LOOK BOTH WAYS: FUTURE-INFORMED AUTOREGRESSIVE HIGH-POLY MESH GENERATION

Anonymous authors
Paper under double-blind review



Figure 1: **High-poly 3D mesh generation results of FAMesh**. Future-part topological cues guide autoregressive decoding toward topologically consistent high-face-count meshes.

ABSTRACT

Existing part-wise autoregressive 3D mesh generation methods predict each next part conditioned only on preceding parts. Without explicit future-informed topological compatibility guidance, early local topological biases can accumulate along the generation chain and eventually lead to topological drift in high-face-count settings. We propose FAMesh, a topologically consistent framework for high-face-count 3D mesh generation. FAMesh formulates target mesh generation as part-wise autoregressive prediction of topological sequences and introduces a global topological scaffold (TopoScaf), which captures the coarse global structure of the target mesh as a compact part-level topological sequence. TopoScaf complements the target-part autoregression with explicit cues about future-part topological compatibility, steering generation along a topologically consistent trajectory. Furthermore, we design Part-Structured Asymmetric Interaction (PSAI), which reformulates information transfer required for TopoScaf modeling and TopoScaf-to-target topological guidance into efficient interactions over discriminative part-level structural summaries. This design achieves a $1.52\times$ training speedup, without compromising TopoScaf modeling fidelity or its topological guidance for target generation. Experiments show that FAMesh outperforms existing methods in generation quality and topological consistency.

1 INTRODUCTION

Recent advances in 3D content generation have substantially improved the quality and complexity of synthesized 3D assets Poole et al. (2023); Li et al. (2026a); Tang et al. (2025c). Among existing 3D

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.

representations, meshes are particularly appealing because they explicitly encode surface geometry, topological connectivity, and structural details, making them well suited for applications in computer vision Ren et al. (2025b); Duggal et al. (2025), virtual reality Rasch et al. (2025); Rahimi et al. (2025), and embodied AI Xia et al. (2026); Cao et al. (2025). Despite this progress, maintaining global topological consistency in high-face-count mesh generation remains challenging.

Traditional mesh generation methods treat the entire mesh as a single generation unit Lai et al. (2026); Hunyuan3D Team et al. (2025); Wu et al. (2025). While effective at capturing global shape, such formulations often struggle to preserve local topological detail and global structural coherence simultaneously, especially for complex objects. To address this limitation, subsequent works adopt part-level modeling, decomposing object generation into local part synthesis and explicit inter-part composition. This paradigm improves fine-grained controllability and structural stability Li et al. (2025b); Yang et al. (2025). Yet, early part-based methods were developed under continuous or implicit representation paradigm, which does not directly model discrete face connectivity. As a result, they remain prone to topological artifacts and non-watertight outputs Li et al. (2026b).

Recent methods tokenize mesh parts into 1D topological sequences and generate vertices and faces part by part with autoregressive models. This paradigm explicitly models discrete topological relationships and therefore improves topological consistency Chen et al. (2025c); Zhao et al. (2025); Weng et al. (2025b). However, their prefix-driven generation predicts each next part solely from preceding parts, without explicit future-informed topological compatibility guidance Yang et al. (2026); Lionar et al. (2025). Early local topological biases therefore propagate through the generation chain and accumulate into topological drift. Subsequent improvements primarily mitigate prefix-induced interference by shortening prefix dependency, either through compact part tokenization or aggressive preceding-context pruning Tang et al. (2025a); Chen et al. (2025c); Weng et al. (2025b); Gao et al. (2025); Xu et al. (2026); Wang et al. (2026a), extending topology-consistent generation to around $30K$ faces. Despite these gains, biased prefixes still govern subsequent predictions, leading to persistent topological failures when scaling to more practical face counts.

To address these limitations, we propose FAMesh, a topologically consistent framework for high-poly 3D mesh generation. FAMesh formulates target mesh generation as part-wise autoregressive prediction of topological sequences and introduces a global topological scaffold (TopoScaf), which captures the coarse global structure of the target mesh as a compact part-level topological sequence. Unlike prefix-only target-part autoregression, TopoScaf complements this process with a part-aligned global topological prior that encodes explicit cues about future-part topological compatibility, counteracting prefix-induced drift and steering generation toward a topologically consistent trajectory. Furthermore, we design Part-Structured Asymmetric Interaction (PSAI), which reformulates information transfer for TopoScaf cross-part modeling and TopoScaf-derived topological guidance for target generation as efficient interactions over discriminative part-level structural summaries. This design preserves TopoScaf modeling fidelity and effective target guidance, while achieving a $1.52\times$ training speedup.

Our contributions are as follows:

- We propose FAMesh, a future-informed autoregressive generation framework, enabling topology-consistent mesh generation at up to $80K$ faces.
- We introduce TopoScaf, a part-aligned global topological prior with explicit future-part compatibility cues, and PSAI, which enables efficient TopoScaf modeling and TopoScaf-to-target topological guidance through part-level structural summaries.
- Experiments show that FAMesh outperforms existing methods in generation quality, topological consistency, and scalability.

2 Related Work

2.1 3D Generation

Recent 3D content generation has advanced from coarse representations, such as point clouds and implicit fields Zhang et al. (2026); Tang et al. (2024); Lan et al. (2026); He et al. (2025), to direct mesh and texture generation Xiang et al. (2025); Lu et al. (2025); Wang et al. (2024); Wu et al. (2024), with architectures also shifting from feed-forward and multi-stage reconstruction Xu et al. (2024a); Yu et al. (2025) to diffusion-based frameworks Li et al. (2025a); Ma et al. (2025); Wang et al. (2026b).

However, most methods treat an object as a holistic generation unit Lai et al. (2025); Tochilkin et al. (2024), making it difficult to preserve local topological detail and global structural coherence under complex compositional structures.

2.2 Part-level 3D Mesh Generation

To better capture object compositionality, recent methods have shifted from whole-object modeling to part-level generation. Early approaches follow a two-stage pipeline that first predicts part segmentation or layout and then generates individual parts Deng et al. (2026); Dong et al. (2025); Tang et al. (2025b); Yang et al. (2025), making them vulnerable to decomposition errors that propagate through generation. Later methods, such as PartCrafter and MoCA, jointly model parts and their interactions to improve detail controllability and structural stability Lin et al. (2025); Li et al. (2025b). However, these early efforts rely on continuous or implicit representations, which cannot explicitly constrain discrete face connectivity, leaving topological artifacts and non-watertight outputs difficult to avoid.

2.3 Part-level Autoregressive Mesh Generation

To explicitly model discrete mesh topology, recent methods tokenize mesh parts into 1D topological sequences and generate them autoregressively part-by-part Weng et al. (2025b); Lionar et al. (2025); Chen et al. (2025a;b). Representative works such as MeshXL show the feasibility of combining mesh tokenization with autoregressive generation Chen et al. (2024). However, relying solely on preceding parts makes these methods prone to bias accumulation, causing structural disconnection and topological drift Arora et al. (2022); Ren et al. (2025a). Existing improvements reduce such prefix interference by shortening or weakening preceding-part dependence. EdgeRunner and Silk improve tokenization compactness, via compressed layer adjacency, to shorten the full prefix sequence and improve sequence learnability Tang et al. (2025a); Song et al. (2026). Meshtron and MeshMosaic instead prune preceding-part context at the modeling level, using compact cues such as boundary information for current-part prediction Xu et al. (2026); Hao et al. (2024). Nevertheless, both directions remain fundamentally prefix-conditioned, leaving cross-part topological failures difficult to avoid when scaling to high-face-count meshes. Although prior works, PivotMesh and MeshArt, introduce coarse global representations to constrain geometry or part-level structure, such guidance does not explicitly specify a consistent topological trajectory, leaving cross-part topological drift unresolved Weng et al. (2025a); Gao et al. (2025). In contrast, FAMesh establishes a global topological scaffold, complementing explicit future-part compatibility cues for topology-consistent generation.

3 Method

3.1 Overall

As shown in Fig. 2, given an input image $\mathcal{I}$, FAMesh generates a high-face-count 3D mesh in a part-organized topological sequence space. The target mesh $\mathcal{M}$ is first decomposed into an ordered set of parts,

$$\mathcal{M} = \left[\mathcal{M}^{(1)}, \mathcal{M}^{(2)}, \ldots, \mathcal{M}^{(N)}\right], \tag{1}$$

where N denotes the number of parts and the part order is fixed during preprocessing. Each part is then tokenized into a 1D topological sequence by a mesh tokenizer $\texttt{Tok}(\cdot)$:

$$\mathcal{S}^{(n)} = \texttt{Tok}\left(\mathcal{M}^{(n)}\right) = \left[s_1^{(n)}, s_2^{(n)}, \ldots, s_{L^{(n)}}^{(n)}\right], \qquad \mathcal{S} = \left[\mathcal{S}^{(1)}, \mathcal{S}^{(2)}, \ldots, \mathcal{S}^{(N)}\right], \tag{2}$$

where, $s_l^{(n)}$ denotes the l-th token of the n-th part, $L^{(n)}$ is its sequence length, $\mathcal{S}$ represents the full target sequence. Therefore, FAMesh aims to generate the part sequences $\{\hat{\mathcal{S}}^{(n)}\}_{n=1}^{N}$ and reconstructs the final mesh by detokenizing $\texttt{Tok}^{-1}(\cdot)$ each part and concatenating $\texttt{Concat}(\cdot)$ them in the predefined order:

$$\hat{\mathcal{M}} = \texttt{Concat}\left(\texttt{Tok}^{-1}(\hat{\mathcal{S}}^{(1)}), \texttt{Tok}^{-1}(\hat{\mathcal{S}}^{(2)}), \ldots, \texttt{Tok}^{-1}(\hat{\mathcal{S}}^{(N)})\right). \tag{3}$$

FAMesh consists of Scaffold Generator (Scafor) and Scaffold-guided Part Autoregressor (ScafPAR). Given an input image $\mathcal{I}$, Scafor predicts TopoScaf, a compact part-level topological sequence that

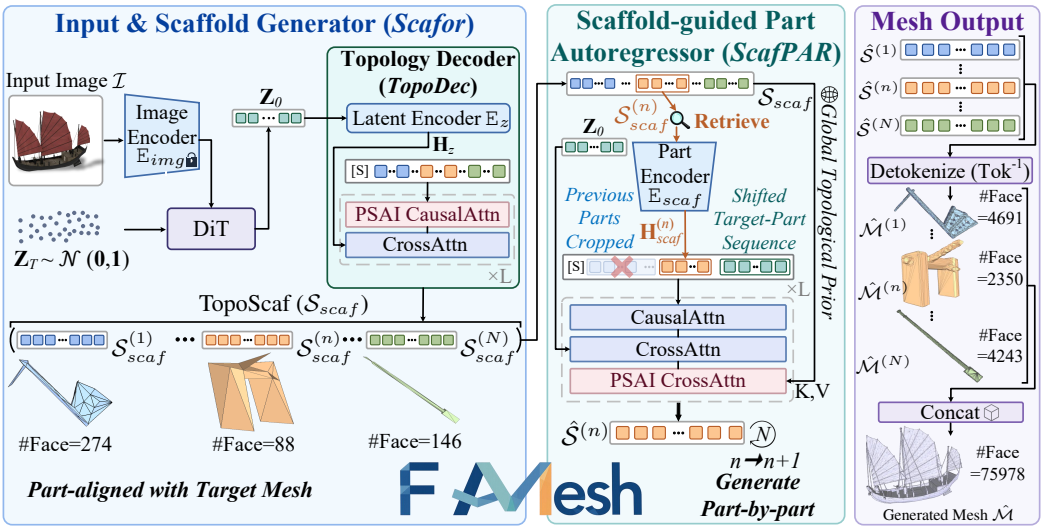


Figure 2: **Overview of FAMesh.** Given an input image $\mathcal{I}$, Scafor autoregressively predicts a part-aligned global topological scaffold (TopoScaf $\mathcal{S}_{scaf}$) with a DiT and a PSAI-equipped TopoDec. ScafPAR then autoregressively generates the high-resolution target part sequence $\hat{\mathcal{S}}^{(n)}$ one at a time, using the aligned TopoScaf part $\mathcal{S}^{(n)}_{scaf}$ as a structural prefix $\mathbf{H}^{(n)}_{scaf}$ and the full TopoScaf as a PSAI-mediated global prior to provide explicit cues about future-part topological compatibility before detokenization and part concatenation.

preserves the coarse global structure of the target mesh $\mathcal{M}$ and is aligned with its part decomposition and ordering. ScafPAR then generates the target sequence $\mathcal{S}$ part by part, using TopoScaf as a global prior to complement explicit cues about future-part topological compatibility and maintain topology-consistent generation.

We further introduce Part-Structured Asymmetric Interaction (PSAI), which reformulates information transfer for TopoScaf cross-part modeling and TopoScaf-to-target guidance into efficient interactions over discriminative part-level structural summaries, improving training efficiency while preserving TopoScaf fidelity and guidance effectiveness.

We introduce Scafor, ScafPAR, and PSAI in Sec. 3.2, Sec. 3.3, and Sec. 3.4, respectively.

3.2 Scaffold Generator

As shown in Fig. 2, Scafor predicts a TopoScaf $\mathcal{S}_{scaf}$ from the input image $\mathcal{I}$. $\mathcal{S}_{scaf}$ is a compact serialized sequence that is explicitly aligned with the target mesh $\mathcal{M}$ in part decomposition and ordering:

$$\mathcal{S}_{scaf} = \left[\mathcal{S}^{(1)}_{scaf}, \mathcal{S}^{(2)}_{scaf}, \cdots, \mathcal{S}^{(N)}_{scaf}\right], \qquad \mathcal{S}^{(n)}_{scaf} = \left[s^{(n)}_{scaf,1}, \ldots, s^{(n)}_{scaf,L^{(n)}_{scaf}}\right], \tag{4}$$

where $L^{(n)}_{scaf} \ll L^{(n)}$, and $s^{(n)}_{scaf,l}$ denotes the l-th topological token of the n-th TopoScaf part. Each $\mathcal{S}^{(n)}_{scaf}$ encodes a face-sparse yet topologically complete structure of the target part $\mathcal{M}^{(n)}$. Thus, $\mathcal{S}_{scaf}$ preserves part order and cross-part topological organization with reduced sequence length, making it an efficient part-aligned structural prior for ScafPAR. Scafor consists of a Diffusion Transformer (DiT) and a Topology Decoder (TopoDec).

DiT. Given an input image $\mathcal{I}$, an image encoder $\mathrm{E}_{img}(\cdot)$ extracts image features, on which DiT conditions to sample an image-conditioned latent code $\mathbf{Z}_0 \in \mathbb{R}^{L_z \times d_z}$ from Gaussian noise. This code serves as the shared object-level condition for both TopoDec and ScafPAR.

TopoDec. Given the image-conditioned latent code $\mathbf{Z}_0$, TopoDec decodes it into the TopoScaf $\mathcal{S}_{scaf}$. We first project $\mathbf{Z}_0$ into an object-level topological representation:

$$\mathbf{H}_z = \mathrm{E}_z(\mathbf{Z}_0), \qquad \mathbf{H}_z \in \mathbb{R}^{L_h \times d_h}. \tag{5}$$

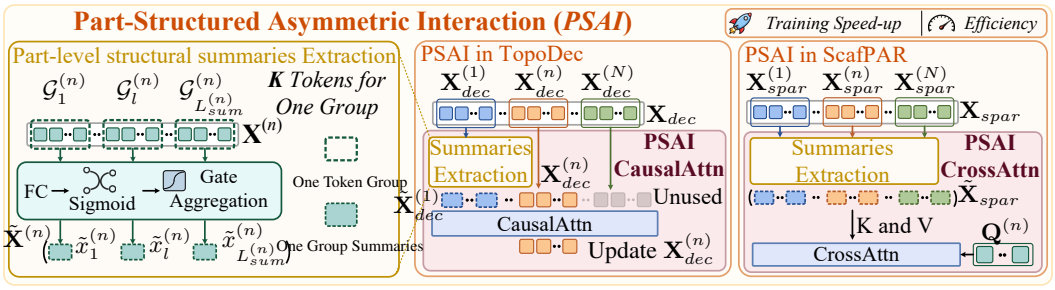


Figure 3: **Illustration of PSAI.** PSAI extracts discriminative part-level structural summaries by grouping every K tokens and applying gated aggregation. In TopoDec, it compresses preceding-part interactions while preserving full local intra-part causal modeling for the current part. In ScafPAR, TopoScaf summaries serve as compact keys and values for cross-attention, enabling efficient TopoScaf-to-target topological guidance.

Conditioned on $\mathbf{H}_z$, TopoDec autoregressively predicts the flattened TopoScaf:

$$p(\mathcal{S}_{scaf} \mid \mathbf{H}_z) = \prod_{l=1}^{L_{scaf}} p(s_{scaf,l} \mid s_{scaf,<l}, \mathbf{H}_z), \qquad L_{scaf} = \sum_{n=1}^{N} L_{scaf}^{(n)}, \tag{6}$$

where $s_{scaf,l}$ denotes the l-th token and $s_{scaf,<l}$ denotes its preceding tokens. TopoDec uses PSAI-based causal attention to improve TopoScaf modeling efficiency while preserving part-level topological fidelity. Further analysis of TopoScaf is provided in the Section 4.3 and B. During training, a point-cloud encoder is used only to provide a KL-regularized latent target for TopoDec and DiT latent learning; it is removed at inference, where $\mathbf{Z}_0$ is sampled by the image-conditioned DiT.

3.3 Scaffold-guided Part Autoregressor

As shown in Fig. 2, ScafPAR generates the high-resolution target sequence $\mathcal{S}$ part by part under the guidance of TopoScaf $\mathcal{S}_{scaf}$. Unlike existing autoregressive methods that condition each part on preceding-part tokens, ScafPAR completely removes cross-part prefix tokens and preserves autoregressive dependency only within the current part. This reduces the quadratic cost of long-prefix attention and mitigates prefix-induced error accumulation.

Since $\mathcal{S}_{scaf}$ is explicitly aligned with $\mathcal{S}$ in part decomposition and ordering, the part index n serves as a structured pointer to retrieve the corresponding TopoScaf part $\mathcal{S}_{scaf}^{(n)}$, rather than as a learnable identity embedding. We encode $\mathcal{S}_{scaf}^{(n)}$ into a fixed-length structural prefix:

$$\mathbf{H}_{scaf}^{(n)} = \mathrm{E}_{scaf}\left(\mathcal{S}_{scaf}^{(n)}\right), \qquad \mathbf{H}_{scaf}^{(n)} \in \mathbb{R}^{L_{pre} \times d}, \qquad L_{pre} \ll L_{scaf}^{(n)}, \tag{7}$$

where $\mathrm{E}_{scaf}(\cdot)$ denotes the TopoScaf part encoder. This prefix provides an aligned structural anchor for the n-th target part without feeding the full TopoScaf subsequence into the decoder. The generation of the n-th target part is formulated as

$$p\left(\hat{\mathcal{S}}^{(n)} \mid \mathbf{H}_{scaf}^{(n)}, \mathcal{S}_{scaf}, \mathbf{Z}_0\right) = \prod_{l=1}^{L^{(n)}} p\left(\hat{s}_l^{(n)} \mid \hat{s}_{<l}^{(n)}, \mathbf{H}_{scaf}^{(n)}, \mathcal{S}_{scaf}, \mathbf{Z}_0\right), \tag{8}$$

where $\hat{\mathcal{S}}^{(n)}$ is the generated part sequence, $\hat{s}_l^{(n)}$ is its l-th token, and $\hat{s}_{<l}^{(n)}$ denotes the within-part autoregressive prefix. The latent code $\mathbf{Z}_0$ provides object-level conditions through standard cross-attention, while $\mathcal{S}_{scaf}$ is injected through PSAI-based cross-attention to transfer cross-part topological organization and future-part compatibility cues. In this way, ScafPAR generates topology-consistent target parts efficiently.

3.4 Part-Structured Asymmetric Interaction

As shown in Fig. 3, PSAI adapts interaction granularity to mesh structure by compressing TopoScaf cross-part modeling and TopoScaf-to-target cross-granularity interactions into discriminative part-level structural summaries. This improves the efficiency of information transfer without weakening

local topological representation. We next define the part-level structural summaries and describe their instantiations in TopoDec and ScafPAR.

Part-level structural summaries. Given a part-organized feature sequence

$$\mathbf{X} = \Big[\mathbf{X}^{(1)}; \cdots ; \mathbf{X}^{(N)}\Big], \qquad \mathbf{X}^{(n)} = \Big[x_1^{(n)}; \cdots ; x_{L_x^{(n)}}^{(n)}\Big] \in \mathbb{R}^{L_x^{(n)} \times d_x}, \tag{9}$$

where $[\cdot;\cdot]$ denotes feature concatenation and $\mathbf{X}^{(n)}$ denotes the token features of the n-th part. We partition each part into $L_{sum}^{(n)} = \lceil L_x^{(n)}/K \rceil$ local groups, with $\mathcal{G}_l^{(n)}$ denoting the l-th group. Each group is compressed into a structural summary token by gated aggregation:

$$\tilde{x}_l^{(n)} = \sum_{x_i^{(n)} \in \mathcal{G}_l^{(n)}} \alpha_i^{(n)} x_i^{(n)}, \qquad \alpha_i^{(n)} = \sigma\Big(f(x_i^{(n)})\Big), \tag{10}$$

where $f(\cdot)$ is a fully connected layer and $\sigma(\cdot)$ is the Sigmoid. The resulting summary sequence is

$$\tilde{\mathbf{X}}^{(n)} = \Big[\tilde{x}_1^{(n)}; \cdots ; \tilde{x}_{L_{sum}^{(n)}}^{(n)}\Big] \in \mathbb{R}^{L_{sum}^{(n)} \times d_x}. \tag{11}$$

Thus, $\tilde{\mathbf{X}}^{(n)}$ provides a K-fold compressed part representation that preserves discriminative structural cues. The generic variables are instantiated differently in TopoDec and ScafPAR.

PSAI in TopoDec. In TopoDec, the generic input $\mathbf{X}$ in Eq. equation 9 is instantiated as the causal hidden states $\mathbf{X}_{dec} = [\mathbf{X}_{dec}^{(1)}; \cdots ; \mathbf{X}_{dec}^{(N)}]$ for TopoScaf prediction, where $\mathbf{X}_{dec}^{(n)} \in \mathbb{R}^{L_{scaf}^{(n)} \times d_x}$ and $L_{scaf}^{(n)} = L_x^{(n)}$. Because fine-grained topological dependencies are primarily intra-part, whereas cross-part dependencies mainly encode coarse structural coordination, PSAI preserves full causal attention within the current part and summarizes preceding parts:

$$\texttt{PSAI}_{causal}(\mathbf{X}_{dec})^{(n)} = \texttt{CausalAttn}\Big(\Big[\tilde{\mathbf{X}}_{dec}^{(<n)}; \mathbf{X}_{dec}^{(n)}\Big]\Big), \qquad \tilde{\mathbf{X}}_{dec}^{(<n)} = \Big[\tilde{\mathbf{X}}_{dec}^{(1)}; \cdots ; \tilde{\mathbf{X}}_{dec}^{(n-1)}\Big], \tag{12}$$

where, $\texttt{PSAI}_{causal}(\mathbf{X}_{dec})^{(n)}$ denotes updated causal hidden states for predicting the n-th TopoScaf part, $\tilde{\mathbf{X}}_{dec}^{(n)}$ denotes structural summaries of the n-th part. This reduces cross-part causal cost while preserving fine-grained intra-part topology for efficient TopoScaf modeling.

PSAI in ScafPAR. In ScafPAR, the generic input $\mathbf{X}$ in Eq. equation 9 is instantiated as the TopoScaf token features $\mathbf{X}_{spar} = [\mathbf{X}_{spar}^{(1)}; \cdots ; \mathbf{X}_{spar}^{(N)}]$ for target guidance, where $\mathbf{X}_{spar}^{(n)} \in \mathbb{R}^{L_{scaf}^{(n)} \times d_x}$ and $L_{scaf}^{(n)} = L_x^{(n)}$. Since ScafPAR mainly exploits coarse cross-part topological cues from $\mathcal{S}_{scaf}$ rather than dense local topological details, PSAI compresses TopoScaf features into part-level structural summaries for efficient cross-granularity guidance:

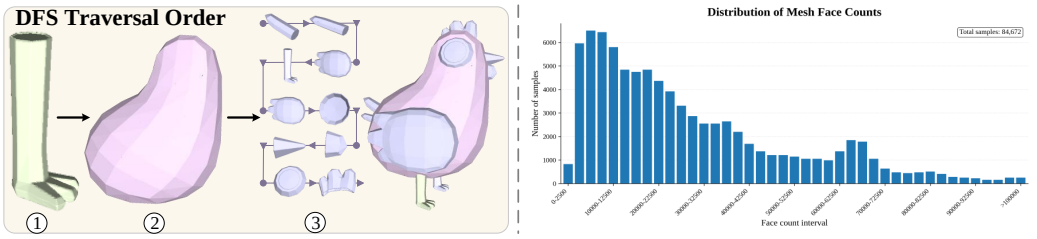
$$\texttt{PSAI}_{cross}\Big(\mathbf{Q}^{(n)}, \mathbf{X}_{spar}\Big) = \texttt{CrossAttn}\Big(\mathbf{Q}^{(n)}, \tilde{\mathbf{X}}_{spar}\Big), \qquad \tilde{\mathbf{X}}_{spar} = \Big[\tilde{\mathbf{X}}_{spar}^{(1)}; \cdots ; \tilde{\mathbf{X}}_{spar}^{(N)}\Big], \tag{13}$$

where $\mathbf{Q}^{(n)}$ denotes the query features for predicting $\hat{\mathcal{S}}^{(n)}$, and $\tilde{\mathbf{X}}_{spar}^{(n)}$ denotes the structural summaries of the n-th TopoScaf part. This avoids dense cross-attention over the full TopoScaf sequence while preserving global topology and future-part compatibility cues.

4 Experiment

4.1 Implementation

Training Details. FAMesh is implemented in PyTorch and trained on 32 NVIDIA H20 GPUs for five weeks. We use the EdgeRunner Tang et al. (2025a) mesh tokenizer with 512-bin coordinate quantization and sequence-control tokens. TopoScaf is obtained by independently simplifying each target part using quadric edge-collapse decimation while preserving fine-to-coarse face ancestry. The recorded ancestry then guides the traversal of simplified faces during EdgeRunner tokenization. The details is provided in Algorithm 1. Training follows three stages: TopoDec is trained with a point-cloud encoder in a VAE-style formulation using shifted cross-entropy and KL regularization, DiT is optimized with an MSE diffusion objective, and ScafPAR is trained with teacher forcing to generate high-resolution mesh tokens conditioned on TopoScaf. The learning rate decays from 10^{-5} to 10^{-6} for DiT and from 10^{-4} to 10^{-6} for TopoDec and ScafPAR. All models are optimized with Adam, accelerated by FlashAttention Dao (2024), and evaluated on a single H20 GPU.



(a) Illustration of parts with DFS order (b) Distribution of face count in our dataset

Figure 4: **Part ordering and dataset statistics**.(a) Mesh parts are ordered by DFS traversal, starting from the part nearest to a fixed 3D spatial anchor, to obtain a consistent part sequence. (b) Face-count distribution of our curated subset, covering $85K$ meshes with diverse topological complexity.

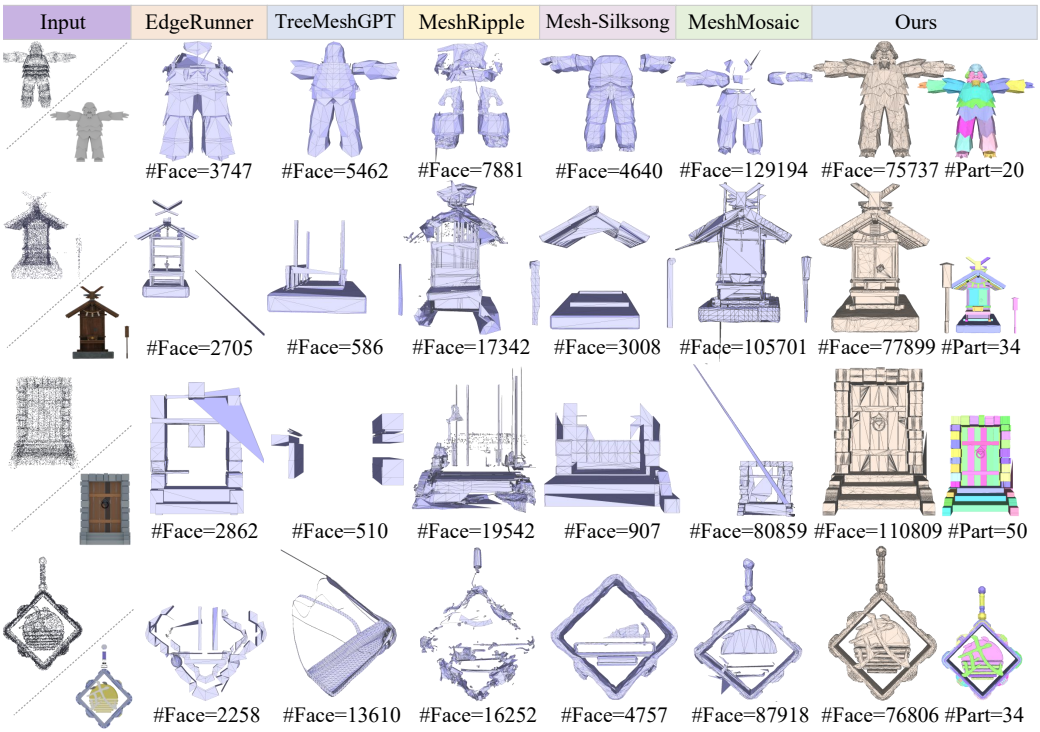


Figure 5: **Qualitative comparison.** Existing autoregressive baselines either preserve only coarse structures or become unstable as face scale increases. In contrast, FAMesh maintains coherent structure, fine detail, and topological consistency at high-face-count settings.

Dataset and augmentation. We curate a $85K$-mesh part-decomposed subset from ObjaverseXL Deitke et al. (2023), randomly selecting 500 meshes for testing and using remainder for training. Part order is fixed by DFS on the part adjacency graph, starting from the part nearest to a fixed 3D spatial anchor. Fig. 4 illustrates the part ordering and dataset statistics. For robustness, we apply random mesh rotations, Gaussian noise to encoder-side point clouds, and input image degradation, and additionally perturb TopoScaf sequences in a topology-preserving manner during ScafPAR training.

4.2 Comparisons

Baselines and Metrics. We compare FAMesh with five representative autoregressive mesh generation baselines. EdgeRunner and MeshRipple support image-conditioned generation Tang et al. (2025a); Lin et al. (2026), while TreeMeshGPT, Mesh-Silksong, and MeshMosaic take point clouds as input Lionar et al. (2025); Song et al. (2026); Xu et al. (2026). We evaluate mesh quality using ground-truth-based metrics and reference-free metrics Xu et al. (2026; 2024b); Chen & Zhang (2021); Ravi et al. (2020); Son et al. (2025); Charton et al. (2021). For ground-truth-based evaluation, Hausdorff distance (HD)

Table 1: **Quantitative comparison of ground-truth-based generation quality**. **Best** results are shown in bold, and second-best results are underlined. FAMesh consistently achieves the best performance across all metrics, showing substantially lower surface and edge reconstruction errors while improving normal consistency and F-score scores.

Method	HD ↓	CD_{L1} ↓	CD_{L2} $\times 10^2$ ↓	GT-NC ↑	F1 ↑	ECD ↓	EF1 ↑
EdgeRunner Tang et al. (2025a) (ICLR2025)	0.220	0.110	0.103	0.556	0.030	0.170	0.012
TreeMeshGPT Lionar et al. (2025) (CVPR2025)	0.223	0.072	0.057	0.564	0.048	0.111	0.041
MeshRipple Lin et al. (2026) (CVPR2026)	0.282	0.161	0.211	0.517	0.022	0.190	0.028
Mesh-Silksong Song et al. (2026) (ICLR2026)	0.096	0.034	0.016	0.659	0.105	0.073	0.112
MeshMosaic Xu et al. (2026) (CVPR2026)	0.400	0.121	0.164	0.523	0.006	0.185	0.082
FAMesh (Ours)	**0.027**	**0.018**	**0.010**	**0.938**	**0.305**	**0.046**	**0.505**

measures worst-case surface deviation, Chamfer distance (CD_{L1} and CD_{L2}) measures bidirectional surface distances, ground-truth normal consistency (GT-NC) evaluates normal agreement with the ground truth, F-score (F1) measures the surface precision–recall balance, and edge Chamfer distance (ECD) and edge F-score (EF1) assess sharp-edge fidelity. For reference-free evaluation, self-normal consistency (SNC) measures local normal consistency within the generated mesh, non-manifold edge ratio (NME) quantifies non-manifold edge violations, aspect ratio (AR) evaluates triangle shape quality, and mean opinion score (MOS) on a 1–5 scale to assess human preference over visual quality.

Table 2: **Quantitative comparison of intrinsic mesh quality.** FAMesh achieves more regular triangle shapes, and the highest human preference.

Method	SNC ↑	NME ↓	AR ↓	MOS ↑
EdgeRunner Tang et al. (2025a)	0.825	0.238	18.09	2.41
TreeMeshGPT Lionar et al. (2025)	0.885	0.027	8.55	1.68
MeshRipple Lin et al. (2026)	0.752	0.136	12.54	2.73
Mesh-Silksong Song et al. (2026)	0.841	**0.012**	27.24	2.74
MeshMosaic Xu et al. (2026)	0.767	0.339	32.28	3.52
FAMesh (Ours)	**0.924**	0.023	**4.55**	**4.32**

Qualitative Comparisons. As shown in Fig. 5, existing autoregressive baselines either retain only coarse object structure at low face counts or become fragmented when scaled to complex shapes. EdgeRunner, TreeMeshGPT, MeshRipple, and Mesh-Silksong frequently miss fine structures and suffer from severe part discontinuities. MeshMosaic increases the generated face count, but the larger prefix-driven generation scale also amplifies instability, resulting in broken topology, spurious components, and weak global alignment. Its quantitative scores are further affected by systematic orientation deviations under the same input condition; we do not apply post-hoc pose correction, since orientation consistency is part of conditional generation fidelity. In contrast, FAMesh consistently preserves coherent object-level structure, dense geometric detail, and well-aligned part organization across diverse categories, achieving finer part granularity without sacrificing topological consistency.

Table 3: **Ablation study.** Removing TopoScaf yields the fastest ScafPAR inference by eliminating TopoScaf-guidance computation, but substantially degrades generation quality, confirming that TopoScaf provides the main quality gains. PSAI improves training and inference efficiency with comparable geometric performance.

Module	TopoScaf	PSAI	HD ↓	CD_{L1} ↓	GT-NC ↑	MOS ↑	Training Time (h) ↓	Inference (Tok/s) ↑
TopoDec	N/A	✗	0.088	0.071	0.853	**3.92**	495	79.1
	N/A	✓	**0.087**	**0.068**	**0.861**	3.81	**336**	**80.6**
ScafPAR	✗	✗	0.145	0.092	0.575	3.47	529	**78.2**
	✓	✗	0.031	0.025	0.924	4.30	582	70.4
	✓	✓	**0.027**	**0.018**	**0.938**	**4.32**	**371**	73.8

Quantitative Comparisons. Tables 1 and 2 report ground-truth-based fidelity and intrinsic mesh quality, respectively. FAMesh achieves the best performance on all ground-truth-based metrics, while substantially improving GT-NC, F1, and EF1. These gains indicate that FAMesh better preserves both surface geometry and edge-level topology. For intrinsic quality, FAMesh achieves the highest SNC and the lowest AR, showing smoother local surface consistency and fewer degenerate triangle shapes. Although Mesh-Silksong obtains a slightly lower NME, its weaker GT alignment and much larger AR indicate that local normal smoothness alone does not translate into faithful mesh generation.

4.3 Ablation Study

Table 3 demonstrates that TopoScaf and PSAI contribute complementary benefits. In TopoDec, PSAI markedly improves training efficiency and slightly increases decoding throughput, while preserving comparable perceptual quality and marginally improving geometric metrics. In ScafPAR, removing TopoScaf gives the fastest inference but substantially degrades all quality metrics; visually, in Fig. 6, this variant exhibits severe topological decay, fragmented parts, and unstable layouts. Adding TopoScaf restores coherent object-level structure and brings large gains in reconstruction quality, showing the importance of explicit future-part topological compatibility cues. Further incorporating PSAI improves the efficiency of TopoScaf-guided generation and achieves the best ScafPAR training efficiency, suggesting that part-level structural summaries allow the model to exploit TopoScaf guidance more effectively than relying on dense cross-granularity interactions.

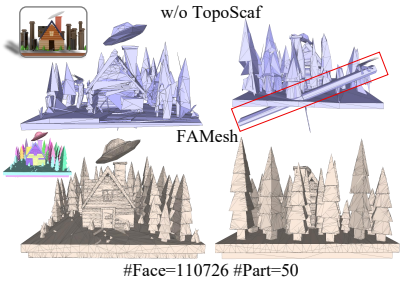


Figure 6: **Ablation visualization.** Without TopoScaf, the model exhibits severe topological decay and fragmented structures, whereas FAMesh preserves consistent topology and well-aligned parts at high face and part counts.

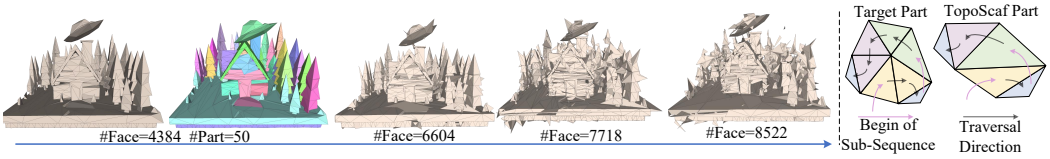


Figure 7: **Analysis of TopoScaf.** Left: a $4K$-face TopoScaf captures part organization and global connectivity with mild prefix-induced drift, serving as a reliable global topological prior. Right: face merging shortens each TopoScaf part but preserves its target-part traversal topology, yielding explicit traversal-consistent topological cues for ScafPAR.

Analysis of TopoScaf. As shown in Fig. 7 left, a TopoScaf with about $4K$ faces already preserves the part organization and global connectivity of a complex mesh, while its short sequence mitigates prefix-induced drift, making it a reliable global topological prior. Densifying TopoScaf reintroduces long-sequence prefix bias, leading to topological drift and missing faces. Right panel shows that each TopoScaf part is a coarsened counterpart of its target part: face simplification that preserves fine-to-coarse face ancestry shortens the sequence but preserves the face traversal topology, providing traversal-consistent alignment and explicit topological cues for ScafPAR.

Robustness to Preprocessing Choices. As shown in Table 4, we repartition parts at inference, replace the default DFS part ordering with BFS, and replace only the face-traversal rule within the same tokenizer. The latter two variants are retrained accordingly. All three remain stable under several preprocessing variations. We further remove traversal consistency by discarding the fine-to-coarse face ancestry used to guide the traversal of the simplified TopoScaf, such that the coarse and target meshes are traversed independently. This causes substantial degradation, even slightly below ScafPAR without TopoScaf. This highlights the advantage of TopoScaf over generic coarse structural priors by directly providing a part-aligned, traversal-consistent topological trajectory to mitigate topological drift.

Table 4: **Robustness to preprocessing choices.** FAMesh is insensitive to reasonable preprocessing variations, while removing traversal consistency substantially degrades performance.

Setting	$CD_{L1}\downarrow$	GT-NC ↑	EF1 ↑	NME ↓
Part Repartition	<u>0.019</u>	<u>0.934</u>	0.496	**0.018**
Part Reordering (BFS)	0.020	0.931	0.488	0.033
Replace Face Traversal	0.029	0.933	<u>0.499</u>	0.026
w/o Traversal Consistency	0.097	0.562	0.058	0.103
FAMesh (default)	**0.018**	**0.938**	**0.505**	<u>0.023</u>

4.4 Conclusion

FAMesh is a future-informed autoregressive framework for high-face-count 3D mesh generation. It introduces TopoScaf, a compact part-aligned global topological scaffold that complements prefix-only generation with explicit cues about future-part topological compatibility, mitigating topological drift and structural disconnection. FAMesh further designs PSAI to enable efficient cross-part and cross-granularity topology transfer, producing dense, coherent, and topologically consistent meshes.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, improving clarity and readability, and limited code debugging during the preparation of this work. All technical content, methodological design, experimental procedures, results, analyses, and conclusions were developed and verified by the authors.

REFERENCES

Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Cheung. Why Exposure Bias Matters: An Imitation Learning Perspective of Error Accumulation in Language Generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 700–710, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.58.

Ziang Cao, Zhaoxi Chen, Liang Pan, and Ziwei Liu. PhysX-3D: Physical-Grounded 3D Asset Generation. *Advances in Neural Information Processing Systems*, 38, 2025.

Jerome Charton, Stephen Baek, and Youngjun Kim. Mesh Repairing Using Topology Graphs. *Journal of Computational Design and Engineering*, 8(1):251–267, 2021. doi: 10.1093/jcde/qwaa076.

Dian Chen, Yansong Qu, Xinyang Li, Ming Li, and Shengchuan Zhang. XSpecMesh: Quality-Preserving Auto-Regressive Mesh Generation Acceleration via Multi-Head Speculative Decoding, 2025a.

Sijin Chen, Xin Chen, Anqi Pang, Xianfang Zeng, Wei Cheng, Yijun Fu, Fukun Yin, Zhibin Wang, Jingyi Yu, Gang Yu, Bin Fu, and Tao Chen. MeshXL: Neural Coordinate Field for Generative 3D Foundation Models. *Advances in Neural Information Processing Systems*, 37:97141–97166, 2024.

Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Zhongang Cai, Lei Yang, Gang Yu, Guosheng Lin, and Chi Zhang. MeshAnything: Artist-Created Mesh Generation with Autoregressive Transformers. In *International Conference on Learning Representations (ICLR)*, 2025b.

Yiwen Chen, Yikai Wang, Yihao Luo, Zhengyi Wang, Zilong Chen, Jun Zhu, Chi Zhang, and Guosheng Lin. MeshAnything V2: Artist-Created Mesh Generation with Adjacent Mesh Tokenization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13922–13931, 2025c.

Zhiqin Chen and Hao Zhang. Neural Marching Cubes. *ACM Transactions on Graphics*, 40(6):1–15, 2021. doi: 10.1145/3478513.3480518.

Tri Dao. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-XL: A Universe of 10M+ 3D Objects. *Advances in Neural Information Processing Systems*, 36, 2023.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6367–6376, 2026.

Shaocong Dong, Lihe Ding, Xiao Chen, Yaokun Li, Yuxin Wang, Yucheng Wang, Qi Wang, Jaehyeok Kim, Chenjian Gao, Zhanpeng Huang, Zibin Wang, Tianfan Xue, and Dan Xu. From One to More: Contextual Part Latents for 3D Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8230–8240, 2025.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Shivam Duggal, Yushi Hu, Oscar Michel, Aniruddha Kembhavi, William T Freeman, Noah A Smith, Ranjay Krishna, Antonio Torralba, Ali Farhadi, and Wei-Chiu Ma. Eval3D: Interpretable and Fine-Grained Evaluation for 3D Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13326–13336, 2025.

Daoyi Gao, Yawar Siddiqui, Lei Li, and Angela Dai. MeshArt: Generating Articulated Meshes with Structure-Guided Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 618–627, 2025.

Zekun Hao, David W Romero, Tsung-Yi Lin, and Ming-Yu Liu. Meshtron: High-Fidelity, Artist-Like 3D Mesh Generation at Scale, 2024.

Xianglong He, Zi-Xin Zou, Chia-Hao Chen, Yuan-Chen Guo, Ding Liang, Chun Yuan, Wanli Ouyang, Yan-Pei Cao, and Yangguang Li. SparseFlex: High-Resolution and Arbitrary-Topology 3D Shape Modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14822–14833, 2025.

Hunyuan3D Team, Bowen Zhang, Chunchao Guo, Haolin Liu, Hongyu Yan, Huiwen Shi, Jingwei Huang, Junlin Yu, Kunhong Li, Linus, Penghao Wang, Qingxiang Lin, Sicong Liu, Xianghui Yang, Yixuan Tang, Yunfei Zhao, Zeqiang Lai, Zhihao Liang, and Zibo Zhao. Hunyuan3D-Omni: A Unified Framework for Controllable Generation of 3D Assets, 2025.

Zeqiang Lai, Yunfei Zhao, Haolin Liu, Zibo Zhao, Qingxiang Lin, Huiwen Shi, Xianghui Yang, Mingxin Yang, Shuhui Yang, Yifei Feng, Sheng Zhang, Xin Huang, Di Luo, Fan Yang, Fang Yang, Lifu Wang, Sicong Liu, Yixuan Tang, Yulin Cai, Zebin He, Tian Liu, Yuhong Liu, Jie Jiang, Linus, Jingwei Huang, and Chunchao Guo. Hunyuan3D 2.5: Towards High-Fidelity 3D Assets Generation with Ultimate Details, 2025.

Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Xin Yang, Xin Huang, Jingwei Huang, Xiangyu Yue, and Chunchao Guo. NaTex: Seamless Texture Generation as Latent Color Diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18619–18629, 2026.

Yushi Lan, Fangzhou Hong, Shangchen Zhou, Shuai Yang, Xuyi Meng, Yongwei Chen, Zhaoyang Lyu, Bo Dai, Xingang Pan, and Chen Change Loy. LN3Diff++: Scalable Latent Neural Fields Diffusion for Speedy 3D Generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3):3002–3017, 2026. doi: 10.1109/TPAMI.2025.3633073.

Jinwei Li, Huan-Ang Gao, Wenyi Li, Haohan Chi, Chenyu Liu, Chenxi Du, Yiqian Liu, Mingju Gao, Guiyu Zhang, Zongzheng Zhang, Li Yi, Yao Yao, Jingwei Zhao, Hongyang Li, Yikai Wang, and Hao Zhao. FB-4D: Spatial-Temporal Coherent Dynamic 3D Content Generation with Feature Banks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5290–5301, 2026a.

Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. CraftsMan3D: High-fidelity Mesh Generation with 3D Native Diffusion and Interactive Geometry Refiner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5307–5317, 2025a.

Yanfeng Li, Tao Tan, Qinquan Gao, Zhiwen Cao, Xiaohong Liu, and Yue Sun. HiFi-Mesh: High-Fidelity Efficient 3D Mesh Generation via Compact Autoregressive Dependence. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(8):6566–6574, 2026b.

Zhiqi Li, Wenhuan Li, Tengfei Wang, Zhenwei Wang, Junta Wu, Haoyuan Wang, Yunhan Yang, Zehuan Huang, Yang Li, Peidong Liu, and Chunchao Guo. MoCA: Mixture-of-Components Attention for Scalable Compositional 3D Generation, 2025b.

Junkai Lin, Hang Long, Huipeng Guo, Jielei Zhang, Jiayi Yang, Tianle Guo, Yang Yang, Jianwen Li, Wenxiao Zhang, Matthias Nießner, and Wei Yang. MeshRipple: Structured Autoregressive Generation of Artist-Meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12706–12718, 2026.

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. *Advances in Neural Information Processing Systems*, 38, 2025.

Stefan Lionar, Jiabin Liang, and Gim Hee Lee. TreeMeshGPT: Artistic Mesh Generation with Autoregressive Tree Sequencing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26608–26617, 2025.

Jiawei Lu, Yingpeng Zhang, Zengjun Zhao, He Wang, Kun Zhou, and Tianjia Shao. GenesisTex2: Stable, Consistent and High-Quality Text-to-Texture Generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(6):5820–5828, 2025.

Zhiyuan Ma, Xinyue Liang, Rongyuan Wu, Xiangyu Zhu, Zhen Lei, and Lei Zhang. Progressive Rendering Distillation: Adapting Stable Diffusion for Instant Text-to-Mesh Generation without 3D Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11036–11050, 2025.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, 2023.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3D Using 2D Diffusion. In *International Conference on Learning Representations (ICLR)*, 2023.

Fatema Rahimi, Abolghasem Sadeghi-Niaraki, and Soo-Mi Choi. Generative AI meets virtual reality: A comprehensive survey on applications, challenges, and future direction. *IEEE Access*, 13: 94893–94909, 2025. doi: 10.1109/ACCESS.2025.3574779.

Julian Rasch, Julia Töws, Teresa Hirzle, Florian Müller, and Martin Schmitz. CreepyCoCreator? Investigating AI Representation Modes for 3D Object Co-Creation in Virtual Reality. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1–14, 2025. doi: 10.1145/3706598.3713720.

Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3D Deep Learning with PyTorch3D, 2020.

Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond Next-Token: Next-X Prediction for Autoregressive Visual Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15781–15791, 2025a.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. GEN3C: 3D-Informed World-Consistent Video Generation with Precise Camera Control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6121–6132, 2025b.

Sanghyun Son, Matheus Gadelha, Yang Zhou, Matthew Fisher, Zexiang Xu, Yi-Ling Qiao, Ming C. Lin, and Yi Zhou. DMesh++: An Efficient Differentiable Mesh for Complex Shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26590–26599, 2025.

Gaochao Song, Zibo Zhao, Haohan Weng, Jingbo Zeng, Rongfei Jia, and Shenghua Gao. Topology-Preserved Auto-regressive Mesh Generation in the Manner of Weaving Silk. In *International Conference on Learning Representations (ICLR)*, 2026.

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *Computer Vision – ECCV*, pp. 1–18. Springer, 2024.

Jiaxiang Tang, Zhaoshuo Li, Zekun Hao, Xian Liu, Gang Zeng, Ming-Yu Liu, and Qinsheng Zhang. EdgeRunner: Auto-regressive Auto-encoder for Artistic Mesh Generation. In *International Conference on Learning Representations (ICLR)*, 2025a.

Jiaxiang Tang, Ruijie Lu, Max 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. *Advances in Neural Information Processing Systems*, 38, 2025b.

Zhenyu Tang, Junwu Zhang, Xinhua Cheng, Wangbo Yu, Chaoran Feng, Yatian Pang, Bin Lin, and Li Yuan. Cycle3D: High-Quality and Consistent Image-to-3D Generation via Generation-Reconstruction Cycle. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(7): 7320–7328, 2025c.

Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. TripoSR: Fast 3D Object Reconstruction from a Single Image, 2024.

Hanxiao Wang, Yuan-Chen Guo, Ying-Tian Liu, Zi-Xin Zou, Biao Zhang, Weize Quan, Ding Liang, Yan-Pei Cao, and Dong-Ming 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 (CVPR)*, pp. 12719–12729, 2026a.

Renke Wang, Zhenyu Zhang, Ying Tai, and Jian Yang. DiffProxy: Multi-View Human Mesh Recovery via Diffusion-Generated Dense Proxies, 2026b.

Zhengyi Wang, Jonathan Lorraine, Yikai Wang, Hang Su, Jun Zhu, Sanja Fidler, and Xiaohui Zeng. LLaMA-Mesh: Unifying 3D Mesh Generation with Language Models, 2024.

Haohan Weng, Yikai Wang, Tong Zhang, CL Chen, and Jun Zhu. PivotMesh: Generic 3D Mesh Generation via Pivot Vertices Guidance. In *International Conference on Learning Representations (ICLR)*, 2025a.

Haohan Weng, Zibo Zhao, Biwen Lei, Xianghui Yang, Jian Liu, Zeqiang Lai, Zhuo Chen, Yuhong Liu, Jie Jiang, Chunchao Guo, Tong Zhang, Shenghua Gao, and C. L. Philip Chen. Scaling Mesh Generation via Compressive Tokenization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11093–11103, 2025b.

Kailu Wu, Fangfu Liu, Zhihan Cai, Runjie Yan, Hanyang Wang, Yating Hu, Yueqi Duan, and Kaisheng Ma. Unique3D: High-Quality and Efficient 3D Mesh Generation from a Single Image. *Advances in Neural Information Processing Systems*, 37:125116–125141, 2024.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, and Yao Yao. Direct3D-S2: Gigascale 3D Generation Made Easy with Spatial Sparse Attention. *Advances in Neural Information Processing Systems*, 38, 2025.

Hongchi Xia, Xuan Li, Zhaoshuo Li, Qianli Ma, Jiashu Xu, Ming-Yu Liu, Yin Cui, Tsung-Yi Lin, Wei-Chiu Ma, Shenlong Wang, Shuran Song, and Fangyin Wei. SAGE: Scalable Agentic 3D Scene Generation for Embodied AI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22358–22368, 2026.

Xiaoyu Xiang, Liat Sless Gorelik, Yuchen Fan, Omri Armstrong, Forrest N. Iandola, Yilei Li, Ita Lifshitz, and Rakesh Ranjan. Make-A-Texture: Fast Shape-Aware Texture Generation in 3 Seconds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4872–4881, 2025.

Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. InstantMesh: Efficient 3D Mesh Generation from a Single Image with Sparse-View Large Reconstruction Models, 2024a.

Rui Xu, Longdu Liu, Ningna Wang, Shuangmin Chen, Shiqing Xin, Xiaohu Guo, Zichun Zhong, Taku Komura, Wenping Wang, and Changhe Tu. CWF: Consolidating Weak Features in High-quality Mesh Simplification. *ACM Transactions on Graphics*, 43(4):1–14, 2024b. doi: 10.1145/3658159.

Rui Xu, Tianyang Xue, Qiujie Dong, Le Wan, Zhe Zhu, Peng Li, Zhiyang Dou, Cheng Lin, Shiqing Xin, Yuan Liu, Wenping Wang, and Taku Komura. MeshMosaic: Scaling Artist Mesh Generation via Local-to-Global Assembly. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20003–20013, 2026.

Yichen Yang, Hong Li, Haodong Zhu, Linin Yang, Guojun Lei, Sheng Xu, and Baochang Zhang. PartDiffuser: Part-wise 3D Mesh Generation via Discrete Diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19940–19949, 2026.

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. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–12, 2025. doi: 10.1145/3757377.3763872.

Qiao Yu, Xianzhi Li, Yuan Tang, Xu Han, Long Hu, Yixue Hao, and Min Chen. Fancy123: One Image to High-Quality 3D Mesh Generation via Plug-and-Play Deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 595–604, 2025.

Ruijie Zhang, Bixin Zeng, Shengpeng Wang, Fuhui Zhou, and Wei Wang. RaLD: Generating High-Resolution 3D Radar Point Clouds with Latent Diffusion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(22):18773–18781, 2026.

Ruowen Zhao, Junliang Ye, Zhengyi Wang, Guangce Liu, Yiwen Chen, Yikai Wang, and Jun Zhu. DeepMesh: Auto-Regressive Artist-Mesh Creation with Reinforcement Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10612–10623, 2025.

A MODEL ARCHITECTURE DETAILS

FAMesh follows the pipeline shown in Fig. 2. The image encoder is a pretrained ViT Dosovitskiy et al. (2021), and the DiT module follows the standard DiT architecture without architectural modification Peebles & Xie (2023). Given an input image, DiT predicts a KL-regularized object-level latent code $\mathbf{Z}_0$, which provides a compact distribution-regularized representation for TopoScaf generation.

In Scafor, $\mathbf{Z}_0$ is further processed by a latent encoder $\mathtt{E}_z(\cdot)$ to obtain discriminative topological features $\mathbf{H}_z$ for TopoDec. Rather than directly decoding from $\mathbf{Z}_0$, we use a 6-layer self-attention encoder for $\mathtt{E}_z(\cdot)$, which refines the $\mathbf{Z}_0$ into topology-aware features. TopoDec is implemented as a 24-layer autoregressive decoder with causal attention, cross-attention to $\mathbf{H}_z$, and PSAI-equipped causal interaction for efficient cross-part modeling.

In ScafPAR, the part encoder $\mathtt{E}_{scaf}(\cdot)$ encodes the retrieved TopoScaf part $\mathcal{S}^{(n)}_{scaf}$ into a fixed-length structural prefix $\mathbf{H}^{(n)}_{scaf}$. We implement $\mathtt{E}_{scaf}(\cdot)$ as a 4-layer self-attention encoder and set the prefix length to $L_{pre} = 64$, as mentioned in Eq. 8. The layer numbers of $\mathtt{E}_z(\cdot)$ and $\mathtt{E}_{scaf}(\cdot)$ follow the empirical design guidance of Meshtron Hao et al. (2024). ScafPAR itself is a 24-layer autoregressive decoder, where preceding target-part tokens are pruned and autoregression is preserved only within the current part. The decoder uses causal attention for current-part generation, cross-attention for image-latent conditioning, and PSAI cross-attention to inject compact TopoScaf-derived global topological guidance.

Group Size. As shown in Table 5, we first ablate the PSAI grouping size K, which controls the granularity of part-level structural summaries used in both TopoScaf modeling and TopoScaf-guided target mesh generation. A smaller K retains finer structural summaries but increases computation: compared with the default $K = 32$, using $K = 8$ increases training time by 39% and slightly reduces inference speed, while bringing no consistent quality improvement. Larger K provides only marginal efficiency gains but can make the summaries overly coarse. For example, $K = 64$ reduces training time by only 3% and improves inference speed by 2%, yet worsens CD_{L1} by 14.9% and decreases MOS from 4.34 to 4.12. Although some individual metrics fluctuate across different group sizes, no alternative setting consistently dominates the default. We therefore adopt $K = 32$, which provides a stable efficiency–quality trade-off for the full FAMesh pipeline. We further evaluate the aggregation rule and part-aware organization in PSAI. Replacing gated aggregation with mean pooling degrades both geometric accuracy and perceptual quality, indicating that learned gates are useful for extracting discriminative structural summaries. Removing explicit part-aware organization and using token-group as pseudo-parts leads to a more pronounced performance drop, especially in GT-NC and MOS, suggesting that PSAI benefits from the semantic and topological consistency of true mesh parts.

Table 5: **Architecture ablation.** We ablate the efficiency-oriented designs in FAMesh, including PSAI and the bottleneck factorization of Q/K/V projections. For PSAI, we study the grouping size K, the aggregation rule for structural summaries, and the explicit part-aware organization. Training time and inference speed are normalized to the default configuration. The default setting uses part-aware gated aggregation with $K = 32$ and a 128-dimensional projection bottleneck, yielding a favorable efficiency–quality trade-off.

Ablated Factor	Variant	Train Time ↓	Inf. Speed ↑	CD_{L1} ↓	GT-NC ↑	HD ↓	MOS ↑
PSAI granularity	$K = 8$	1.39×	0.96×	0.051	0.872	0.056	4.33
PSAI granularity	$K = 16$	1.12×	0.98×	0.048	0.891	0.051	4.31
PSAI granularity	$K = 32$ **(default)**	1.00×	1.00×	0.047	0.884	0.053	4.34
PSAI granularity	$K = 64$	0.97×	1.02×	0.054	0.922	0.049	4.12
PSAI aggregation	Mean pooling, $K = 32$	1.04×	1.01×	0.055	0.876	0.061	4.18
PSAI structure	Token-group as pseudo-parts	0.96×	1.06×	0.060	0.761	0.067	4.05
Q/K/V projection	Full projection	1.07×	0.78×	0.052	0.887	0.059	4.35
Q/K/V projection	**Bottleneck projection (default)**	1.00×	1.00×	0.047	0.884	0.053	4.34

Attention Projections. We further factorize the $Q/K/V$ projections in all attention blocks of TopoDec and ScafPAR, including causal attention, cross-attention, and PSAI cross-attention. Each original linear projection is replaced by a two-step bottleneck projection that maps features to a

Algorithm 1 Traversal-Consistent TopoScaf Construction

Require: Target part $M^{(n)}$, target face budget $B^{(n)}$, EdgeRunner tokenizer $\mathrm{Tok}(\cdot)$
Ensure: TopoScaf sequence $S^{(n)}_{\mathrm{scaf}}$

$(S^{(n)}, \tau^{(n)}) \leftarrow \mathrm{Tok}(M^{(n)}, \texttt{return_face_order})$ ▷ target-part traversal
$\widetilde{M}^{(n)} \leftarrow M^{(n)}$
$\mathcal{A}(f) \leftarrow \{f\},\ \forall f \in F(M^{(n)})$ ▷ initialize face ancestry
while $|F(\widetilde{M}^{(n)})| > B^{(n)}$ **do**
  $e^\star \leftarrow \arg\min_e \mathrm{QEMCost}(e)$
  $(\widetilde{M}^{(n)}, \mathcal{R}, \mathcal{N}) \leftarrow \mathrm{Collapse}(\widetilde{M}^{(n)}, e^\star)$
  for $f \in \mathcal{R}$ **do**
    $g \leftarrow \mathrm{Owner}(f, \mathcal{N})$
    $\mathcal{A}(g) \leftarrow \mathcal{A}(g) \cup \mathcal{A}(f)$
  end for
end while
for $\bar{f} \in F(\widetilde{M}^{(n)})$ **do**
  $r(\bar{f}) \leftarrow \min\{i \mid f_i^{(n)} \in \mathcal{A}(\bar{f})\}$
end for
$\tau^{(n)}_{\mathrm{scaf}} \leftarrow \mathrm{Sort}\left(F(\widetilde{M}^{(n)}); r(\cdot)\right)$ ▷ ancestry-projected traversal
$S^{(n)}_{\mathrm{scaf}} \leftarrow \mathrm{Tok}\left(\widetilde{M}^{(n)}; \texttt{guide} = \tau^{(n)}_{\mathrm{scaf}}\right)$
return $S^{(n)}_{\mathrm{scaf}}$

128-dimensional intermediate space before projecting them back to the target dimension:

$$\phi(\mathbf{X}) = \mathbf{X}\mathbf{W}_{down}\mathbf{W}_{up}, \qquad \mathbf{W}_{down} \in \mathbb{R}^{d_{in}\times 128}, \quad \mathbf{W}_{up} \in \mathbb{R}^{128\times d_{out}}. \tag{14}$$

Compared with full Q/K/V projections, the bottleneck design reduces training time from $1.07\times$ to $1.00\times$ and improves inference speed from $0.78\times$ to $1.00\times$, corresponding to a 28.2% relative speedup. Meanwhile, generation quality remains stable: CD_{L1} and HD are improved by 9.6% and 10.2%, respectively, while GT-NC and MOS show only negligible changes. These results indicate that factorized attention projections provide a more efficient feature extraction scheme without evident degradation in mesh generation quality.

Traversal-Consistent TopoScaf Construction. As shown in Algorithm 1, given the target-part traversal $\tau^{(n)}$ produced by the EdgeRunner tokenizer, we simplify $M^{(n)}$ using QEM while tracking the fine-to-coarse face ancestry. Each surviving coarse face is assigned the earliest traversal position among its fine-face ancestors, which defines the ancestry-projected traversal $\tau^{(n)}_{\mathrm{scaf}}$. The simplified mesh is then serialized by the same EdgeRunner tokenizer following this traversal order; when the next guided face cannot be reached by a valid traversal transition, a new sub-sequence is started. This construction maintains four invariants: First, QEM preservation: ancestry tracking does not alter the QEM collapse cost or collapse order; Second, ancestry completeness: the ancestry sets of all surviving faces form a partition of the original target-part faces; Third, traversal consistency: $\tau^{(n)}_{\mathrm{scaf}}$ preserves the relative order induced by the target traversal $\tau^{(n)}$ after face merging; and Finally, serialization validity: the resulting sequence always follows the original EdgeRunner token grammar, with a new valid sub-sequence introduced whenever direct traversal cannot continue.

B More Experiment Results

Table 6: **Effect of TopoScaf resolution on target mesh generation.** We report target-mesh quality together with the computational costs of TopoDec and ScafPAR under different TopoScaf resolutions.

#Faces	$\mathrm{CD}_{L1}\downarrow$	GT-NC ↑	EF1 ↑	NME ↓	TopoDec (Training/Inference) ↓	ScafPAR (Training/Inference) ↓	ScafPAR Tok/s ↑
∼2.5K	0.064	0.692	0.136	0.081	15.3G/7.5G	26.71G/12.83G	75.2
∼4.5K	**0.018**	**0.938**	**0.505**	**0.023**	18.6G/8.2G	28.92G/14.41G	73.8
∼7.5K	0.036	0.730	0.286	0.049	25.8G/11.4G	33.05G/17.96G	69.9

TopoScaf Resolution. As shown in Table 6, $2.5K$ TopoScaf removes excessive structural details during simplification, providing insufficient guidance for ScafPAR and leading to clearly degraded target-mesh quality. Increasing the resolution to $4.5K$ preserves richer geometric and topological cues while maintaining a compact scaffold representation. In contrast, $7.5K$ further increases autoregressive sequence complexity and computational cost, which reduces scaffold-generation stability without yielding additional quality gains. We therefore adopt $4.5K$ faces as the best trade-off between structural fidelity, generation stability, and efficiency. All results use a batch size of 1.

Runtime and Throughput. Exact face-count matching across different methods is difficult because each method has its own generation granularity and inference procedure. Consequently, wall-clock time alone does not provide a directly comparable measure of generation efficiency when the output resolutions differ substantially. We therefore report the output face count together with token throughput, runtime, and face throughput in Table 7. FAMesh generates 75,737 faces, substantially more than most baselines, which naturally results in a longer absolute generation time. To account for this difference, we additionally report Faces/s as an output-normalized, geometry-level efficiency measure. Since FAMesh follows a two-stage generation pipeline, its end-to-end face throughput is computed using the number of faces in the final high-resolution mesh divided by the total runtime of Scafor and ScafPAR. This result shows that the longer absolute runtime of FAMesh is largely associated with its higher output resolution rather than low average generation efficiency.

Table 7: **Runtime and throughput comparisons.** For FAMesh, Tok/s and time are reported as Scafor/ScafPAR. Faces/s is computed using the final mesh face count divided by the total two-stage runtime.

Method	Faces	Tok/s ↑	Time (s)	Faces/s ↑
EdgeRunner	3,747	52.8	745	5.03
TreeMeshGPT	5,462	<u>78.4</u>	627	8.71
MeshRipple	7,881	31.3	2,304	3.42
Mesh-Silksong	4,640	13.49	633	7.34
MeshMosaic	129,194	42.7	8,703	**14.84**
FAMesh	75,737	**80.6**/73.8	248/5,404	<u>13.41</u>

Inference Cost Analysis. Table 8 reports the stage-wise inference cost of FAMesh. Scafor is executed once per object to generate the complete ∼4.5K-face TopoScaf, requiring 8.2 GB peak memory and 248 s inference time. ScafPAR then generates the high-resolution target mesh sequentially at the part level. Since preceding target-part tokens are removed from the autoregressive context, its peak memory is governed primarily by the currently decoded part, rather than the accumulated face count of the whole target mesh. As the current part grows from ∼1K to ∼4.5K faces, peak memory increases moderately from 9.7 GB to 14.4 GB. In contrast, decoding time grows more rapidly because a larger part introduces an increasingly long within-part causal prefix during autoregressive generation. Consequently, scaling to a higher-face-count mesh mainly increases the accumulated decoding time across parts, while the peak memory remains bounded by the current-part generation cost under a fixed TopoScaf configuration.

Table 8: **Inference cost and part-wise scaling.** Scafor generates the complete TopoScaf once, whereas ScafPAR decodes high-resolution target parts sequentially.

Stage	Generation Unit	#Faces	Peak Mem. (GB)	Time (s)
Scafor	Full TopoScaf	∼4.5K	8.2	248
ScafPAR	Current target part	∼1K	9.7	59
	Current target part	∼2K	11.1	131
	Current target part	∼3K	12.4	214
	Current target part	∼4.5K	14.4	332

Inter-Part Consistency. FAMesh targets artist-authored meshes composed of independently indexed connected components, combined by concatenation without cross-part vertex or edge welding. Therefore, we further adopt three established metrics to comprehensively evaluate inter-part contact consistency: (1) Adjacency Precision and (2) Adjacency Recall, measuring the correctness and completeness of the induced inter-part contact graph, respectively. (3) Mesh Collision Rate, the percentage of parts involved in at least one significant interpenetration. As shown in Table 9, TopoScaf consistently improves contact-graph fidelity, interface connectivity, and geometric contact quality in the generated high-resolution meshes.

Table 9: **Inter-part contact consistency.** Higher adjacency precision and recall indicate better inter-part connectivity, while a lower mesh collision rate is preferred.

Method	Adjacency Precision ↑	Adjacency Recall ↑	Mesh Collision Rate ↓
MeshMosaic	72.3	69.9	51.5
Mesh-Silksong	68.6	61.3	43.5
FAMesh (w/o TopoScaf)	56.0	44.0	61.0
FAMesh	**83.9**	**83.8**	**15.4**

Table 10: **User study.** We report average preference scores from 24 participants under four perceptual criteria. Higher scores indicate stronger user preference.

Method	Neatness ↑	Artistry ↑	Similarity to GT ↑	Detail Recovery ↑	Avg. ↑
EdgeRunner	0.70	0.65	<u>0.96</u>	0.56	0.72
MeshRipple	0.93	0.88	0.90	0.74	0.86
TreeMeshGPT	0.21	0.29	0.37	0.32	0.30
Mesh-Silksong	0.36	0.45	0.49	0.42	0.43
MeshMosaic	<u>1.48</u>	<u>1.35</u>	0.82	<u>1.55</u>	<u>1.30</u>
FAMesh	**2.32**	**2.38**	**2.46**	**2.41**	**2.39**

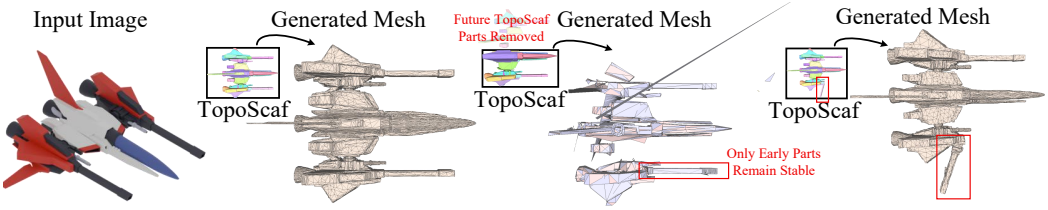


Figure 8: **Functional analysis of TopoScaf guidance.** Complete TopoScaf guidance yields a coherent high-resolution mesh with well-preserved part layout and global structure. When TopoScaf is truncated after a selected part, the earliest pre-cutoff parts remain relatively stable, but the generation soon drifts even before the removed scaffold region is fully reached, indicating that TopoScaf serves not merely as a per-part coarse template, but as a part-aligned global prior providing past, current, and future-part compatibility cues. Local TopoScaf perturbations further induce corresponding structural changes in the generated mesh, showing that ScafPAR actively follows the scaffold guidance.

User Study. Although the mean opinion score (MOS) provides a compact overall perceptual assessment, a single score cannot fully explain which visual aspects contribute to the preference of generated meshes. We therefore conduct a fine-grained user study. Specifically, we compare FAMesh with EdgeRunner, MeshRipple, TreeMeshGPT, Mesh-Silksong, and MeshMosaic on four perceptual criteria: Neatness, which measures whether the generated mesh is clean, well organized, and free of noisy artifacts or fragmented structures; Artistry, which evaluates whether the mesh exhibits visually plausible, artist-like structure and editable surface layout; Similarity to GT, which measures the agreement with the ground-truth shape in global pose, part arrangement, and object proportions; and Detail Recovery, which evaluates the preservation of fine structures, sharp features, and local geometric details. We randomly sample 30 test objects covering diverse categories and geometry complexities. For each object, results from all methods are rendered under the same camera views, lighting condition, and visualization style. The input image and ground-truth mesh are shown as references, while method names are hidden and the display order is randomized. A total of 24 participants with experience in computer graphics, 3D modeling, or visual inspection are invited to evaluate the results. For each criterion, participants select the top three methods, which receive 3, 2, and 1 points, respectively, while the remaining methods receive 0 points. The final score of each method is averaged over all objects and participants. As shown in Table 10, FAMesh achieves the highest preference scores across all four criteria, indicating that the proposed TopoScaf-guided generation improves not only geometric fidelity but also perceptual mesh quality and fine-detail recovery.

Functional analysis of TopoScaf. We conduct a qualitative intervention study to examine whether TopoScaf acts as an effective structural prior for ScafPAR beyond a simple per-part coarse template. As shown in Fig. 8, the complete TopoScaf leads to a coherent high-resolution mesh with well-preserved part layout and global structure. When TopoScaf is truncated after a selected part, ScafPAR still receives the corresponding scaffold segments for the pre-cutoff parts, and the earliest generated parts remain relatively stable. However, the generation soon degrades even before the removed scaffold region is fully reached, producing part drift, sparse-looking surfaces, and local topological collapse. This is because each current scaffold segment only specifies a coarse local topology, while

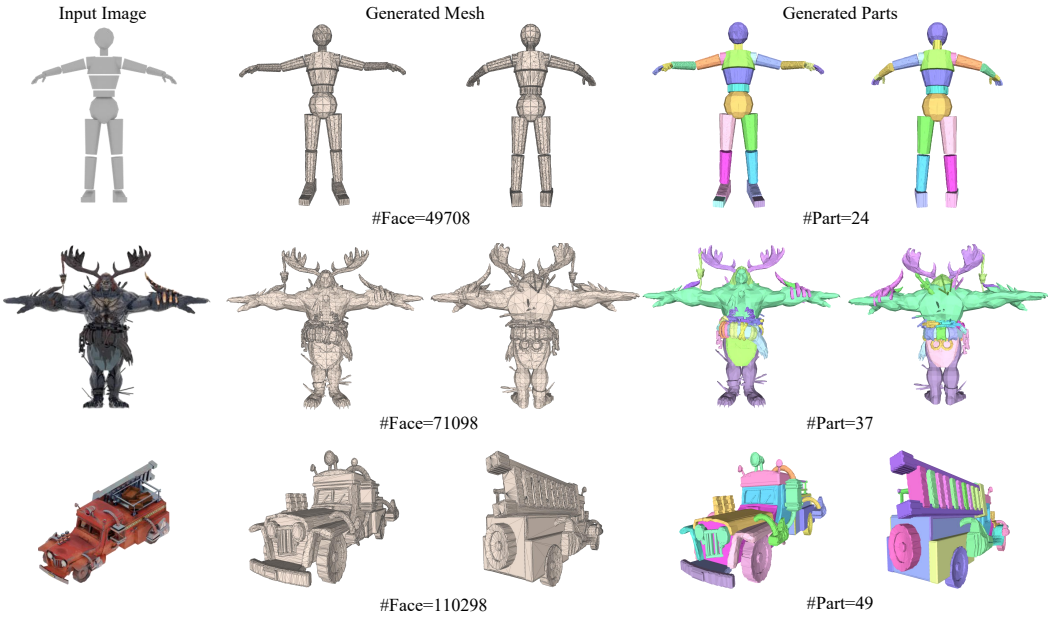


Figure 9: **Additional qualitative results.** For each example, we show the input image, two views of the generated mesh, and two views of the generated parts. FAMesh preserves coherent global structure, fine geometric details, and part-level organization across diverse high-face-count objects; the generated face and part counts are reported below each sample.

high-resolution part generation must also remain compatible with future parts in terms of boundary placement, orientation, symmetry, and cross-part connectivity. Removing later TopoScaf segments therefore weakens the global compatibility constraint and causes the autoregressive trajectory to drift. This provides qualitative evidence that TopoScaf is not merely used for per-part coarse-to-fine refinement, but serves as a part-aligned global prior that provides past, current, and future-part compatibility cues. Moreover, locally perturbing TopoScaf induces a corresponding structural change in the generated mesh, indicating that ScafPAR follows the scaffold guidance rather than treating it as a weak auxiliary condition.

More FAMesh Results. To further complement the qualitative comparisons in the main paper, we provide additional results on diverse and structurally complex objects in Fig. 9. Each example includes the input image, two rendered views of the generated mesh, and the corresponding part-level visualization. These examples cover articulated characters and vehicles with substantially different topology, geometry, and part complexity. FAMesh consistently preserves the global object layout while recovering fine-grained structures such as limbs, antlers, accessories, wheels, and vehicle components. The part-colored renderings further show that the generated meshes maintain coherent part organization and cross-part connectivity, demonstrating the scalability of FAMesh to high-face-count generation under image conditioning.

C LIMITATIONS

Although FAMesh substantially improves topological consistency for high-face-count mesh generation, it remains subject to the inherent limitations of sequence-based autoregressive modeling. In particular, the quadratic complexity of long-sequence attention prevents the generated meshes from fully matching the extremely high face counts of the ground-truth meshes under practical computational budgets. As a result, some fine-scale geometric details may still be under-resolved. In addition, our implementation adopts an existing mesh tokenization scheme that serializes mesh geometry and connectivity into discrete topological sequences. Such tokenization is effective for autoregressive mesh modeling, but it does not explicitly optimize the regularity of the output triangulation, such as edge-length uniformity, triangle aspect ratio, or remeshing quality. Consequently, the raw generated meshes may contain non-uniform face distributions or elongated triangles in local regions. Nevertheless, FAMesh effectively mitigates prefix-induced error accumulation by conditioning part-wise

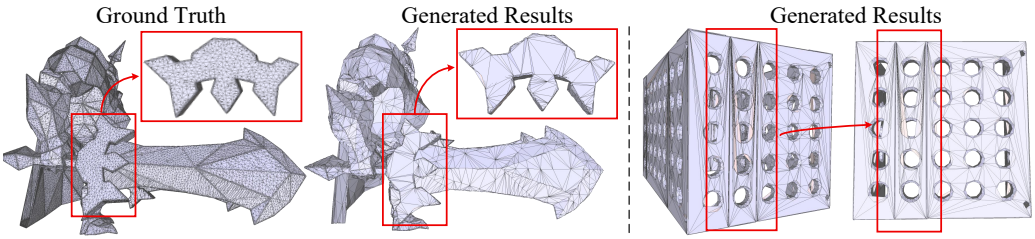


Figure 10: **Limitations of raw mesh triangulation.** Left: compared with the dense ground truth, generated meshes may under-resolve fine local structures and exhibit non-uniform face distributions under practical autoregressive sequence budgets. Right: reference-free examples further reveal intrinsic triangulation artifacts, such as elongated triangles and irregular tessellation around repeated holes. Red boxes highlight representative regions.

autoregressive generation on a part-aligned global topological scaffold. This design enables the generated token sequence to follow a globally coherent topological organization, including part ordering, local connectivity, and cross-part structural compatibility. Within feasible computational budgets, FAMesh therefore supports larger-scale and more topology-consistent mesh generation than prefix-driven autoregressive alternatives, while mesh regularity remains a complementary direction for future improvement.