Seg3DParts: Segmentation-Grounded Controllable Part-Level 3D Generation

Jiantao Lin[1,*] **Meixi Chen**[1,*] **Yingjie Xu**[1,3,*] **Chenbo Fu**[1] **Leyi Wu**[1]
Hao Chen[2] **Yinchuan Li**[3] **Ying-Cong Chen**[1,2,†]
[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]The Hong Kong University of Science and Technology [3]Knowin AI
[*]Equal contribution. [†]Corresponding author.



Figure 1: Qualitative results across diverse object categories. Seg3DParts generates spatially aligned part meshes for everyday objects, articulated figures, and organic shapes. Colors distinguish individual parts assembled directly in a shared canonical space without post-hoc alignment.

Abstract

Part-level 3D assets are essential for editing, reassembly, and interaction, yet recovering such structure from a single image remains challenging due to occlusion, ambiguous boundaries, and the need for coherent multi-part reasoning. Existing approaches struggle to achieve both controllable part-level generation and coherent multi-part structure, as part identity and spatial allocation are typically inferred implicitly. We present Seg3DParts, a segmentation-grounded framework for controllable part-level 3D generation from a single image. By treating segmentation as an explicit grounding signal, our method defines part identity during generation, enabling each component to be anchored to a corresponding image region. To ensure coherent assemblies, we introduce structured cross-part interaction that allows components to exchange global context throughout the generative process. As a result, Seg3DParts directly generates well-aligned part meshes in a shared canonical space without post-hoc alignment, supporting flexible and controllable decomposition. We further introduce PartObjectNet, a large-scale dataset with over 200K objects and 1M annotated parts. Experiments demonstrate that Seg3DParts achieves superior geometry quality, cross-part coherence, and part-level controllability over existing methods.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

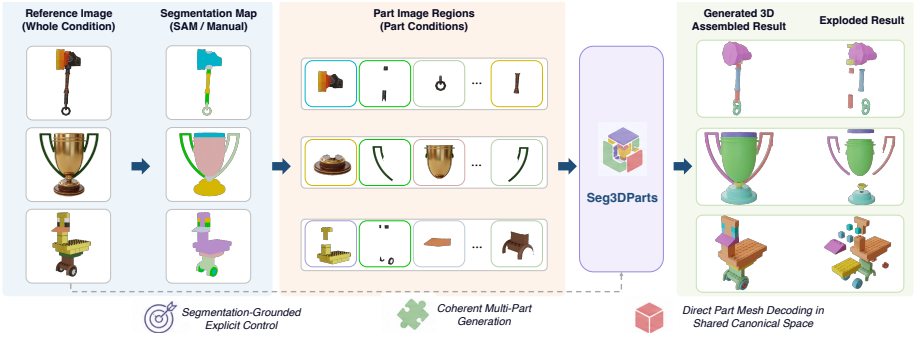


Figure 2: Conditioned on reference image and per-part image regions derived from a segmentation map , **Seg3DParts** jointly generates all part meshes in a shared canonical space, yielding controllable decomposition and coherent multi-part assemblies.

1 Introduction

Generating part-level 3D structure from a single RGB image is fundamentally ill-posed. Local appearance alone is often insufficient to determine precise part boundaries, especially when adjacent components share similar materials or exhibit weak shading cues. Occlusion further complicates the problem, as important functional components may be partially or entirely invisible in the input view, providing little or no direct 2D evidence for their geometry. Beyond individual parts, a model must also reason about cross-part spatial relationships in 3D, ensuring that components attach correctly and maintain plausible relative scale and placement. A practical system therefore needs to recover missing geometry, localize parts reliably, and reason coherently across multiple interacting components under severe single-view ambiguity.

Existing approaches to part-level 3D generation generally follow two paradigms. Decomposition-based pipelines explicitly segment an object into parts and reconstruct each component independently by completing missing geometry before assembling them into a full shape [1–6]. While this design provides direct control over part decomposition, reconstruction is driven primarily by local geometric cues within each segment, with limited modeling of cross-part relationships. As a result, these methods often produce inconsistent scale, misalignment, or implausible assemblies, especially when parts are heavily occluded or contain large missing regions. In contrast, joint part-structured generative models represent multiple components within a unified latent space and generate them in an end-to-end manner [7–10]. By jointly modeling all components, these methods encourage global coherence across parts. However, parts are encoded as implicit latent slots without explicit semantic grounding or spatial anchoring. Consequently, part identity, correspondence, and placement must be inferred during generation, which can lead to ambiguous decomposition and limited controllability.

Despite their differences, both paradigms share a common limitation: part identity and spatial allocation are treated as implicit variables that must be inferred rather than explicitly specified. This implicit formulation makes it difficult to achieve both precise part-level control and coherent multi-part generation, particularly under occlusion or limited visual evidence.

We instead introduce a formulation where part identity is explicitly specified and controllable via segmentation, rather than implicitly inferred during generation. Specifically, we reformulate part-level 3D generation as a segmentation-grounded conditional generation problem, where segmentation defines part identity and anchors each component to its corresponding image region. Under this formulation, Seg3DParts enables controllable and coherent multi-part 3D generation from a single image (Fig. 2).

At its core, Seg3DParts conditions each component on localized appearance cues derived from segmentation, allowing part-specific geometry to be generated from its corresponding image region. This explicit grounding provides direct control over part identity and improves robustness under occlusion and ambiguous boundaries.

To ensure coherent multi-part structure, Seg3DParts further introduces a part-level interaction mechanism that enables components to exchange global structural context throughout the generative process. This structured interaction allows each part to reason about the shape, scale, and placement of others while preserving explicit part identities.

As a result, Seg3DParts learns a coherent multi-part representation from which independent part meshes are directly decoded in a shared canonical space, eliminating post-hoc alignment and enabling flexible, segmentation-driven, controllable generation. We also construct PartObjectNet, a large-scale dataset of part-separated 3D objects with rich part-level annotations. Together, these design principles shift part-level 3D generation from implicit allocation to explicitly grounded conditional generation.

Our contributions are summarized as follows:

- **A segmentation-grounded formulation for part-level 3D generation.** We introduce a formulation that explicitly anchors part identity to segmentation regions, enabling controllable and spatially grounded generation.
- **A unified framework for controllable and coherent multi-part generation.** We instantiate this formulation with segmentation-grounded conditioning and structured cross-part interaction, enabling joint reasoning over shape, placement, and inter-part relationships.
- **A large-scale dataset of part-separated 3D objects.** We construct *PartObjectNet*, a dataset containing over 200K objects with more than 1M annotated parts, providing rich and structured supervision for learning complete and coherent part-level geometry.

2 Related Work

2.1 3D Generation

Recent advances in 3D generation have enabled the synthesis of high-quality 3D objects from text, images, or learned shape distributions [11–22]. Early approaches often relied on category-specific models or 2D-driven pipelines, while more recent methods adopt 3D-native generative models operating in geometry-aware latent spaces, such as 3D latent diffusion frameworks [12, 13, 23]. These models significantly improve geometric fidelity and scalability, and some recent works further enable direct mesh generation with well-structured topology and clean connectivity, without post-processing [24, 25]. Despite this progress, most existing 3D generation methods focus on whole-object synthesis and produce monolithic representations without explicit part structure. While effective for visualization and rendering, such representations offer limited support for fine-grained editing, reassembly, or part-level control, motivating growing interest in part-level 3D generation.

2.2 Part-Level 3D Generation

Recent work on part-level 3D generation moves beyond monolithic object synthesis toward explicit component-based modeling. Existing methods can be broadly categorized into two paradigms: decomposition-based pipelines, which explicitly segment a whole object into surface parts and process each component separately; and part-structured generative models, which represent multiple components jointly within a unified generative framework.

2.2.1 Decomposition-Based Part Generation Pipelines

Decomposition-based pipelines approach part-level 3D generation by explicitly dividing a complete object into surface parts and reconstructing each component through part-level geometry completion. Given an whole object mesh, these methods first perform surface segmentation and then recover missing geometry for individual parts before assembling them into a full shape [1, 2, 5, 6, 26]. Representative work such as HoloPart [1] exemplifies this paradigm by performing holistic surface segmentation followed by part-wise geometry completion. While explicit decomposition enables controllable manipulation of individual components, part reconstruction is primarily guided by local surface cues within each segment. Cross-part consistency is enforced only after each part has been completed, rather than influencing how parts are reconstructed in the first place. As a result, these methods struggle when parts are heavily occluded or contain large missing surface regions, where local geometry alone provides insufficient constraints to infer complete and semantically consistent components. This often leads to misaligned assemblies or inconsistent relative scales in the reconstructed shapes.

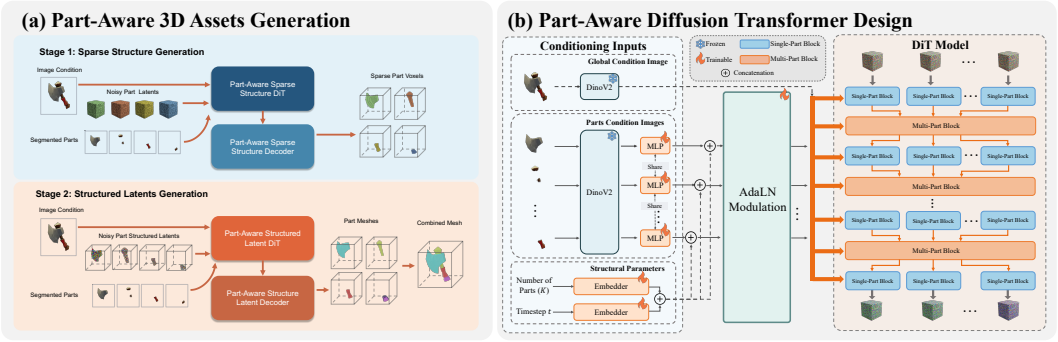


Figure 3: **Overview of Seg3DParts. (a) Part-aware 3D asset generation pipeline.** Seg3DParts adopts a two-stage framework. Stage 1 predicts part-aware sparse voxel structures from the input image and segmentation. Stage 2 generates geometry-aware latents for each part voxel, which are decoded into meshes aligned in a shared canonical space for direct assembly. **(b) Part-aware DiT.** The DiT used in both stages follows the same part-aware design. Global image features from frozen DINOv2 [29] are injected via cross-attention, while part-specific features and structural parameters modulate the network through AdaLN [30]. The DiT alternates single-part blocks for intra-part modeling and multi-part blocks for cross-part interaction, enabling coherent multi-part generation.

2.2.2 Part-Structured Generative Models

Part-structured generative models synthesize objects by jointly representing multiple components within a unified latent space and generating them in an end-to-end manner [3, 7, 9, 10, 27]. By coupling parts during generation, these methods encourage global structural coherence and produce plausible multi-part assemblies.

Despite these advantages, parts are typically encoded as implicit latent slots without explicit semantic specification. As a result, there is no stable correspondence between latent slots and interpretable object components, making it difficult to explicitly assign, constrain, or manipulate individual parts. The number, identity, and granularity of parts are therefore often implicitly fixed by model design, limiting flexibility under alternative decompositions.

Recent work such as OmniPart [28] explores incorporating segmentation into part-level generation by partitioning a voxel representation using global segmentation (e.g., SAM). However, segmentation is used as a global partitioning signal rather than explicitly defining part identities during generation. As a result, part allocation and spatial assignment remain implicitly determined, which can lead to ambiguity in part boundaries and merged components, especially under occlusion.

Together, these methods rely on implicit modeling of part identity and spatial allocation, rather than explicitly grounding parts to observable signals.

3 Method

We formulate part-level 3D generation as a segmentation-grounded conditional generation problem, where segmentation explicitly defines part identity and provides a direct interface for controllable generation. To this end, we propose **Seg3DParts**, a segmentation-grounded multi-part 3D generation framework built upon the structured latent representation and rectified-flow pipeline of TRELLIS [13]. As illustrated in Fig. 3 (a), given an input image and part segmentation, Seg3DParts conditions each component on localized appearance cues for part-specific geometry generation. We first introduce the structured latent backbone based on TRELLIS, followed by segmentation-grounded part conditioning and multi-part latent interaction for coherent generation. Finally, we describe canonical-space decoding and training objectives.

3.1 Preliminaries

Seg3DParts builds upon the structured latent 3D generation framework of TRELLIS, which represents 3D shapes using sparse voxel structures and synthesizes geometry through a two-stage generative process.

TRELLIS consists of *Sparse Structure Generation* and *Structured Latents Generation*. In the first stage, a VAE together with a rectified-flow DiT models sparse voxel occupancy, defining the global spatial structure without encoding geometric details. In the second stage, conditioned on the sparse structure, a separate VAE and DiT generate structured latents that capture fine-grained geometry and are decoded into surface meshes. Following TRELLIS, we refer to these components as the *sparse structure VAE/DiT* and *structure latent VAE/DiT*, respectively.

In this work, we adopt the sparse voxel representation and the two-stage VAE-based rectified-flow framework of TRELLIS as our backbone. Unless otherwise specified, all VAEs and DiTs described in the following sections are instantiated in a part-aware manner, where latent representations are maintained per part rather than for the whole object.

3.2 Segmentation-Grounded Part Conditioning

Seg3DParts grounds each generated component in localized image evidence via segmentation-conditioned features, enabling explicit part identity specification during generation, as illustrated in Fig. 3 (b). This design enables explicit part-level specification and provides direct control over decomposition and generation from a single input view, and is applied to both *part-aware sparse structure DiT* and *part-aware structure latent DiT*.

Given an input RGB image I, for each part k, we obtain a part segmentation either from an external segmentation model (e.g., SAM [31]) or from user-provided annotations. We extract the corresponding region from I to form a part-conditioning image I^k_{part}, which provides localized visual evidence for that component.

Each part-conditioning image I^k_{part} is encoded by a frozen DINOv2 backbone to extract high-level visual features, followed by a lightweight trainable MLP that maps these features into a compact appearance embedding $\mathbf{v}_k$. The embedding captures the visible attributes of the corresponding part and remains robust to partial or missing observations caused by occlusion. We concatenate $\mathbf{v}_k$ with the diffusion timestep embedding and an encoding of the total number of parts K to form a part-specific conditioning vector $\mathbf{c}_k$. This vector is projected to the modulation dimension and injected into all DiT layers via Adaptive Layer Normalization (AdaLN).

In addition to part-wise conditioning, the global-conditioning image I is encoded by DINOv2 to provide global contextual information, which is injected to the model via cross-attention.

By separating part-level modulation from global contextual conditioning, Seg3DParts ensures that each component is grounded in its corresponding image region while remaining consistent with the overall object structure. This design enables fine-grained control over part identity and appearance, supports varying decomposition granularities, and allows the model to handle occluded or invisible components.

3.3 Multi-Part Latent Interaction

While segmentation-grounded part conditioning specifies the appearance of individual components, coherent 3D assemblies further require explicit reasoning across parts during generation. Relative placement, scale consistency, and structural compatibility cannot be reliably inferred from per-part cues alone, especially under occlusion or limited visual evidence.

Seg3DParts introduces *multi-part latent interaction* throughout the generative pipeline. All VAEs and rectified-flow DiTs operate on part-separated latent representations and periodically enable information exchange across components via *multi-part cross-attention*. Unlike standard cross-attention that treats all tokens uniformly, multi-part interaction is explicitly structured at the part level, preserving clear part identities while allowing global coordination across components.

Formally, let $\mathbf{X}^k \in \mathbb{R}^{N_k \times d}$ denote the latent tokens of part k. Multi-part interaction is applied after intra-part self-attention as an explicit inter-part information exchange step. For each part, contextual information from other components is incorporated by attending to their latent tokens:

$$\mathbf{X}^k_{\text{out}} = \mathbf{X}^k + \text{Attn}\big(\mathbf{Q} = \mathbf{X}^k,\ \mathbf{K} = [\mathbf{X}^j]_{j\neq k},\ \mathbf{V} = [\mathbf{X}^j]_{j\neq k}\big)\,, \tag{1}$$

where $[\cdot]$ denotes concatenation along the token dimension. The residual formulation preserves each part's internal representation while augmenting it with structural context from other components,

enabling coordinated reasoning over relative placement and inter-part relationships without collapsing parts into a shared representation.

By integrating multi-part latent interaction across all stages of generation, Seg3DParts produces coherent multi-part assemblies directly in a shared canonical space, eliminating the need for post-hoc alignment. Implementation details of the interaction blocks are provided in the appendix.

3.4 Canonical-Space Part Decoding

A key property of Seg3DParts is that all generated part meshes are decoded directly into a shared canonical space, without any post-hoc alignment such as translation, scaling, or optimization.

This is achieved by consistently preserving part-level spatial alignment throughout data preparation and model training. Parts are obtained by decomposing complete objects while retaining their original scale and placement, and all stages of encoding, generation, and decoding operate directly on these aligned representations.

As a result, at inference time, independently decoded part meshes are already correctly positioned with respect to one another. This design eliminates the need for explicit assembly and enables stable multi-part generation under varying decomposition granularity and partial observations.

3.5 Optimization Loss

Seg3DParts is trained using a combination of reconstruction and generative objectives across the two stages of the framework.

VAE Training. Both stage-1 and stage-2 VAEs are trained with a standard variational objective that combines reconstruction losses with KL regularization to encourage a smooth latent distribution.

In the sparse structure stage, the VAE focuses on capturing coarse part-level occupancy, providing a structural prior for downstream generation. In the structured latent stage, the VAE emphasizes accurate surface geometry reconstruction conditioned on the generated sparse structure.

For stable supervision, decoded representations are normalized before computing reconstruction losses. Detailed loss formulations and training configurations are provided in the appendix.

DiT Training. The rectified-flow DiTs in both stages are trained using a flow matching objective. Specifically, the DiTs learn to predict velocity fields that transform noise samples into target latent representations under the rectified-flow formulation.

4 Experiments

4.1 Dataset

Existing 3D datasets either focus on whole-object reconstruction without explicit part structure, or provide part annotations that are noisy, weakly aligned, and unsuitable for training generative models. To address this gap, we construct **PartObjectNet**, a large-scale dataset of high-quality part-separated 3D objects designed specifically for part-level 3D generation.

PartObjectNet is built by integrating data from Objaverse [32], Texverse [33], and PartNet [34]. We first automatically select objects with explicit part-level decompositions, retaining only instances with 2–15 parts to balance compositional diversity and structural complexity. We further enforce semantic and geometric quality by removing low-quality cases, including white-mesh placeholders, noisy scanned results, large scene-level assets, and objects with unreasonable or semantically inconsistent part decompositions.

After automated filtering followed by manual curation, the final dataset contains approximately 200K high-quality objects spanning diverse categories and decomposition styles. Each object is represented as a set of part meshes that preserve their original relative scale and spatial placement, providing spatially aligned part-level supervision suitable for learning inter-part relationships and controllable generation. For evaluation, we randomly hold out 500 objects from PartObjectNet as a test set, with no overlap with the training data. In addition, to assess generalization beyond the training distribution,

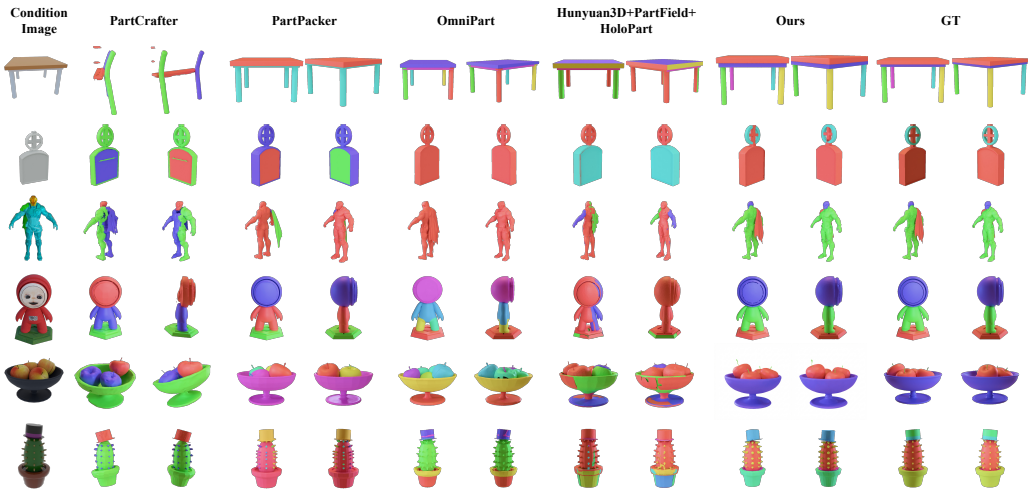


Figure 4: **Qualitative comparison of part-level 3D generation.** We compare Seg3DParts with decomposition-based pipelines and part-structured generative models. GT part segmentations are provided as input to Seg3DParts and OmniPart. Different colors indicate different semantic parts.

we further evaluate our method on an external benchmark, PartObjaverse-Tiny [5], which contains objects independent of our training data.

4.2 Implementation Details

Seg3DParts is trained following the two-stage pipeline of TRELLIS, with both stages extended to support part-level generation. In both the sparse structure stage and the structured latents stage, multi-Part Latent Interaction layers are inserted into the VAEs and diffusion transformers to enable cross-part information exchange. For the structured latents stage, we replace the voxel-feature projection used in TRELLIS with a mesh-based latent encoder similar to TripoSF [35], which improves geometric expressiveness and benefits the reconstruction of occluded regions. Both VAEs are trained using 8 NVIDIA A800 GPUs for approximately two days. The diffusion transformers in the two stages are trained using rectified-flow objectives on 16 NVIDIA A800 GPUs for approximately one weeks.

4.3 Evaluation Protocol

We evaluate Seg3DParts on 3D geometric quality at both the global object level and the part level. All metrics are computed on the held-out test set of 500 objects and PartObjaverse-Tiny.

Global-Level Geometry Evaluation. To assess overall geometric fidelity, we concatenate all generated part meshes into a single mesh and compare it with the ground-truth object. Generated and ground-truth meshes are normalized into a common $[-1, 1]^3$ space. We report the Chamfer Distance (CD) and F-Score (FS) at a threshold of 0.1, computed by uniformly sampling 16K points from each mesh surface. Lower CD and higher FS indicate better global geometric accuracy.

Part-Level Geometry Evaluation. To evaluate individual component quality, each generated part mesh is compared with its corresponding ground-truth part. We report per-part CD and FS using the same evaluation protocol as the global-level metrics. In addition, we measure volumetric consistency using Intersection over Union (IoU), where both generated and ground-truth parts are voxelized into a $64 \times 64 \times 64$ grid within the shared canonical space.

Part Overlap Evaluation. To assess part disentanglement and spatial separation, we compute the average pairwise IoU between generated parts using the same 64^3 voxelization. Lower overlap indicates reduced interpenetration and better geometric decoupling between components.

4.4 Comparison with State-of-the-Art Methods

We conduct a comprehensive comparison with state-of-the-art methods following two representative pipelines for part-level 3D generation and reconstruction.

Table 1: Global geometry (FS, CD) and part-overlap (IoU) evaluation on both PartObjectNet and PartObjaverse-Tiny.

Method	PartObjectNet			PartObjaverse-Tiny		
	FS@0.1↑	CD↓	IoU↓	FS@0.1↑	CD↓	IoU↓
PartCrafter	0.698	0.248	0.050	0.731	0.182	0.049
PartPacker	0.876	0.115	0.033	0.802	0.138	0.033
OmniPart	0.885	0.108	0.057	0.763	0.168	0.041
Hunyuan3D2.1 + PartField	0.860	0.120	0.031	0.793	0.144	0.028
Hunyuan3D2.1 + PartField + HoloPart	0.858	0.121	0.067	0.796	0.143	0.041
Ours	0.917	0.085	0.012	0.810	0.128	0.018

Decomposition-Based Part Generation Pipelines. This class of methods follows a sequential strategy that first obtains a complete object mesh and then decomposes it into parts for further processing. We adopt a representative pipeline combining **PartField** [6] and **HoloPart** [1]. Specifically, PartField is used to perform surface-based part segmentation on a reconstructed whole-object mesh, while HoloPart completes the geometry of each segmented part. To obtain the initial whole-object mesh, we employ the state-of-the-art single-view reconstruction model **Hunyuan3D 2.1** [14]. This pipeline reflects a commonly used practice where part-level generation is achieved by post-hoc decomposition and completion based on a reconstructed full shape.

Part-Structured Generative Models. We further compare against recent generative models that explicitly incorporate part structure into the generative process. **OmniPart** [28] is built entirely upon the TRELLIS framework and performs part-level generation by autoregressively predicting part bounding boxes in the sparse voxel space produced by Stage-1, followed by fine-tuning Stage-2 to synthesize geometry for each predicted part. **PartCrafter** [7] builds upon a pretrained whole-object generative model and extends it to part-aware generation by introducing explicit part tokens and structured attention. It alternates local and global attention across DiT layers, enabling parts to be generated jointly while modeling their interactions. **PartPacker** [9] is an end-to-end part-level 3D generation framework that utilizes a dual-volume packing strategy. It organizes an arbitrary number of parts into two complementary volumetric latent spaces based on a bipartite contraction formulation. This strategy maintains geometric separation between contacting parts while promoting inter-part consistency through a fixed-length latent representation.

Quantitative Results. We quantitatively evaluate Seg3DParts against state-of-the-art baselines in terms of global geometry fidelity, inter-part overlap, and part-level geometric accuracy. As shown in Table 1 and Table 2, Seg3DParts achieves the best overall performance, obtaining the highest F-Score and lowest Chamfer Distance among all evaluated methods, indicating superior global geometric fidelity. In addition, our method produces a substantially lower part-overlap IoU, suggesting that individual components are generated with clearer spatial separation and less interpenetration. These improvements stem from generating multiple parts directly in a shared canonical space, where part geometry and spatial relationships are modeled jointly, while segmentation-aware conditioning explicitly specifies which component is being generated, and multi-part latent interaction allows each part to adapt its geometry in response to other components.

Table 2: Part-level geometry evaluation on PartObjectNet and PartObjaverse-Tiny. Only methods with explicit part correspondence are reported.

Method	PartObjectNet			PartObjaverse-Tiny		
	FS@0.1↑	CD↓	IoU↑	FS@0.1↑	CD↓	IoU↑
OmniPart	0.565	0.417	0.551	0.455	0.418	0.429
Ours	0.774	0.192	0.781	0.639	0.310	0.702

We further evaluates part-level geometric accuracy for methods that preserve explicit part correspondence. Among existing approaches, only OmniPart supports conditioning on a given part segmentation and maintains a stable correspondence between generated parts and ground-truth components, making it suitable for fair part-level comparison. Seg3DParts significantly outperforms OmniPart across all metrics, with large gains in both F-Score and Chamfer Distance. These results indicate that Seg3DParts not only improves global shape quality, but also produces more accurate and well-aligned individual components, validating its effectiveness for controllable part-level 3D generation.

Qualitative Results. Figure 4 presents qualitative comparisons of part-level 3D generation from a single image. PartCrafter is able to generate diverse component geometries, but often struggles to maintain consistent semantic part separation, leading to fragmented or ambiguous part boundaries. PartPacker and decomposition-based pipelines Hunyuan3D + PartField + HoloPart generally produce plausible overall shapes; however, their part-level controllability is unstable, with inconsistent part

Table 3: Ablation study of multi-part latent interaction and segmentation-aware part conditioning. Metrics are reported for global geometry, part overlap, and part-level accuracy.

Method	Global-Level		Part-Overlap	Part-Level		
	FS@0.1↑	CD↓	IoU↓	FS@0.1↑	CD↓	IoU↑
Full Model	0.856	0.129	0.054	0.631	0.467	0.649
w/o Multi-Part Interaction	0.838	0.138	0.103	0.531	0.534	0.434
Single Whole-Object Conditioning	0.840	0.137	0.126	0.488	0.589	0.444

identities and noticeable variations in relative placement across instances. Although OmniPart explicitly models part structure by conditioning on a whole-object segmentation map, it frequently exhibits incomplete or merged parts, indicating challenges in robust part separation and identity preservation. In contrast, Seg3DParts consistently generates complete, well-separated part meshes with accurate alignment and coherent spatial relationships across diverse object categories. These qualitative results demonstrate the effectiveness of segmentation-aware conditioning and explicit cross-part interaction for reliable part-level 3D generation.

4.5 Ablation Study

We conduct ablations to evaluate each component of our framework. For efficiency, we train all variants on 1K randomly sampled objects for 5K steps under a consistent setup. Quantitative and qualitative results are shown in Table 3 and Fig. 5, respectively.

Effect of Multi-Part Latent Interaction. We first remove the proposed multi-part latent interaction, replacing all multi-part blocks with single-part blocks while keeping all other components unchanged. As shown in Table 3 and Fig. 5, this variant exhibits a consistent degradation across all metrics. While the drop in global geometry quality is moderate, the generated parts tend to intersect or overlap in 3D space, as visualized in Fig. 5. This indicates that without explicit cross-part interaction, parts are generated more independently, leading to weaker structural coordination and increased interpenetration.

Effect of Segmentation-Aware Part Conditioning. We further replace part-wise segmentation conditioning with a single whole-object segmentation map. Although global shape quality remains comparable, Fig. 5 shows that part boundaries become ambiguous and several components are incorrectly shaped or misplaced. This results in noticeably degraded part-level fidelity and overlap metrics, highlighting the importance of localized, part-specific conditioning for accurate part synthesis.

Controllability via Segmentation Input. Beyond improving part fidelity, segmentation-aware conditioning also enables explicit control over part placement. As shown in Fig. 6, we fix the input image and vary only the part segmentation. The generated results faithfully follow the provided segmentation, producing distinct and consistent part layouts for the same object. This demonstrates that Seg3DParts allows precise, segmentation-driven control over part decomposition and spatial configuration, rather than relying on a fixed or implicit part structure.

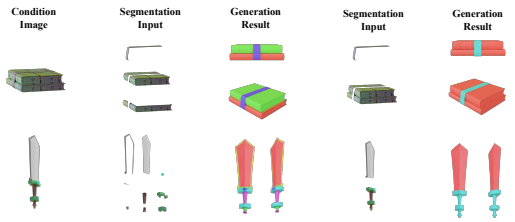


Figure 6: **Segmentation-driven part control.** Different segmentation inputs on the same image produce distinct and accurate part layouts.

5 Conclusion

We presented Seg3DParts, a segmentation-based framework for controllable part-level 3D generation, which explicitly leverages 2D semantic segmentation to guide structured 3D part synthesis from a single RGB image. By combining explicit part grounding with joint multi-part reasoning, our method directly produces coherent and well-aligned part meshes without post-hoc alignment. Experiments demonstrate that Seg3DParts achieves stronger geometric fidelity and part-level controllability than existing baselines.

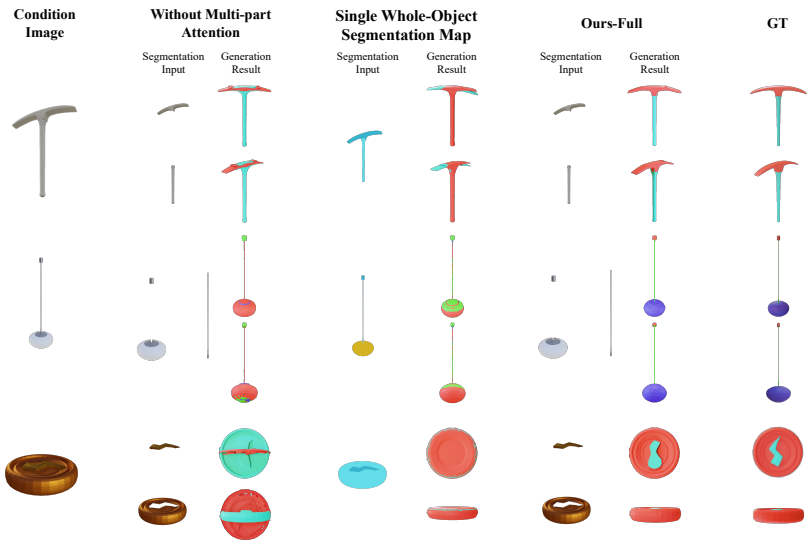


Figure 5: **Ablation study on part-level generation.** Removing multi-part attention leads to interpenetrating and incoherent parts, while replacing part-wise segmentation with a single object mask causes unstable decomposition. The full model produces well-separated, structurally coherent parts that closely match the ground truth.

Acknowledgments and Disclosure of Funding

This research received no specific grant from any funding agency. Computational resources were provided by the Envision Lab, led by Prof. Ying-Cong Chen, at The Hong Kong University of Science and Technology (Guangzhou). The authors declare no competing interests.

References

[1] Y. Yang, Y.-C. Guo, Y. Huang, Z.-X. Zou, Z. Yu, Y. Li, Y.-P. Cao, and X. Liu, "Holopart: Generative 3d part amodal segmentation," *arXiv preprint arXiv:2504.07943*, 2025.

[2] X. Yan, J. Xu, Y. Li, C. Ma, Y. Yang, C. Wang, Z. Zhao, Z. Lai, Y. Zhao, Z. Chen *et al.*, "X-part: High fidelity and structure coherent shape decomposition," *arXiv preprint arXiv:2509.08643*, 2025.

[3] M. Chen, J. Wang, R. Shapovalov, T. Monnier, H. Jung, D. Wang, R. Ranjan, I. Laina, and A. Vedaldi, "Autopartgen: Autoregressive 3d part generation and discovery," *arXiv preprint arXiv:2507.13346*, 2025.

[4] M. Chen, R. Shapovalov, I. Laina, T. Monnier, J. Wang, D. Novotny, and A. Vedaldi, "Partgen: Part-level 3d generation and reconstruction with multi-view diffusion models," *arXiv preprint arXiv:2412.18608*, 2024.

[5] Y. Yang, Y. Huang, Y.-C. Guo, L. Lu, X. Wu, E. Y. Lam, Y.-P. Cao, and X. Liu, "Sampart3d: Segment any part in 3d objects," *arXiv preprint arXiv:2411.07184*, 2024.

[6] M. Liu, M. A. Uy, D. Xiang, H. Su, S. Fidler, N. Sharp, and J. Gao, "Partfield: Learning 3d feature fields for part segmentation and beyond," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 9704–9715.

[7] Y. Lin, C. Lin, P. Pan, H. Yan, Y. Feng, Y. Mu, and K. Fragkiadaki, "Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers," *arXiv preprint arXiv:2506.05573*, 2025.

[8] S. Dong, L. Ding, X. Chen, Y. Li, Y. Wang, Y. Wang, Q. Wang, J. Kim, C. Gao, Z. Huang, Z. Wang, T. Xue, and D. Xu, "From one to more: Contextual part latents for 3d generation," *arXiv preprint arXiv:2507.08772*, 2025.

[9] J. Tang, R. Lu, Z. Li, Z. Hao, X. Li, F. Wei, S. Song, G. Zeng, M.-Y. Liu, and T.-Y. Lin, "Efficient part-level 3d object generation via dual volume packing," *arXiv preprint arXiv:2506.09980*, 2025.

[10] L. Ding, S. Dong, Y. Li, C. Gao, X. Chen, R. Han, Y. Kuang, H. Zhang, B. Huang, Z. Huang, Z. Wang, D. Xu, and T. Xue, "Fullpart: Generating each 3d part at full resolution," *arXiv preprint arXiv:2510.26140*, 2025.

[11] J. Lin, X. Yang, M. Chen, Y. Xu, D. Yan, L. Wu, X. Xu, L. Xu, S. Zhang, and Y.-C. Chen, "Kiss3dgen: Repurposing image diffusion models for 3d asset generation," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 5870–5880.

[12] L. Zhang, Z. Wang, Q. Zhang, Q. Qiu, A. Pang, H. Jiang, W. Yang, L. Xu, and J. Yu, "Clay: A controllable large-scale generative model for creating high-quality 3d assets," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 4, pp. 1–20, 2024.

[13] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang, "Structured 3d latents for scalable and versatile 3d generation," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 21 469–21 480.

[14] Z. Zhao, Z. Lai, Q. Lin, Y. Zhao, H. Liu, S. Yang, Y. Feng, M. Yang, S. Zhang, X. Yang *et al.*, "Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation," *arXiv preprint arXiv:2501.12202*, 2025.

[15] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot one image to 3d object," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2023, pp. 9264–9275.

[16] X. Long, Y.-C. Guo, C. Lin, Y. Liu, Z. Dou, L. Liu, Y. Ma, S.-H. Zhang, M. Habermann, C. Theobalt *et al.*, "Wonder3d: Single image to 3d using cross-domain diffusion," *arXiv preprint arXiv:2310.15008*, 2023.

[17] T. Jia, D. Yan, D. Hao, Y. Li, K. Zhang, X. He, L. Li, J. Chen, L. Jiang, Q. Yin *et al.*, "Ultrashape 1.0: High-fidelity 3d shape generation via scalable geometric refinement," *arXiv preprint arXiv:2512.21185*, 2025.

[18] S. Wu, Y. Lin, F. Zhang, Y. Zeng, Y. Yang, Y. Bao, J. Qian, S. Zhu, X. Cao, P. Torr *et al.*, "Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention," *arXiv preprint arXiv:2505.17412*, 2025.

[19] J. Yang, T. Shang, W. Sun, X. Song, Z. Cheng, S. Wang, S. Chen, W. Liu, H. Li, and P. Ji, "Pandora3d: A comprehensive framework for high-quality 3d shape and texture generation," *arXiv preprint arXiv:2502.14247*, 2025.

[20] Y. Li, Z.-X. Zou, Z. Liu, D. Wang, Y. Liang, Z. Yu, X. Liu, Y.-C. Guo, D. Liang, W. Ouyang *et al.*, "Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models," *arXiv preprint arXiv:2502.06608*, 2025.

[21] S. Wu, Y. Lin, F. Zhang, Y. Zeng, J. Xu, P. Torr, X. Cao, and Y. Yao, "Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer," *Advances in Neural Information Processing Systems*, vol. 37, pp. 121 859–121 881, 2024.

[22] B. Zhang, J. Tang, M. Niessner, and P. Wonka, "3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models," *ACM Transactions On Graphics (TOG)*, vol. 42, no. 4, pp. 1–16, 2023.

[23] Z. Li, Y. Wang, H. Zheng, Y. Luo, and B. Wen, "Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling," *arXiv preprint arXiv:2505.14521*, 2025.

[24] Y. Siddiqui, A. Alliegro, A. Artemov, T. Tommasi, D. Sirigatti, V. Rosov, A. Dai, and M. Nießner, "Meshgpt: Generating triangle meshes with decoder-only transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 19 615–19 625.

[25] B. Dai, L. R. Luo, Q. Tang, J. Wang, X. Lian, H. Xu, M. Qin, X. Xu, B. Dai, H. Wang *et al.*, "Meshcoder: Llm-powered structured mesh code generation from point clouds," *arXiv preprint arXiv:2508.14879*, 2025.

[26] K. Deng, Y. Yang, J. Sun, X. Liu, Y. Liu, D. Liang, and Y.-P. Cao, "Geosam2: Unleashing the power of sam2 for 3d part segmentation," *arXiv preprint arXiv:2508.14036*, 2025.

[27] Z. Li, W. Li, T. Wang, Z. Wang, J. Wu, and H. Wang, "Moca: Mixture-of-components attention for scalable compositional 3d generation," *arXiv preprint arXiv:2512.07628*, 2025.

[28] Y. Yang, Y. Zhou, Y.-C. Guo, Z.-X. Zou, Y. Huang, Y.-T. Liu, H. Xu, D. Liang, Y.-P. Cao, and X. Liu, "Omnipart: Part-aware 3d generation with semantic decoupling and structural cohesion," *arXiv preprint arXiv:2507.06165*, 2025.

[29] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," in *International Conference on Learning Representations (ICLR)*, 2024.

[30] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 8780–8794.

[31] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[32] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, "Objaverse: A universe of annotated 3d objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 142–13 153.

[33] Y. Zhang, L. Zhang, R. Ma, and N. Cao, "Texverse: A universe of 3d objects with high-resolution textures," *arXiv preprint arXiv:2508.10868*, 2025.

[34] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 909–918.

[35] X. He, Z.-X. Zou, C.-H. Chen, Y.-C. Guo, D. Liang, C. Yuan, W. Ouyang, Y.-P. Cao, and Y. Li, "Sparseflex: High-resolution and arbitrary-topology 3d shape modeling," *arXiv preprint arXiv:2503.21732*, 2025.

[36] T. Shen, J. Munkberg, J. Hasselgren, K. Yin, Z. Wang, W. Chen, Z. Gojcic, S. Fidler, N. Sharp, and J. Gao, "Flexible isosurface extraction for gradient-based mesh optimization," *ACM Trans. Graph.*, vol. 42, no. 4, jul 2023. [Online]. Available: https://doi.org/10.1145/3592430

[37] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," vol. 13, no. 4, 2004, pp. 600–612.

[38] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern Recognition*, 2018, pp. 586–595.

Appendix

A Implementation and Architectural Details

This appendix provides detailed descriptions of the architectural design, training configuration, and inference procedure of Seg3DParts. These details complement the main paper and are omitted from the core sections for clarity.

B Part-Aware Variational Autoencoders

Seg3DParts adopts a two-stage variational autoencoding scheme for modeling part-level 3D geometry, following the structured latent generation paradigm of TRELLIS [13]. Both stages operate on part-separated representations, where each semantic component is encoded and decoded independently while sharing network parameters. All VAEs are trained with standard variational objectives that combine reconstruction losses with KL divergence regularization. Decoded representations are normalized for stable supervision, while the canonical alignment of parts is preserved throughout training.

B.1 Stage-1 Part-Aware Sparse Structure VAE

The stage-1 VAE models coarse part-level structure using a sparse voxel occupancy representation at 64^3 resolution. To support part-level generation, each semantic part is treated as an independent element along the batch dimension. This design allows all parts to be processed in parallel with shared encoder–decoder weights, while preserving explicit separation between components.

Architecture. The encoder maps each part's sparse voxel grid into a compact latent representation using a hierarchical 3D convolutional backbone. For each part, a Gaussian posterior is inferred by predicting the mean and variance of the latent distribution. The decoder mirrors the encoder with symmetric upsampling layers and reconstructs binary voxel occupancy for each part at the original resolution.

Multi-Part Interaction. To capture global structural relationships among components, explicit cross-part interaction is introduced at the latent bottleneck. At this stage, part-wise latent feature grids exchange information via attention along the part dimension, allowing each component to reason about the presence, relative placement, and spatial extent of other parts while maintaining distinct identities. Multi-part interaction is restricted to the bottleneck, where representations are compact and semantically meaningful. Higher-resolution encoder and decoder stages operate independently on each part to preserve local geometric detail. The interaction is applied as a residual refinement and reduces to standard self-attention when only a single part is present.

Training Objective. The stage-1 VAE is trained to reconstruct sparse voxel occupancy using a Dice loss, which is well suited for supervising highly sparse structures. A KL divergence term regularizes the latent distribution. The training objective is

$$\mathcal{L}_{\text{stage1}} = \mathcal{L}_{\text{dice}} + \lambda_{\text{KL}}^{\text{ss}} \mathcal{L}_{\text{KL}}, \tag{2}$$

where the KL weight is set to $\lambda_{\text{KL}}^{\text{ss}} = 10^{-3}$. Decoded voxel grids are normalized before loss computation for numerical stability, without affecting the canonical alignment of part representations.

B.2 Stage-2 Part-Aware Structured Latent VAE

The stage-2 VAE models fine-grained surface geometry for each part. It adopts a mesh-based structured latent encoding strategy inspired by TripoSF [35], directly encoding surface geometry into sparse latent tokens. Compared to voxel-feature projection, this design provides stronger geometric expressiveness and improved robustness under partial or occluded observations.

Architecture. For each part, surface geometry is encoded into sparse structured latent tokens, which are processed by a sparse transformer encoder–decoder. Both the encoder and decoder consist of 12 transformer blocks and operate on part-separated latent representations, preserving explicit part identities throughout the network. The encoder predicts a Gaussian posterior for each part's structured latent code.

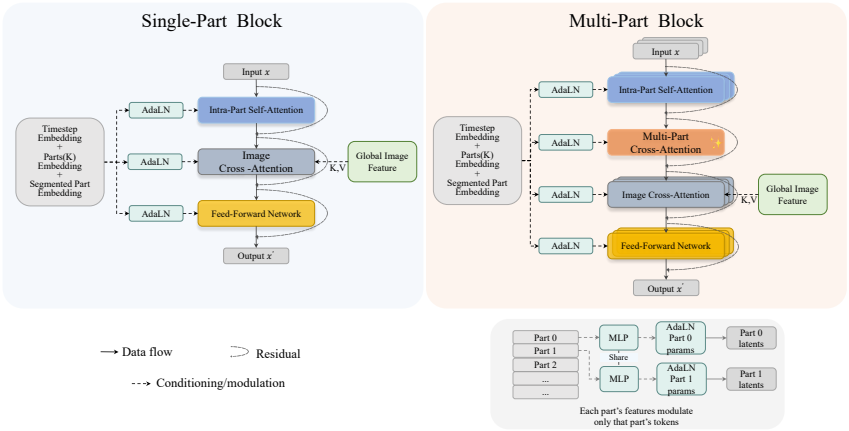


Figure 1: **Single-part and multi-part transformer block in Seg3DParts.** Single-part block model each semantic part independently using intra-part self-attention, image cross-attention, and feed-forward layers with AdaLN conditioning. Multi-part block insert an additional cross-part attention layer to enable information exchange across parts while preserving explicit part identities. AdaLN modulation is applied in a part-specific manner, so each part's conditioning affects only its own latent tokens.

Multi-Part Latent Interaction. To promote geometric coherence across components, multi-part latent interaction is integrated symmetrically into both the encoder and decoder. A subset of transformer blocks is replaced with multi-part interaction blocks, in which latent tokens of each part attend to those of other parts via cross-part attention, while the remaining blocks process parts independently. This periodic interaction enables global coordination among parts without collapsing them into a shared representation, and reduces to standard self-attention in the single-part case.

Mesh Decoding. The decoder progressively upsamples sparse latent features and reconstructs explicit surface geometry for each part using a FlexiCubes-based mesh extraction module. FlexiCubes [36] provides a differentiable and topology-adaptive surface representation, allowing high-quality mesh reconstruction while maintaining training stability. All part meshes are decoded directly into a shared canonical space, enabling coherent multi-part assemblies without post-hoc alignment.

Training Objective. The stage-2 part-aware structured latent VAE focuses exclusively on geometric reconstruction. Its training objective combines multiple geometry-aware supervision signals together with KL regularization on the structured latent distribution:

$$\begin{aligned}\mathcal{L}_{\text{VAE}}^{\text{sl}} = \mathcal{L}_{\text{mask}} + \lambda_{\text{depth}}\,\mathcal{L}_{\text{depth}} + \lambda_{\text{tsdf}}\,\mathcal{L}_{\text{tsdf}} \\ + \mathcal{L}_{\text{normal}}^{\text{perceptual}} + \lambda_{\text{KL}}^{\text{sl}}\,\mathcal{L}_{\text{KL}},\end{aligned} \tag{3}$$

where $\mathcal{L}_{\text{mask}}$ supervises the rendered silhouette, $\mathcal{L}_{\text{depth}}$ is computed using a Smooth-ℓ_1 loss on rendered depth maps, and $\mathcal{L}_{\text{tsdf}}$ supervises the truncated signed distance field.

The perceptual normal loss is defined as

$$\mathcal{L}_{\text{normal}}^{\text{perceptual}} = \mathcal{L}_{\text{normal}}^{\ell_1} + \lambda_{\text{ssim}}\,\mathcal{L}_{\text{ssim}}^{\text{normal}} + \lambda_{\text{lpips}}\,\mathcal{L}_{\text{lpips}}^{\text{normal}}, \tag{4}$$

which enforces both low-level and perceptual consistency on rendered surface normals using a combination of ℓ_1, SSIM [37], and LPIPS [38] losses.

The loss weights are set to $\lambda_{\text{depth}} = 10.0$, $\lambda_{\text{tsdf}} = 0.01$, $\lambda_{\text{ssim}} = 0.2$, $\lambda_{\text{lpips}} = 0.2$, and $\lambda_{\text{KL}}^{\text{sl}} = 10^{-6}$. Decoded geometry is normalized to $[-1, 1]$ before loss computation for numerical stability, without altering the canonical alignment of part representations.

C Transformer Architecture Details

The rectified-flow Diffusion Transformers (DiTs) used in both stages of Seg3DParts share a unified backbone architecture. Each DiT consists of 24 transformer blocks with a hidden dimension of

1024 and 16 attention heads. The two stages differ primarily in their input resolution and data representation: part-aware sparse structure DiT (Stage 1) operates on voxel grids at a resolution of 16^3 to model coarse part structure, while part-aware structure latent DiT (Stage 2) processes sparse voxel features embedded in a 64^3 coordinate space to capture fine-grained surface geometry. In both stages, part-level conditioning and cross-part coordination are integrated into the transformer through a mixture of *single-part blocks* and *multi-part blocks*, as detailed below.

C.1 Single-Part and Multi-Part Transformer Blocks

Single-Part Blocks. As illustrated in Fig. 1, single-part blocks operate independently on each semantic part. Each block consists of intra-part self-attention, image-conditioned cross-attention, and a feed-forward network, with residual connections throughout. Self-attention is confined to tokens within the same part, allowing local geometric structure to be modeled without interference from other components. These blocks constitute the majority of the network and primarily focus on refining per-part geometric details.

Multi-Part Blocks. As illustrated in Fig. 1, multi-part blocks extend the single-part design by inserting an additional *multi-part attention* layer immediately after intra-part self-attention. This layer enables information exchange across different parts, allowing the model to coordinate relative placement, scale, and structural compatibility among components. The remaining layers, including image-conditioned cross-attention and the feed-forward network, retain the same structure as in single-part blocks.

Multi-part blocks are sparsely interleaved throughout the transformer according to a fixed schedule, providing periodic global coordination while preserving part-level specialization. When only a single part is present, the multi-part attention naturally reduces to standard self-attention.

C.2 Conditioning with AdaLN

As illustrated in Fig. 1, both single-part and multi-part blocks use AdaLN for conditioning. The conditioning signal is formed by combining the timestep embedding, a parts-count embedding, and a per-part embedding extracted from the segmented image region. These per-part embeddings are processed by a shared MLP to produce AdaLN parameters (e.g., scale, shift, and gating) for each part, and are then applied to the corresponding part tokens via indexed broadcasting, so that the conditioning of part k affects only the tokens belonging to part k.

D Inference and Sampling Procedure

At inference time, Seg3DParts follows the same two-stage generation pipeline as training. Both sparse structure and structured latent DiTs are sampled using rectified-flow integration with sampling steps 50 and 30 respectively. Classifier-free guidance (CFG) is applied to image conditioning with a 3.0 guidance scale across all experiments. No test-time optimization or post-processing is used. All parts are decoded independently and naturally aligned in a shared canonical space.

E Dataset Construction Details

PartObjectNet is constructed by aggregating part-decomposed 3D objects from Objaverse, Texverse, and PartNet. We retain only objects with 2–15 semantic parts, which covers common composite structures while avoiding trivial or excessively fragmented cases. All candidates are further filtered through automatic checks and manual curation to remove low-quality meshes, scanned artifacts, scene-level assets, and semantically inconsistent part decompositions.

For each retained object, all parts preserve their original scale and relative spatial placement, providing aligned part-level supervision suitable for training part-aware generative models. The final dataset contains approximately 200K objects spanning diverse categories and decomposition styles.

F Fully Occluded Parts

Seg3DParts can also generate parts that are fully occluded in the input image. For a fully occluded part, we supply an all-black image crop as its part image condition. This is consistent with the training procedure: whenever a part is entirely hidden in the training view, it likewise receives an all-black image condition. The model therefore learns to interpret a blank image condition as the signal that no local visual evidence is available for that component. Its geometry is inferred from the global image context, the learned 3D prior, and the visible part streams through cross-part interaction (Fig. 2).

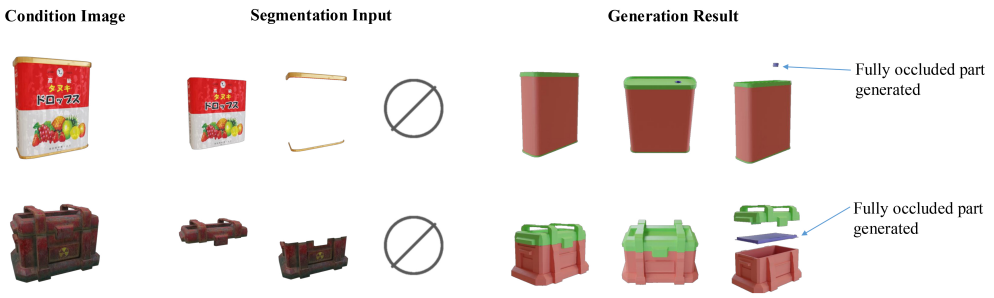


Figure 2: **Generation of fully occluded parts.** Given a single condition image (left), Seg3DParts generates a complete part decomposition (right) even for parts fully hidden in the input view, which receive an all-black image condition. Arrows indicate the recovered occluded parts: the circular pour-out opening on top of the juice tin (top) and the inner lid of the toolbox (bottom).

To quantify this behavior, we stratify parts by the fraction of their mask visible in the input view and compare the full model with the model without multi-part interaction. As shown in Table 4, interaction improves all three metrics in every visibility bin. At zero visibility, it increases part F-score by 23.5% and part IoU by 69.3%, while reducing part CD by 12.2%. The lower absolute scores in this bin are expected because the target geometry is underdetermined without local appearance evidence; nevertheless, the consistent improvements show that information from the other part streams contributes to recovering occluded geometry.

Table 4: **Part-averaged reconstruction by input-view visibility.** The full model and the model without multi-part interaction use the same controlled ablation setting. Best results within each visibility bin are shown in bold.

Visibility ratio	Method	Part $FS_{0.1}$ ↑	Part CD ↓	Part IoU ↑
0 (fully occluded)	Full	**0.384**	**0.539**	**0.171**
	w/o interaction	0.311	0.614	0.101
$(0, 0.25]$	Full	**0.487**	**0.495**	**0.258**
	w/o interaction	0.389	0.559	0.171
$(0.25, 0.50]$	Full	**0.533**	**0.490**	**0.318**
	w/o interaction	0.401	0.529	0.207
$(0.50, 0.75]$	Full	**0.560**	**0.466**	**0.570**
	w/o interaction	0.414	0.510	0.262
$(0.75, 1.00]$	Full	**0.676**	**0.377**	**0.750**
	w/o interaction	0.547	0.498	0.452

G Input-View Segmentation Adherence

We further evaluate whether the generated 3D parts follow the decomposition specified by the input masks. After applying the same global alignment used in our 3D evaluation, we render each output

from the input viewpoint without any part-wise alignment. Part-count match measures whether the number of generated meshes equals the number of input masks. Silhouette IoU measures the overlap between the rendered foreground and the reference object silhouette. Matched part-mask IoU measures the part-averaged overlap between the rendered parts and the input masks after permutation-invariant matching; when multiple rendered parts match the same input mask, their projections are merged before computing IoU.

Table 5: **Input-view segmentation adherence.** Both methods use the same ground-truth input masks and the same global alignment. No part-wise alignment is applied. Best results on each benchmark are shown in bold.

Dataset	Method	Part-count match ↑	Silhouette IoU ↑	Matched part-mask IoU ↑
PartObjectNet	OmniPart	38.8%	0.690	0.317
	Ours	**100.0%**	**0.788**	**0.548**
PartObjaverse-Tiny	OmniPart	40.5%	0.526	0.229
	Ours	**100.0%**	**0.680**	**0.482**

Seg3DParts preserves the requested part count on all evaluated objects and substantially improves both silhouette and matched part-mask IoU over OmniPart. These results show that the generated meshes more faithfully follow the supplied segmentation, rather than only reconstructing a plausible whole object.

H Limitations

While Seg3DParts enables flexible and coherent part-level generation, its performance depends on the quality of the input part segmentation. When the segmentation is of low quality or the part boundaries are semantically ambiguous, the generated geometry may degrade accordingly (Fig. 3).

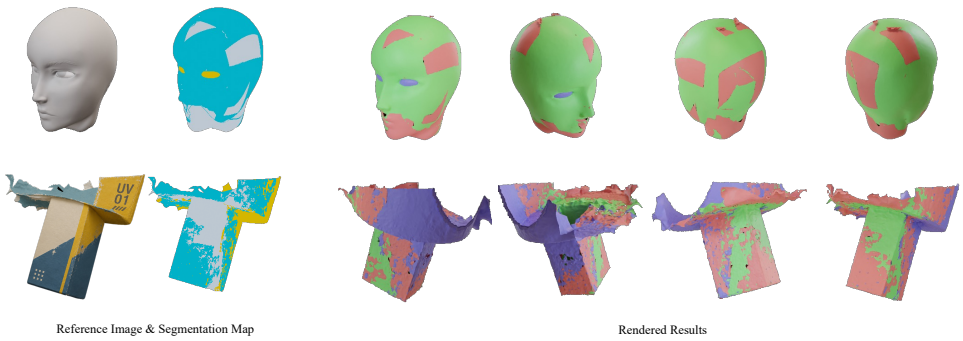


Figure 3: **Failure cases.** Seg3DParts may produce degraded geometry when the input part segmentation is of low quality or when part boundaries are semantically ambiguous.

I More Results

Figures 4 and 5 show additional qualitative results of Seg3DParts across diverse object categories and part decompositions, further demonstrating the generalization and coherence of our method.

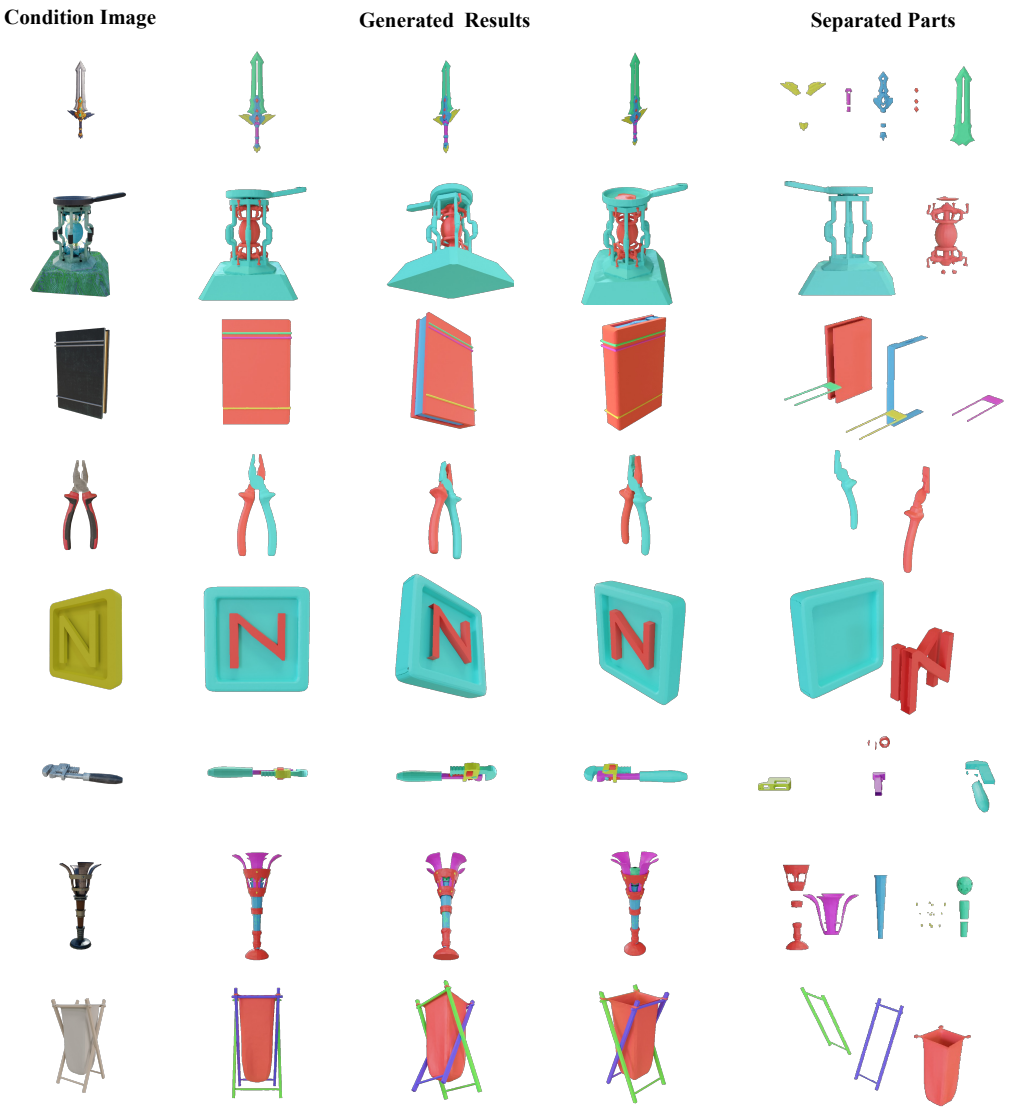


Figure 4: **Additional qualitative results (Part I)**.

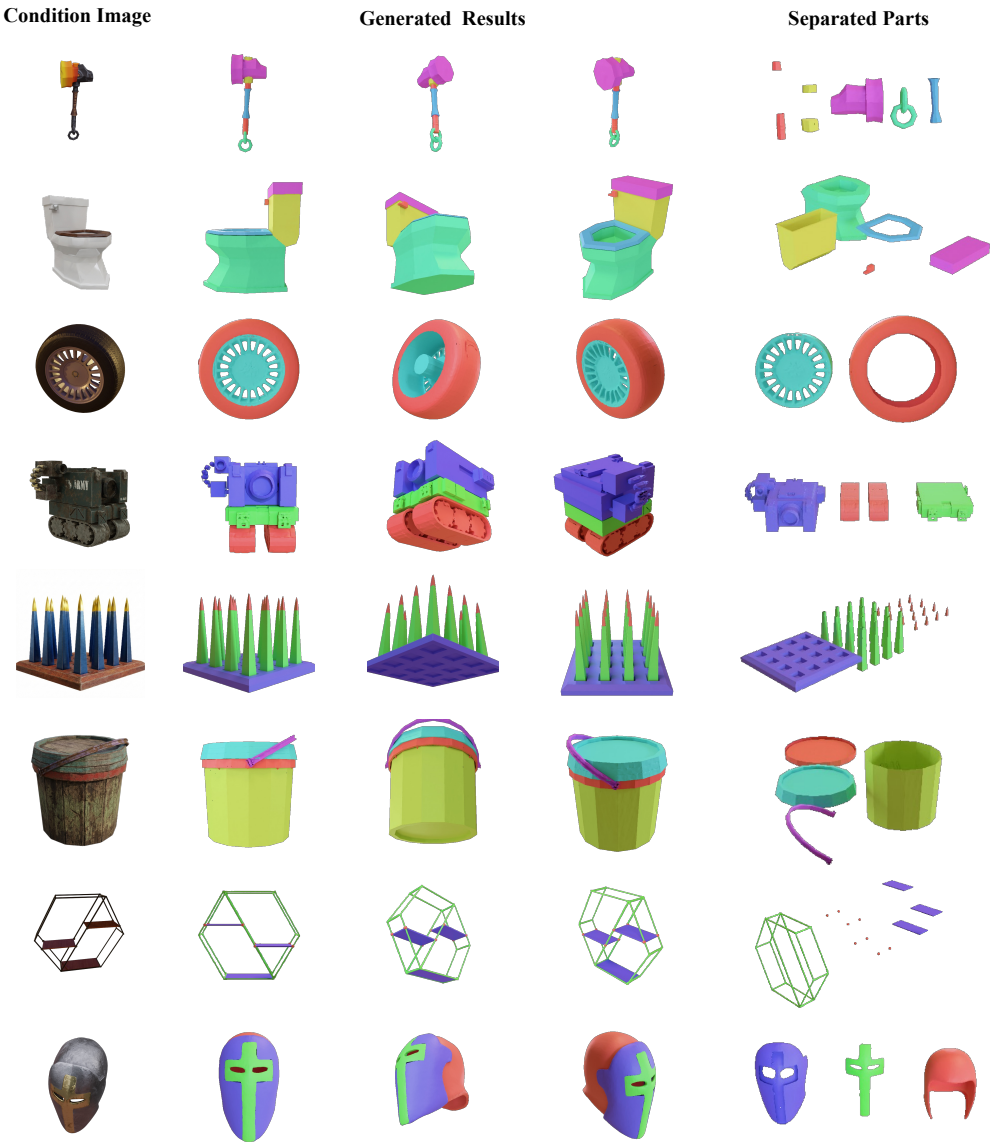


Figure 5: **Additional qualitative results (Part II).**.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes. The main claims in the abstract and introduction are limited to controllable part-level 3D generation, shared canonical-space assembly, and improved multi-part coherence, which are supported by the method description and experiments.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Yes. We discuss limitations including the dependence on segmentation quality.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not include theoretical results such as formal theorems or proofs. The contributions are primarily empirical and architectural.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Yes. We provide sufficient details of the model architecture, two-stage generation pipeline, part-aware conditioning mechanism, training setup, and evaluation protocol to enable reproduction of the main results.

Guidelines:

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

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

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

   Answer: [No]

   Justification: The code and data are currently being organised in preparation for publication.

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Yes. We provide details on the datasets, training and evaluation settings, model configuration, optimization strategy, and key hyperparameters in the main paper and Appendix, enabling a clear understanding of the experimental setup.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: We do not report error bars or statistical significance tests due to the high computational cost of training multiple 3D generative models. Instead, we support our conclusions with standard quantitative metrics and extensive qualitative comparisons.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Yes. We provide information on the compute resources used for training, including GPU type and approximate training time, allowing readers to estimate the computational requirements for reproducing the experiments.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Yes. The research conforms to the NeurIPS Code of Ethics and does not involve sensitive data, human subjects, or high-risk applications.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

Justification: Controllable part-level 3D generation can benefit downstream applications such as 3D content creation, asset editing, animation, robotics simulation, and education, by lowering the cost of producing structured 3D assets. As with other generative models, potential negative uses include the creation of misleading or unauthorized 3D replicas of real objects; we encourage responsible use and provenance tracking for generated assets.

Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release high-risk datasets or deployable generative models at submission time, and therefore no specific safeguards are required.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All existing assets used in this work, including the 3D object collections (Objaverse, Texverse, PartNet) and prior model architectures and baselines, are properly cited in the paper, and we use them in accordance with their respective licenses and terms of use.

Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: We introduce PartObjectNet, a part-separated 3D object dataset, whose construction, filtering, statistics, and intended use are documented in the main paper and the Appendix. Documentation will accompany the dataset upon its release.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

Answer: [N/A]

Justification: The paper does not involve research with human subjects, and therefore no IRB approval or equivalent review is required.

Guidelines:

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

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as an important, original, or non-standard component of the core methodology in this research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.