MESHOCTAVE: VERTEX SPLIT-AND-REWIRE CASCADES FOR NATIVE MESH GENERATION

Anonymous authors
Paper under double-blind review

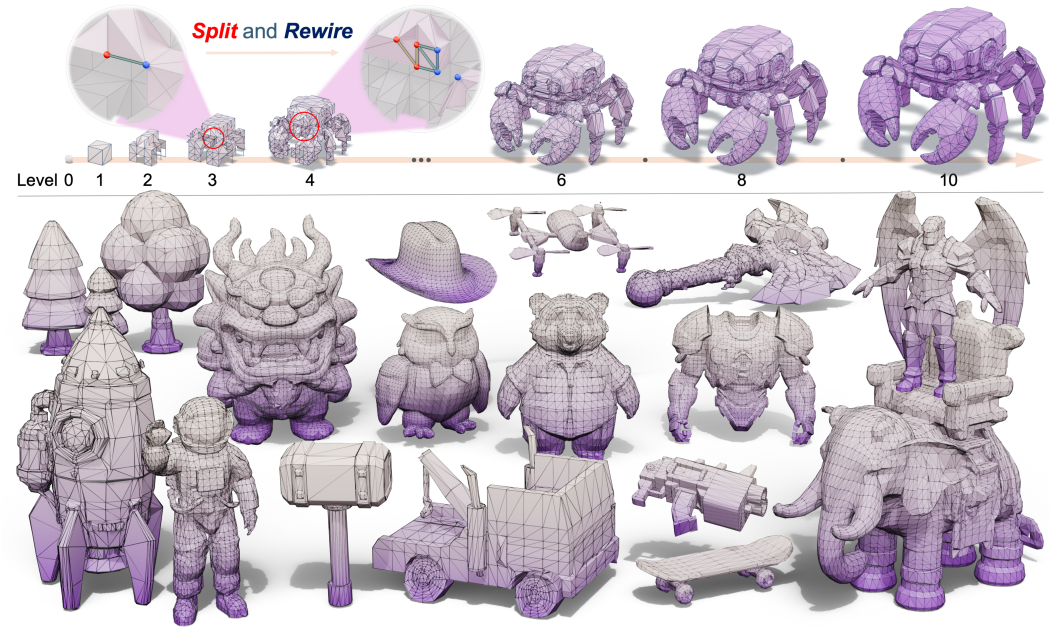

Figure 1: **MeshOctave generates meshes via cascading resolution transitions.** Top: Starting from a single voxel, each step doubles vertex resolution by predicting child vertex instantiation (**split**) and reconnection (**rewire**) for all parent faces in parallel. The number of steps scales logarithmically with resolution, independent of face count. Bottom: Meshes generated by MeshOctave.

ABSTRACT

Generating compact, artist-style meshes with explicit topology typically relies on autoregressive models which incur prohibitive sequential per-token costs, or continuous flow models that depend on heuristic connectivity decoders. Next-scale generation paradigms offer a compelling alternative by enabling parallel intra-scale token prediction and coarse-to-fine refinement from global structure to local topology; yet, existing methods derive hierarchical scales via progressive mesh simplification and invert them sequentially. This eliminates intra-scale parallelism and scales generation steps linearly with face count. In this paper, we propose **MeshOctave**, which instead defines scale through dyadic spatial grid resolutions, framing coarsening as a deterministic collapse that merges vertices sharing a voxel cell and inherits connectivity. Its inverse operation, **split-and-rewire**, determines which octant sub-vertices are instantiated for each coarse face and resolves local connectivity using discrete structural tokens. These per-face operations require no serialization, each scale transition is modeled as an unordered set that adds one bit of coordinate precision, naturally supporting dynamic-length meshes and adaptive resolution refinement. We construct a scale-conditioned masked-uniform discrete diffusion model to learn split-and-rewire operation from resolution collapse hierarchies. MeshOctave outperforms strong baselines in geometric fidelity and topological validity by a non-trivial margin, while supporting adaptive resolution refinement and extending naturally to mesh subdivision tasks.

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

Meshes are the foundational geometric representation for computer graphics, animation, and simulation, where production assets require purposeful vertex layouts and well-structured edge connectivity. Recent 3D foundation models based on implicit fields (Zhang et al., 2023; 2024; Zhao et al., 2023; Xiang et al., 2025; 2026; Zhao et al., 2025b; Wu et al., 2025; Li et al., 2025a; Lai et al., 2026; Li et al., 2025b) achieve impressive shape fidelity, they decouple surface geometry from topology. Extracting explicit surfaces requires heuristic isosurface extraction (Lorensen & Cline, 1998), yielding over-tessellated meshes that necessitate costly manual retopology before downstream deployment. This disconnect has driven the shift toward native mesh generation with explicit topology. Existing native generators broadly fall into two paradigms. Autoregressive (AR) models serialize meshes into sequential vertex and face tokens (Nash et al., 2020; Siddiqui et al., 2024; Chen et al., 2025a;b; Hao et al., 2024; Weng et al., 2025b; Zhao et al., 2025a; Lin et al., 2026). While preserving discrete connectivity, their inference latency scales linearly with face count, compounding error propagation and imposing artificial ordering heuristics onto permutation-invariant graphs. Conversely, continuous diffusion and flow-matching models denoise mesh elements in parallel. They bypass discrete topology by converting topology to continuous surface fields (Zhao et al., 2026; Song et al., 2026a) or per-vertex embedding with spacetime supervision (Li et al., 2026; Wang et al., 2026) or relaxed half-edges (Shen et al., 2024; Xu et al., 2026). However, relying on post-hoc thresholding of continuous latents makes them fragile: minor predictive deviations precipitate missing or spurious faces. Furthermore, connectivity is resolved only at the terminal vertex set, even in coarse-to-fine architectures (Wang et al., 2026). While discrete triangle-soup diffusion (Alliegro et al., 2023; Song et al., 2025) avoids continuous relaxations, it reduces topology to mere coordinate coincidence. Crucially, all existing approaches resolve connectivity at a single monolithic scale: forcing models to predict macro-level geometry and micro-level topology simultaneously, inherently impairs their ability to balance global structural coherence with local geometric fidelity.

Next-scale prediction (Tian et al., 2024) establishes a coarse-to-fine hierarchy that enables parallel intra-scale generation conditioned on coarse structural priors. While successfully extended to voxels (Chen et al., 2025c; Medi et al., 2025), octrees (Wei et al., 2025; Feng et al., 2026), point clouds (Meng et al., 2026), and neural subdivision surfaces (Guo et al., 2026), adapting this paradigm to topology-preserving meshes remains an open challenge. It requires a coarsening operator that satisfies three key properties: (i) global parallelism, operating uniformly across the entire mesh so that scale transition is a single parallel pass; (ii) canonicity, the operation is an intrinsic, deterministic function of geometry rather than heuristic scheduling; and (iii) order-agnostic decodability, factorizes inverse refinement into locally independent decisions for concurrent decoding. Recent coarse-to-fine models instead derive hierarchies from classical mesh simplification: VertexRegen (Zhang et al., 2025) inverts QEM-guided progressive edge collapses (Garland & Heckbert, 1997; Hoppe, 1999), while ARMesh (Lei et al., 2025) sequentially expands simplicial complexes from a base point. Both define scale as a sequence of individual local operations, violating all three criteria: the hierarchy reflects cost-metric tie-breaking rather than intrinsic shape, and refinements depend strictly on sequential decisions reinstating autoregressive latency,

To establish a canonical, parallelizable hierarchy for artist meshes, we define scale directly by spatial resolution. Halving the quantization grid resolution merges vertices falling within the same voxel, a grid-aligned form of vertex clustering (Rossignac & Borrel, 1993; Luebke & Erikson, 1997) we term resolution collapse, with each merged vertex inheriting the combined connectivity of its constituents. Its inverse, **split-and-rewire**, factorizes into per-face decisions that determine: (i) which of the eight octant children of each vertex are instantiated at the finer resolution, and (ii) how these children connect internally and to the children of the face's other two vertices. Each transition adds one bit of coordinate precision per axis while simultaneously splitting merged vertices and rewiring their edges, reaching an L^3 grid from a single root voxel in at most $\log_2 \mathrm{L}$ levels. Leveraging this hierarchy, we introduce **MeshOctave**, a discrete diffusion model that generates meshes scale by scale (Fig. 1). For each coarse face, the split-and-rewire decisions are parameterized as a nine-element discrete structural tokens: three 8×1 binary occupancy vectors (one per parent vertex), three 8×8 intra-parent adjacency matrices, and three 8×8 inter-parent adjacency matrices. Because tokens are spawned directly from the faces generated at previous level, the sequence length dynamically adapts to geometric complexity across shapes and scales. Following TSSR (Song et al., 2025), we corrupt these tokens with combined mask-based and uniform discrete noise (Austin et al., 2021; Sahoo et al., 2024; Schiff et al., 2025) and denoise them using an hourglass transformer. Vertex

occupancy tokens are decoded via categorical classification heads, whereas the six adjacency matrices are predicted row by row using a lightweight autoregressive head. Given a coarse mesh, the structural tokens for decoding next-scale mesh is then generated from a fully masked sequence, progressively refined through our propose-and-refine sampling loop until all structural tokens are confidently resolved. Extensive experiments demonstrate that MeshOctave outperforms strong autoregressive and flow-matching baselines by a substantial margin, demonstrating the feasibility and advantages of next-scale generation paradigm for native meshes. Our primary contributions are then mainly: (1) **the formulation of split-and-rewire cascades** for canonical, globally parallel, and order-agnostic multi-scale mesh formulation; (2) **a scale adaptive discrete diffusion model** that supports next-scale native mesh generation and adaptive mesh resolution refinement.

2 Related Work

Autoregressive Mesh Generation. Autoregressive approaches formulate mesh synthesis as causal sequence modeling over quantized geometric and topological primitives. PolyGen (Nash et al., 2020) and MeshGPT (Siddiqui et al., 2024) established representative vertex-, face-, and latent-token formulations, followed by improvements in model capacity, conditioning (Chen et al., 2024; 2025a; Weng et al., 2025a). To avoid repeatedly encoding shared vertices, recent methods exploit vertex reuse (Chen et al., 2025b), connectivity-aware traversal (Tang et al., 2025; Rossignac, 1999), block and patch aggregation (Weng et al., 2025b), hierarchical BFS encoding (Song et al., 2026b), or tree-structured sequencing (Lionar et al., 2025). Other directions reduce decoding costs through hierarchical architectures (Hao et al., 2024), infer faces after vertex generation (Kim et al., 2026), apply preference or reinforcement-based post-training (Zhao et al., 2025a; Liu et al., 2025), or extend autoregressive modeling to quadrilateral meshes (Liu et al., 2026). MeshRipple (Lin et al., 2026) further introduces frontier-aware BFS tokenization, expansive prediction, and NSCA for structured artist-mesh generation. These methods directly produce meshes with explicit connectivity, but retain sequential dependencies at the token, vertex, or local-update level, limiting generation parallelism as mesh complexity increases.

Table 1: **Comparison with prior works.** MeshOctave jointly refines vertices and topology via multi-scale raw-space masked diffusion. **Notation:** Rep. (R): raw; (L): latent; Method (A): autoregressive; (D): diffusion/flow; (M): masked diffusion; Arch. (N): next scale; (W): whole mesh.

Method	Orderless	Vertex	Topology
MeshGPT	×	L + A W	
BPT	×	R + A W	
VertexRegen	×	R + A N	
FastMesh	×	R + A W	–
ARMesh	×	R + A N	
TSSR	×	R + M W	
MeshFlow	✓	L + D W	
LATO	✓	L + D W	
Nexus	✓	R + D N	L + D W
LATO.2	✓	L + D W	L + D W
Meshy T2	✓	L + D W	
MeshOctave	✓	R + M N	

Diffusion-Based Mesh Generation. Diffusion and flow-matching models generate mesh representations through iterative updates that operate on multiple elements in parallel. PolyDiff (Alliegro et al., 2023) applies discrete diffusion to quantized triangle soups, while TSSR (Song et al., 2025) combines masked topology sculpting with uniform-noise shape refinement over mesh token sequences. Continuous approaches learn latent representations of geometry and connectivity: LATO (Zhao et al., 2026) uses structured voxel latents, MeshFlow (Li et al., 2026) combines a mesh VAE with flow matching, and Meshy T2 (Xu et al., 2026) jointly generates geometry and connectivity through vertex-set latents. Geometry and topology can also be modeled in separate stages, as in LATO.2 (Long et al., 2026), which generates vertices before sampling their conditional connectivity. These methods avoid token-by-token decoding, but typically recover target-scale connectivity without conditioning it on an explicit coarser mesh.

Next-Scale Mesh Generation. VAR (Tian et al., 2024) replaces next-token image prediction with next-scale prediction, generating all tokens at a scale conditioned on coarser scales. Related 3D approaches explore multiscale latent representations and octree hierarchies, as in SAR3D (Chen et al., 2025c) and OctGPT (Wei et al., 2025). For explicit meshes, progressive generation provides control

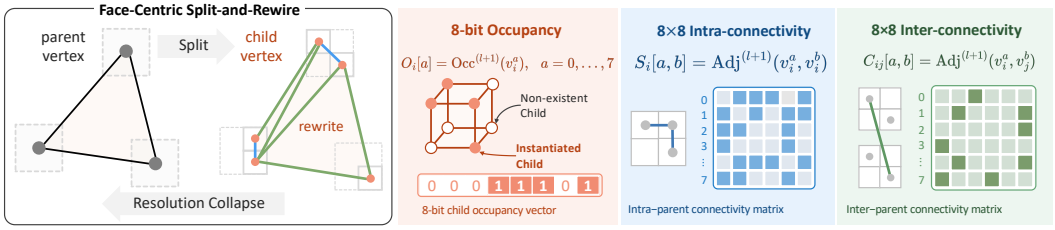

Figure 2: **Face-centric split-and-rewire.** Resolution collapse merges vertices fall in the same grid cell; its inverse, split-and-rewire specifies which of the eight child voxels are occupied and how the instantiated children are connected. For each parent face, 3 8-bit occupancy vectors encode vertex splitting, and 3 intra-parent and 3 inter-parent 8×8 connectivity matrices encode rewiring.

over the level of detail: VertexRegen (Zhang et al., 2025) learns to reverse edge collapses through vertex splits, while ARMesh (Lei et al., 2025) grows simplicial complexes through local remeshing. Both increase mesh detail through sequences of local operations. Nexus (Wang et al., 2026) instead generates vertices through a spatial octree hierarchy and subsequently models topology conditioned on the generated vertices. Our method defines mesh scales by dyadic spatial resolution and uses masked diffusion to predict split-and-rewire configurations in parallel within each scale.

3 Vertex Split-and-Rewire Cascades

Generating a whole scale in parallel requires a global coarsening operator whose inverse decomposes into local, order-free decisions. We derive from nested quantization grids.

3.1 Resolution Collapse

Let a mesh at hierarchy level l be defined as $\mathcal{M}^l = (\mathbf{V}^l, \mathbf{F}^l)$, where $\mathbf{V}^l \subset \mathbb{R}^3$ represents the set of vertices and $\mathbf{F}^l$ denotes the set of triangular faces. $\mathcal{E}(\mathcal{M}^l)$ denotes the induced edge set. Each face is defined as a triplet of vertices, $\mathbf{f}^l = \langle \mathbf{v}_0^l, \mathbf{v}_1^l, \mathbf{v}_2^l \rangle \in \mathbf{V}^l$. Vertex coordinates are quantized to a uniform grid of resolution 2^l; thus, $l = 10$ corresponds to a 1024^3 grid. The resolution collapse operator $\mathbb{C}$ downsamples $\mathcal{M}^l$ to $\mathcal{M}^{l-1}$ by halving the grid resolution. During this process, each coarse voxel at level $l-1$ subsumes an octant of 2^3 fine voxels at level l, denoted as mapping $\pi : \mathbf{V}^l \to \mathbf{V}^{l-1}$ that maps each fine vertex to its corresponding coarse level vertex. $\mathbf{F}^{l-1}$ are obtained by directly projecting each fine face via π, removing duplicates while preserving degenerated faces:

$$\mathbf{V}^{l-1} \leftarrow \pi(\mathbf{V}^l), \quad \mathbf{F}^{l-1} \leftarrow \left\{ \left\langle \pi(\mathbf{v}_0^l), \pi(\mathbf{v}_1^l), \pi(\mathbf{v}_2^l) \right\rangle \;\middle|\; \langle \mathbf{v}_0^l, \mathbf{v}_1^l, \mathbf{v}_2^l \rangle \in \mathbf{F}^l \right\}. \tag{1}$$

We use operator $\mathbb{C}$ to denote Eqn. 1, which operates independently and uniformly across all grid voxels in a single pass, the resulting multiscale hierarchy,

$$\mathcal{M}^{\mathrm{L}} \xrightarrow{\mathbb{C}} \mathcal{M}^{\mathrm{L}-1} \xrightarrow{\mathbb{C}} \cdots \xrightarrow{\mathbb{C}} \mathcal{M}^0, \tag{2}$$

is canonical and determined by the initial mesh and the spatial grid.

3.2 Face-centric Split-and-Rewire Cascades

Next-scale mesh generation then is the inversion process of resolution collapse, hence it is essential to define the inverse of $\mathbb{C}$. This inverse process $\mathbb{S}$ doubles the spatial resolution and recover the information discarded: which children of each parent vertex exist, and how they connect.

Vertex Split. With grid resolution doubled, a given parent vertex $\mathbf{v}^{l-1}$ is divided into 2^3 sub-voxels (potentially contain child-vertex) in level l, yielding an octant $\mathcal{O}(\mathbf{v}^{l-1})$. We directly encode the instantiation of child-vertex into an 8-bit binary vector $\mathbf{O}(\mathbf{v}^{l-1}) \in \{0,1\}^8$ with $\mathbf{O}(\mathbf{v}^{l-1}) = \mathbf{1}[\mathcal{O}(\mathbf{v}^{l-1}) \in \mathbf{V}^l]$, i.e., depending on whether the sub-voxels presents as vertices in $\mathbf{V}^l$. Thus, a single vector specifies the complete subdivision pattern for a given parent vertex.

Rewire. We must also recover the connectivity of the split vertices. Since each parent vertex yields up to eight children, the connectivity between the children of any two parent vertices can be modeled

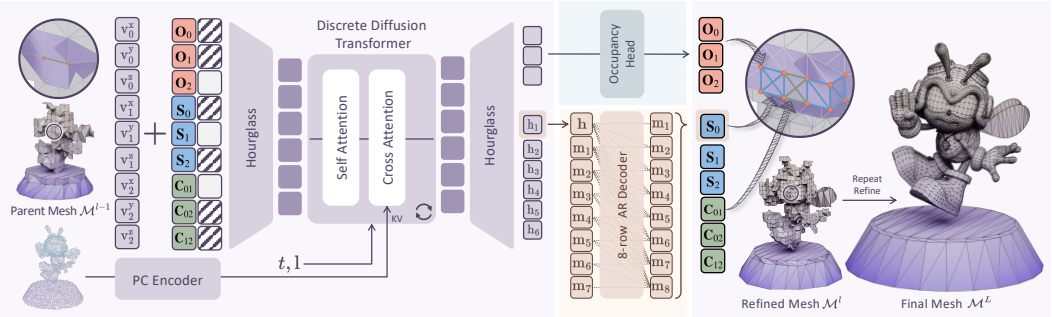

Figure 3: **Scale-conditioned discrete diffusion model.** The generator refines a mesh from scale l to $l+1$, the split-and-rewire structural tokens $\mathbf{z}$ is concatenated with vertex coordinates. The elements of $\mathbf{z}$ are masked according to diffusion timestep t. An hourglass Transformer processes these tokens, modulated by timestep t and scale l via adaLN. Finally, categorical heads predict child-vertex occupancy while an autoregressive MLP head decodes row-factorized adjacency matrices, assembling the refined mesh at scale $l+1$.

using an 8×8 binary matrix $\mathbf{M}$. For a given parent face $\mathbf{f}^{l-1} = \langle \mathbf{v}_0^{l-1}, \mathbf{v}_1^{l-1}, \mathbf{v}_2^{l-1} \rangle$, we define the connectivity matrix $\mathbf{M}_{ij}$ between the children of parent vertices $\mathbf{v}_i^{l-1}$ and $\mathbf{v}_j^{l-1}$ as:

$$\mathbf{M}_{ij}(a,b) = \mathbf{1}\left[\left\langle \mathcal{O}_a(\mathbf{v}_i^{l-1}), \mathcal{O}_b(\mathbf{v}_j^{l-1}) \right\rangle \in \mathcal{E}(\mathcal{M}^l)\right], \; 0 \le i \le j \le 2, \tag{3}$$

where $a, b \in \{0, \ldots, 7\}$ index the sub-voxel (child vertex).When $i = j$, the matrix records connections among the children of the same parent vertex, which we denote as the intra-parent connectivity matrix $\mathbf{S}_i$. When $i < j$, the matrix records connections between the children of two distinct parent vertices, denoted as the inter-parent connectivity matrix $\mathbf{C}_{ij}$. Consequently, a parent face contains three intra-parent and three inter-parent connectivity matrices.

Together, the inversion of collapsing any parent face $\mathbf{f}^{l-1}$ is fully represented by nine-element structural token $\mathbf{z}$ (three vertex occupancy vectors and six connectivity matrices).

$$\mathcal{M}^{l-1} \xrightarrow{\mathbb{S}(\mathbf{z})} \mathcal{M}^{l}; \quad \mathbf{z}_{\mathbf{f}^{l-1}} = \big(\mathbf{O}_0, \mathbf{O}_1, \mathbf{O}_2; \, \mathbf{S}_0, \mathbf{S}_1, \mathbf{S}_2; \, \mathbf{C}_{01}, \mathbf{C}_{02}, \mathbf{C}_{12}\big). \tag{4}$$

During each resolution collapse, we record $\mathbf{z}$ for every face. And fine mesh $\mathcal{M}^{\mathrm{L}}$ can by fully recovered from a single voxel at level 0 through face-wise split-and-rewire operation $\mathbb{S}$.

4 Scale Adaptive Discrete Diffusion

We adopt the split-and-rewire cascade (Eqn. 4) for next-scale mesh generation: given a coarse mesh $\mathcal{M}^{l-1}$, the generator predicts a structural token $\mathbf{z}$ for each face $\mathbf{f}^{l-1}$. Since $\mathbf{z}$ contains only binary vectors and matrices, we formulate its prediction as a discrete diffusion process, in which the structural token of each face is progressively revealed from masked tokens.

To input the coarse model $\mathcal{M}^{l-1}$, we fuse $\mathbf{z}$ with the vertex coordinates of its corresponding face in $\mathcal{M}^{l-1}$. As both contain nine elements per face, they are added directly after embedding: coordinate slot k of face f add with structural token slot $z_{f,k}$: $\mathbf{x}_{f,k} = \mathrm{Embed}(z_{f,k}) + \mathrm{Embed}(\mathbf{f}[k])$, we omit $l-1$ for clearity. where $\mathrm{Embed}(\cdot)$ is embedding, and $\mathbf{x}_{f,k}$ is the resulting slot-wise input to the network. An hourglass encoder Hao et al. (2024) then compresses $\mathbf{x}_{f,k}$ into vertex-level and subsequently face-level tokens. A transformer backbone processes the face-level tokens, and an hourglass decoder expands its output back to nine tokens per face. Finally, an occupancy decoder $\mathcal{D}_o$ and a connectivity decoder $\mathcal{D}_c$ map the hidden states to $\mathbf{z}$. $\mathcal{D}_o$ is a 256-way classifier whose output index maps to the 8×1 binary occupancy $\mathbf{O}$. $\mathcal{D}_c$ is a small MLP autoregressively predicts the eight rows $\mathbf{m}_r, r = 1...8$ of the connectivity matrix $\mathbf{M}$ (Eq. 3), where each step conditions on output hidden $\mathbf{h}_m$ corresponding to $\mathbf{S}$ and $\mathbf{C}$, and the previously decoded rows:

$$\mathcal{D}_c : \; p_c(\mathbf{M} \mid \mathbf{h}_m) = \prod_{r=1}^{8} p_c(\mathbf{m}_r \mid \mathbf{m}_{<r}, \mathbf{h}_m). \tag{5}$$

In the diffusion process, each connectivity token $\mathbf{M}$ is nonetheless treated as a single slot, with all eight rows masked or revealed together.

4.1 MASKED-UNIFORM DISCRETE DIFFUSION.

We adopt a discrete-state diffusion framework (Austin et al., 2021), in which each slot in $\mathbf{z}$ is corrupted independently with probability $r(t) = \cos(\pi t/2)$, $t \sim \mathcal{U}(0,1)$, so that $t = 0$ corresponds to full corruption and $t = 1$ to clean tokens. We adopt the standard $\mathbf{z}_0$-parameterization and train the network to recover the clean tokens directly from the corrupted sequence $\tilde{\mathbf{z}}$:

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\,l,\,\tilde{\mathbf{z}}}\Big[\sum_{(f,k)\in\Omega} -\log p_\theta\big(z^\star_{f,k} \mid \tilde{\mathbf{z}}, \mathbf{F}^{l-1}, t, l, \mathbf{c}\big)\Big], \tag{6}$$

where $z^\star_{f,k}$ denotes the ground-truth value of the k-th slot of $\mathbf{z}$ corresponding to face f, and p_θ is parameterized per slot by either the categorical occupancy head or the autoregressive connectivity head (Eq. 5). Following standard practice in mesh generation (Chen et al., 2025a; Wang et al., 2026), we condition the generator on point-cloud $\mathbf{c}$, and Ω denotes the set of slots on which the loss is computed.

Mask-Uniform Noise Strategy. In standard masked diffusion, unmasked tokens are permanently fixed, incurring early prediction errors to irreversibly propagate into the final mesh. Inspired by TSSR (Song et al., 2025), we adopt a mask-uniform noising strategy: a mask pathway that predicts structure from partial tokens, and a uniform pathway that refines the fully populated sequence. Both share a single backbone and are conditioned via a mode flag [MODE] appended to the point-cloud feature $\mathbf{c}$.

The mask pathway replaces corrupted slots $f, k \in \mathcal{B}$ with [MASK] and is supervised only on those slots, then we set $\Omega = \mathcal{B}$ and compute the loss $\mathcal{L}_{\text{mask}}$ from Eq. 6. The uniform pathway instead replaces corrupted slots with values drawn uniformly from the slot's state space and is supervised on all slots, i.e., $\Omega = \{1, \ldots, 9n\}$, and we compute $\mathcal{L}_{\text{unif}}$ from Eq. 6. Since the corrupted positions are not indicated, the model must learn to detect and correct any errors. Both losses use a multi-class focal loss (Lin et al., 2017) instead of cross-entropy to mitigate class imbalance by down-weighting well-classified slots, details see A.4. The uniform pathway additionally trains a confidence head $\mathcal{C}_\phi$ to predict whether its own prediction at each slot is correct, with label $y_{f,k} = \mathbf{1}\big[\arg\max p_\theta(\cdot \mid \mathbf{h}_{f,k}) = z^\star_{f,k}\big]$. The head operates on $\mathrm{sg}[\mathbf{h}_{f,k}]$, so that $\mathcal{L}_{\text{conf}}$ does not update the backbone:

$$\mathcal{L}_{\text{conf}} = \frac{1}{9n}\sum_{f,k} \kappa_{f,k}\,\mathrm{BCE}\big(\mathcal{C}_\phi(\mathrm{sg}[\mathbf{h}_{f,k}]),\, y_{f,k}\big), \tag{7}$$

where n is the number of faces, and $\kappa_{f,k} = \kappa > 1$ for the minority incorrect class ($y_{f,k} = 0$) and 1 otherwise. To expose the uniform pathway to realistic errors, a fraction ρ of its training steps replaces uniform noise with the model's own greedy predictions from a gradient-free mask-pathway forward pass. At each step, we draw $s \sim \mathrm{Bernoulli}(p)$ with $p = 0.5$ and minimize

$$\mathcal{L}_{\text{total}} = s\,\mathcal{L}_{\text{mask}} + (1-s)\big(\mathcal{L}_{\text{unif}} + \lambda\,\mathcal{L}_{\text{conf}}\big), \tag{8}$$

where λ balances the confidence loss.

Vertex-anchored 3D RoPE. Unlike TSSR's 1D indexing, we represent meshes via orderless per-face token blocks to preserve block permutation equivariance. We inject structure via 64d RoPE, allocating 30 frequency pairs to 3D voxel coordinates and 2 to intra-face roles τ (occupancy, intra-/inter-parent connectivity). Tokens anchor to vertex pairs (v_a, v_b) or (v_a, v_a) for vertices and connectivity resprectively (Table 4). Sharing rotation angles across half their geometric frequencies makes vertex-edge incidence an explicit phase alignment. Finally, lexicographical (z-y-x) parent sorting canonicalizes shared edges, and reduced coarse tokens remain anchored to all three parent vertices.

Inference. During inference, the two pathways alternate within an iterative propose-and-correct loop. Initialized from a fully masked sequence, the mask pathway first generates candidate structures for all uncommitted regions of the mesh. The uniform pathway then evaluates the entire sequence, simultaneously refining token representations and assigning confidence scores via $\mathcal{C}_\phi$. Rather than permanently freezing unmasked tokens with a hard threshold, slots are stochastically re-masked ([MASK]) based on their confidence scores and an annealed decoding schedule. Tokens with lower

Table 2: **Quantitative comparison.** *Type* denotes the generative paradigm: autoregressive generation (AR), continuous flow matching (Flow), or discrete diffusion (DD). **N** marks methods that generate meshes over a coarse-to-fine resolution hierarchy and expose intermediate scales. Best results are in **bold**, and second-best results are underlined.

Metric	Mesh Any.V2	Vertex Regen	AR Mesh	Fast Mesh	Mesh Silk.	BPT	Deep Mesh	Mesh Ripple	Mesh Flow	LATO	LATO.2	MESH OCTAVE
Type	AR	AR+**N**	AR+**N**	AR	AR	AR	AR	AR	Flow	Flow	Flow	DD+**N**
CD-L2 ↓	0.1103	0.0874	0.0804	0.0647	0.0622	0.0610	0.0505	0.0468	0.0452	0.0430	<u>0.0406</u>	**0.0392**
CD-L1 ↓	0.1539	0.1238	0.1146	0.0925	0.0895	0.0873	0.0729	0.0691	0.0672	0.0621	<u>0.0595</u>	**0.0581**
HD ↓	0.2317	0.1875	0.1569	0.1151	0.1440	0.1111	0.0957	0.0948	0.0765	0.0741	<u>0.0665</u>	**0.0645**
|NC| ↑	0.6807	0.6655	0.6058	0.7413	0.7728	0.7976	0.8187	0.8141	0.8237	0.8273	<u>0.8333</u>	**0.8478**

confidence are re-masked with higher probability, enabling the network to re-evaluate early errors while progressively committing to stable structures in later iterations. This cycle of proposal and refinement continues until all sampling steps are completed.

4.2 Implementation.

We instantiate our discrete diffusion backbone as a symmetric hourglass Transformer with $9/3/1$ tokens per face, totaling 24 blocks across stage depths $[2, 4, 6, 6, 4, 2]$, width 1024, 16 heads, SwiGLU FFNs of width 2816, and RMSNorm, for 624M parameters. We adopt vertex-anchored 3D RoPE and variable-length FlashAttention-2 throughout. Conditioning combines adaLN-Zero modulation, the sum of timestep t and scale level l embeddings, with cross-attention over point-cloud features $\mathbf{c}$. Condition $\mathbf{c}$ is derived via an 8-layer Michelangelo-style encoder of width 768 with 12 heads and 1024 latents, totaling 193M parameters and fine-tuned jointly with the backbone. Inputs consist of 40960 point-normal pairs subsampled from 50k surface samples, augmented with random scaling and Gaussian jitter of $\sigma = 0.01$ applied with probability 0.5. We train on 350K meshes filtered from the Objaverse and Toys4K datasets, up to 15k faces and 135k tokens each at the finest 9-token/face resolution, for 7 days on $8\times$NVIDIA H800 80GB GPUs using bfloat16 mixed precision, and gradient checkpointing.

5 Experiments

Evaluation Protocol. To ensure a fair comparison of MeshOctave with existing methods without bias, we select 300 meshes from ObjaverseXL and Toys4K, generate 200 meshes using large-scale 3D generative models (Xiang et al., 2025; Hunyuan3D et al., 2025; Wu et al., 2025) as our test set. We uniformly sample 4,096 surface points from each generated mesh and its reference mesh. Geometric fidelity is measured using CD-L1, CD-L2, and Hausdorff distance. We additionally report absolute normal consistency to evaluate surface orientation and local structure.

Baselines. We compare with representative mesh generation methods across three paradigms. The autoregressive baselines include MeshAnythingV2 Chen et al. (2025b), MeshSilksong Song et al. (2026b), BPT Weng et al. (2025b), DeepMesh Zhao et al. (2025a), FastMesh Kim et al. (2026), and MeshRipple Lin et al. (2026). The next-scale baselines include VertexRegen Zhang et al. (2025) and ARMesh Lei et al. (2025). The flow-matching baselines include MeshFlow Li et al. (2026), LATO Zhao et al. (2026), and LATO.2 Long et al. (2026).

5.1 Quantitative Analysis.

Quality MeshOctave outperforms all autoregressive, next-scale, and flow-matching baselines across every metric (Table 2). Although flow-matching methods generally surpass autoregressive models by avoiding surface fragmentation and ensuring completeness, MeshOctave outperforms the strongest flow models by a non-trivial margin.

Inference Latency. MeshOctave takes $\sim$8 s per scale (20 diffusion steps) and $\sim$90 s across 10 scales, nearly independent from face count. In contrast, baselines such as ARMesh and VertexRegen

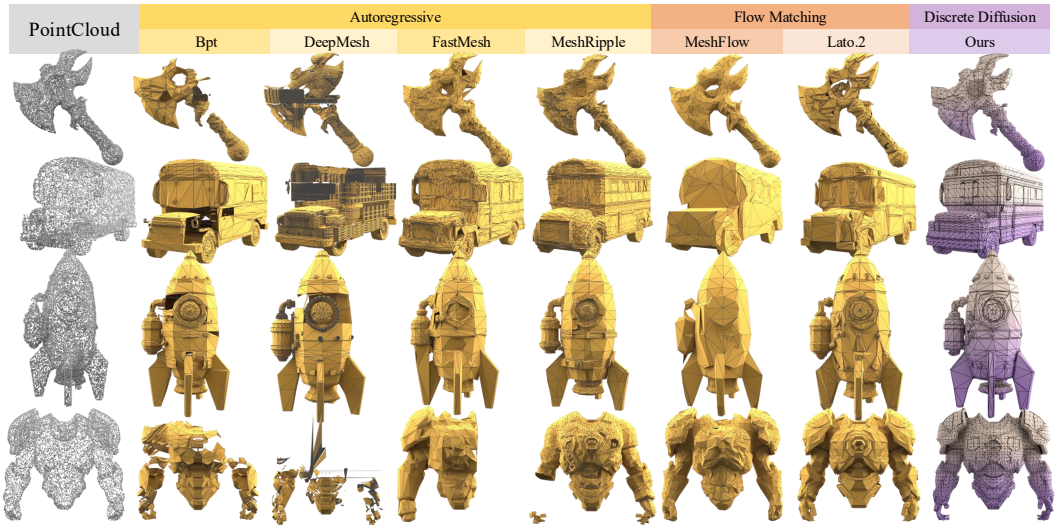

Figure 4: **Qualitative comparison.** Each row shows one input point cloud and the final meshes generated by autoregressive, flow-matching, and diffusion-based methods.

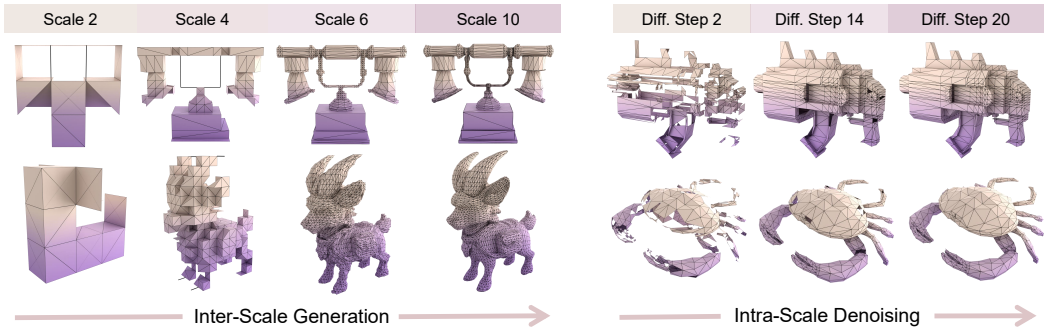

Figure 6: **Generation process visualization.** Meshes are progressively generated across scales, while predictions within a scale are iteratively refined through masked and uniform denoising.

require $\sim$15 minutes for 5k faces, a $\sim 10\times$ speedup that highlights the efficiency of our parallel framework (Appendix A.7).

5.2 Qualitative Analysis.

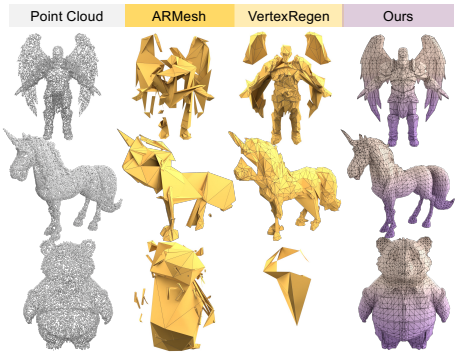

Figure 5: **Comparison with level-of-detail baselines.**

Figure 4 presents qualitative comparisons. While autoregressive baselines frequently show broken surfaces, flow-matching approaches tend to oversmooth fine geometric details. In contrast, MeshOctave faithfully preserves intricate shape features and delicate parts.

Figure 6 visualizes both inter-scale generation and intra-scale denoising process.

Fig. 5 compares MeshOctave with VertexRegen and ARMesh, the coarse-to-fine baselines closest to our paradigm. Lacking public weights, we retrained both from scratch on 200k meshes ($<$ 5k faces) for five days under their official configurations. Relying on sequential, single-element refinements, both baselines suffer from compounding errors that cause severe geometric distortions. Conversely, MeshOctave resolves split-and-rewire decisions for entire resolution levels in parallel, producing substantially more coherent and watertight meshes.

5.3 Ablation Study.

We ablate our core design choices in Table 3 by evaluating four variants: replacing discrete with continuous diffusion (Continuous), omitting the uniform pathway (w/o Uniform), predicting adjacency rows independently without the AR head (w/o AR Head), and substituting 3D RoPE with 1D RoPE (1D RoPE). Continuous diffusion performs worst, as relaxing discrete constraints causes severe topological collapse. Mask-only diffusion lacks a refinement mechanism to correct early commitments. Predicting connectivity rows independently results in spurious edge connections. 1D RoPE discards explicit 3D coordinates, substantially degrading the model's awareness of local geometric proximity.

Table 3: **Ablation study.** One component replaced per row; best in **bold**.

Variant	CD-L2 ↓	HD ↓	|NC| ↑
Full	**0.0430**	**0.0739**	**0.8379**
Continuous	0.0512	0.0983	0.6685
w/o Uniform	0.0453	0.0884	0.7955
w/o AR head	0.0496	0.1005	0.7907
w/o 3D RoPE	0.0469	0.0915	0.7911

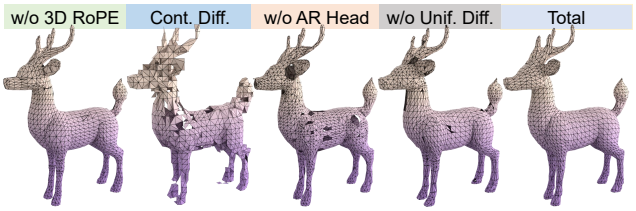

Figure 7: **Ablation, qualitative.** The same shape produced by each variant, labelled above each panel.

5.4 Application.

Mesh Subdivision MeshOctave naturally performs mesh subdivision without retraining by conditioning on points sampled directly from the coarse input (Fig. 8). Unlike baselines (Loop, 1987; Liu et al., 2020; Guo et al., 2026) that enforce uniform, fixed-topology subdivision, MeshOctave adaptively predicts child occupancy and connectivity—modifying topology and concentrating faces on complex regions for superior geometric fidelity.

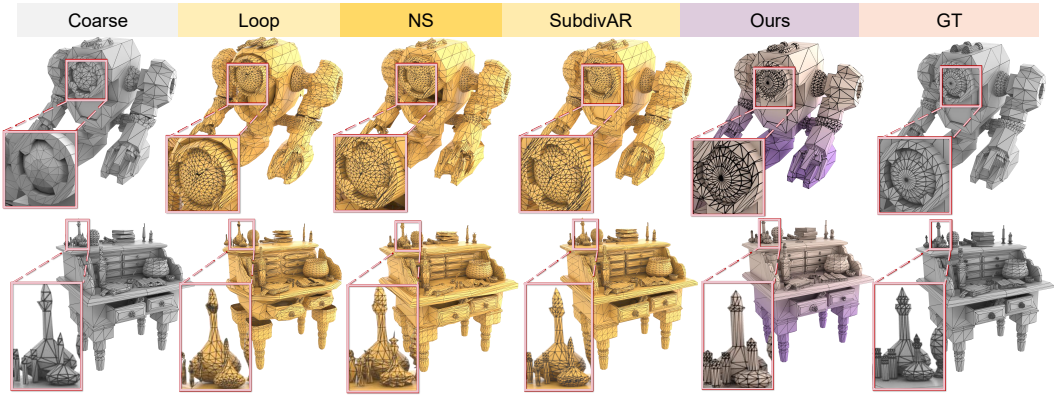

Figure 8: **Application in mesh subdivision.** MeshOctave adaptively refines geometry and connectivity given an existing coarse mesh, producing more detailed and artist-like subdivision results.

6 Conclusion

We present MeshOctave, a next-scale generative framework for native 3D meshes centered on a globally parallel, order-agnostic split-and-rewire cascade. Driven by a mask-uniform discrete diffusion model across resolution scales, MeshOctave predicts intra-scale structural tokens in parallel, bypassing the sequential bottlenecks of prior multiscale methods. While generating sequentially across scales yields higher latency than single-scale flow matching and risks propagating coarse topological errors without backtracking, future work will investigate diffusion distillation for faster inference and bidirectional transitions to dynamically revise coarse topology.

AI USE STATEMENT

In preparing this manuscript, the authors employed ChatGPT exclusively as an editorial aid to polish English expression, correct grammatical errors, and enhance clarity. It was not involved in data collection or processing, conceptual or theoretical development, hypothesis formulation, mathematical reasoning, methodology or experimental design, method implementation, translation, qualitative analysis, or the interpretation of experimental findings. All suggested revisions were carefully examined and, when necessary, modified by the authors to maintain technical accuracy and the original intent. The authors assume full responsibility for the integrity and final content of this manuscript.

REFERENCES

Antonio Alliegro, Yawar Siddiqui, Tatiana Tommasi, and Matthias Nießner. Polydiff: Generating 3d polygonal meshes with diffusion models. *arXiv preprint arXiv:2312.11417*, 2023.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Sijin Chen, Xin Chen, Anqi Pang, Xianfang Zeng, Wei Cheng, Yijun Fu, Fukun Yin, Zhibin Wang, Jingyi Yu, Gang Yu, et al. 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, Xin Chen, 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*, volume 2025, pp. 51369–51389, 2025a.

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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13922–13931. IEEE, 2025b.

Yongwei Chen, Yushi Lan, Shangchen Zhou, Tengfei Wang, and Xingang Pan. Sar3d: Autoregressive 3d object generation and understanding via multi-scale 3d vqvae. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28371–28382. IEEE, 2025c.

Qi-Yuan Feng, Zi-Xin Zou, HaoXiang Chen, Yuanchen Guo, Yan-Pei Cao, Tai-Jiang Mu, and Shimin Hu. Grow3d: Hierarchical next-scale octree prediction for fast and high-fidelity 3d shape generation. *ACM Transactions on Graphics*, 2026.

Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 209–216, 1997.

Huipeng Guo, Zikai Song, Hang Long, Jielei Zhang, Wenbing Li, Junkai Lin, Tianhao Zhao, Jinshen Zhang, Tianle Guo, and Wei Yang. Subdivar: Autoregressive next-scale prediction for neural mesh subdivision. *arXiv preprint arXiv:2606.27088*, 2026.

Zekun Hao, David W Romero, Tsung-Yi Lin, and Ming-Yu Liu. Meshtron: High-fidelity, artist-like 3d mesh generation at scale. *arXiv preprint arXiv:2412.09548*, 2024.

Hugues Hoppe. New quadric metric for simplifying meshes with appearance attributes. In *Proceedings Visualization'99 (Cat. No. 99CB37067)*, pp. 59–510. IEEE, 1999.

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

Jeonghwan Kim, Yushi Lan, Armando Fortes, Yongwei Chen, and Xingang Pan. Fastmesh: Efficient artistic mesh generation via component decoupling. In *2026 International Conference on 3D Vision (3DV)*, pp. 1478–1488. IEEE, 2026.

Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Qingxiang Lin, Jingwei Huang, Chunchao Guo, and Xiangyu Yue. Lattice: Democratize high-fidelity 3d generation at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19982–19992, 2026.

Jiabao Lei, Kewei Shi, Zhihao Liang, and Kui Jia. Armesh: Autoregressive mesh generation via next-level-of-detail prediction. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 161613–161633. Curran Associates, Inc., 2025. doi: 10.52202/085713-5397. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/ec7d99ca50540a9c6994197f4355776f-Paper-Conference.pdf.

Weiyu Li, Antoine Toisoul, Tom Monnier, Roman Shapovalov, Rakesh Ranjan, Ping Tan, and Andrea Vedaldi. Meshflow: Efficient artistic mesh generation via meshvae and flow-based diffusion transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5849–5858, 2026.

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

Zhihao Li, Yufei Wang, Heliang Zheng, Yihao Luo, and Bihan Wen. Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. *Advances in Neural Information Processing Systems*, 38:118582–118600, 2025b.

Junkai Lin, Hang Long, Huipeng Guo, Jielei Zhang, JiaYi Yang, Tianle Guo, Yang Yang, Jianwen Li, Wenxiao Zhang, Matthias Nießner, et al. Meshripple: Structured autoregressive generation of artist-meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12706–12718, 2026.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Stefan Lionar, Jiabin Liang, and Gim Hee Lee. Treemeshgpt: Artistic mesh generation with autoregressive tree sequencing. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26608–26617. IEEE, 2025.

Hsueh-Ti Derek Liu, Vladimir G Kim, Siddhartha Chaudhuri, Noam Aigerman, and Alec Jacobson. Neural subdivision. *ACM Transactions on Graphics*, 39(4):124:1–124:16, 2020. doi: 10.1145/3386569.3392418.

Jian Liu, Jing Xu, Song Guo, Jing Li, Jingfeng Guo, Jiaao Yu, Haohan Weng, Biwen Lei, Xianghui Yang, Zhuo Chen, Fangqi Zhu, Tao Han, and Chunchao Guo. Mesh-rft: Enhancing mesh generation via fine-grained reinforcement fine-tuning. *Advances in Neural Information Processing Systems*, 38:156033–156058, 2025.

Jian Liu, Chunshi Wang, Song Guo, Haohan Weng, Zhen Zhou, Zhiqi Li, Jiaao Yu, Yiling Zhu, Jing Xu, Biwen Lei, et al. Quadgpt: Native quadrilateral mesh generation with autoregressive models. In *International Conference on Learning Representations*, volume 2026, pp. 22114–22139, 2026.

Hang Long, Tianhao Zhao, Junkai Lin, Youjia Zhang, Huipeng Guo, Rendong Liang, Jiale Xu, Jozef Hladkỳ, Matthias Nießner, Yuanming Hu, et al. Lato. 2: Factorized 3d mesh generation with vertex and topology flow. *arXiv preprint arXiv:2607.10623*, 2026.

Charles Loop. Smooth subdivision surfaces based on triangles. Master's thesis, The University of Utah, 1987.

William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353. ACM, 1998.

David Luebke and Carl Erikson. View-dependent simplification of arbitrary polygonal environments. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 199–208, 1997.

Tejaswini Medi, Arianna Rampini, Pradyumna Reddy, Pradeep Kumar Jayaraman, and Margret Keuper. 3d-wag: hierarchical wavelet-guided autoregressive generation for high-fidelity 3d shapes. In *36th British Machine Vision Conference (BMVC)*. BMVA, 2025.

Ziqiao Meng, Qichao Wang, Zhiyang Dou, Zixing Song, Zhipeng Zhou, Irwin King, and Peilin Zhao. Pointnsp: Autoregressive 3d point cloud generation with next-scale level-of-detail prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31450–31461, 2026.

Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*, pp. 7220–7229. PMLR, 2020.

Jarek Rossignac. Edgebreaker: Connectivity compression for triangle meshes. *IEEE transactions on visualization and computer graphics*, 5(1):47–61, 1999.

Jarek Rossignac and Paul Borrel. Multi-resolution 3d approximations for rendering complex scenes. In *Modeling in computer graphics: methods and applications*, pp. 455–465. Springer, 1993.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Yair Schiff, Subham Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-Torre, Bernardo Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 43776–43821, 2025.

Tianchang Shen, Zhaoshuo Li, Marc Law, Matan Atzmon, Sanja Fidler, James Lucas, Jun Gao, and Nicholas Sharp. Spacemesh: A continuous representation for learning manifold surface meshes. In *SIGGRAPH Asia 2024 conference papers*, pp. 1–11, 2024.

Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19615–19625. IEEE, 2024.

Gaochao Song, Haohan Weng, Luo Zhang, Zibo Zhao, and Shenghua Gao. Mesh bdf: Barycentric dominance field for 3d native mesh generation. *arXiv preprint arXiv:2606.31777*, 2026a.

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 *The Fourteenth International Conference on Learning Representations*, 2026b.

Kaiyu Song, Hanjiang Lai, Yaqing Zhang, Chuangjian Cai, Yan Pan, Kun Yue, and Jian Yin. Topology sculptor, shape refiner: Discrete diffusion model for high-fidelity 3d meshes generation, 2025. URL `https://arxiv.org/abs/2510.21264`.

Jiaxiang Tang, Max 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*, volume 2025, pp. 35913–35934, 2025.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Hanxiao Wang, Ying-Tian Liu, Yuan-Chen Guo, Qi-Yuan Feng, Zi-Xin Zou, Ding Liang, Biao Zhang, and Yan-Pei Cao. Nexus: Native mesh generation with diffusion. *ACM Transactions on Graphics (TOG)*, 45(4):1–14, 2026.

Si-Tong Wei, Rui-Huan Wang, Chuan-Zhi Zhou, Baoquan Chen, and Peng-Shuai Wang. Octgpt: Octree-based multiscale autoregressive models for 3d shape generation. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2025.

Haohan Weng, Yikai Wang, Tong Zhang, C. L. Philip Chen, and Jun Zhu. Pivotmesh: Generic 3d mesh generation via pivot vertices guidance. In *International Conference on Learning Representations*, pp. 46180–46199, 2025a.

Haohan Weng, Zibo Zhao, Biwen Lei, Xianghui Yang, Jian Liu, Zeqiang Lai, Zhuo Chen, Yuhong Liu, Jie Jiang, Chunchao Guo, et al. Scaling mesh generation via compressive tokenization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11093–11103, 2025b.

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. In *Advances in Neural Information Processing Systems*, volume 38, pp. 170778–170804. Curran Associates, Inc., 2025. doi: 10.52202/085713-5688.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480. IEEE, 2025.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, et al. Native and compact structured latents for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14419–14429, 2026.

Jiale Xu, Rendong Liang, Yuhao Long, Siyuan Shen, Zangyueyang Xian, Xiaofei Wu, Zeyi Xu, and Yuanming Hu. Meshy t2: Fast native mesh generation with flow matching. *arXiv preprint arXiv:2607.28675*, 2026.

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

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

Xiang Zhang, Yawar Siddiqui, Armen Avetisyan, Chris Xie, Jakob Engel, and Henry Howard-Jenkins. Vertexregen: Mesh generation with continuous level of detail. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12570–12580. IEEE, 2025.

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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10612–10623. IEEE, 2025a.

Tianhao Zhao, Youjia Zhang, Hang Long, Jinshen Zhang, Wenbing Li, Yang Yang, Gongbo Zhang, Jozef Hladkỳ, Matthias Nießner, and Wei Yang. Lato: 3d mesh flow matching with structured topology preserving latents. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in neural information processing systems*, 36:73969–73982, 2023.

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025b.

A Appendix

A.1 Details of the Split-and-Rewire Tokenizer

Octant-absolute indexing. Rows and columns of $\mathbf{S}_i$ and $\mathbf{C}_{ij}$ are indexed by the child's fixed octant id $a \in \{0, \dots, 7\}$ rather than by a compacted position among the children actually present. This makes every matrix a full, lossless 8×8 byte regardless of how many children a parent vertex has, and removes any need to merge, reorder, or budget children when a parent has close to the maximum of eight – a parent with all eight children behaves identically to one with a single child, just with a denser matrix.

Occupancy disambiguates connectivity. $\mathbf{S}_i$ and $\mathbf{C}_{ij}$ are only meaningful for octants that $\mathbf{O}$ marks active: at decode time every row and column of $\mathbf{S}_i$ is masked by $\mathbf{O}_i$, and every row of $\mathbf{C}_{ij}$ by $\mathbf{O}_i \otimes \mathbf{O}_j$, before any edge is read out. This keeps occupancy and connectivity strictly separable in the representation: the generator can predict a connectivity byte for an octant slot without that byte being mistaken for evidence that the slot is occupied.

Degenerate faces. A predicted edge does not always close into a triangle at a given level – a thin feature that occupies a single voxel presents as a dangling edge or an isolated active octant rather than three mutually connected ones. Rather than dropping such connectivity, we decode it as a degenerate, zero-area face: an unclosed edge becomes (A, B, B) and an isolated active octant becomes (A, A, A), using the same $\mathbf{O}/\mathbf{S}_i/\mathbf{C}_{ij}$ prediction and the same closed-triangle rule as any ordinary face. This keeps a placeholder for the feature in the octree hierarchy so that $\mathbb{S}$ still has an ancestor to refine at the next level, instead of silently erasing structure the current resolution cannot yet resolve into a proper triangle.

A.2 Training and Inference Algorithms

We provide pseudocode for the mask-uniform training step and the corresponding mask→token→token inference loop of the mask-uniform noise strategy in Sec. 4.

Algorithm 1 details one training step of the mask-uniform noise strategy of Sec. 4, and Algorithm 2 the corresponding mask→token→token decoding loop used at generation time. Both are written for a single sample; in practice all steps are batched over the variable-length face sequence via the varlen attention packing described in Sec. 4. Two details are elided from the pseudocode for brevity: (i) classifier-free guidance, when enabled, evaluates every f_θ call twice (with $\mathbf{c}$ and with $\mathbf{c}{=}\varnothing$) and linearly interpolates the two logit sets before the $\arg\max$; (ii) the $\arg\max$ over a connectivity slot's 8-row distribution is not a single categorical draw but the autoregressive roll-out of Eq. 5, decoding one row at a time conditioned on the rows already emitted.

A.3 Geometry-Anchored RoPE Anchor Table

Unlike TSSR's 1D sequence indexing, we represent meshes via orderless per-face token blocks. To provide spatial and topological awareness, we condition rotary position embeddings (RoPE) on each token's 3D voxel coordinates and its intra-face role τ (vertex occupancy, intra-parent, or inter-parent connectivity), maintaining permutation equivariance across face blocks. For RoPE embedding dimension 64, we allocate 2 frequency pairs to τ and 30 to geometry. Each anchor is an explicit vertex (Table 4). At the coordinate level of the hourglass transformer, tokens are anchored to an ordered pair (v_a, v_b): vertex-related tokens carry (v_a, v_a) and connection tokens carry (v_a, v_b). Sharing rotation angles across half their geometric frequencies turns vertex-edge incidence into an explicit phase alignment rather than an inferred latent relation. Sorting parent vertices lexicographically z-y-x ensures canonical edge pairs across shared faces. After token reduction, each coarse token remains anchored across all three parent vertices.

Table 4 lists, for every token type at every hourglass scale, the explicit vertex (or vertex pair) it is anchored to and its role τ, referenced from Sec. 4.1.

Algorithm 1 Mask-uniform training step

Input: ground-truth slots $z^\star$ at level l, point cloud $\mathbf{c}$, mask probability p, self-generation probability p_{selfgen}, condition-drop probability p_{drop}, focal exponent γ, confidence loss weight λ, wrong-class weight κ

1. $\mathbf{c} \leftarrow \varnothing$ w.p. p_{drop} *(classifier-free guidance dropout)*
2. sample $t \sim \mathcal{U}(0,1)$; $b_{f,k} \overset{\text{iid}}{\sim} \text{Bernoulli}(r(t))$ for every slot *(cosine schedule)*
3. sample $s \sim \text{Bernoulli}(p)$
4. **if** $s = 1$ **(mask pathway):**
   (a) $\tilde{z}_{f,k} \leftarrow$ [MASK] if $b_{f,k}$=1 else $z^\star_{f,k}$, for every slot
   (b) $h \leftarrow f_\theta(\tilde{z}, t, l, \mathbf{c};\ \text{mode}{=}1)$
   (c) **return** $\mathcal{L}_{\text{mask}} = \mathcal{L}_{\text{CE}}(\{(f,k) : b_{f,k}{=}1\})$ *(Eq. **??**, focal CE on corrupted slots only)*
5. **else (uniform pathway):**
   (a) **if** $\text{rand}() < p_{\text{selfgen}}$:
      i. sample fresh $t' \sim \mathcal{U}(0,1)$, $b'_{f,k} \sim \text{Bernoulli}(r(t'))$
      ii. $z^{\text{mask}}_{f,k} \leftarrow$ [MASK] if $b'_{f,k}$=1 else $z^\star_{f,k}$
      iii. $\tilde{z} \leftarrow \arg\max\ p_\theta\big(\cdot \mid f_\theta(z^{\text{mask}}, t', l, \mathbf{c}; \text{mode}{=}1)\big)$ *(no-grad self-generated corruption)*
   (b) **else:** $\tilde{z}_{f,k} \leftarrow u_{f,k} \sim \mathcal{U}\{0,\dots,255\}^{K_k}$ if $b_{f,k}$=1 else $z^\star_{f,k}$
   (c) $h \leftarrow f_\theta(\tilde{z}, t, l, \mathbf{c};\ \text{mode}{=}0)$
   (d) $y_{f,k} \leftarrow \mathbb{1}[\arg\max\ p_\theta(\cdot \mid h_{f,k}) = z^\star_{f,k}]$ for every slot
   (e) $\mathcal{L}_{\text{conf}} \leftarrow \frac{1}{9n} \sum_{f,k} \kappa_{f,k}\, \text{BCE}\big(g_\phi(\text{sg}[h_{f,k}]), y_{f,k}\big)$ *(Eq. 7)*
   (f) **return** $\mathcal{L}_{\text{unif}} + \lambda\, \mathcal{L}_{\text{conf}}$, with $\mathcal{L}_{\text{unif}} = \mathcal{L}_{\text{CE}}(\text{all slots})$

Algorithm 2 Mask→token→token inference

Input: parent conditioning, level l, point cloud $\mathbf{c}$, steps T, re-mask schedule $\text{sched}(s)$ linearly annealed $0.2 \to 0.9$

1. $z_{f,k} \leftarrow$ [MASK] for every slot *(fully corrupted canvas)*
2. **for** $s = 0, \dots, T-1$:
   (a) $t \leftarrow s/T$
   (b) $h \leftarrow f_\theta(z, t, l, \mathbf{c};\ \text{mode}{=}1)$
   (c) $z_{f,k} \leftarrow \arg\max\ p_\theta(\cdot \mid h_{f,k})$ for every slot *(mask-fill: overwrites all slots, no freezing)*
   (d) $h \leftarrow f_\theta(z, t, l, \mathbf{c};\ \text{mode}{=}0)$
   (e) $z_{f,k} \leftarrow \arg\max\ p_\theta(\cdot \mid h_{f,k})$ for every slot *(uniform-refine, reusing h below)*
   (f) **if** $s < T-1$:
      i. $\text{conf}_{f,k} \leftarrow \sigma(g_\phi(h_{f,k}))$ for every slot
      ii. $z_{f,k} \leftarrow$ [MASK] w.p. $(1 - \text{conf}_{f,k}) \cdot \text{sched}(s)$ *(Eq. **??**, stochastic re-mask)*
3. **return** z

A.4 Focal Loss for Connectivity Class Imbalance

Both $\mathcal{L}_{\text{mask}}$ and $\mathcal{L}_{\text{unif}}$ (Sec. 4) replace the plain cross-entropy of Eq. 6 with a multi-class focal loss (Lin et al., 2017). With $p_{f,k}$ the softmax probability assigned to the ground-truth class $z^\star_{f,k}$,

$$-\log p_\theta(z^\star_{f,k} \mid \cdot) \ \longrightarrow\ -(1 - p_{f,k})^\gamma \log p_{f,k}, \tag{9}$$

which automatically down-weights slots the model already predicts confidently (γ=0 recovers plain cross-entropy) and keeps the $\sim$92% empty rows of the connectivity matrix $\mathbf{M}$ from dominating the gradient.

Table 4: Geometry-anchored rotary encoding. The 30 geometric slots are split as 2 anchors $\times$ 3 axes $\times$ 5 slots at 9/face, and as 3 vertices $\times$ 3 axes $\times$ 3 slots after reduction. τ separates tokens that share an anchor.

Hourglass level	Token Corresponds to	Anchor	Type τ
9/face, coordinate-level	F_0, F_1, F_2	$(v_0, v_0), (v_1, v_1), (v_2, v_2)$	0
	$M_0^{\text{intra}}, M_1^{\text{intra}}, M_2^{\text{intra}}$	$(v_0, v_0), (v_1, v_1), (v_2, v_2)$	1
	$M_{01}^{\text{inter}}, M_{02}^{\text{inter}}, M_{12}^{\text{inter}}$	$(v_0, v_1), (v_0, v_2), (v_1, v_2)$	2
3/face, vertex-level	subdivision / intra / inter aggregate	(v_0, v_1, v_2)	0/1/2
1/face, face-level	face feature	(v_0, v_1, v_2)	3

A.5 Denoising Within a Resolution Level

Figure 9 visualizes intra-scale denoising within a single resolution transition, starting from the resolution-32 parent mesh. At each selected diffusion step, we decode the current prediction immediately before low-confidence tokens are re-masked for the next iteration. The initial prediction contains fragmented and disconnected geometry. As denoising proceeds, iterative prediction and re-masking progressively improve the object silhouette and local connectivity, leading to a coherent mesh at the final step.

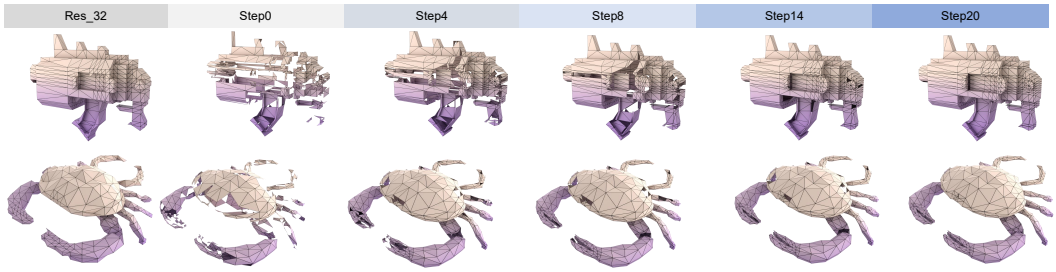

Figure 9: **Intra-scale denoising.** The left column shows the resolution-32 parent mesh. The remaining columns show the decoded predictions immediately before re-masking at diffusion steps 0, 4, 8, 14, and 20 of the same resolution transition.

A.6 Generation Across Resolutions

Figure 10 illustrates the coarse-to-fine generation trajectory across successive spatial resolutions. At lower resolutions, the mesh captures the overall silhouette and the arrangement of major parts with relatively few vertices and faces. Each subsequent split-and-rewire transition doubles the coordinate-grid resolution, introducing finer geometric features and more detailed connectivity while building on the mesh produced at the preceding level. The resulting meshes form a sequence of usable levels of detail: generation may stop at an intermediate resolution when a coarser mesh is sufficient.

A.7 Inference Speed Comparison

We measure wall-clock inference time as a function of output mesh complexity (number of faces in the generated mesh) against a next-scale autoregressive baseline (armesh) and a token-by-token autoregressive vertex-split baseline (VertexRegen Zhang et al. (2025)), as well as MeshOctave. All methods were run on a single GPU. Neither baseline was trained on meshes beyond 5k faces; for the $[5\text{k}, 10\text{k}]$ range (dashed in Figure 11) we still measure the wall-clock time each baseline takes to reach that many faces, but do not verify that the resulting mesh is valid at that scale – these points reflect real, measured decoding cost, not usable output quality. For MeshOctave, every point across $[2\text{k}, 10\text{k}]$ is likewise a directly measured average wall-clock time over batches of generated trellis test-set meshes with real samples in every bin, with the decoded mesh verified at every point.

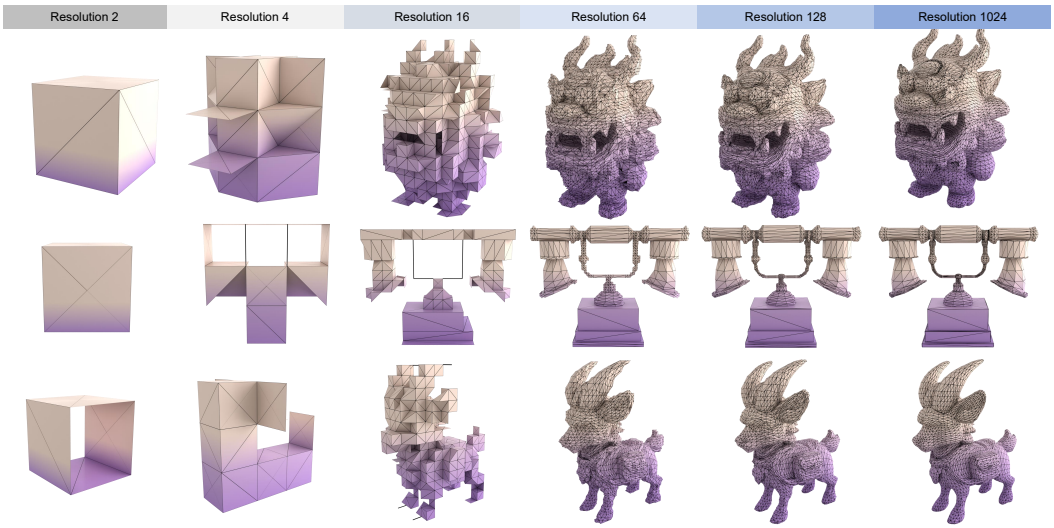

Figure 10: **Mesh generation across spatial resolutions.** Intermediate meshes from one coarse-to-fine generation trajectory. Each column represents the completed mesh at a different grid resolution. Increasing resolution progressively adds geometric detail and refines connectivity; the result at each level can be used as a standalone level-of-detail mesh.

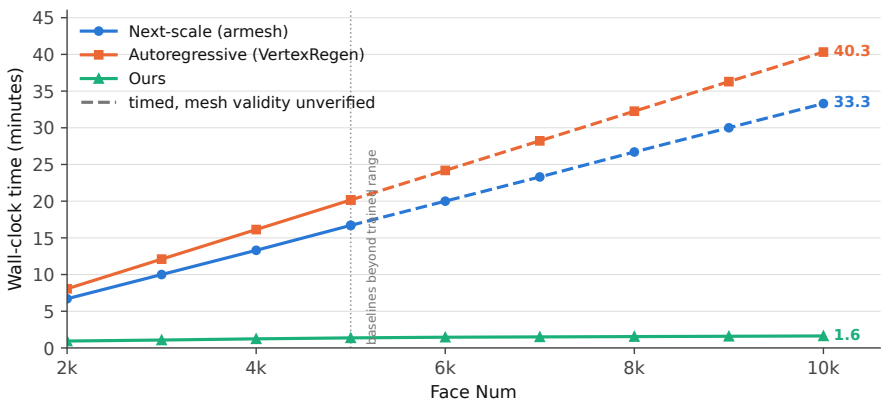

Figure 11: Inference wall-clock time versus output mesh complexity (number of faces), for the next-scale baseline (armesh), the autoregressive vertex-split baseline (VertexRegen), and MESHOCTAVE. MESHOCTAVE generates comparably complex meshes over an order of magnitude faster than both baselines. All curves are directly measured. Neither baseline was trained beyond 5k faces (dotted vertical line); their dashed segments are still measured wall-clock time, but the decoded mesh's validity at that scale is unverified. The MESHOCTAVE curve is solid throughout: every bin is a directly measured, verified generation.