MESHCRAFTER: STRUCTURALLY ROBUST AUTOREGRESSIVE MESH GENERATION

Anonymous authors
Paper under double-blind review

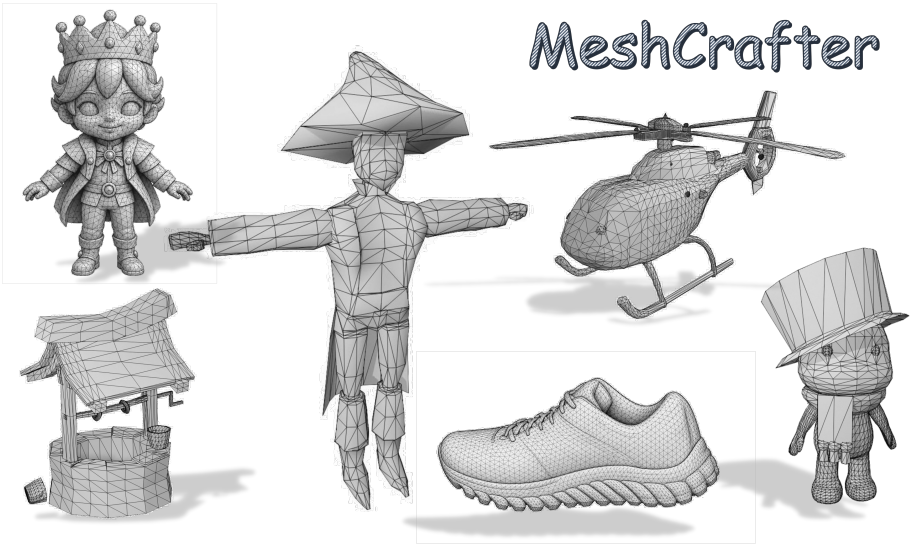

Figure 1: **MeshCrafter** generates all kinds of high-quality 3D assets.

ABSTRACT

Autoregressive mesh generation becomes increasingly challenging as mesh complexity and sequence length grow. Although recent methods improve scalability through compact representations or localized generation, geometric structure is typically imposed through discrete partitioning and fixed serialization, which discards ambiguous transitions between neighboring regions. Moreover, standard next-token training optimizes local prediction likelihood rather than the geometric and structural quality of the complete generated mesh.

We present **MeshCrafter**, a structure-aware autoregressive framework that incorporates geometric organization throughout region modeling, serialization, and post-training. MeshCrafter first constructs latent geometric regions from pretrained geometry features and refines their organization with a triplet objective, while explicitly estimating a continuous interface-proximity field to preserve structural transitions discarded by hard assignments. Based on this representation, we introduce soft-interface-guided hierarchical mesh tokenization, where discrete region ownership constrains the traversal space and continuous interface proximity determines the traversal priority. A hierarchical coarse-to-fine spatial encoding further provides a compact representation of 3D vertices. Finally, we introduce structure-aware reinforcement post-training, which compares groups of complete mesh trajectories using geometric and latent structural feedback and directly optimizes the generation policy at the sequence level.

Experiments on a 1k-mesh benchmark demonstrate consistent improvements over representative autoregressive mesh generators. MeshCrafter achieves state-of-art results compared among strongest evaluated baselines, while presenting its robustness and effectiveness.

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.

1 INTRODUCTION

Autoregressive mesh generators represent geometry as a sequence of discrete decisions (Nash et al., 2020; Siddiqui et al., 2023; Chen et al., 2024). Although these approaches improve scalability by designing compact representations or localized generation strategies (Chen et al., 2025b; Weng et al., 2025a; Zhao et al., 2025; Xu et al., 2025), the underlying geometric structure is usually learned only implicitly after tokenization. In particular, the serialization order is mainly determined by predefined connectivity, local patch decomposition, or geometric heuristics (Siddiqui et al., 2023; Chen et al., 2025b; Xu et al., 2025), while structural ambiguity and transition information between different geometric regions are not explicitly modeled. As a result, for complex meshes, the model must recover local surface connectivity through long-range dependencies, leading to accumulated errors such as topology defects, surface discontinuities, and missing details. Moreover, standard next-token prediction objectives optimize local token likelihood rather than the quality of the complete generated mesh. A sequence with locally plausible token predictions does not necessarily produce globally consistent geometry and topology after decoding. Therefore, supervised autoregressive training alone cannot fully exploit the structural feedback contained in complete mesh trajectories.

To address these challenges, we propose MeshCrafter, a structure-aware autoregressive mesh generation framework for complex 3D assets. MeshCrafter first introduces Latent Region Structuring, which derives discrete region ownership P and continuous interface proximity $\hat{\mathcal{B}}$ from geometric features, capturing both regional organization and local structural transitions. Based on this representation, we develop Soft-Interface-Guided Hierarchical Mesh Tokenization, where P constrains the traversal space and $\hat{\mathcal{B}}$ determines traversal priority, producing structurally coherent mesh sequences with compact hierarchical spatial encoding. Finally, we introduce Structure-Aware Reinforcement Post-Training, which optimizes generation policies using geometric and structural feedback from complete mesh trajectories, aligning token-level learning with mesh-level quality (Rafailov et al., 2023b; Schulman et al., 2017a; Shao et al., 2024).

Our contributions are summarized as follows:

- We introduce a structure-aware mesh generation framework that integrates geometric structure into region modeling, sequence construction, and generation optimization.
- We propose Soft-Interface-Guided Hierarchical Mesh Tokenization, which jointly leverages discrete region constraints and continuous interface guidance for structure-preserving mesh serialization.
- We propose Structure-Aware Reinforcement Post-Training, which optimizes complete mesh trajectories using structural feedback, improving geometric and topological quality on complex meshes.

2 RELATED WORK

3D Generation. Recent advances in 3D generation have explored diffusion-based and latent-based models operating on point clouds, implicit fields, voxels, and intermediate surface representations (Nichol et al., 2022; Poole et al., 2023; Jun and Nichol, 2023; Lin et al., 2023; Wang et al., 2023; Metzer et al., 2023; Hong et al., 2023b;a; Li et al., 2024; Xu et al., 2024a; Tang et al., 2024b; Bai et al., 2025; Lan et al., 2025; Xiang et al., 2025; Luo et al., 2025; Shen et al., 2026; Zhang et al., 2026). These approaches have demonstrated strong capability in modeling global 3D shapes and semantic priors. However, the final mesh is often obtained through additional surface extraction or reconstruction procedures (Mildenhall et al., 2020; Wang et al., 2021b;a; Li et al., 2023), which may introduce extra complexity and limit direct control over mesh connectivity. In contrast, MeshCrafter focuses on autoregressive mesh generation from geometric conditioning signals and they can be handled by upstream 3D foundation models such as TRELLIS (Xiang et al., 2025).

Autoregressive Mesh Generation and Tokenization. Autoregressive mesh generation models formulate mesh synthesis as sequence modeling over discrete geometric tokens. Early works directly predict vertices and faces (Nash et al., 2020), while subsequent methods introduce learned geometric vocabularies, connectivity-aware serialization, and hierarchical representations to improve generation quality and scalability (Siddiqui et al., 2023; Chen et al., 2024; 2025a; Tang et al., 2024a; Gao et al.,

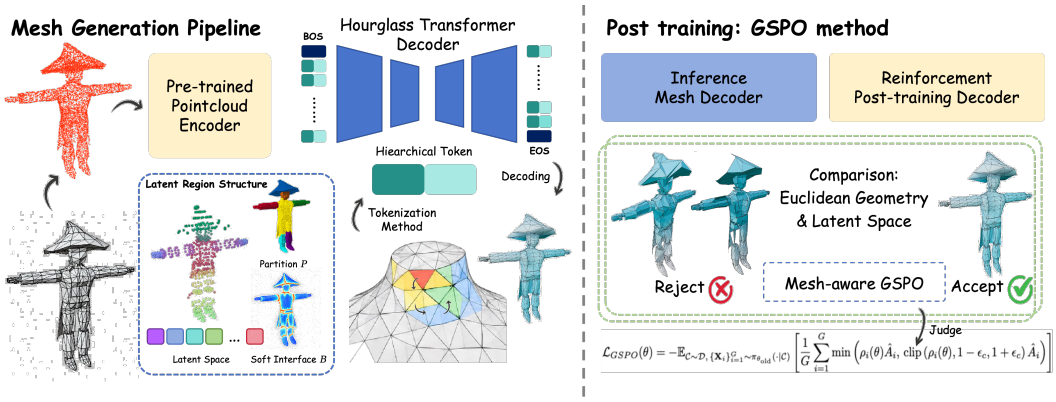

Figure 2: **The overall framework of MeshCrafter.** Given an input point cloud, MeshCrafter first constructs latent geometric regions and interface-aware structural representations, then performs soft-interface-guided hierarchical mesh tokenization, and finally optimizes mesh generation through structure-aware reinforcement post-training.

2025; Lionar et al., 2025; Zhao et al., 2025; Weng et al., 2025a; Xu et al., 2025; Shen et al., 2026; Zhang et al., 2026).

To handle increasingly complex meshes, recent approaches investigate more compact representations and localized generation strategies. DeepMesh (Zhao et al., 2025) introduces hierarchical mesh representation and structural conditioning, while MeshMosaic (Xu et al., 2025) explores patch-based autoregressive generation with local-to-global assembly. Other works further study efficient mesh sequence modeling and structured token spaces (Weng et al., 2025b; Wang et al., 2025; Chen et al., 2025b; Xu et al., 2026; Wang et al., 2026; Lin et al., 2026). Beyond mesh-specific methods, discrete representation learning and vector-quantized modeling have also influenced modern geometric tokenization strategies (van den Oord et al., 2017; Esser et al., 2021; Yu et al., 2024; 2023; Chang et al., 2022; Zhang et al., 2023).

Although these approaches improve sequence efficiency, the structural transition information between different geometric regions is not explicitly modeled during token construction. MeshCrafter addresses this limitation by incorporating latent geometric regions and continuous interface information directly into mesh serialization.

Structure-aware Representation and Generation Alignment. Geometric structure modeling has been widely explored in 3D analysis, reconstruction, and generation. Previous approaches often rely on semantic parts or explicit decomposition to capture object-level organization (Liu et al., 2024; Yang et al., 2024; Lin et al., 2025; Tertikas et al., 2023).

Meanwhile, recent geometric foundation models show that pretrained representations can provide meaningful latent geometric structures without requiring semantic annotations (Peng et al., 2023; Zhao et al., 2023; Yu et al., 2022; Zhang et al., 2022; Xu et al., 2024b).

Beyond representation learning, recent advances in generative model alignment demonstrate that post-training can further improve generation quality. Preference optimization methods such as DPO (Rafailov et al., 2023a) and policy optimization methods including PPO (Schulman et al., 2017b) and GRPO (Shao et al., 2024) enable models to incorporate feedback beyond maximum likelihood training. Similar ideas have also been explored in large-scale generative systems through reinforcement learning from human feedback and preference optimization (Ouyang et al., 2022; Bai et al., 2022). However, applying such strategies to mesh generation remains challenging because the quality of a mesh depends on global geometry and structural validity. MeshCrafter combines geometry-aware structural representation with mesh-level reinforcement post-training, allowing complete mesh trajectories to be optimized using geometric and structural feedback.

3 METHOD

MeshCrafter is a structurally robust autoregressive method for mesh generation from conditioning geometry. Sec. 3.1 introduces **Latent Region Structuring**, which organizes the conditioning geometry into discrete regions while preserving continuous interface proximity. Sec. 3.2 presents **Interface-Conditioned Mesh Tokenization**, which combines coarse-to-fine coordinate codes with vertex-level interface cues and region-level spatial context. Sec. 3.3 develops **Structure-Aware Reinforcement Post-Training**, which directly optimizes complete mesh trajectories using 3D geometric and latent structural rewards. An overview of the full pipeline is presented in Figure 2.

3.1 LATENT REGION STRUCTURING

Recent approaches directly assign regions and influence context generation. As shown in figure 3, neighbor surface elements with similar geometries may receive different discrete assignments during partition, causing abrupt changes in local context, serialization order, and subsequent patch assembly. To solve these problems, we retain discrete regions for efficient generation while explicitly modeling the transition zones between them.

Given a conditioning point cloud $\mathcal{Q} = \{\mathbf{q}_i\}_{i=1}^{N}$, we extract geometry features using a pretrained Michelangelo point encoder (Zhao et al., 2023) with basic k-means initialization:

$$\mathbf{z}_i = E_{\text{geo}}(\mathbf{q}_i), \ell_i = \arg\min_{k\in\{1,\dots,K\}} \|\mathbf{z}_i - \boldsymbol{\mu}_k\|_2^2, \quad (1)$$

where $\boldsymbol{\mu}_k$ denotes the k-th cluster center. The number of regions is selected adaptively from input complexity using the same rule during training and inference.

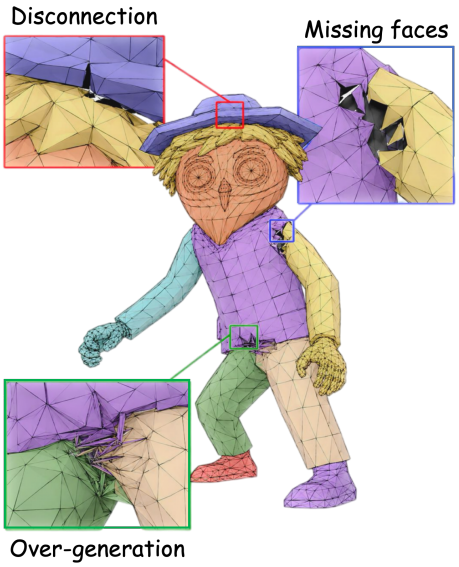

Figure 3: **Mesh Problem in direct region partition.** Problems include mesh disconnection, missing faces and mesh over-generation.

While base k-means assignment disregards explicit margin between neighboring regions. We refine the latent metric using them as pseudo-labels. For each sampled surface element a, we select a positive point p from the same region and a negative point n from a different region, and optimize

$$\mathcal{L}_{\text{tri}} = \mathbb{E}_{(a,p,n)} \left[\max\big(0,\ \|\mathbf{z}_a - \mathbf{z}_p\|_2^2 - \|\mathbf{z}_a - \mathbf{z}_n\|_2^2 + \delta\big)\right], \quad (2)$$

The loss is minimized when the negative feature is at least a margin δ farther from the anchor than the positive feature. It therefore tightens the feature distribution within each region while separating different regions, yielding a more structured latent space for subsequent interface estimation.

Despite the improved inter-region separation, residual feature ambiguity persists around region interfaces. Yet we define $D_{\text{in}}(f)$ and $D_{\text{out}}(f)$ as average feature distance to same-region and adjacent-region neighbors of face f with feature $\varphi(f)$

$$D_{\text{in}}(f) = \frac{1}{|\mathcal{N}_{\text{in}}(f)|} \sum_{f'\in\mathcal{N}_{\text{in}}(f)} \|\varphi(f) - \varphi(f')\|_2, \quad (3)$$

$$D_{\text{out}}(f) = \frac{1}{|\mathcal{N}_{\text{out}}(f)|} \sum_{f'\in\mathcal{N}_{\text{out}}(f)} \|\varphi(f) - \varphi(f')\|_2. \quad (4)$$

We measure the local separation by

$$A(f) = |D_{\text{out}}(f) - D_{\text{in}}(f)|, \qquad \hat{\mathcal{B}}(f) = 1 - \min\left(1, \frac{A(f)}{A_{\text{ref}}}\right). \quad (5)$$

A large separation indicates a stable region interior, whereas $D_{\text{in}}(f) \approx D_{\text{out}}(f)$ produces a large $\hat{\mathcal{B}}(f)$ and identifies a location close to a region transition.

The resulting representation therefore keeps two complementary quantities $\mathcal{P}(\{)$ and $\hat{\mathcal{B}}(f)$ The former determines where local generation is performed, while the latter preserves information that would otherwise be discarded by hard partitioning.

3.2 SOFT-INTERFACE-GUIDED HIERARCHICAL MESH TOKENIZATION

Existing mesh tokenization methods mainly improve efficiency by changing the basic representation unit or the coordinate encoding scheme. However, when applied to complex 3D assets with large face counts, their efficiency and scalability remain limited by long token sequences. Based on the discrete region partition P and continuous interface proximity $\hat{\mathcal{B}}$ obtained in Sec. 3.1, we introduce a two-stage hierarchical tokenizer: P determines where the traversal is structurally allowed to expand, while $\hat{\mathcal{B}}$ determines the visiting priority among the currently accessible faces.

Structure-Guided Sequence Construction. For region k, we denote its face set as

$$\mathcal{F}_k = \{f \in \mathcal{F} \mid P(f) = k\}. \tag{6}$$

We first select the face with the highest interface proximity as the entry of the sequence:

$$f_{k,1} = \arg\max_{f \in \mathcal{F}_k} \hat{\mathcal{B}}(f). \tag{7}$$

The soft interface field then continuously guides the remaining serialization process. Let $\mathcal{S}_{k,t}$ denote the set of faces visited in the first t steps. We construct the current frontier using unvisited faces that remain in region k and are adjacent to the visited surface:

$$\mathcal{C}_{k,t} = \{f \in \mathcal{F} \setminus \mathcal{S}_{k,t} \mid P(f) = k,\ \exists g \in \mathcal{S}_{k,t},\ f \sim g\}, \tag{8}$$

where $f \sim g$ indicates that two faces share an edge in the original mesh. The next face is selected as

$$f_{k,t+1} = \arg\max_{f \in \mathcal{C}_{k,t}} \hat{\mathcal{B}}(f). \tag{9}$$

This produces an ordered face sequence

$$\sigma_k = (f_{k,1}, \ldots, f_{k,N_k}). \tag{10}$$

We then unfold the vertices of these faces using a consistent winding order, so that the structural ordering jointly determined by P and $\hat{\mathcal{B}}$ is directly propagated to the resulting vertex stream.

Hierarchical Spatial Encoding. We next map each 3D vertex to a compact discrete representation. Encoding the three coordinate axes independently not only increases the sequence length, but also splits a single 3D position into multiple independent coordinate symbols. We therefore decompose each quantized vertex $\mathbf{q}_{k,t}$ into two spatial levels:

$$q_{k,t,d} = N_f q^c_{k,t,d} + q^f_{k,t,d}, d \in \{x, y, z\}, \tag{11}$$

where $q^c_{k,t,d}$ specifies the coordinate of the coarse 3D spatial cell, and $q^f_{k,t,d}$ denotes the discrete offset within that cell.

The three-axis information at the two spatial levels is then jointly packed as

$$\alpha_{k,t} = \sum_{d=0}^{2} q^c_{k,t,d} N_c^{2-d}, \beta_{k,t} = \sum_{d=0}^{2} q^f_{k,t,d} N_f^{2-d}. \tag{12}$$

Each 3D vertex is thus represented by two discrete symbols,

$$(\alpha_{k,t}, \beta_{k,t}), \tag{13}$$

corresponding to its coarse spatial location and intra-cell position, respectively. Region k is finally encoded as

$$\mathbf{X}_k = [(\alpha_{k,1}, \beta_{k,1}), \ldots, (\alpha_{k,T_k}, \beta_{k,T_k})]. \tag{14}$$

The quantization resolution, spatial hierarchy configuration, and inverse mapping are detailed in the appendix.

3.3 STRUCTURE-AWARE REINFORCEMENT POST-TRAINING

We already establish a region-level structural representation of the conditioning geometry together with an interface-aware coarse-to-fine coordinate sequence through preceding sections. Inspired by Group Sequence Policy Optimization (GSPO), we instead optimize the generation policy directly from groups of complete mesh rollouts.

For a conditioning geometry $\mathcal{Q}$, the region partition $\mathcal{P}$, the continuous interface field $\hat{\mathcal{B}}$, and the corresponding token-aligned structural conditions $\mathbf{c}_t$ are determined before generation. Starting from the supervised model, the rollout policy $\pi_{\theta_{\text{old}}}$ samples a group of G complete candidate sequences under the same structural condition and provide a common reference for comparing alternative autoregressive mesh trajectories.:

$$\mathbf{X}_i = \{(\alpha_{i,t}, \beta_{i,t})\}_{t=1}^{T_i} \sim \pi_{\theta_{\text{old}}}\left(\cdot \mid \mathcal{Q}, \mathcal{P}, \hat{\mathcal{B}}\right), \qquad i = 1, \ldots, G. \tag{15}$$

Structure-aware scoring. We evaluate each rollout from the resulting 3D asset and its geometry-aware latent representation, compare the generated geometry with its conditioning geometry both in Euclidean space and in the structure-aware latent space established in Sec. 3.1. The reward function reflects global geometric and the underlying regional structure preservation. We denote the resulting scalar reward by

$$R_i = \mathcal{R}_{\text{3D}}\left(\mathcal{M}_i, \mathcal{Q}, \mathbf{Z}_i, \mathbf{Z}_{\mathcal{Q}}\right), \tag{16}$$

where $\mathbf{Z}_i$ and $\mathbf{Z}_{\mathcal{Q}}$ denote geometry-aware latent representations of the generated and conditioning geometries, respectively.

For the G rollouts generated from the same condition, we compute a group-relative advantage by normalizing their rewards:

$$\hat{A}_i = \frac{R_i - \mu_R}{\sigma_R + \epsilon}, \qquad \mu_R = \frac{1}{G}\sum_{j=1}^{G} R_j, \tag{17}$$

where σ_R is the standard deviation of the rewards within the group. More construction details and scoring term normalizations are provided in the appendix.

Mesh-aware sequence optimization. According to the representation in Sec. 3.2, we therefore treat the coarse–fine pair as one geometric decision and define its conditional log-likelihood as

$$\ell_{i,t}(\theta) = \log p_\theta\left(\alpha_{i,t} \mid \mathcal{H}_{i,t}, \mathbf{c}_{i,t}\right) + \log p_\theta\left(\beta_{i,t} \mid \mathcal{H}_{i,t}, \alpha_{i,t}, \mathbf{c}_{i,t}\right). \tag{18}$$

Based on this vertex-level factorization, the policy ratio for the complete rollout is defined as

$$\rho_i(\theta) = \exp\left[\frac{1}{T_i}\sum_{t=1}^{T_i}\left(\ell_{i,t}(\theta) - \ell_{i,t}(\theta_{\text{old}})\right)\right]. \tag{19}$$

While vertex-level normalization aligns the policy update with the coarse-to-fine geometric decision defined in Sec. 3.2, it also normalizes the contribution of trajectories with different mesh lengths rather than individual coordinate symbols.

Finally, following the group-sequence optimization principle, we denote the conditioning geometry and its structural information as

$$\mathcal{C} = \left(\mathcal{Q}, \mathcal{P}, \hat{\mathcal{B}}\right), \tag{20}$$

with $\mathcal{C} \sim \mathcal{D}$. For each condition, the old policy $\pi_{\theta_{\text{old}}}$ samples a group of complete mesh trajectories:

$$\{\mathbf{X}_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}\left(\cdot \mid \mathcal{C}\right). \tag{21}$$

Based on the group-relative advantage and the mesh-aware sequence ratio defined above, we optimize the following clipped sequence-level objective:

$$\mathcal{L}_{GSPO}(\theta) = -\mathbb{E}_{\mathcal{C}\sim\mathcal{D},\, \{\mathbf{X}_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|\mathcal{C})}\left[\frac{1}{G}\sum_{i=1}^{G}\min\left(\rho_i(\theta)\hat{A}_i,\ \text{clip}\left(\rho_i(\theta), 1-\epsilon_c, 1+\epsilon_c\right)\hat{A}_i\right)\right]. \tag{22}$$

This objective jointly compares multiple complete mesh trajectories under the same structural condition and constrains the policy update through clipping, thereby exploiting relative quality differences within each group while maintaining stable optimization.

4 EXPERIMENTS

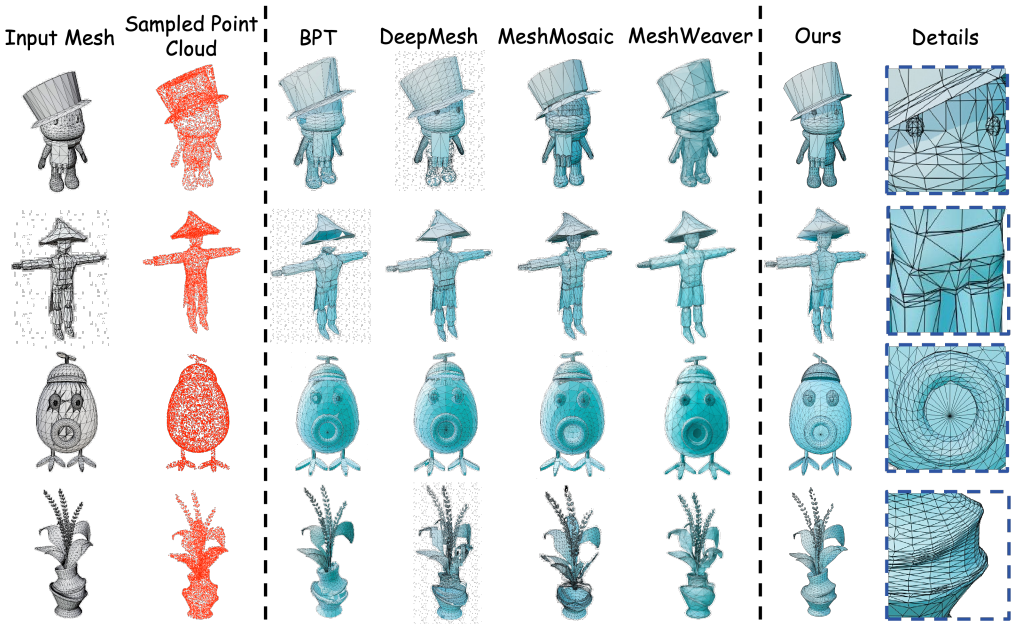

Figure 4: **Qualitative comparison.** MeshCrafter generates coherent topology and fine detail.

4.1 EXPERIMENTAL SETUP

Datasets. We construct the training corpus from Objaverse (Deitke et al., 2023b), Objaverse-XL (Deitke et al., 2023a), ABO (Collins et al., 2022), HSSD (Khanna et al., 2024), 3D-FUTURE (Fu et al., 2021), and ShapeNetV2 (Chang et al., 2015). After data cleaning, we obtain approximately 1M training meshes; detailed data statistics and the filtering pipeline are provided in the appendix. Our test set contains 1k meshes, including 400 meshes generated by TRELLIS from held-out prompts and the remaining 600 meshes sampled from the aforementioned datasets. The test meshes contain an average of approximately 5.1K faces. Both the training and test sets cover a broad range of mesh complexities. TRELLIS-generated meshes are used exclusively for evaluation and are never included in training.

Baselines and Evaluation Metrics. We compare MeshCrafter with representative state-of-the-art autoregressive mesh generation methods, including BPT (Weng et al., 2025a), DeepMesh (Zhao et al., 2025), MeshMosaic (Xu et al., 2025), and MeshWeaver (Xu et al., 2026). All methods are evaluated on the same test set using official implementations and recommended configurations whenever available.We evaluate all methods from four perspectives: geometric fidelity, local surface quality, topological validity, and generation efficiency. Geometric fidelity is measured using Chamfer Distance (CD) and Hausdorff Distance (HD), while local surface quality is evaluated using F1-score and Normal Consistency (NC). All geometric metrics are computed in the same normalized coordinate space. For topological validity, we primarily report the Watertight Rate (WT) as well.

Implementation Details. MeshCrafter adopts a hierarchical hourglass Transformer with 551M parameters. The backbone consists of 6 hierarchical levels and 48 self-attention layers, with a hidden dimension of 1536 and 16 attention heads. We employ RoPE positional embeddings and SwiGLU feed-forward networks. The training procedure uses a sliding window of length 9,000 with a stride of 4,500, while inference employs a rolling KV cache to retain historical context.

Each training sample is conditioned on 8,192 surface points. The base model is optimized using AdamW with an initial learning rate of 10^{-4}, cosine decay to 10^{-5}, 10K warm-up steps, a weight decay of 0.01, and gradient clipping with a maximum norm of 1.0. Following our previous training configuration, the model is trained for 1.25M steps on $64\times$ NVIDIA A800 GPUs.

Table 1: Quantitative comparison and user study.

Method	Quantitative						User Study		
	CD↓	HD↓	F1↑	NC↑	Time(s)↓	WT%↑	Art.↑	Detail↑	Plaus.↑
BPT	0.114	0.268	0.647	0.71	165	42	0.79	1.73	0.95
DeepMesh	0.135	0.287	0.588	0.74	113	51	1.21	0.88	1.51
MeshMosaic	0.107	0.226	0.716	0.78	125	58	1.66	1.23	1.35
MeshWeaver	0.074	0.163	0.821	0.84	96	80	2.08	2.11	2.06
Ours w/o GSPO	0.066	0.141	0.853	0.89	**88**	85	2.45	2.32	2.48
Ours w/ GSPO	**0.058**	**0.125**	**0.879**	**0.93**	91	**94**	**2.66**	**2.54**	**2.75**

4.2 Main Results

Quantitative Results. Table 1 summarizes the quantitative comparison between MeshCrafter and representative autoregressive mesh generation methods. Overall, our method achieves better performance in geometric fidelity, local surface quality, and topological integrity. Compared with BPT, DeepMesh, MeshMosaic, and MeshWeaver, MeshCrafter obtains lower Chamfer Distance (CD) and Hausdorff Distance (HD), while achieving higher F1-score and Normal Consistency (NC), indicating more accurate global geometry and local surface reconstruction.

MeshCrafter also shows a clear advantage in topological integrity. Compared with the strongest baseline, the Watertight Rate (WT) improves from **80%** to **94%**. This suggests that the improvement is not limited to point-wise geometric fitting, but also extends to the structural completeness and connectivity of the generated surfaces. Overall, MeshCrafter maintains high geometric accuracy while effectively reducing local missing regions, surface discontinuities, and topological artifacts in complex mesh generation.

Qualitative Results. Figure 4 presents qualitative comparisons on complex 3D assets. BPT and DeepMesh are more prone to geometric omissions, irregular triangulation, and local disconnections around thin structures, high-curvature regions, and complex local connections. MeshMosaic and MeshWeaver recover more complete global shapes, but local discontinuities can still be observed around structural transitions and fine-grained connectivity.

In contrast, MeshCrafter preserves the overall geometry while producing more stable local connectivity around thin structures, high-curvature details, and region interfaces. The zoomed-in comparisons further show that our method generates more continuous surface structures and preserves finer geometric details, while reducing local holes, broken connections, and irregular tessellation. These qualitative observations are consistent with the geometric and topological metrics reported in Table 1.

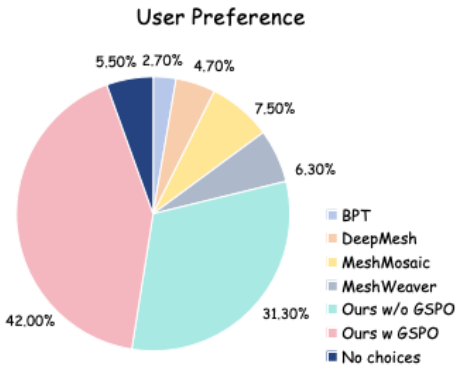

Figure 5: **User study results.** MeshCrafter receives the most preference comparing with other sota methods.

User Study. We recruited 20 volunteers (12 CG/vision researchers, 8 with $\geq$2 years 3D modeling experience). According to generated meshes, raters scored each method on a 4-point ordinal scale (0=poor, 1=acceptable, 2=good, 3=excellent) across three criteria: artistic quality, detail richness, and structural plausibility. Figure 5 and table 1 shows that MeshCrafter significantly outperforms all baselines from all aspects we

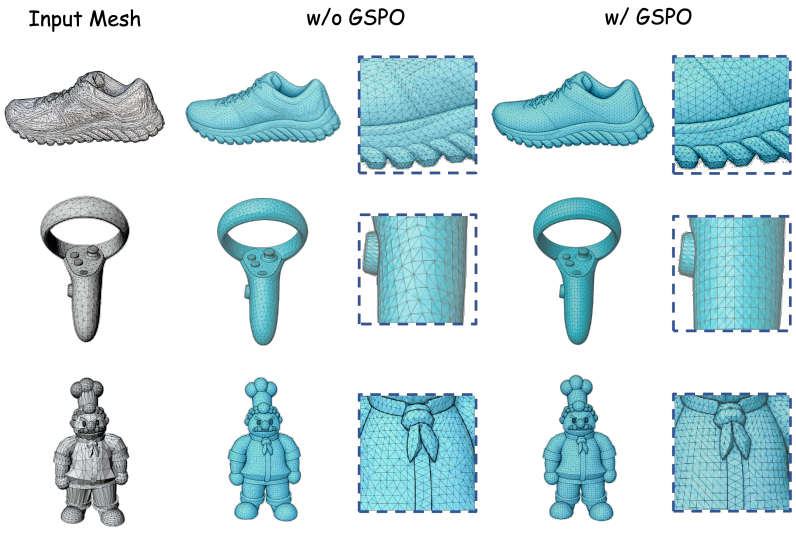

Figure 6: **Effect of Structure-Aware Reinforcement Post-Training.** The proposed post-training stage improves surface regularity and structural consistency with fine geometric details. Zoomed-in regions highlight the local structural artifacts.

4.3 Ablation Study

Tokenization Method Ablation. We further compare different mesh tokenization strategies. We adopt the *Compression Ratio* (CR) to measure sequence compactness and introduce a new measurement *Face Adjacency Preservation* (FAP) to evaluate the preservation of local face adjacency. The detailed definition of FAP is provided in the appendix.

As shown in Table 2, our method achieves a compression ratio of **0.14**, reducing the normalized sequence length by approximately **22%** compared with MeshWeaver, the most compact competing method. Meanwhile, our method obtains an FAP of **0.88**, outperforming all evaluated state-of-the-art tokenization schemes. This indicates that the improved compression does not come at the cost of local topological continuity. Instead, our tokenization encourages consecutively generated faces to remain adjacent on the original mesh, yielding shorter autoregressive sequences while better preserving local mesh connectivity.

Table 2: Comparison of mesh tokenization methods.

Method	CR↓	FAP↑	Time(s)↓
MeshAnythingV2	0.46	0.71	142
EdgeRunner	0.47	0.73	128
BPT	0.26	0.79	96
DeepMesh	0.28	0.81	84
MeshWeaver	0.18	0.85	73
Ours	**0.14**	**0.91**	**61**

Effect of Reinforcement Post-Training. The quantitative results are summarized in Table 1. Our model shows consistent improvements in both geometric fidelity and structural validity, indicating that reinforcement signals defined more complete mesh trajectories. As illustrated in Fig. 6, compared with the model without post-training, the GSPO-based variant produces meshes with more coherent local structures and more stable surface organization, while reducing local topological artifacts and irregular geometry. These results show that the benefit of post-training is reflected not only in quantitative metrics but also directly in the structural quality of the generated meshes.

5 Conclusion

MeshCrafter introduces latent region structuring and soft-interface-guided hierarchical mesh tokenization, while also incorporates geometric organization and interface information directly into mesh serialization. Furthermore, our structure-aware reinforcement post-training aligns sequence optimization with mesh-level geometric and topological quality through feedback from complete mesh trajectories. Extensive experiments demonstrate that MeshCrafter achieves improved fidelity, structural validity, and scalability on complex mesh generation tasks.

REFERENCES

W. Bai, Y. Li, W. Chen, W. Luo, and H. Sun. Dive3d: Diverse distillation-based text-to-3d generation via score implicit matching. *arXiv preprint arXiv:2506.13594*, 2025.

Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, C. Chen, C. Olsson, C. Olah, D. Hernandez, D. Drain, D. Ganguli, D. Li, E. Tran-Johnson, E. Perez, J. Kerr, J. Mueller, J. Ladish, M. Landau, K. Ndousse, S. Johnston, B. Greene, J. Leike, and G. Irving. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022.

Y. Chen, T. He, D. Huang, W. Ye, S. Chen, J. Tang, X. Chen, Z. Cai, L. Yang, G. Yu, G. Lin, and C. Zhang. Meshanything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*, 2024.

Y. Chen, Y. Wang, Y. Luo, Z. Wang, Z. Chen, J. Zhu, C. Zhang, and G. Lin. Meshanything v2: Artist-created mesh generation with adjacent mesh tokenization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13922–13931, 2025a.

Y. Chen, Y. Wang, Y. Luo, Z. Wang, Z. Chen, J. Zhu, C. Zhang, and G. Lin. Meshanything v2: Artist-created mesh generation with adjacent mesh tokenization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13922–13931, 2025b.

J. Collins, S. Goel, K. Deng, A. Luthra, L. Xu, E. Gundogdu, X. Zhang, T. F. Y. Vicente, T. Dideriksen, H. Arora, M. Guillaumin, and J. Malik. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21126–21136, 2022.

M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, E. VanderBilt, A. Kembhavi, C. Vondrick, G. Gkioxari, K. Ehsani, L. Schmidt, and A. Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. In *Advances in Neural Information Processing Systems*, volume 36, 2023a.

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 (CVPR)*, pages 13142–13153, 2023b.

P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021.

H. Fu, R. Jia, L. Gao, M. Gong, B. Zhao, S. J. Maybank, and D. Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129(12):3313–3337, 2021. doi: 10.1007/s11263-021-01534-z.

D. Gao, Y. Siddiqui, L. Li, and A. Dai. Meshart: Generating articulated meshes with structure-guided transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 618–627, 2025.

Y. Hong, F. Zhan, Y. He, et al. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023a.

Y. Hong, F. Zhan, Y. He, et al. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023b.

H. Jun and A. Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.

M. Khanna, Y. Mao, H. Jiang, S. Haresh, B. Shacklett, D. Batra, A. Clegg, E. Undersander, A. X. Chang, and M. Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16384–16393, 2024.

Y. Lan, F. Hong, S. Zhou, S. Yang, X. Meng, Y. Chen, Z. Lyu, B. Dai, X. Pan, and C. C. Loy. Ln3diff++: Scalable latent neural fields diffusion for speedy 3d generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–18, 2025. ISSN 1939-3539. doi: 10.1109/tpami.2025.3633073. URL http://dx.doi.org/10.1109/TPAMI.2025.3633073.

J. Li, J. Tan, et al. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In *International Conference on Learning Representations*, 2024.

Z. Li, T. Müller, A. Evans, R. H. Taylor, M. Unberath, M.-Y. Liu, and C.-H. Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023.

C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023.

J. Lin, H. Long, H. Guo, J. Zhang, J. Yang, T. Guo, Y. Yang, J. Li, W. Zhang, M. Nießner, and W. Yang. Meshripple: Structured autoregressive generation of artist-meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12706–12718, June 2026.

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.

S. Lionar, J. Liang, and G. H. Lee. Treemeshgpt: Artistic mesh generation with autoregressive tree sequencing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26608–26617, 2025.

A. Liu, C. Lin, Y. Liu, X. Long, Z. Dou, H.-X. Guo, P. Luo, and W. Wang. Part123: part-aware 3d reconstruction from a single-view image. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–12, 2024.

Y. Luo, Z. Liu, Y. Zhao, S. Li, H. Cai, K. Kawaguchi, T.-S. Chua, Y. Zhang, and X. Wang. Towards unified and lossless latent space for 3d molecular latent diffusion modeling. *arXiv preprint arXiv:2503.15567*, 2025.

G. Metzer, E. Richardson, M. Sela, T. Shaharabany, R. Giryes, and D. Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes. In *CVPR Workshops*, 2023.

B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision*, pages 405–421. Springer, 2020.

C. Nash, Y. Ganin, S. M. A. Eslami, and P. Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International Conference on Machine Learning*, pages 7220–7228. PMLR, 2020.

A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022.

S. Peng, C. Jiang, Y. Liao, M. Niemeyer, M. Pollefeys, and A. Geiger. Shape as points: A differentiable 3d shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11930–11939, 2023.

B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *International Conference on Learning Representations*, 2023.

R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741, 2023a.

R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741, 2023b.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017a.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017b.

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, and D. Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

T. Shen, Y. Zhang, C. Tang, C. Ping, Z. Zhao, L. Wan, Y. Wang, R. Wang, and S. He. Flashmesh: Faster and better autoregressive mesh synthesis via structured speculation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27052–27061, June 2026.

Y. Siddiqui, J. Thies, F. Ma, and M. Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Advances in Neural Information Processing Systems*, volume 36, pages 15088–15100, 2023.

J. Tang, Z. Li, Z. Hao, X. Liu, G. Zeng, M.-Y. Liu, and Q. Zhang. Edgerunner: Auto-regressive auto-encoder for artistic mesh generation. *arXiv preprint arXiv:2409.18114*, 2024a.

J. Tang, X. Ren, et al. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024b.

K. Tertikas, D. Paschalidou, B. Pan, J. J. Park, M. A. Uy, I. Emiris, Y. Avrithis, and L. Guibas. Generating part-aware editable 3d shapes without 3d supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4466–4478, 2023.

A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

H. Wang, Y.-C. Guo, Y.-T. Liu, Z.-X. Zou, B. Zhang, W. Quan, D. Liang, Y.-P. Cao, and D.-M. Yan. Face: A face-based autoregressive representation for high-fidelity and efficient mesh generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12719–12729, June 2026.

L. Wang, Y. Liu, L. Liu, C. Theobalt, T. Komura, and W. Wang. Volsdf: Volume rendering neural implicit surfaces. In *Advances in Neural Information Processing Systems*, volume 34, 2021a.

P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems*, volume 34, pages 27171–27183, 2021b.

Y. Wang, X. Yi, H. Weng, Q. Xu, X. Wei, X. Yang, C. Guo, L. Chen, and H. Zhang. Nautilus: Locality-aware autoencoder for scalable mesh generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10961–10970, 2025.

Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in neural information processing systems*, 36:8406–8441, 2023.

H. Weng, Z. Zhao, B. Lei, X. Yang, J. Liu, Z. Lai, Z. Chen, Y. Liu, J. Jiang, C. Guo, T. Zhang, S. Gao, and C. L. P. Chen. Scaling mesh generation via compressive tokenization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11093–11103, 2025a.

H. Weng, Z. Zhao, B. Lei, X. Yang, J. Liu, Z. Lai, Z. Chen, Y. Liu, J. Jiang, C. Guo, et al. Scaling mesh generation via compressive tokenization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11093–11103, 2025b.

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*, pages 21469–21480, 2025.

J. Xu, W. Cheng, et al. Instantmesh: Efficient 3d mesh generation from a single image. *arXiv preprint arXiv:2404.07191*, 2024a.

J. Xu, W. Cheng, et al. Partfield: Learning 3d part segmentation by a neural field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

J. Xu, W. Zhao, and Y. Shan. Meshweaver: Sparse-voxel-guided surface weaving for autoregressive mesh generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5912–5922, June 2026.

R. Xu, T. Xue, Q. Dong, L. Wan, Z. Zhu, P. Li, Z. Dou, C. Lin, S. Xin, Y. Liu, and T. Komura. Meshmosaic: Scaling artist mesh generation via local-to-global assembly. *arXiv preprint arXiv:2509.19995*, 2025.

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.

L. Yu, J. Lezama, A. Gunduz, et al. Magvit: Masked generative video transformer. In *International Conference on Learning Representations*, 2023.

L. Yu, J. Lezama, A. Gunduz, et al. Magvit-v2: Language model beats diffusion – tokenizer is key to visual generation. In *International Conference on Learning Representations*, 2024.

X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19313–19322, 2022.

B. Zhang, J. Tang, M. Niessner, and P. Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models, 2023. URL `https://arxiv.org/abs/2301.11445`.

R. Zhang, Z. Guo, W. Zhang, K. Li, Z. Miao, B. Cui, Y. Qiao, P. Gao, and H. Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8552–8562, 2022.

X. Zhang, S. Yoo, H. Wu, C. Li, J. Xie, and Z. Tu. Pixarmesh: Autoregressive mesh-native single-view scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5881–5891, June 2026.

R. Zhao, J. Ye, Z. Wang, G. Liu, Y. Chen, Y. Wang, and J. Zhu. Deepmesh: Auto-regressive artist-mesh creation with reinforcement learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10612–10623, 2025.

Z. Zhao, W. Liu, X. Chen, X. Zeng, R. Wang, P. Cheng, B. Fu, T. Chen, G. Yu, and S. Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. In *Advances in Neural Information Processing Systems*, volume 36, pages 44725–44739, 2023.

A Detailed Implementation of Soft-Interface-Guided Hierarchical Mesh Tokenization

This section provides additional implementation details of the soft-interface-guided hierarchical mesh tokenization introduced in Sec. 3.2. The tokenizer consists of two stages: interface-guided face traversal and hierarchical spatial encoding.

A.1 Interface-guided Face Traversal

Given a mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F})$, where $\mathcal{V}$ and $\mathcal{F}$ denote the vertex and face sets, respectively, the region assignment obtained from Sec. 3.1 is defined as

$$P : \mathcal{F} \to \{1, \dots, K\}. \tag{23}$$

For region k, the corresponding face subset is

$$\mathcal{F}_k = \{f \in \mathcal{F} | P(f) = k\}. \tag{24}$$

Unlike conventional connectivity-based traversal, we use the continuous interface proximity field $\hat{\mathcal{B}}$ as a structural guidance signal. The initial face is selected as

$$f_{k,1} = \arg\max_{f \in \mathcal{F}_k} \hat{\mathcal{B}}(f). \tag{25}$$

At step t, the visited faces are denoted as $\mathcal{S}_{k,t}$. The traversal frontier only contains unvisited faces that belong to the current region and are adjacent to the visited surface:

$$\mathcal{C}_{k,t} = \{f \in \mathcal{F}_k \setminus \mathcal{S}_{k,t} | \exists g \in \mathcal{S}_{k,t}, (f, g) \in \mathcal{E}_f\}. \tag{26}$$

The next face is selected according to:

$$f_{k,t+1} = \arg\max_{f \in \mathcal{C}_{k,t}} \hat{\mathcal{B}}(f). \tag{27}$$

The traversal terminates when all faces in region k are visited, producing an ordered sequence:

$$\sigma_k = (f_{k,1}, ..., f_{k,N_k}). \tag{28}$$

A.2 Vertex Stream Construction

Each triangular face is unfolded according to a fixed winding order. To avoid repeated vertices caused by shared faces, we adopt a first-occurrence rule. A vertex is appended to the vertex stream only when it first appears during traversal.

The resulting vertex sequence is:

$$\mathcal{V}_k = (v_{k,1}, ..., v_{k,T_k}). \tag{29}$$

Therefore, the structural order determined by P and $\hat{\mathcal{B}}$ is directly propagated to the final token sequence.

A.3 Hierarchical Coordinate Quantization

For each vertex $v = (x, y, z)$, coordinates are first normalized:

$$\bar{v}_d = \frac{v_d - v_d^{min}}{v_d^{max} - v_d^{min}}, \quad d \in \{x, y, z\}. \tag{30}$$

The normalized coordinates are quantized into discrete values:

$$q_d = \lfloor \bar{v}_d (R - 1) \rfloor. \tag{31}$$

Instead of independently encoding three axes, we adopt hierarchical coarse-to-fine encoding. Given resolutions N_c and N_f:

$$R = N_c N_f, \tag{32}$$

each coordinate is decomposed as:

$$q_d = N_f q_d^c + q_d^f. \tag{33}$$

Here, q_d^c represents the coarse spatial cell and q_d^f represents the local offset inside that cell.

Algorithm 1: Soft-Interface-Guided Hierarchical Mesh Tokenization

[1] Mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F})$, region assignment P, interface proximity $\hat{\mathcal{B}}$, quantization resolution $R = N_c N_f$ Token sequences $\{\mathbf{X}_k\}_{k=1}^{K}$
for $k = 1, \ldots, K$ **do**
  $\mathcal{F}_k \leftarrow \{f \in \mathcal{F} \mid P(f) = k\}$
  $f_{k,1} \leftarrow \arg\max_{f \in \mathcal{F}_k} \hat{\mathcal{B}}(f)$
  $\mathcal{S} \leftarrow \{f_{k,1}\}$, $\sigma_k \leftarrow [f_{k,1}]$
  while $|\mathcal{S}| < |\mathcal{F}_k|$ **do**
    $\mathcal{C} \leftarrow \{f \in \mathcal{F}_k \setminus \mathcal{S} \mid \exists g \in \mathcal{S},\ f \sim g\}$
    $f^\star \leftarrow \arg\max_{f \in \mathcal{C}} \hat{\mathcal{B}}(f)$
    $\sigma_k \leftarrow \sigma_k \| [f^\star]$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{f^\star\}$
  $\mathcal{V}_k \leftarrow \mathrm{Unique}_{\mathrm{ord}} \left(\|_{f \in \sigma_k} \omega(f) \right)$
  $\mathbf{X}_k \leftarrow [\,]$
  for $\mathbf{v}_{k,t} \in \mathcal{V}_k$ **do**
    Quantize $\mathbf{v}_{k,t}$ into $\mathbf{q}_{k,t} = (q_{k,t,x}, q_{k,t,y}, q_{k,t,z})$
    for $d \in \{x, y, z\}$ **do**
      $q^c_{k,t,d} \leftarrow \lfloor q_{k,t,d} / N_f \rfloor$
      $q^f_{k,t,d} \leftarrow q_{k,t,d} \bmod N_f$
    $\alpha_{k,t} \leftarrow q^c_{k,t,x} N_c^2 + q^c_{k,t,y} N_c + q^c_{k,t,z}$
    $\beta_{k,t} \leftarrow q^f_{k,t,x} N_f^2 + q^f_{k,t,y} N_f + q^f_{k,t,z}$
    $\mathbf{X}_k \leftarrow \mathbf{X}_k \| [(\alpha_{k,t}, \beta_{k,t})]$
return $\{\mathbf{X}_k\}_{k=1}^{K}$

A.4 Coordinate Token Packing

The three spatial dimensions are jointly packed:

$$\alpha = q_x^c N_c^2 + q_y^c N_c + q_z^c, \tag{34}$$

$$\beta = q_x^f N_f^2 + q_y^f N_f + q_z^f. \tag{35}$$

Therefore, each vertex is represented by two discrete symbols:

$$(\alpha, \beta). \tag{36}$$

A.5 Inverse Quantization

During decoding, coarse and fine coordinates are recovered by unpacking:

$$(q_x^c, q_y^c, q_z^c) = \mathrm{unpack}(\alpha), \tag{37}$$

$$(q_x^f, q_y^f, q_z^f) = \mathrm{unpack}(\beta). \tag{38}$$

The original quantized coordinate is reconstructed as:

$$q_d = N_f q_d^c + q_d^f. \tag{39}$$

Finally, the continuous coordinate is recovered by inverse normalization:

$$v_d = \frac{q_d}{R-1}(v_d^{max} - v_d^{min}) + v_d^{min}. \tag{40}$$

This hierarchical encoding preserves 3D spatial coherence while reducing the sequence length required for large-scale mesh generation. The algorithm is presented above.

B Additional Experimental Settings

B.1 Data Curation

Since basic raw 3D assets contain a substantial amount of low-quality, degenerate, or structurally corrupted geometry, we adopt a two-stage data curation pipeline consisting of geometry-based filtering followed by Qwen-based visual quality screening.

Geometry-based Filtering. We first perform a unified set of geometric validity checks on the raw meshes. We retain samples whose face counts fall within 1K–40K and remove meshes that are non-manifold, non-watertight, or contain evident self-intersections. This stage eliminates samples with geometric and topological defects that can be directly detected using explicit mesh properties, preventing invalid connectivity and severely degenerate geometry from entering the training corpus.

Multi-view Rendering. For meshes that pass the geometry-based filtering stage, we further render each sample into a set of multi-view images for subsequent visual quality assessment. Specifically, each mesh is normalized into a common coordinate space and rendered from several predefined viewpoints using a fixed camera configuration and consistent rendering settings. The same background, illumination, and camera parameters are used across all samples to reduce variation introduced by the rendering process. These multi-view renderings provide complementary observations of the global silhouette, local geometry, and surface completeness of each mesh.

Qwen-based Quality Screening. The rendered multi-view images of each mesh are then provided to **Qwen** for a second-stage fine-grained quality screening. Unlike the preceding rule-based filtering, this stage targets visual and structural defects that are difficult to identify using simple topology or face statistics alone. In particular, Qwen is used to detect samples with evident geometric corruption, missing structures, floating components, implausible local connections, severe surface distortions, or generally poor geometric quality. The decision is made jointly from multiple rendered views to better assess the global completeness and structural plausibility of each asset.

Only meshes that pass both the geometric validity checks and the Qwen-based visual quality screening are retained for training. After the complete curation process, the resulting training corpus contains approximately 1M high-quality meshes, with face counts ranging from 1K to 40K. Detailed statistics of the data sources and face-count distribution are provided in Fig. 7.

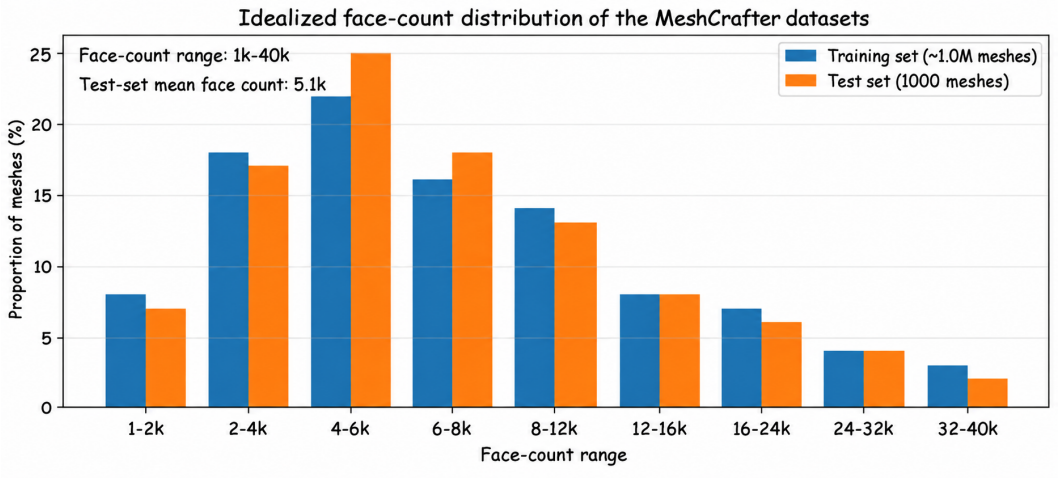

Figure 7: **Face-count distribution of the curated datasets.** Distribution of mesh complexity in the training and test sets, with face counts ranging from 1K to 40K.

B.2 Definition of Face Adjacency Preservation

We introduce Face Adjacency Preservation (FAP) to measure how well a mesh serialization preserves local face connectivity. Let

$$\mathcal{E}_F = \{(f_i, f_j) \mid f_i \sim f_j\} \tag{41}$$

denote the set of face pairs that share an edge in the original mesh, and let $\pi(f)$ denote the position of face f in the serialized sequence. FAP is defined as

$$\mathrm{FAP} = \frac{1}{|\mathcal{E}_F|} \sum_{(f_i,f_j)\in\mathcal{E}_F} 1\left[|\pi(f_i) - \pi(f_j)| \le w\right], \tag{42}$$

where w is a fixed local sequence window shared by all methods. A higher FAP indicates that more originally adjacent faces remain close after serialization, and therefore reflects better preservation of local mesh connectivity.

B.3 LoRA Configuration for Reinforcement Post-Training

During structure-aware reinforcement post-training, we freeze the pretrained MeshCrafter backbone and only optimize the inserted LoRA adapters. We use a LoRA rank of $r = 16$, a scaling factor of $\alpha_{\mathrm{L}} = 32$, and a dropout rate of 0.05. LoRA modules are inserted into the Q/K/V/O projections of self-attention and the up-/gate-/down-projections of the SwiGLU feed-forward layers. For each conditioning input, we sample $G = 8$ complete mesh trajectories. The sequence-level clipping coefficient is set to $\epsilon = 0.2$, the KL regularization coefficient to $\beta_{\mathrm{KL}} = 0.01$, and reinforcement post-training is performed for 50K optimization steps.

C More Experiments

C.1 Ablation on Interface-Aware Region Structuring



Figure 8: **Qualitative comparison of different region partition strategies.** Our full region structuring retains discrete region ownership while further modeling continuous interface proximity, providing a more stable representation for subsequent structure-aware serialization.

The region representation introduced in Sec. 3.1 does not discard discrete partitioning. Instead, it retains the region assignment $P(f)$ while complementing it with the continuous interface proximity $\hat{\mathcal{B}}(f)$. To analyze the necessity of this design, we compare several region partition strategies that rely only on hard assignments, with representative results shown in Fig. 8.

With purely discrete partitioning, each local surface element must be assigned to a single region. Consequently, neighboring elements with similar geometry may receive different labels due to small

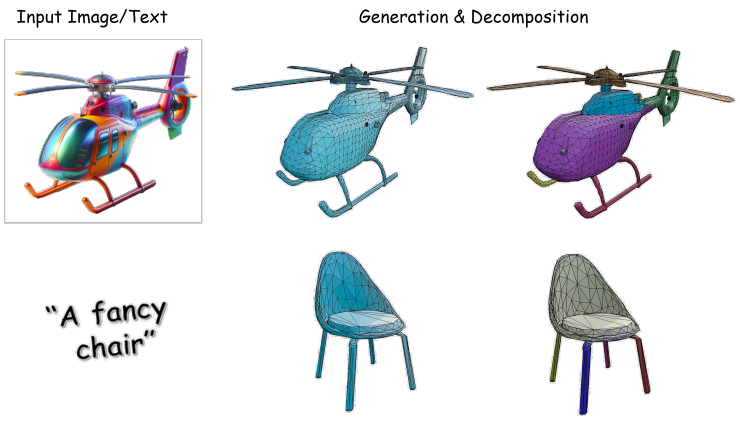

Figure 9: **Multi-modal condition generation.** Image and text inputs are first converted into intermediate 3D geometry by TRELLIS, from which point-cloud conditions are sampled for MeshCrafter.

Table 3: **Multi-modal generation results on a 40-mesh subset.**

Conditioning	CD↓	F1↑	WT%↑
Point cloud	0.059±0.015	0.881±0.062	95%
Image → TRELLIS	0.062±0.016	0.871±0.058	93%
Text → TRELLIS	0.066±0.019	0.858±0.071	89%

variations around the clustering boundary. Such abrupt assignment changes are particularly common around complex connections, thin structures, and high-curvature regions, leading to fragmented regions, unstable boundaries, or incorrect separation of locally coherent structures. Although refining the latent metric improves intra-region compactness and inter-region separation, hard assignments alone still discard the remaining ambiguity around region interfaces.

Our region structuring therefore adopts a complementary discrete–continuous representation. The discrete assignment $P(f)$ preserves the regional organization required for efficient localized generation, while $\hat{\mathcal{B}}(f)$ explicitly measures how close a local surface element is to a structural transition. The visualization in Fig. 8 focuses on the discrete component of different partition strategies, where hard partitioning tends to produce fragmented or unstable region boundaries around complex geometry. In the full model, these discrete regions are further complemented by continuous interface proximity, allowing subsequent serialization to avoid relying solely on rigid region labels. The advantage of our formulation therefore comes from jointly preserving *discrete region organization* and *continuous transition information*, rather than replacing hard partitioning with a fully soft assignment.

C.2 Multi-Modal Condition Generation

Specifically, we use TRELLIS to convert image or text inputs into intermediate 3D geometry, and then sample point clouds from the generated geometry as the conditioning input of MeshCrafter. In this way, different input modalities are mapped into a unified geometric conditioning space without modifying the autoregressive mesh generator.

Figure 9 presents representative examples, while Table 3 reports results on a 40-mesh subset. Compared with direct point-cloud conditioning, image- and text-conditioned generation introduce only a moderate degradation, mainly due to the additional modality-to-geometry conversion stage. Nevertheless, MeshCrafter maintains stable geometric fidelity and watertightness across all three conditioning settings, indicating that the proposed generation framework can effectively operate on geometric conditions obtained from different upstream modalities.

C.3 Ablation of Parameter Sensitivity and Robustness

We further study the robustness of the interface-proximity signal and the sensitivity to representative hyperparameters in the region structuring and reinforcement post-training stages.

Table 4: Robustness to perturbations of the interface-proximity signal.

Noise level σ	CD↓	F1↑	WT%↑	Stitch error↓
0.0	0.062	0.863	89	2.9
0.1	0.065	0.856	86	3.8
0.3	0.071	0.841	81	5.4
0.5	0.082	0.819	76	8.1
1.0	0.098	0.786	68	11.7

We first perturb the estimated interface proximity $\hat{\mathcal{B}}$ with different noise levels σ. As shown in Table 4, the performance degrades gradually as the perturbation becomes stronger. Even at $\sigma = 0.5$, the model still achieves a WT of 76%, indicating that the generation process does not rely on an exact interface estimate. Meanwhile, the increasing stitch error suggests that inaccurate transition information mainly affects local connections between neighboring regions.

Table 5: **Hyperparameter sensitivity.** Default values are underlined.

A_{ref}	CD↓	WT%↑	G	CD↓	WT%↑
0.50	0.071	83%	2	0.068	86%
0.75	0.066	86%	4	0.065	87%
<u>1.00</u>	<u>0.062</u>	<u>89%</u>	<u>8</u>	<u>0.062</u>	<u>89%</u>
1.25	0.064	87%	12	0.063	88%
1.50	0.069	83%	16	0.063	88%

We also vary the reference scale A_{ref} in Sec. 3.1 and the rollout group size G in Sec. 3.3. For A_{ref}, overly small values produce a more concentrated interface response, while overly large values make the transition signal less discriminative. The default $A_{\text{ref}} = 1.0$ achieves the best performance in this sweep. For reinforcement post-training, increasing G from 2 to 8 consistently improves generation quality by providing more stable group-relative comparisons. Further increasing G brings little additional gain while increasing the rollout cost. We therefore use $A_{\text{ref}} = 1.0$ and $G = 8$ in all experiments.

D AI Use Statement

Generative AI tools were used in this work for several research and writing tasks. In data curation, we used Qwen to assist with fine-grained visual quality screening of multi-view renderings of candidate 3D meshes, as described in Appendix B.1. We also used large language models to assist with language polishing, translation, organization and presentation of experimental results, formatting of tables and references, and refinement of parts of the manuscript. LLMs were additionally used to provide feedback on the presentation of the methodology and experimental design. All AI-assisted content, including dataset filtering decisions, technical descriptions, figures, references, and manuscript text, was reviewed and verified by the authors. The authors made the final decisions on the methodology, experiments, analyses, and conclusions, and take full responsibility for all content of this work.