PARTS BY BOUNDARIES: NATIVE PART-AWARE 3D GENERATION WITH SEG-VOXEL REPRESENTATION

Anonymous authors
Paper under double-blind review



Figure 1: **Teaser.** Our method natively integrates part decomposition into 3D generation pipeline, jointly producing high-fidelity geometry and part-separation cues that explicitly encode the object's decomposable structure and ultimately yield coherent part assets.

ABSTRACT

Recent 3D generative models produce high-quality assets but typically treat objects as monolithic shapes. Existing part-aware approaches address this limitation through external segmentation, predefined layouts, or additional planning, leaving shape generation and part discovery largely decoupled. In this work, we propose a native part-aware 3D generation framework that models inter-part separation instead of explicit part assignments. Our key observation is that recovering part structure does not require explicit part assignments over the entire shape; it is sufficient to identify the interfaces where contacting parts should be separated. We therefore cast part decomposition as a local boundary prediction task, replacing open-ended part assignments with simple binary separation cues. To support this formulation, we introduce **Seg-Voxel**, a sparse voxel representation built upon the voxel design of TRELLIS.2 that jointly encodes geometry and inter-part boundaries. We further develop a native 3D generative pipeline that synthesizes Seg-Voxel to jointly generate object geometry and the separation structure required to partition it into individual components. The predicted boundaries are refined to ensure reliable geometric separation, then reassigned to adjacent components to recover coherent part assets. Experiments demonstrate that our method outperforms existing part-aware approaches in geometric quality, decomposition accuracy, and efficiency, enabling high-quality native part-aware generation without external segmentation or complex post-generation planning.

1 INTRODUCTION

Recent advances Xiang et al. (2025); Luo et al. (2026); Li et al. (2026b); Hunyuan3D et al. (2025) in 3D generative modeling have enabled the synthesis of increasingly high-quality 3D assets with detailed geometry and realistic appearance. Nevertheless, most existing generative models still represent an object as a single monolithic shape, limiting their applicability to downstream scenarios that require structured editing, animation, rearrangement, and fine-grained interaction. Part-aware 3D generation aims to address this limitation by generating objects together with meaningful com-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ponent structures. Existing approaches commonly obtain such structures through an additional decomposition process. Some methods Yang et al. (2024); Liu et al. (2025b); Li et al. (2026a) first generate a complete shape and subsequently segment it using pretrained 2D or 3D segmentation models, while others Yang et al. (2025); Ding et al. (2025); Dong et al. (2025) rely on predefined part layouts or planning modules to guide the independent or joint generation of multiple parts.

However, these paradigms commonly separate *shape generation* from *part discovery*: the object geometry is first determined, while its part structure is introduced through an additional prediction or planning stage. A more natural formulation is to incorporate part structure directly into the native 3D generation process, such that the model generates geometry and discovers its decomposition jointly. Yet doing so raises a key challenge: directly modeling explicit part assignments requires the generative model to account for highly variable and open-ended part compositions, including different numbers, identities, and spatial arrangements of parts.

In this work, we address this challenge by reformulating part-aware generation as modeling inter-part separation rather than explicit part assignments. We observe that the key to part decomposition lies in identifying the interfaces between contacting parts, which specify where the geometry should be disconnected. This allows part structure to be recovered without explicit per-surface part assignments. Based on this insight, we formulate part decomposition as a *local boundary prediction problem*, where the model predicts separation boundaries between contacting parts instead of explicit part labels over the entire shape. This transforms global part configurations into a set of local binary decisions, providing a simple and geometry-aligned representation of part structure that can be naturally integrated into native 3D generation.

Building on this formulation, we introduce **Seg-Voxel**, a sparse voxel representation derived from the voxel design of TRELLIS.2 Xiang et al. (2025) to jointly model object geometry and local inter-part boundaries. By encoding part structure through local separation cues rather than explicit part identities, Seg-Voxel naturally accommodates diverse part configurations within the same geometric framework. We further develop a native 3D generative pipeline that jointly synthesizes geometry and separation boundaries in a shared latent space, allowing the generated shape to be directly partitioned into disconnected components. A subsequent boundary processing stage ensures reliable separation and assigns the boundary regions to adjacent components to recover part assets. Together, these designs integrate part decomposition directly into the 3D generation process, enabling the native generation of coherent part-aware 3D assets.

We train and evaluate our method on HY3D-Bench Hunyuan3D et al. (2026) benchmarks under diverse generation settings. Across all evaluations, our approach consistently outperforms existing part-aware generation methods in terms of geometric quality, part decomposition accuracy, and overall efficiency. Without relying on external segmentation models or complex post-generation planning, our method integrates native part decomposition directly into the 3D generation process while preserving the high-quality geometry of the underlying TRELLIS.2 Xiang et al. (2025) framework, providing a simple and effective paradigm for generating coherent part-aware 3D assets.

2 Related Works

3D Shape Generation. Recent 3D generative models have demonstrated the ability to produce highly faithful 3D shape representations. Early methods Lin et al. (2023); Gao et al. (2022); Liu et al. (2023b;a) mainly relied on lifting powerful 2D generative priors into 3D, with DreamFusion Poole et al. (2022) being a representative example that introduced Score Distillation Sampling (SDS) to optimize implicit 3D representations from pretrained 2D diffusion models Podell et al. (2024), effectively addressing the lack of 3D training data at the time. More recently, 3D-native diffusion models built upon 3D-aligned latent representations have become a dominant paradigm for 3D generation. In particular, 3DShape2VecSet Zhang et al. (2023)-based methods Zhang et al. (2024); Li et al. (2025); Hunyuan3D et al. (2025); Li et al. (2026b) are representative approaches in this direction, which encode 3D shapes into a compact latent space for diffusion and decoding the generated representations into SDF fields, enabling high-quality end-to-end 3D generation. However, recovering the final 3D assets typically requires iso-surface extraction from the generated SDF via Marching Cubes Lorensen & Cline (1987), requiring both training and generated shapes to be watertight. Such watertight conversion can alter the native geometry of 3D assets and lead to geometric fidelity loss. Recent methods such as TRELLIS.2 Xiang et al. (2025) and FaithC Luo et al.

(2026) instead explore voxel-based representations that partition the original 3D shape into voxel-level surface elements and extract the attributes of the surface within each voxel, avoiding watertight conversion and enabling more faithful, near-lossless reconstruction of the original geometry.

3D Part Generation. As 3D assets are increasingly expected to support structured manipulation and interactive applications, decomposing objects into meaningful parts has become an important research direction. Existing part decomposition methods mainly build upon either 2D priors or native 3D supervision. For example, some methods Chen et al. (2024); Liu et al. (2024); Deng et al. (2025) like SaMesh Tang et al. (2024) extend 2D segmentation models Kirillov et al. (2023); Ravi et al. (2025) to 3D through multi-view post-processing, while more recent approaches Zhu et al. (2025); Ma et al. (2025; 2024); Jin et al. (2026) like SAMPart3D Yang et al. (2024) and PartField Liu et al. (2025a) directly learn point-wise semantic features from large-scale part-annotated meshes and perform part decomposition through feature clustering. Beyond decomposition, part-aware 3D generation aims to produce individual part assets while preserving their semantic structures. HoloPart Yang et al. (2026) generates parts independently from segmentation results, whereas OmniPart Yang et al. (2025), FullPart Ding et al. (2025), and others Dong et al. (2025); Liao et al. (2026); Wang et al. (2025) jointly generate multiple parts, with part layouts specified externally or predicted by dedicated layout planning modules. However, these approaches typically separate shape generation from part discovery, requiring a generated global shape first and an additional segmentation or planning stage to obtain the final parts. Developing 3D generative models that can discover and generate semantic parts within the native generation pipeline, without relying on external segmentation models or complex planning modules, is therefore a valuable direction for advancing part-aware 3D generation.

Native 3D Part Generation. Recently, several methods have explored incorporating part decomposition directly into the native 3D generation pipeline. PartCrafter Lin et al. (2025) jointly denoises concatenated part latents, but simultaneously resolving part assignments and inter-part geometric relationships can destabilize part decomposition. PartPacker Tang et al. (2025) simplifies generation through dual-volume packing, but requires bipartite part connectivity and risks incomplete decomposition due to geometric adhesion. SegVisGen Li et al. (2026a) takes a different perspective by representing parts through colors, whereas part decomposition is inherently tied to geometric structure and thus more naturally modeled within geometric generation. UniPart He et al. (2026) jointly generates geometry and segmentation through a VecSet-based VAE, but requires watertight shapes and imposes an additional learning burden on the VAE through part-label feature modeling. Rather than predicting explicit part labels, our method formulates part decomposition as robust separation-boundary generation that divides the geometry into disconnected components, followed by boundary reassignment to recover coherent part assets. This formulation simplifies the learning objective, making joint geometry and segmentation learning more amenable. Building upon the voxel representation of TRELLIS.2 Xiang et al. (2025), our approach enables the joint generation of geometrically consistent shapes and corresponding part assets.

3 METHODS

Our method aims to reformulate part-aware 3D generation into two stages that can be naturally incorporated into the native 3D generation pipeline: (1) jointly generating the shape and separation boundaries used to partition the geometry into disconnected components, and (2) refining the separation boundaries to ensure complete and robust separation between adjacent parts, and subsequently reassigning these boundaries to the corresponding neighboring components to recover part assets. To this end, building upon the sparse voxel representation introduced by TRELLIS.2 Xiang et al. (2025), we develop a novel sparse voxel representation, termed **Seg-Voxel**, to achieve the first objective. We introduce the relevant preliminaries and the design of Seg-Voxel in Sec. 3.1. And in Sec. 3.2, we introduce the VAE that maps Seg-Voxel to the latent space and the diffusion model for its generation. Finally, we refine the generated separation boundaries to improve their connectivity and ensure complete geometric separation, before reassigning the boundary regions to adjacent components to recover the final part assets. We detail this process in Sec. 3.3.

3.1 PRELIMINARIES AND PROBLEM FORMULATION

TRELLIS.2 Xiang et al. (2025) introduces a sparse voxel representation **O-Voxel** for modeling the 3D assets. Given a 3D shape, O-Voxel represents it as a collection of feature tuples associated with

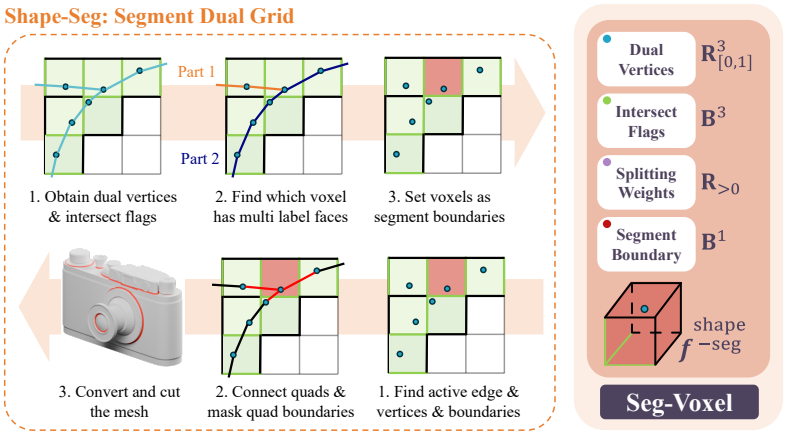

Figure 2: Illustration of Seg-Voxel and the instant bidirectional conversion between part-structured 3D asset and Seg-Voxel.

sparse voxels on a regular 3D grid of resolution $N \times N \times N$: $f = \{(f_i^{\text{shape}}, f_i^{\text{mat}}, p_i)\}_{i=1}^{L}$, where $p_i \in \{0, 1, \dots, N-1\}^3$ denotes the coordinate to the i-th active voxel, and empty voxels that do not intersect with the asset are set inactive. For local geometric information f_i^{shape}, the O-Voxel represents surfaces with *Flexible Dual Grid* formulation, where one vertex is defined per primal cell and one quadrilateral face per primal edge, connecting dual vertices in adjacent primal cells. This formulation computes in a Dual Contouring (Ju et al., 2002) like manner: edge that intersects the mesh activates the corresponding dual face and assigns the associated Hermite data to adjust the positions of the dual vertices following a quadratic error function:

$$\min_{v \in \text{voxel}} e(v) = \sum_i d^2_{\Pi,i} + \lambda_{\text{bound}} \sum_j d^2_{L,j} + \lambda_{\text{reg}}\, d^2_{\hat{\boldsymbol{q}}}. \tag{1}$$

Based on the above algorithm description, a mesh can be converted to a series of active voxels with geometric feature f_i^{shape} that comprises:

- **Dual vertex** $\boldsymbol{v}_i$, a vertex within the grid to represent local surface shape.
- **Edge intersection flags** $\boldsymbol{\delta}_i$, which determine quad connections among neighboring dual vertices. The intersections are saved on 3 predefined voxel edges along the X, Y, and Z axes respectively (The flags for the other 9 voxel edges are stored in neighboring voxels).
- **Splitting weights** γ_i, controlling how quadrilateral faces are subdivided into triangles.

And the mesh can be recovered from the O-Voxel by connecting dual vertices across intersected edges, forming quadrilateral faces among neighboring active voxels, and adaptively subdivided into two triangles guided by the splitting weights.

We propose **Seg-Voxel**, which incorporates native part structure into O-Voxel. To avoid modeling an open-ended space of part compositions, we formulate part-aware structure modeling as a *Boundary Prediction* problem. If two parts are not in contact in 3D space, they naturally form separate connected components in the extracted mesh and can be directly isolated. Therefore, we only need to identify the boundaries along which contacting parts should be disconnected, such that each part forms an independent connected component. In this way, part compositions can be represented through local binary decisions, requiring only one additional geometric feature:

- **Segment Boundary** $\boldsymbol{c}_i$, indicating whether the current voxel lies on an inter-part boundary.

The conversion between a part-structured mesh with per-face labels and its Seg-Voxel representation is summarized as follows (see Fig. 2 for an illustration).

Mesh $\rightarrow$ Seg-Voxel. Given a mesh, we first identify all voxel edges intersecting with the mesh surface and mark their neighboring voxels as active. While collecting Hermite data and calculating each dual vertex through the QEF in Eq. 1, we additionally record the part label of the mesh face associated with each surface intersection. If the intersected edges of an active voxel are associated with more than one part label, the voxel is marked as a segment-boundary voxel.

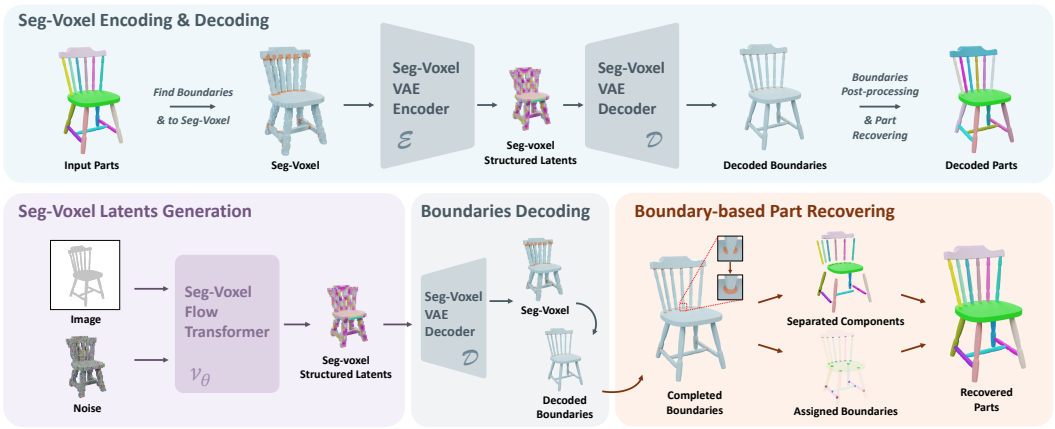

Figure 3: **Overall framework.** We first convert part meshes into Seg-Voxel representations and learn a structured latent space with a Seg-Voxel VAE. A flow-based generator synthesizes Seg-Voxel latents conditioned on the input prompt, starting from noise defined on the sparse voxel structure. The resulting latents are then decoded into geometry together with initial separation boundaries. The predicted boundaries are then refined to improve local continuity, used to separate the geometry into disconnected components, and finally reassigned to neighboring components to recover well-structured part assets

Seg-Voxel → Mesh. From a Seg-Voxel, we first follow the O-Voxel to recover the mesh faces. A face is marked as a segment-boundary face if it is associated with a segment-boundary voxel. We temporarily remove these boundary faces, thereby disconnecting adjacent parts. The resulting connected components are identified as individual parts. Finally, the removed boundary faces are reassigned to the neighboring parts through region growing, yielding the final part-labeled mesh.

3.2 Seg-Voxel VAEs and Generation

We then apply a VAE to learn a latent representation of Seg-Voxel for diffusion-based generation. The overall pipeline is illustrated in Fig. 3. Existing part-annotated datasets typically provide discrete part labels rather than explicit separation boundaries. We therefore first convert the original part annotations into local boundary labels. For each active voxel i, let $\mathcal{L}_i$ denote the set of unique part labels associated with the surface intersections within the voxel. The segment-boundary indicator is then defined as $\{c_i = \mathbb{I}\,(|\mathcal{L}_i| > 1)\}$, where $c_i = 1$ indicates that the surface within the voxel belong to multiple parts, and $c_i = 0$ otherwise. In this way, the original part annotations are converted into local binary boundary signals, which are incorporated into Seg-Voxel and modeled jointly with the geometric features. Building upon the original TRELLIS.2 Xiang et al. (2025) VAE architecture, we extend its input and output features to accommodate the additional segment-boundary attribute and train the VAE on the resulting Seg-Voxel data. Given a Seg-Voxel representation f^{seg}, the encoder $\mathcal{E}$ maps it into a compact latent representation $\boldsymbol{z}$, while the decoder $\mathcal{D}$ reconstructs the corresponding Seg-Voxel:

$$\boldsymbol{z} = \mathcal{E}(f^{\text{seg}}), \qquad \hat{f}^{\text{seg}} = \mathcal{D}(\boldsymbol{z}). \tag{2}$$

We follow the training procedure and VAE objective of TRELLIS.2, with an additional boundary reconstruction loss. Since boundary voxels are substantially less frequent than non-boundary voxels, we adopt a balanced binary cross-entropy loss:

$$\mathcal{L}_{\text{bdry}} = -\frac{1}{2|\mathcal{V}_+|}\sum_{i\in\mathcal{V}_+}\log\hat{p}_i - \frac{1}{2|\mathcal{V}_-|}\sum_{i\in\mathcal{V}_-}\log(1-\hat{p}_i), \tag{3}$$

where $\hat{p}_i$ is the predicted boundary probability for active voxel i; $\mathcal{V}_+$ and $\mathcal{V}_-$ denote the sets of active voxels with ground-truth labels $c_i = 1$ and $c_i = 0$, respectively. The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\text{VAE}}^{\text{base}} + \lambda_{\text{bdry}}\mathcal{L}_{\text{bdry}}, \tag{4}$$

where $\mathcal{L}_{\text{VAE}}^{\text{base}}$ denotes the original VAE objective and λ_{bdry} controls the contribution of boundary reconstruction. This joint objective yields a latent representation that captures both shape geometry and separation boundaries, providing the basis for subsequent Seg-Voxel generation.

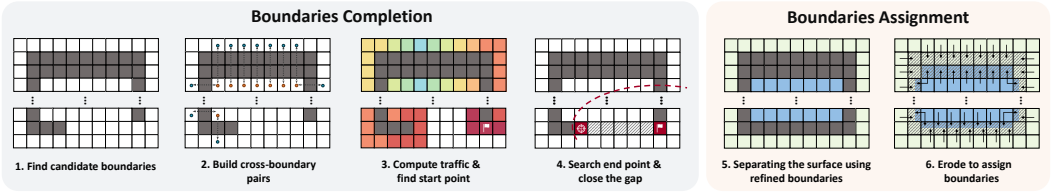

Figure 4: **Boundary-based part recovering.** We first (1) identify incomplete boundaries and (2) build cross-boundary pairs, then (3) compute traffic of each face as the number of shortest paths between cross-boundary pairs that pass through it. (higher face traffic indicated by the blue-to-red color scale above.) (4) Starting from the highest-traffic face (flag), we expand the search radius to identify open boundary ends (crosshairs) and connect them to close the gap. (5) Removing the refined boundary faces yields separate components, whose labels then (6) propagate synchronously to assign the boundary surface, forming the final coherent part surfaces.

We further build a flow-matching generation pipeline on top of the learned Seg-Voxel latent space. TRELLIS.2 factorizes 3D generation into two sequential stages: sparse structure synthesis and O-Voxel feature generation. We retain the sparse structure synthesis stage and adapt the original feature generator to generate Seg-Voxel features. Specifically, given the synthesized sparse structure $\int$ and condition $\boldsymbol{y}$, the flow model generates the Seg-Voxel latent $\hat{\boldsymbol{z}}^{\text{seg}}$ by evolving:

$$\hat{\boldsymbol{z}}^{\text{seg}} = \boldsymbol{v}_\theta \left(\boldsymbol{z}_t, t \mid \int, \boldsymbol{y} \right), \tag{5}$$

where $\boldsymbol{v}_\theta$ denotes the learned velocity field. The generated latent features $\hat{\boldsymbol{z}}^{\text{seg}}$ are then decoded by our VAE into Seg-Voxels as $\hat{f}^{\text{seg}} = \mathcal{D}(\hat{\boldsymbol{z}}^{\text{seg}})$, jointly recovering the 3D shape $\mathcal{S}$ and its separation boundaries $\mathcal{B}$. The predicted boundaries $\mathcal{B}$ provide the initial separation cues for the subsequent boundary-based part recovering stage.

3.3 Boundary-based Part Recovering

To recover individual parts from the generated geometry $\mathcal{S}$ and separation boundaries $\mathcal{B}$, we first complete potentially incomplete boundaries to separate the geometry into disconnected components and then assign boundary faces to the resulting components, as illustrated in Fig. 4.

Boundary Completion. The generated separation boundaries $\mathcal{B}$ provide the basis for decomposing the complete geometry $\mathcal{S}$. However, these boundaries may contain local gaps. Under our connected-component formulation, even a small gap can leave otherwise separate parts connected, causing them to merge into a single component. We therefore introduce a *path-traffic-guided boundary completion* procedure to identify and close such gaps.

Removing a valid separation boundary should disconnect its two sides into distinct connected components. An incomplete boundary violates this property: although its two sides may be locally separated by the predicted boundary, they can remain globally connected through residual paths around the boundary gap. We therefore identify boundary regions whose surrounding faces still belong to a single connected component, and construct cross-boundary face pairs whose connectivity remains intact after excluding the predicted boundary faces (as shown in step 2 in Fig 4). These pairs represent residual connections that should be blocked by the completed boundary. For each selected pair, we compute the shortest path within a neighborhood of the existing boundary while excluding boundary faces. We define the *path-traffic* of a face as the number of these residual paths traversing it, indicating locations where multiple residual connections converge. Starting from the highest-path-traffic face, we progressively expand the search radius to identify candidate paths that connect the open ends of the same boundary component. We rank these candidates according to their path length and the number of selected face pairs whose residual connectivity is removed by the candidate. The selected path is added to the boundary set to close the gap. This process yields the completed boundaries $\tilde{\mathcal{B}}$.

Boundary Assignment. After completion, removing $\tilde{\mathcal{B}}$ separates the geometry into disconnected components but leaves the boundary faces unassigned. Boundary bands between different parts may form a single connected region spanning multiple components. Copying such a region to every adjacent component would assign unrelated boundary faces to individual parts and introduce extraneous geometry. We therefore adopt an *erosion-based boundary assignment* procedure that assigns

Table 1: Quantitative comparison of VAE reconstruction. All values are multiplied by 10^2.

Method	CD ↓	$\text{F1}_{0.01}$ ↑	$\text{F1}_{0.005}$ ↑	mCD ↓	$\text{mF1}_{0.01}$ ↑	$\text{mF1}_{0.005}$ ↑
TRELLIS.2 - 512	0.0056	95.14	65.68	-	-	-
TRELLIS.2 - 1024	0.0055	95.17	65.84	-	-	-
Seg-Voxel - 512 (Ours)	0.0056	95.14	65.73	0.473	84.24	75.35
Seg-Voxel - 1024 (Ours)	**0.0055**	**95.17**	**65.87**	**0.456**	**86.73**	**79.12**

faces according to local connectivity. Specifically, neighboring component labels propagate synchronously into the boundary bands, progressively shrinking the unassigned region. When multiple labels reach an unassigned face in the same iteration, we retain and propagate all of them, allowing shared ownership without arbitrarily selecting a single label. The procedure continues until all boundary faces are assigned, yielding $\mathcal{B}^{\text{assign}}$. Following these steps, we obtain the final part assets from the initially separated geometry:

$$\mathcal{P}_i = \mathcal{S}_i \cup \mathcal{B}_i^{\text{assign}}, \qquad i = 1, \ldots, M, \tag{6}$$

where $\mathcal{B}_i^{\text{assign}}$ denotes the boundary elements assigned to the i-th disconnected component $\mathcal{S}_i$, and $\mathcal{P}_i$ represents the final part asset.

4 Experiments

4.1 Experimental Setup

Implementation Details. We adopt TRELLIS.2 (Xiang et al., 2025) as our base model and initialize both the Seg-VAE and Seg-SLAT flow model from their counterparts. Both models are fine-tuned using the AdamW optimizer. The Seg-VAE is trained for 350K steps with a batch size of 16 and a learning rate of 1×10^{-5}. The weight of the boundary-voxel loss is set to 0.1. The Seg-SLAT flow model is trained for 100K steps with a batch size of 32 and a learning rate of 1×10^{-4}. All experiments are conducted on eight NVIDIA A100 GPUs.

Datasets and Baseline. We construct our dataset by filtering the HY3D-Bench Part dataset (Hunyuan3D et al., 2026). The resulting dataset contains 110K 3D objects with approximately 1.6M annotated parts. We randomly hold out 300 objects for testing and use the remaining objects for training. We compare our method against five publicly available baselines: (i) PartPacker (Tang et al., 2025), (ii) PartCrafter (Lin et al., 2025), (iii) OmniPart (Yang et al., 2025), (iv) XPart (Yan et al., 2025), and (v) SegviGen (Li et al., 2026a). For the image-conditioned methods (i, ii, iii, and v), we provide the same input image as our method. For methods requiring a holistic input mesh (iv and v), we generate the mesh using TRELLIS.2, consistent with the base model used by our method. All baselines are evaluated using their publicly released implementations.

Evaluation Metrics. We evaluate the results at both whole-object and part levels, measuring overall coherence and geometric quality of individual parts. For whole-object, we merge all parts and report the chamfer distance (CD) and F-score with a distance threshold of 0.1. For part level, since direct correspondence between generated and ground-truth parts is unavailable, we adopt a best-match evaluation protocol following prior work (Liu et al., 2025b; Ma et al., 2025). We use an F-score threshold of 0.1 for generation and stricter thresholds of 0.01 and 0.005 for VAE reconstruction.

4.2 Main Results

Seg-Voxel VAE Reconstruction We first evaluate the reconstruction performance of our Seg-Voxel VAE across objects with varying part complexity. As shown in Fig. 5, the model faithfully reconstructs both the overall geometry and the part decomposition. In particular, the predicted segmentation boundaries closely follow the interfaces between adjacent parts, validating our formulation of part-structure modeling through boundary prediction. As reported in Tab. 1, our Seg-Voxel VAE achieves accurate part-level reconstruction while maintaining whole-object geometric fidelity comparable to the original TRELLIS.2 shape VAE (Xiang et al., 2025). Trained only at resolution of 512, Seg-Voxel generalizes directly to higher-resolution inference, where finer geometry and more localized boundary prediction further improve part reconstruction. Together, these results show that the proposed representation effectively preserves complex part structures without substantially degrading geometric reconstruction quality.



Figure 5: Reconstruction results of our Seg-Voxel VAE across increasing levels of part complexity.

Table 2: Quantitative comparison of part generation. All values except time are multiplied by 10^2.

Method	CD↓	$F1_{0.1}$ ↑	mCD↓	$mF1_{0.1}$ ↑	Time(s)↓
PartPacker (Tang et al., 2025)	0.506	94.74	5.869	71.14	23.76
PartCrafter (Lin et al., 2025)	0.802	90.21	7.772	63.52	104.99
OmniPart (Yang et al., 2025)	0.432	95.74	1.599	85.46	38.61
XPart (Yan et al., 2025)	0.407	95.74	4.441	69.17	442.89
SegviGen (Li et al., 2026a)	0.203	98.13	1.796	88.35	157.75
Seg-Voxel (Ours)	**0.203**	**98.14**	**1.242**	**91.03**	**20.68**

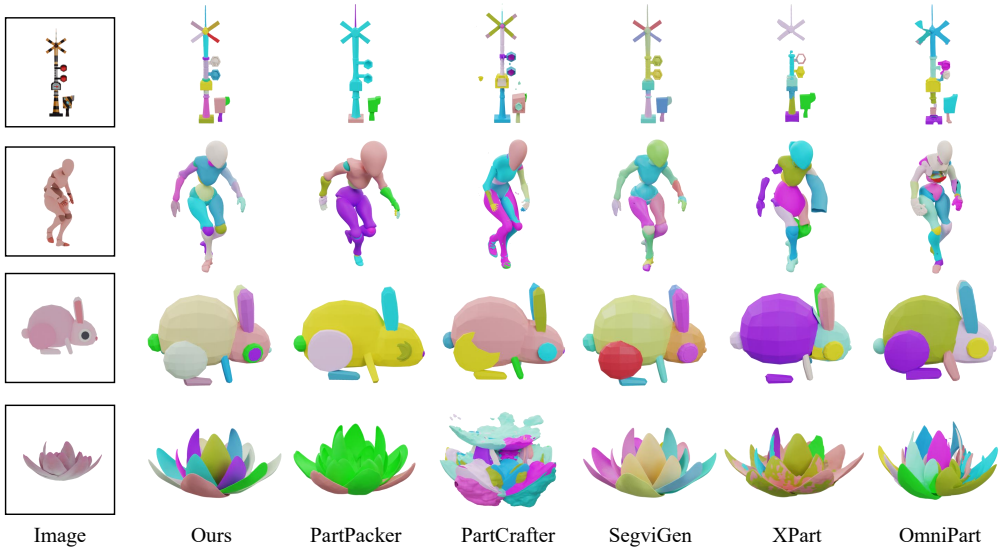

Figure 6: Qualitative results of part-level 3D generation for a given image.

Part Generation Results We compare our method with five representative methods for image-conditioned part-level 3D generation in Tab. 2, with qualitative comparisons presented in Fig. 6. At the whole-object level, our method achieves CD and F1 scores comparable to SegviGen, indicating that it preserves the base model's geometric fidelity while incorporating part structure. At the part level, our method achieves the lowest mCD and the highest $\mathrm{mF1}_{0.1}$ among all compared methods, indicating improved geometric fidelity of individual parts. In terms of computational efficiency, our method achieves the lowest inference time among the compared methods, demonstrating its efficiency in practical generation. As shown in Fig. 6, our method produces coherent overall geometry with distinct and structurally meaningful part decompositions. The generated parts fit together closely while retaining clear boundaries, as illustrated by the flower petals and the rabbit's body components. By contrast, existing methods can exhibit several failure modes. Methods that

Table 3: Ablation of the boundary completion.

Refinement	mCD ↓	mF1@0.1 ↑
No-completion	1.973	87.62
Dilation	1.328	90.55
Ours	**1.242**	**91.03**

Table 4: Ablation of the boundary assignment.

Assignment	mCD ↓	mF1@0.1 ↑
N-Copy	2.281	85.27
N-Max	1.311	89.78
Ours	**1.242**	**91.03**

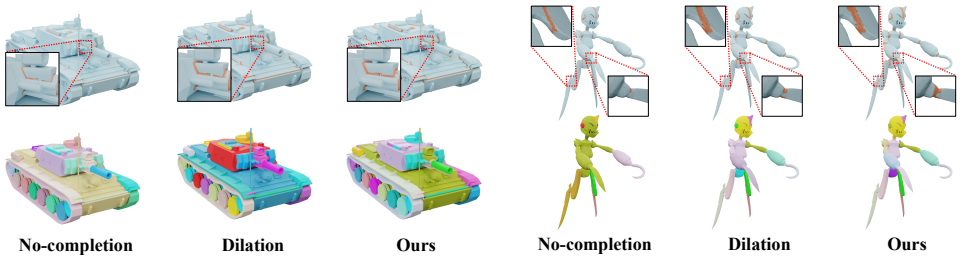

Figure 7: Ablation comparison of the boundary ablation methods.

jointly generate separate part geometries may produce gaps or overlaps between parts. For methods relying on layout planning, inaccurate part layouts can lead to missing small parts or fragmented components. SegviGen may merge geometrically adjacent parts when similar predicted label colors make them difficult to distinguish. These results further validate the effectiveness of our unified boundary-aware representation for jointly modeling global geometry and part-level structure.

4.3 Ablation Studies

Boundary Completion. To evaluate the effectiveness of boundary completion, we compare our path-traffic-guided method with two alternatives: *no completion* and *boundary dilation* with a fixed number of iterations. As shown in Tab. 3, both completion strategies improve part-level generation quality, while our method achieves the best results on both metrics. Since boundary gaps vary in size across different geometries, a fixed number of dilation iterations may leave larger gaps unresolved or excessively expand boundaries around smaller gaps. In contrast, our method uses path-traffic to identify locations where residual connections are concentrated and guide targeted boundary completion. As illustrated in Fig. 7, the path-traffic-guided strategy can close larger gaps that remain open after dilation, thereby separating parts that would otherwise remain connected.

Boundary Assignment. We also compare our erosion-based boundary assignment with two simple alternatives: *N-Copy*, which duplicates each boundary region across all adjacent parts, and *N-Max*, which assigns the entire region to the adjacent part with the largest number of adjacency connections. As shown in Tab. 4, our method achieves the lowest mCD and the highest $\mathrm{mF1}_{0.1}$. N-Copy can introduce extraneous geometry into individual parts by duplicating entire boundary regions, while N-Max assigns each boundary region to a single adjacent part regardless of its local connectivity. In contrast, our method progressively propagates component labels through the boundary region according to local connectivity, allowing multiple neighboring components to share boundary faces when appropriate and yielding better part-level reconstruction quality.

5 Conclusion

In this work, we presented a native part-aware 3D generation framework that integrates part decomposition directly into the 3D generation process. Instead of predicting explicit part assignments over the entire shape, we formulated part decomposition as local boundary prediction, where inter-part separation is represented through simple binary cues. Based on this formulation, we introduced **SegVoxel** to jointly model object geometry and inter-part boundaries, together with a native generative pipeline and a boundary post-processing procedure for recovering coherent part assets. Experiments demonstrate that our approach achieves improved geometric quality, decomposition accuracy, and efficiency compared with existing part-aware generation methods. By avoiding external segmentation models and complex post-generation planning while preserving high-quality geometry, our method provides a simple and effective framework for generating structured, coherent, and directly usable part-aware 3D assets.

DISCLOSURE OF AI USE

In this work, we used generative AI tools to assist with routine engineering tasks, such as implementing GPU acceleration for Seg-Voxel operations. For example, when implementing GPU acceleration, we first developed and validated CPU versions for easier debugging, and then used AI coding tools to assist in translating the verified operations into CUDA implementations. We checked output consistency between the AI-assisted implementations and the author-written implementations. The research ideas and methodology were developed independently by the authors. Synthetic dataset generation and mathematical proof development do not apply to this work. We take responsibility for the final content of this work, including all artifacts produced with the assistance of generative AI.

REFERENCES

Minghao Chen, Roman Shapovalov, Iro Laina, Tom Monnier, Jianyuan Wang, David Novotny, and Andrea Vedaldi. Partgen: Part-level 3d generation and reconstruction with multi-view diffusion models. *arXiv preprint arXiv:2412.18608*, 2024.

Ken Deng, Yunhan Yang, Jingxiang Sun, Xihui Liu, Yebin Liu, Ding Liang, and Yan-Pei Cao. Geosam2: Unleashing the power of sam2 for 3d part segmentation. *arXiv preprint arXiv:2508.14036*, 2025.

Lihe Ding, Shaocong Dong, Yaokun Li, Chenjian Gao, Xiao Chen, Rui Han, Yihao Kuang, Hong Zhang, Bo Huang, Zhanpeng Huang, Zibin Wang, Dan Xu, and Tianfan Xue. Fullpart: Generating each 3d part at full resolution, 2025.

Shaocong Dong, Lihe Ding, Xiao Chen, Yaokun Li, Yuxin Wang, Yucheng Wang, Qi Wang, Jaehyeok Kim, Chenjian Gao, Zhanpeng Huang, et al. From one to more: Contextual part latents for 3d generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8230–8240. IEEE, 2025.

Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances in neural information processing systems*, 35:31841–31854, 2022.

Xufan He, Yushuang Wu, Xiaoyang Guo, Chongjie Ye, Jiaqing Zhou, Tianlei Hu, Xiaoguang Han, and Dong Du. Unipart: Part-level 3d generation with unified 3d geom-seg latents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34227–34236, 2026.

Team Hunyuan3D, Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, Sheng Zhang, Zebin He, Di Luo, Haolin Liu, Yunfei Zhao, et al. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. *arXiv preprint arXiv:2506.15442*, 2025.

Team Hunyuan3D, :, Bowen Zhang, Chunchao Guo, Dongyuan Guo, Haolin Liu, Hongyu Yan, Huiwen Shi, Jiaao Yu, Jiachen Xu, Jingwei Huang, Kunhong Li, Lifu Wang, Linus, Penghao Wang, Qingxiang Lin, Ruining Tang, Xianghui Yang, Yang Li, Yirui Guan, Yunfei Zhao, Yunhan Yang, Zeqiang Lai, Zhihao Liang, and Zibo Zhao. Hy3d-bench: Generation of 3d assets, 2026. URL https://arxiv.org/abs/2602.03907.

Li Jin, Weikai Chen, Yujie Wang, Yingda Yin, Zeyu Hu, Runze Zhang, Keyang Luo, Shengju Qian, Xin Wang, and Xueying Qin. Cosmo3d: Open-world promptable 3d semantic part segmentation through llm-guided canonical spatial modeling, 2026. URL https://arxiv.org/abs/2603.01205.

Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. Dual contouring of hermite data. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 339–346, 2002.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Lin Li, Haoran Feng, Zehuan Huang, Haohua Chen, Wenbo Nie, Shaohua Hou, Keqing Fan, Pan Hu, Sheng Wang, Buyu Li, and Lu Sheng. Segvigen: Repurposing 3d generative model for part segmentation. *arXiv preprint arXiv:2603.16869*, 2026a.

Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Zhihao Li, Yufei Wang, Heliang Zheng, Yihao Luo, and Bihan Wen. Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. *Advances in Neural Information Processing Systems*, 38:118582–118600, 2026b.

Manwen Liao, Xinyu Lian, Jian Mao, Kaixu Chen, Li Luo, Jinghao Yan, Wanshui Gan, Qiao Yu, Weitian Zhang, Chunhua Shen, Guang Chen, Bo Dai, Xudong Xu, and Zhaoyang Lyu. Megaparts: Scaling part-aware 3d object generation to 300 parts via token-efficient autoregressive modeling, 2026. URL https://arxiv.org/abs/2608.14783.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 300–309. IEEE, 2023.

Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Yiqiang Feng, Yadong Mu, and Katerina Fragkiadaki. Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers, 2025. URL https://arxiv.org/abs/2506.05573.

Anran Liu, Cheng Lin, Yuan Liu, Xiaoxiao Long, Zhiyang Dou, Hao-Xiang Guo, Ping Luo, and Wenping Wang. Part123: Part-aware 3d reconstruction from a single-view image. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–12, 2024.

Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36:22226–22246, 2023a.

Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. Partfield: Learning 3d feature fields for part segmentation and beyond. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9704–9715. IEEE, 2025a.

Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. Partfield: Learning 3d feature fields for part segmentation and beyond. *arXiv preprint arXiv:2504.11451*, 2025b.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9264–9275. IEEE, 2023b.

William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pp. 163–169, New York, NY, USA, 1987. Association for Computing Machinery. ISBN 0897912276. doi: 10.1145/37401.37422.

Yihao Luo, Xianglong He, Chuanyu Pan, Yiwen Chen, Jiaqi Wu, Yangguang Li, Wanli Ouyang, Yuanming Hu, Guang Yang, and ChoonHwai Yap. Faithful contouring: near-lossless 3d voxel representation free from iso-surface. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14408–14418, 2026.

Changfeng Ma, Yang Li, Xinhao Yan, Jiachen Xu, Yunhan Yang, Chunshi Wang, Zibo Zhao, Yanwen Guo, Zhuo Chen, and Chunchao Guo. P3-sam: Native 3d part segmentation, 2025. URL https://arxiv.org/abs/2509.06784.

Ziqi Ma, Yisong Yue, and Georgia Gkioxari. Find any part in 3d, 2024. URL https://arxiv.org/abs/2411.13550.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

George Tang, William Zhao, Logan Ford, David Benhaim, and Paul Zhang. Segment any mesh. *arXiv preprint arXiv:2408.13679*, 2024.

Jiaxiang Tang, Ruijie Lu, Zhaoshuo Li, Zekun Hao, Xuan Li, Fangyin Wei, Shuran Song, Gang Zeng, Ming-Yu Liu, and Tsung-Yi Lin. Efficient part-level 3d object generation via dual volume packing. *arXiv preprint arXiv:2506.09980*, 2025.

Chunshi Wang, Junliang Ye, Yunhan Yang, Yang Li, Zizhuo Lin, Jun Zhu, Zhuo Chen, Yawei Luo, and Chunchao Guo. Part-x-mllm: Part-aware 3d multimodal large language model, 2025. URL https://arxiv.org/abs/2511.13647.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation. *Tech report*, 2025.

Xinhao Yan, Jiachen Xu, Yang Li, Changfeng Ma, Yunhan Yang, Chunshi Wang, Zibo Zhao, Zeqiang Lai, Yunfei Zhao, Zhuo Chen, et al. X-part: high fidelity and structure coherent shape decomposition. *arXiv preprint arXiv:2509.08643*, 2025.

Yunhan Yang, Yukun Huang, Yuan-Chen Guo, Liangjun Lu, Xiaoyang Wu, Edmund Y Lam, Yan-Pei Cao, and Xihui Liu. Sampart3d: Segment any part in 3d objects. *arXiv preprint arXiv:2411.07184*, 2024.

Yunhan Yang, Yufan Zhou, Yuan-Chen Guo, Zi-Xin Zou, Yukun Huang, Ying-Tian Liu, Hao Xu, Ding Liang, Yan-Pei Cao, and Xihui Liu. Omnipart: Part-aware 3d generation with semantic decoupling and structural cohesion. *arXiv preprint arXiv:2507.06165*, 2025.

Yunhan Yang, Yuanchen Guo, Yukun Huang, Zi-Xin Zou, Zhipeng Yu, Yangguang Li, Yanpei Cao, and Xihui Liu. Holopart: Generative 3d part amodal segmentation. In *International Conference on Learning Representations*, volume 2026, pp. 52657–52678, 2026.

Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 42(4):1–16, 2023.

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions On Graphics (TOG)*, 43(4):1–20, 2024.

Zhe Zhu, Le Wan, Rui Xu, Yiheng Zhang, Honghua Chen, Zhiyang Dou, Cheng Lin, Yuan Liu, and Mingqiang Wei. Partsam: A scalable promptable part segmentation model trained on native 3d data. *arXiv preprint arXiv:2509.21965*, 2025.