.

A straightforward solution is to introduce more efficient attention mechanisms, such as window attention, as explored in the context of large language models [1]. However, as illustrated in Fig. 2, these methods cannot be directly applied to our case. These attention strategies often adopt a certain fixed pattern and leads to attention boundaries that misalign with the semantic layout of the object, leading to inconsistent styles and degraded performance. We find that the fundamental reason is the structural complexity of 3D assets: unlike text sequences, 3D sparse voxel cannot be easily divided into semantically meaningful local blocks using fixed partitioning schemes.

The above observations motivate the introduction of a new 3D attention design, **Part Attention**. Part Attention performs grouping based on externally provided part information. Specifically, for each active voxel  $p_i$ , we assign a part index  $a_i \in \{1, \dots, K\}$ , where  $K$  denotes the number of part groups. Based on this grouping, we apply attention computations that are restricted within each part. Part Attention respects semantic boundaries and object structure, thereby preserving consistency and high-quality output. We next illustrate the designs of Part Self Attention and Part Cross Attention.

**Part Self Attention.** During self-attention computation, each token only attends to other tokens that belong to the same part. Let  $\text{Attn}(i, j)$  denote the attention mask value from the token located at  $p_i$  to the token at  $p_j$ . The attention is masked as:

$$\text{Attn}(i, j) = 0 \quad \text{if} \quad a_i \neq a_j. \quad (1)$$

This masking strategy enforces part-level locality by restricting attention to within each part group, enabling more structured and efficient attention computation. Under the assumption that each part group contains a similar number of tokens, the computation cost of attention is reduced by nearly a factor of  $K$ .

**Part Cross Attention.** In the image-to-3D setting, cross-attention between the 3D sparse voxel and the image features becomes increasingly expensive at high resolutions. To reduce this cost, we also leverage the part index  $a_i \in \{1, \dots, K\}$  to constrain attention across modalities.

Our goal is to allow each 3D voxel to only interact with 2D image regions that correspond to the same part. To implement this, we project each 3D part group onto the condition image using externally provided camera parameters, and assign the part index to the pixels onto which the 3D voxels are projected. Since multiple 3D parts may project onto the same pixel, each 2D image token is associated with a set of part indices.

Let  $\text{Attn}(i, j)$  denote the attention mask value from the voxel token located at  $\mathbf{p}_i$  to the image token indexed by  $j$ , and let  $\mathcal{A}_j$  denote the set of part indices assigned to the  $j$ -th image token. The attention is masked as:

$$\text{Attn}(i, j) = 0 \quad \text{if} \quad a_i \notin \mathcal{A}_j, \quad (2)$$

ensuring that each voxel token only attends to pixels aligned with its own part group. This masking reduces the overall cross-attention cost while preserving semantic consistency between 3D and 2D tokens.

In ULTRA3D, the part labels required by Part Attention are provided by PartField [31] during both training and inference. For camera information, we use ground-truth cameras during training and obtain estimated cameras from the VecSet decoder during inference.

## 4.2 Ultra3D

Following [59], our generation pipeline is divided into two stages: the first stage generates sparse voxel coordinates that capture the overall object structure, and the second stage refines these coordinates by producing high-quality per-voxel latent features for final mesh reconstruction.

**VecSet-based Sparse Voxel Generation.** In our initial experiments, we adopted the sparse voxel generation architecture from Trellis [59], as outlined in Sec. 3.1. However, we found that this design struggles to balance quality and efficiency at high resolutions. As shown in Fig. 4, when aiming to generate meshes at a resolution of 1024, high-quality results typically require sparse voxels at 128 resolution. Under Trellis [59] formulation, this corresponds to applying diffusion-based generation over a  $32^3$  dense grid, equivalent to full attention over 32K tokens, which incurs prohibitive computational cost during both training and inference. Alternatively, downsampling to a  $16^3$  grid reduces the token count but leads to a significant drop in generation quality, making it inadequate for high-fidelity outputs.

Faced with this dilemma, we turn to VecSet [68] for its compact representation and generation efficiency, making it a suitable choice for modeling coarse object structures at scale. As the vector set typically contains only a few thousand tokens, its computational cost is significantly lower than that of sparse voxel-based methods. Specifically, we first use it to generate a 512-resolution mesh, which is then voxelized into sparse voxel. Since the sparse voxel typically has a resolution of 64 or 128, the surface fidelity limitations of VecSet-based models are largely mitigated and have minimal impact on downstream quality.

As illustrated in Sec. 3.2, our implementation of VecSet [68] generation pipeline follows [72]. The shape VAE encodes point clouds sampled from mesh surfaces into a fixed-length vector set, and decodes them into Sign Distance Function (SDF). In addition, since Part Cross Attention requires camera information, we also encode the camera parameters into the latent vector.

**Sparse Latent Generation.** Despite the efficiency of VecSet [68], its inability to capture fine surface geometry necessitates a subsequent refinement stage using sparse voxel-based method. We follow Trellis [59] and adopt a 3D sparse VAE, along with its corresponding flow-matching DiT, to generate per-voxel latents  $\{z_i\}_{i=1}^L$ . We adopt the Sparconv-VAE introduced by [26] as our 3D sparse VAE. It compresses 3D shapes into sparse voxel with latents  $\{(z_i, p_i)\}_{i=1}^L$  as illustrated in Sec. 3.1.

For DiT modeling, we replace most attention blocks with Part Attention to improve efficiency. A small number of full attention layers are retained to align styles across part groups. However, since full attention at high resolutions is computationally expensive, we introduce a residual block that performs full attention at a lower resolution. Specifically, as shown in Fig. 3, we downsample the sparse voxels, apply full attention, and then upsample the features to fuse them back, enabling efficient cross-part communication at low cost. In practice, we stack one full-attention block followed by three Part Attention blocks repeatedly to build the DiT architecture.

The part labels are provided by an external part annotation model [31] during both training and inference. Note that recent advances in VecSet-based part-aware 3D generation [29, 45, 63] offer an alternative for directly generating sparse voxels with part annotations, which can be integrated into our framework.

It is worth noting that, for annotation efficiency, we use a fixed 8-part grouping during training. However, this does not limit the model’s flexibility at inference time, as it can accept inputs with varying numbers of part groups. Besides, our experiments in the supplementary material show that our model consistently produces high-quality outputs even when the number of part groups differs from that used during training.

### 4.3 Sparse Voxel Part Annotation

In this section, we illustrate our data preprocessing pipeline for producing part annotations required by Part Attention. While existing large-scale datasets [3, 10, 11] provide a wealth of 3D assets, most of them lack part-level annotations. Therefore, we construct a part annotation pipeline that efficiently transforms raw meshes into sparse voxel with part annotations.

Since our goal is to annotate millions of 3D assets, methods that require several minutes per shape are impractical. We therefore adopt PartField [31], an efficient and high-quality part segmentation model. PartField is a feedforward network that takes a point-sampled 3D shape as input and predicts a feature field in the form of a triplane, which is subsequently clustered to obtain part-level segmentation. Its fast inference speed and reasonably accurate predictions make it well-suited for large-scale preprocessing.

To process raw mesh into sparse voxels with part annotations, we first uniformly sample point clouds from the mesh surface and input them into Part-Field [31] to obtain a triplane feature field. We then query per-point features of the sampled point clouds from the triplane and voxelize the raw mesh. For each voxel, we average the features of the sampled points within it to produce a sparse voxel with part-aware voxel features. Finally, we apply Agglomerative Clustering to segment the voxels into parts based on these features [31].

A key challenge during clustering is determining the appropriate number of part groups. Since Part Attention benefits from part segmentation that aligns with geometric structures, the ideal number of clusters can vary across different objects. Using too many clusters results in over-segmentation with noisy and fragmented parts, while too few clusters limit the computational advantage of Part Attention. Although one could design heuristics to adaptively determine the optimal number of clusters per object, we found this significantly increases the preprocessing cost. As a practical compromise, we apply a fixed number of clusters to all meshes and empirically set the number to 8. This setting yields reasonable part segmentations for the vast majority of samples.

After clustering, we apply two filtering criteria to discard samples with sub-optimal segmentations. First, we heuristically assume that samples dominated by a single part group tend to exhibit poor segmentation quality. To identify such cases, we compute the voxel ratio of each part group, square these ratios, and sum them as a measure of imbalance. A higher score indicates a more uneven part distribution. Samples exceeding a fixed threshold are removed to eliminate those with overly dominant part groups. Second, well-segmented shapes typically exhibit strong spatial consistency, where most voxels are surrounded by neighbors that share the same part label. To quantify this, we define a metric called neighborhood inconsistency, the proportion of voxels whose neighbors have different part labels. Samples with low full-neighborhood consistency are discarded, as they often reflect fragmented or noisy segmentations. With filtering included, the full preprocessing pipeline is highly efficient, processing each mesh in just 2 seconds on an A800 GPU. As shown in our supplementary material, most processed samples exhibit high-quality segmentations, demonstrating the reliability of our annotation pipeline.

During inference, we first pass the input image through our sparse voxel generation pipeline to obtain a coarse mesh. We then apply the same part annotation procedure to obtain a part-labeled sparse voxel, which is required by the Part Attention.

## 5 Experiments

### 5.1 Implementation Details

ULTRA3D adopts a sparse voxel resolution of 128 and produces output meshes at a resolution of 1024. Unless otherwise specified, all results reported in this paper are generated with this setting. Our 3D sparse VAE adopts the settings



from [26]. The VecSet-based generator follows the configuration of [72], while our part-aware DiT is built upon [59], incorporating the modifications described in Sec. 4.2, and scaling up the model size to 1.8B parameters.

All models are trained on a private 3D dataset with 1000k high-quality 3D objects, annotated using the part annotation pipeline detailed in the main paper. Following the procedure described in the method, we exclude samples where the sum of squared part ratios exceeds 25% or the neighborhood inconsistency exceeds 25%. The SparConv-VAE [26] is trained for 2 days on 32 A800 GPUs; the VecSet-DiT [38,68] is trained for 15 days on 128 GPUs; and the part-aware DiT is trained for 15 days on 256 GPUs with a total batch size of 256.

For training the part-aware DiT, we use the AdamW [19] optimizer with a learning rate of  $1e-4$  and no weight decay. An exponential moving average (EMA) with a rate of 0.9999 is applied. The unconditional guidance probability is set to 0.1, and the timestep sampling schedule follows a Logit-Normal distribution with a mean of 0 and standard deviation of 1. The model is conditioned on DINOv2 [37] image embeddings. At inference time, we sample with a classifier-free guidance [15] scale of 3.5 over 25 steps.

## 5.2 Qualitative Experiments

**Comparison with Concurrent Methods.** As shown in Fig. 5, ULTRA3D consistently outperforms prior methods in visual quality, demonstrating notable improvements in geometric detail and surface accuracy. These gains highlight the effectiveness of our framework and the benefits of our efficient and high resolution design.

**Ablation of Part Attention.** As shown in Fig. 2, Part Attention achieves quality comparable to full global attention, while 3D window attention results in significantly degraded geometry. This is because 3D window attention relies on fixed partitions, which often misalign with object semantics and lead to fragmented attention computation. In contrast, our Part Attention respects semantic boundaries and preserves geometric continuity, enabling both accurate surface modeling and efficient computation. Note that both the Full Attention and 3D Window Attention models are finetuned from the Part Attention checkpoints with sufficient training.

**Experiments on Robustness of Part Annotation.** We use a fixed 8-part grouping during training. However, we note that this does not limit the model’s flexibility at inference time, as it can accept inputs with varying numbers of part groups. Fig. 6 show that our model consistently produces high-quality outputs even when the number of part groups differs from that used during training.

## 5.3 Quantitative Experiments

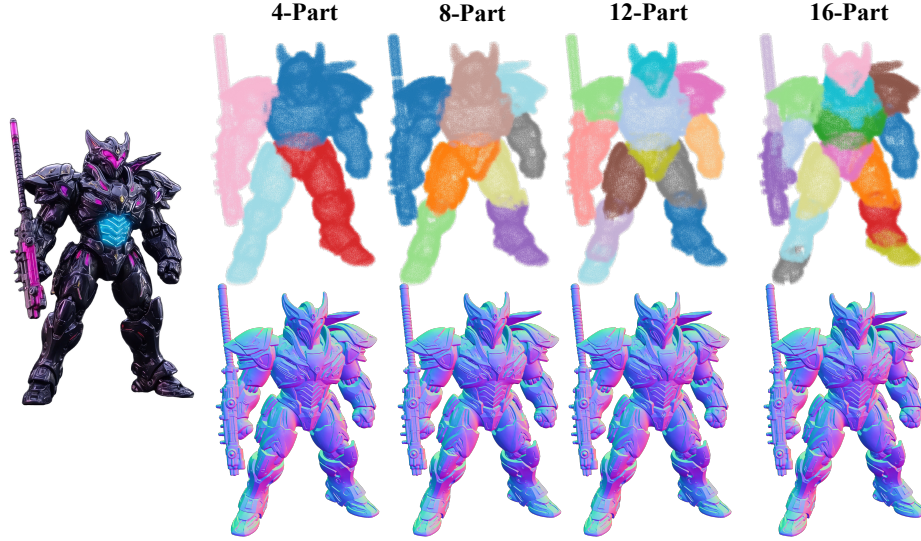
**Acceleration Benefits of Part Attention.** We evaluate the acceleration benefits of Part Attention under the unified use of FlashAttention-2. As shown in Tab. 1, both Part Self Attention and Cross Attention significantly reduce computation cost. Consequently, the overall training and inference pipeline is also



**Fig. 5: Comparison with Prior Methods.** Our method produces higher fidelity and richer surface details. As highlighted in the red boxes, our results align more closely with the input image compared to other methods.

accelerated. Depending on the number of active voxels, the full attention baseline often requires over 15 minutes to generate a single mesh, an impractical cost, whereas our pipeline averages just 4 minutes per sample.

We provide additional quantitative experiments in the supplementary material.



**Fig. 6: Robustness of Part Annotation.** Although our method is trained using data with exactly 8 part groups, we find it to be robust to variations in part annotation. Varying the number of part groups has little impact on generation quality, suggesting that increasing the number of annotated part groups can further accelerate computation without compromising performance.

## 6 Limitations and Future Works

Although Ultra3D achieves efficient, high-quality 3D generation, inference speed remains limited. Future work could reduce latency by improving the 3D representation and attention. The sparse voxel representation still uses many tokens, increasing sequence length and attention cost at high resolutions. Moreover, attention can remain near-dense despite locality heuristics, leading to high runtime and memory. Sparsity-friendly attention for 3D sparse voxel is a promising direction.

## 7 Conclusion

In this work, we aim to address the computational bottlenecks of sparse voxel-based 3D generation. We identify the inefficiency of traditional two-stage pipelines and propose a more scalable alternative by first generating a coarse mesh using VecSet and then voxelizing it into sparse voxels for refinement. To further alleviate the cost of full attention in the second stage, we introduce Part Attention, a geometry-aware attention mechanism that restricts computation to semantically coherent part groups. This design preserves detail while significantly reducing computation. To support this, we construct a large-scale, efficient part annotation pipeline, enabling high-quality part annotation at scale. Through extensive

**Table 1: Efficiency Comparison.** Part Attention significantly accelerates computation over full attention. Substituting most DiT layers with Part Attention yields a substantial speed-up in both training and inference. All attention blocks are implemented with FlashAttention-2 [9].

|                     | Part Self Attention | Part Cross Attention | DiT Training | DiT Inference |
|---------------------|---------------------|----------------------|--------------|---------------|
| <b>Speedup Rate</b> | 6.7×                | 4.1×                 | 3.1×         | 3.3×          |

experiments, we demonstrate that our framework ULTRA3D achieves state-of-the-art visual quality with substantially improved efficiency in both training and inference, making high-resolution 3D generation more tractable and practical.

## Acknowledgements

This research is supported by the MoE AcRF Tier 2 grant (MOE-T2EP20223-0001) and the MoE AcRF Tier 1 grant (RG14/22). This work was supported by the National Natural Science Foundation of China (No. 6250070674).

## References

1. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150 (2020)
2. Chan, E.R., Nagano, K., Chan, M.A., Bergman, A.W., Park, J.J., Levy, A., Aittala, M., Mello, S.D., Karras, T., Wetzstein, G.: Generative novel view synthesis with 3d-aware diffusion models. In: IEEE/CVF ICCV. pp. 4194–4206 (2023)
3. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
4. Chen, S., Chen, X., Pang, A., Zeng, X., Cheng, W., Fu, Y., Yin, F., Wang, B., Yu, J., Yu, G., et al.: Meshxl: Neural coordinate field for generative 3d foundation models. Advances in Neural Information Processing Systems (NeurIPS) (2024)
5. Chen, Y., He, T., Huang, D., Ye, W., Chen, S., Tang, J., Chen, X., Cai, Z., Yang, L., Yu, G., Lin, G., Zhang, C.: Meshanything: Artist-created mesh generation with autoregressive transformers (2024), <https://arxiv.org/abs/2406.10163>
6. Chen, Y., Wang, Y., Luo, Y., Wang, Z., Chen, Z., Zhu, J., Zhang, C., Lin, G.: Meshanything v2: Artist-created mesh generation with adjacent mesh tokenization (2024), <https://arxiv.org/abs/2408.02555>
7. Chen, Z., Wang, F., Liu, H.: Text-to-3d using gaussian splatting. arXiv preprint arXiv:2309.16585 (2023)
8. Chen, Z., Wang, Y., Sun, W., Wang, F., Chen, Y., Liu, H.: Meshgen: Generating pbr textured mesh with render-enhanced auto-encoder and generative data augmentation (2025), <https://arxiv.org/abs/2505.04656>
9. Dao, T.: Flashattention-2: Faster attention with better parallelism and work partitioning. arXiv preprint arXiv:2307.08691 (2023)

10. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., et al.: Objaverse-xl: A universe of 10m+ 3d objects. arXiv preprint arXiv:2307.05663 (2023)
11. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., Vanderbilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: CVPR (2023)
12. Hao, Z., Romero, D.W., Lin, T.Y., Liu, M.Y.: Meshtron: High-fidelity, artist-like 3d mesh generation at scale (2024), <https://arxiv.org/abs/2412.09548>
13. He, X., Zou, Z.X., Chen, C.H., Guo, Y.C., Liang, D., Yuan, C., Ouyang, W., Cao, Y.P., Li, Y.: Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. arXiv preprint arXiv:2503.21732 (2025)
14. He, X., Zou, Z.X., Chen, C.H., Guo, Y.C., Liang, D., Yuan, C., Ouyang, W., Cao, Y.P., Li, Y.: Sparseflex: High-resolution and arbitrary-topology 3d shape modeling (2025), <https://arxiv.org/abs/2503.21732>
15. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications (2021)
16. Hong, F., Tang, J., Cao, Z., Shi, M., Wu, T., Chen, Z., Wang, T., Pan, L., Lin, D., Liu, Z.: 3dtopia: Large text-to-3d generation model with hybrid diffusion priors. CoRR **abs/2403.02234** (2024)
17. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d. arXiv preprint arXiv:2311.04400 (2023)
18. Hunyuan3D, T., Yang, S., Yang, M., Feng, Y., Huang, X., Zhang, S., He, Z., Luo, D., Liu, H., Zhao, Y., Lin, Q., Lai, Z., Yang, X., Shi, H., Zhao, Z., Zhang, B., Yan, H., Wang, L., Liu, S., Zhang, J., Chen, M., Dong, L., Jia, Y., Cai, Y., Yu, J., Tang, Y., Guo, D., Yu, J., Zhang, H., Ye, Z., He, P., Wu, R., Wei, S., Zhang, C., Tan, Y., Sun, Y., Niu, L., Huang, S., Zheng, B., Liu, S., Chen, S., Yuan, X., Yang, X., Liu, K., Zhu, J., Chen, P., Liu, T., Wang, D., Liu, Y., Linus, Jiang, J., Huang, J., Guo, C.: Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material (2025), <https://arxiv.org/abs/2506.15442>
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
20. Lai, Z., Zhao, Y., Liu, H., Zhao, Z., Lin, Q., Shi, H., Yang, X., Yang, M., Yang, S., Feng, Y., Zhang, S., Huang, X., Luo, D., Yang, F., Yang, F., Wang, L., Liu, S., Tang, Y., Cai, Y., He, Z., Liu, T., Liu, Y., Jiang, J., Linus, Huang, J., Guo, C.: Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details (2025), <https://arxiv.org/abs/2506.16504>
21. Lan, Y., Hong, F., Yang, S., Zhou, S., Meng, X., Dai, B., Pan, X., Loy, C.C.: Ln3diff: Scalable latent neural fields diffusion for speedy 3d generation. In: ECCV (2024)
22. Li, J., Tan, H., Zhang, K., Xu, Z., Luan, F., Xu, Y., Hong, Y., Sunkavalli, K., Shakhnarovich, G., Bi, S.: Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In: ICLR (2024)
23. Li, P., Liu, Y., Long, X., Zhang, F., Lin, C., Li, M., Qi, X., Zhang, S., Luo, W., Tan, P., Wang, W., Liu, Q., Guo, Y.: Era3d: High-resolution multiview diffusion using efficient row-wise attention. CoRR **abs/2405.11616** (2024)
24. Li, W., Liu, J., Chen, R., Liang, Y., Chen, X., Tan, P., Long, X.: Craftsman: High-fidelity mesh generation with 3d native generation and interactive geometry refiner. arXiv preprint arXiv:2405.14979 (2024)

25. Li, Y., Zou, Z.X., Liu, Z., Wang, D., Liang, Y., Yu, Z., Liu, X., Guo, Y.C., Liang, D., Ouyang, W., et al.: Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. arXiv preprint arXiv:2502.06608 (2025)
26. Li, Z., Wang, Y., Zheng, H., Luo, Y., Wen, B.: Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling (2025), <https://arxiv.org/abs/2505.14521>
27. Liang, Y., Yang, X., Lin, J., Li, H., Xu, X., Chen, Y.: Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In: IEEE/CVF CVPR. pp. 6517–6526 (2024)
28. Lin, C., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M., Lin, T.: Magic3d: High-resolution text-to-3d content creation. In: IEEE CVPR. pp. 300–309 (2023)
29. Lin, Y., Lin, C., Pan, P., Yan, H., Feng, Y., Mu, Y., Fragkiadaki, K.: Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers (2025), <https://arxiv.org/abs/2506.05573>
30. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: International Conference on Learning Representations (ICLR) (2023)
31. Liu, M., Uy, M.A., Xiang, D., Su, H., Fidler, S., Sharp, N., Gao, J.: Partfield: Learning 3d feature fields for part segmentation and beyond. arXiv preprint arXiv:2504.11451 (2025)
32. Liu, M., Xu, C., Jin, H., Chen, L., Varma T, M., Xu, Z., Su, H.: One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. In: NeurIPS (2024)
33. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
34. Liu, Y., Lin, C., Zeng, Z., Long, X., Liu, L., Komura, T., Wang, W.: Syncdreamer: Generating multiview-consistent images from a single-view image. In: ICLR (2024)
35. Long, X., Guo, Y.C., Lin, C., Liu, Y., Dou, Z., Liu, L., Ma, Y., Zhang, S.H., Habermann, M., Theobalt, C., et al.: Wonder3d: Single image to 3d using cross-domain diffusion. arXiv preprint arXiv:2310.15008 (2023)
36. Nash, C., Ganin, Y., Eslami, S.A., Battaglia, P.: Polygen: An autoregressive generative model of 3d meshes. In: International Conference on Machine Learning (ICML) (2020)
37. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
38. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV. pp. 4195–4205 (2023)
39. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. In: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net (2023), <https://openreview.net/pdf?id=FjNys5c7VyY>
40. Ren, X., Huang, J., Zeng, X., Museth, K., Fidler, S., Williams, F.: Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
41. Shi, Y., Wang, P., Ye, J., Mai, L., Li, K., Yang, X.: Mvdream: Multi-view diffusion for 3d generation. In: ICLR (2024)

42. Siddiqui, Y., Alliegro, A., Artemov, A., Tommasi, T., Sirigatti, D., Rosov, V., Dai, A., Nießner, M.: Meshgpt: Generating triangle meshes with decoder-only transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
43. Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In: ECCV (2024)
44. Tang, J., Li, Z., Hao, Z., Liu, X., Zeng, G., Liu, M.Y., Zhang, Q.: Edgerunner: Auto-regressive auto-encoder for artistic mesh generation (2024), <https://arxiv.org/abs/2409.18114>
45. Tang, J., Lu, R., Li, Z., Hao, Z., Li, X., Wei, F., Song, S., Zeng, G., Liu, M.Y., Lin, T.Y.: Efficient part-level 3d object generation via dual volume packing (2025), <https://arxiv.org/abs/2506.09980>
46. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In: International Conference on Learning Representations (ICLR) (2024)
47. Tochilkin, D., Pankratz, D., Liu, Z., Huang, Z., Letts, A., Li, Y., Liang, D., Laforte, C., Jampani, V., Cao, Y.P.: Triposr: Fast 3d object reconstruction from a single image. arXiv preprint arXiv:2403.02151 (2024)
48. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)
49. Wang, H., Du, X., Li, J., Yeh, R.A., Shakhnarovich, G.: Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In: IEEE CVPR. pp. 12619–12629 (2023)
50. Wang, P., Shi, Y.: Imagedream: Image-prompt multi-view diffusion for 3d generation. CoRR **abs/2312.02201** (2023)
51. Wang, P., Tan, H., Bi, S., Xu, Y., Luan, F., Sunkavalli, K., Wang, W., Xu, Z., Zhang, K.: Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. arXiv preprint arXiv:2311.12024 (2023)
52. Wang, Z., Lorraine, J., Wang, Y., Su, H., Zhu, J., Fidler, S., Zeng, X.: Llama-mesh: Unifying 3d mesh generation with language models (2024), <https://arxiv.org/abs/2411.09595>
53. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. arXiv preprint arXiv:2305.16213 (2023)
54. Wei, X., Zhang, K., Bi, S., Tan, H., Luan, F., Deschaintre, V., Sunkavalli, K., Su, H., Xu, Z.: Meshlrm: Large reconstruction model for high-quality mesh. arXiv preprint arXiv:2404.12385 (2024)
55. Weng, H., Zhao, Z., Lei, B., Yang, X., Liu, J., Lai, Z., Chen, Z., Liu, Y., Jiang, J., Guo, C., Zhang, T., Gao, S., Chen, C.L.P.: Scaling mesh generation via compressive tokenization (2024), <https://arxiv.org/abs/2411.07025>
56. Wu, K., Liu, F., Cai, Z., Yan, R., Wang, H., Hu, Y., Duan, Y., Ma, K.: Unique3d: High-quality and efficient 3d mesh generation from a single image. CoRR **abs/2405.20343** (2024)
57. Wu, S., Lin, Y., Zhang, F., Zeng, Y., Xu, J., Torr, P., Cao, X., Yao, Y.: Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. arXiv preprint arXiv:2405.14832 (2024)
58. Wu, S., Lin, Y., Zhang, F., Zeng, Y., Yang, Y., Bao, Y., Qian, J., Zhu, S., Cao, X., Torr, P., Yao, Y.: Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention (2025), <https://arxiv.org/abs/2505.17412>

59. Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., Yang, J.: Structured 3d latents for scalable and versatile 3d generation (2025), <https://arxiv.org/abs/2412.01506>
60. Xiong, B., Wei, S.T., Zheng, X.Y., Cao, Y.P., Lian, Z., Wang, P.S.: Octfusion: Octree-based diffusion models for 3d shape generation (2025), <https://arxiv.org/abs/2408.14732>
61. Xu, J., Cheng, W., Gao, Y., Wang, X., Gao, S., Shan, Y.: Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. arXiv preprint arXiv:2404.07191 (2024)
62. Xu, Y., Tan, H., Luan, F., Bi, S., Wang, P., Li, J., Shi, Z., Sunkavalli, K., Wetzelstein, G., Xu, Z., et al.: Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. In: International Conference on Learning Representations (ICLR) (2024)
63. Yang, Y., Guo, Y.C., Huang, Y., Zou, Z.X., Yu, Z., Li, Y., Cao, Y.P., Liu, X.: Holopart: Generative 3d part amodal segmentation (2025), <https://arxiv.org/abs/2504.07943>
64. Yang, Y., Huang, Y., Guo, Y.C., Lu, L., Wu, X., Lam, E.Y., Cao, Y.P., Liu, X.: Sampart3d: Segment any part in 3d objects. arXiv preprint arXiv:2411.07184 (2024)
65. Ye, C., Wu, Y., Lu, Z., Chang, J., Guo, X., Zhou, J., Zhao, H., Han, X.: Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. arXiv preprint arXiv:2503.22236 (2025)
66. Yi, T., Fang, J., Wang, J., Wu, G., Xie, L., Zhang, X., Liu, W., Tian, Q., Wang, X.: Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In: IEEE CVPR. pp. 6796–6807 (2024)
67. Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., Ahmed, A.: Big bird: Transformers for longer sequences (2021), <https://arxiv.org/abs/2007.14062>
68. Zhang, B., Tang, J., Niessner, M., Wonka, P.: 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. ACM Transactions On Graphics (TOG) **42**(4), 1–16 (2023)
69. Zhang, K., Bi, S., Tan, H., Xiangli, Y., Zhao, N., Sunkavalli, K., Xu, Z.: Gs-lrm: Large reconstruction model for 3d gaussian splatting. In: ECCV (2024)
70. Zhang, L., Wang, Z., Zhang, Q., Qiu, Q., Pang, A., Jiang, H., Yang, W., Xu, L., Yu, J.: Clay: A controllable large-scale generative model for creating high-quality 3d assets. ACM Transactions on Graphics (TOG) **43**(4), 1–20 (2024)
71. Zhao, R., Ye, J., Wang, Z., Liu, G., Chen, Y., Wang, Y., Zhu, J.: Deepmesh: Auto-regressive artist-mesh creation with reinforcement learning (2025), <https://arxiv.org/abs/2503.15265>
72. Zhao, Z., Lai, Z., Lin, Q., Zhao, Y., Liu, H., Yang, S., Feng, Y., Yang, M., Zhang, S., Yang, X., Shi, H., Liu, S., Wu, J., Lian, Y., Yang, F., Tang, R., He, Z., Wang, X., Liu, J., Zuo, X., Chen, Z., Lei, B., Weng, H., Xu, J., Zhu, Y., Liu, X., Xu, L., Hu, C., Yang, S., Zhang, S., Liu, Y., Huang, T., Wang, L., Zhang, J., Chen, M., Dong, L., Jia, Y., Cai, Y., Yu, J., Tang, Y., Zhang, H., Ye, Z., He, P., Wu, R., Zhang, C., Tan, Y., Xiao, J., Tao, Y., Zhu, J., Xue, J., Liu, K., Zhao, C., Wu, X., Hu, Z., Qin, L., Peng, J., Li, Z., Chen, M., Zhang, X., Niu, L., Wang, P., Wang, Y., Kuang, H., Fan, Z., Zheng, X., Zhuang, W., He, Y., Liu, T., Yang, Y., Wang, D., Liu, Y., Jiang, J., Huang, J., Guo, C.: Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation (2025), <https://arxiv.org/abs/2501.12202>



73. Zhao, Z., Liu, W., Chen, X., Zeng, X., Wang, R., Cheng, P., Fu, B., Chen, T., Yu, G., Gao, S.: Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in Neural Information Processing Systems* **36** (2024)
74. Zou, Z.X., Yu, Z., Guo, Y.C., Li, Y., Liang, D., Cao, Y.P., Zhang, S.H.: Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 10324–10335 (2024)