LATO.2: FACTORIZED 3D MESH GENERATION WITH VERTEX AND TOPOLOGY FLOW

Anonymous authors
Paper under double-blind review



Figure 1: We present LATO.2, which factorizes mesh generation into a vertex flow (V-Flow) generating vertex positions under a controllable vertex count, and a topology flow (T-Flow) predicting connectivity from realized vertices. It supports high-quality generation (bottom left), part-wise generation at scalable resolution (bottom middle), and topology-adaptive editing (bottom right).

ABSTRACT

Flow matching over carefully designed latent representations has recently emerged as a powerful paradigm for topology-aware mesh generation. Existing approaches, however, model vertices and connectivity jointly in a joint latent space, entangling continuous vertex geometry with discrete combinatorial structure; this complicates flow learning and manifests as drifting vertices and broken surfaces. We present **LATO.2**, a factorized flow matching framework that decomposes mesh generation into a vertex flow followed by a connectivity flow conditioned on the realized vertices, with both stages anchored to a shared coarse voxel scaffold. Dedicated VAEs underpin the two stages, recovering vertices at sub-voxel precision and embedding discrete connectivity into a continuous latent space. We demonstrate two advantages unique to this factorization: (i) part-wise generation, in which the scaffold is partitioned and each part synthesized at full latent capacity, yielding substantially higher-resolution meshes than a monolithic latent permits; and (ii) topology-adaptive editing, in which manipulating first-stage vertices induces the corresponding connectivity without re-optimization. Experiments show that LATO.2 surpasses state-of-the-art topology-aware mesh generators in geometric fidelity and connectivity quality.

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

Artist-created meshes exhibit compact, well-structured topology, characterized by adaptive vertex placement and clean, coherent face connectivity. Such structure is essential in production pipelines, where it supports reliable rigging and deformation for animation, high-quality shading, and efficient storage and rendering. Yet while topology-agnostic 3D generation has advanced rapidly, whether through VecSet representations (Zhang et al., 2023; 2024; Zhao et al., 2025b; Lai et al., 2026) or structured latents (Xiang et al., 2025; Wu et al., 2026; Li et al., 2026b), these pipelines produce geometry as neural fields and extract surfaces via iso-surfacing, yielding dense, irregular tessellations devoid of artist-like structure. Directly generating meshes with such structure remains difficult because meshes are inherently hybrid representations: vertex coordinates are coupled with discrete, combinatorial connectivity of variable cardinality, not naturally captured by conventional generative models formulated over fixed-dimensional continuous spaces. To confront this discreteness, a prominent line of work tokenizes meshes into sequences and trains autoregressive models to generate faces token by token, often conditioned on point clouds (Siddiqui et al., 2024; Chen et al., 2024; 2025b; Hao et al., 2024; Weng et al., 2025). By modeling connectivity through sequential token prediction, these approaches produce structurally coherent, production-oriented meshes. However, serializing large, detailed meshes yields prohibitively long token streams, which inflates training cost, slows inference, and caps the attainable face count.

Flow-based topology-aware mesh generation has recently emerged as a compelling alternative. By modeling mesh elements in parallel within a continuous space rather than through sequential prediction, flow-based approaches offer substantially faster inference, admit stronger adherence to global conditions such as input images, and represent vertices and topology explicitly. Early attempts, including PolyDiff (Alliegro et al., 2023) and MeshCraft (He et al., 2025), apply diffusion to per-face features, demonstrating the feasibility of continuous mesh generation but scaling poorly with mesh complexity. MeshFlow (Li et al., 2026a) instead adopts an edge-centric view, constructing a continuous topology space by concatenating vertex-pair features with dynamic padding over varying vertex counts and degrees. LATO (Zhao et al., 2026) introduces a continuous topology representation via a vertex displacement field, compresses it into a structured latent space, and learns its distribution with flow matching; notably, vertices and connectivity are jointly encoded in a single unified latent, from which meshes are reconstructed through vertex subdivision, pruning, and a dedicated connectivity-prediction head.

Nevertheless, encoding vertices and connectivity in a single shared latent has a fundamental drawback: one flow model must jointly capture two statistically dissimilar signals, spatial positions and discrete combinatorial structure. This entanglement complicates flow learning and manifests as the broken surfaces observed in prior work. Our key observation is that the discreteness of connectivity is an artifact of modeling it *jointly* with geometry. Once factorized, each factor admits a natural continuous representation: vertex positions are inherently continuous quantities, and, conditioned on realized vertices, connectivity reduces to relations among known points in space, expressible as continuous per-vertex features decoded into pairwise edge probabilities.

Building on this observation, we propose LATO.2, a factorized explicit mesh generation pipeline that models vertices and connectivity with separate flow matching stages over two structured latent spaces. For vertices, we adopt the VAE architecture of LATO (Zhao et al., 2026) to compress vertex positions alone, and additionally predict a per-vertex drift vector that refines each vertex from its voxel center at the finest resolution, compensating for quantization error. For connectivity, given the realized vertices, a topology latent is constructed as per-vertex features, sampled by a vertex-conditioned flow model and decoded into pairwise edge probabilities. Under this factorization, each flow models a simpler, homogeneous distribution, easing learning and alleviating the broken surfaces of shared-latent designs. The factorization further makes topology adaptive to vertex-level intervention, which we demonstrate in two settings: (i) part-wise generation, in which the voxel scaffold is partitioned and each part synthesized at full latent capacity, yielding substantially higher-resolution meshes than a monolithic latent permits; and (ii) topology-adaptive editing, in which manipulating first-stage vertices induces the corresponding connectivity without re-optimization. Notably, Nexus (Wang et al., 2026), a concurrent and independent work, likewise generates vertices before connectivity, yet differs in both stages. For vertices, Nexus employs a coarse-to-fine octree diffusion model that generates level by level, whereas LATO.2 compresses vertices into a structured latent decoded in a single pass. For topology, Nexus formulates connectivity learning as

graph autoencoding supervised by a SpaceTime loss, whereas LATO.2 trains the topology latent with a contrastive objective and reconstructs faces via loop detection. Extensive experiments show that LATO.2 surpasses state-of-the-art topology-aware mesh generators in geometric fidelity and connectivity quality, while uniquely supporting part-wise high-resolution generation and topology-adaptive editing.

Our contributions are summarized as follows:

- We propose LATO.2, a factorized flow matching framework that decomposes explicit mesh generation into a vertex flow and a connectivity flow over structured latents anchored to a shared voxel scaffold,.
- We introduce a per-vertex drift vector that compensates finest-resolution quantization error in vertex decoding, and a per-vertex topology latent can be decoded into pairwise edge probabilities.
- LATO.2 surpasses state-of-the-art topology-aware mesh generators while enabling vertex induced part-wise high-resolution generation and topology-adaptive editing.

2 RELATED WORK

3D Shape Generation. Early 3D generation methods often relied on 2D diffusion priors due to the scarcity of large-scale 3D data. Representative approaches either optimize 3D representations through Score Distillation Sampling (SDS) (Poole et al., 2022; Chen et al., 2023; Lin et al., 2023; Tang et al., 2024; Yi et al., 2024), or synthesize multi-view images before reconstructing 3D geometry (Liu et al., 2023a; 2024; Long et al., 2024; Shi et al., 2023; Voleti et al., 2024; Shi et al., 2024; Wang & Shi, 2023). Feed-forward reconstructors such as LRM (Hong et al., 2024) and subsequent variants improve efficiency and quality with multi-view supervision and alternative representations such as NeRFs and 3D Gaussians (Li et al., 2024; Kerbl et al., 2023). In parallel, 3D-native latent models generate point clouds or compact shape latents directly (Gao et al., 2022; Liu et al., 2023b; Nichol et al., 2022; Zhang et al., 2023; 2024), and recent structured latent methods further adopt sparse or high-resolution voxel representations (Xiang et al., 2025; Wu et al., 2026; Li et al., 2026b; Xiang et al., 2026). Despite their strong geometric modeling ability, these methods typically represent shapes as implicit fields, volumetric grids, Gaussians, or other non-mesh structures, and obtain meshes only through post-hoc surface extraction such as Marching Cubes (Lorensen & Cline, 1998). As a result, the extracted meshes are often dense and irregular, and their connectivity is not modeled as a first-class generative target.

Autoregressive Mesh Generation. A line of work directly generates meshes by modeling vertices, edges, or faces as discrete sequences. PolyGen (Nash et al., 2020) and MeshGPT (Siddiqui et al., 2024) established the auto-regressive mesh generation paradigm, while later methods improve architecture design, conditioning, and tokenization (Chen et al., 2024; 2025a; Wang et al., 2024; Weng et al., 2024). Since naive face-wise encoding can require up to 9 times face number coordinate tokens, many recent methods focus on more compact mesh serialization, including adjacent mesh tokenization (Chen et al., 2025b), EdgeBreaker-style traversal (Tang et al., 2025; Rossignac, 1999), patchified block-wise indexing (Weng et al., 2025), hierarchical BFS edge tokenization (Song et al., 2025), tree-based sequencing (Lionar et al., 2025), and frontier-aware expansion (Lin et al., 2026). Other works reduce the sequential burden with hierarchical Transformers (Hao et al., 2024; Nawrot et al., 2022), deterministic face prediction from generated vertices (Kim et al., 2026), reinforced fine-tuning (Zhao et al., 2025a; Liu et al., 2026), or quadrilateral mesh generation (Liu et al., 2025). These methods make mesh topology explicit, but they remain constrained by long token sequences, coordinate quantization, ordering sensitivity, and the difficulty of maintaining long-range topological consistency.

Flow-based Explicit Mesh Generation. Continuous and non-autoregressive methods provide an alternative to sequential mesh token generation. DMesh (Son et al., 2024) and DMesh++ (Son et al., 2025) optimize differentiable face-existence probabilities, but rely on Delaunay-based candidate complexes. PolyDiff (Alliegro et al., 2023) performs discrete denoising diffusion over quantized triangle soups, MeshCraft (He et al., 2025) introduces continuous face-level tokens with a flow-based Diffusion Transformer, and SpaceMesh (Shen et al., 2024) represents topology with continuous vertex embeddings while still requiring connectivity recovery. More recent latent diffusion or flow methods, such as MeshFlow (Li et al., 2026a) and LATO (Zhao et al., 2026), move explicit mesh generation into compact latent spaces and generate geometry and connectivity more efficiently than long

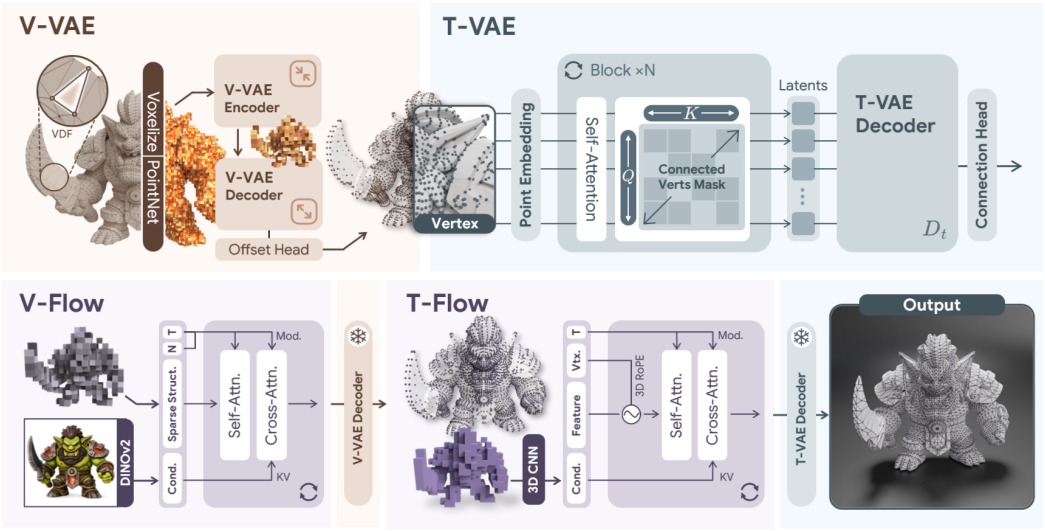

Figure 2: **Overview of LATO.2 pipeline.** LATO.2 factorizes explicit mesh generation into vertex and topology synthesis. V-VAE learns sparse vertex latents and reconstructs high-resolution vertex positions with sub-voxel offsets, while T-VAE learns topology latents through vertex-aware attention. At inference, V-Flow generates vertices conditioned on images with controllable vertex counts, and T-Flow predicts connectivity conditioned on the generated vertices.

auto-regressive sequences. However, these methods still tend to couple vertex geometry and mesh connectivity within a shared representation or generate them in parallel. In contrast, our method explicitly factorizes mesh generation into high-resolution vertex generation and vertex-conditioned topology generation. This separation matches the different structures of the two problems: vertex geometry is decoded through local sparse refinement, while topology is modeled as a relational latent distribution conditioned on the generated vertex set.

3 METHOD

We present LATO.2, a factorized sparse latent framework for explicit mesh generation, illustrated in Figure 2. The framework comprises two representation stages and two generative stages. The Vertex VAE (V-VAE) takes Vertex Displacement Field (VDF) as input (Zhao et al., 2026), compress VDF into a sparse structured latent; and decode into vertex voxels through iterative subdivision and pruning. To compensate for quantization at the finest voxel resolution, the decoder additionally predicts a per-vertex drift vector that refines each vertex away from its voxel center. Given realized vertices, the Topology VAE (T-VAE) encodes connectivity as per-vertex features and recovers pairwise edge probabilities, from which faces are reconstructed via loop detection. Over these two latent spaces, we train a vertex flow (V-Flow) and a vertex-conditioned connectivity flow (T-Flow).

3.1 VERTEX AND TOPOLOGY VAE

We represent a mesh $\mathcal{M} = (\mathbf{V}, \mathbf{E})$ by its vertices $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^{N} \subset \mathbb{R}^3$ and symmetric binary adjacency matrix $\mathbf{A} = [\mathbf{e}_{ij}]$, where $\mathbf{e}_{ij} = 1$ iff vertices i and j are connected, and $\mathbf{e}_{ii} = 0$. We encode geometry and connectivity in separate latent spaces using the Vertex VAE (V-VAE) and Topology VAE (T-VAE), with encoder–decoder pairs $(\mathcal{E}_{\mathbf{v}}, \mathcal{D}_{\mathbf{v}})$ and $(\mathcal{E}_{\mathbf{t}}, \mathcal{D}_{\mathbf{t}})$, respectively.

Vertex VAE. Following LATO (Zhao et al., 2026), we encode the unordered, variable-size vertex set through a Vertex Displacement Field (VDF). We sample surface points $\mathcal{P} = \{\mathbf{p}_k\}_{k=1}^{K}$, each with a normal $\mathbf{n}_k$ and displacement $\mathbf{d}_k$ to a vertex of its containing face. PointNet (Qi et al., 2017) embeds these point features, which are aggregated on a sparse 1024^3 voxel grid and downsampled through sparse 3D convolution and attention to obtain $\mathbf{z}_{\mathbf{v}}$.

The decoder progressively subdivides voxels and prunes those without vertices, refining surviving features after each pruning step by sparse cross-attention with $\mathbf{z_v}$. At the finest level, it predicts an offset $\boldsymbol{\delta}_i$ from each surviving voxel center $\mathbf{c}_i$ to compensate for quantization error:

$$\mathbf{z_v} = \mathcal{E}_\mathbf{v}(\mathcal{P}), \qquad \{\mathbf{c}_i, \boldsymbol{\delta}_i\}_{i=1}^N = \mathcal{D}_\mathbf{v}(\mathbf{z_v}), \qquad \hat{\mathbf{v}}_i = \mathbf{c}_i + \boldsymbol{\delta}_i. \tag{1}$$

Topology VAE. The T-VAE represents connectivity as per-vertex latent features $\mathbf{z_t} = \mathcal{E}_\mathbf{t}(\hat{\mathbf{V}}; \mathbf{A})$ aligned with $\hat{\mathbf{V}} = \{\hat{\mathbf{v}}_i\}_{i=1}^N$. An eight-layer transformer encodes positional embeddings of the vertices; ground-truth connectivity enters only through a mask restricting attention in every layer to each vertex itself and its mesh neighbors:

$$M_{ij} = \begin{cases} 0, & \mathbf{e}_{ij} = 1 \text{ or } i = j, \\ -\infty, & \text{otherwise}, \end{cases} \qquad \mathrm{Attn}(i,j) = \mathrm{softmax}_j\left(\frac{q_i^\top k_j}{\sqrt{d}} + M_{ij}\right), \tag{2}$$

The decoder uses unmasked self-attention to reconstruct connectivity from $\mathbf{z_t}$. Following LATO (Zhao et al., 2026), an MLP predicts edge probabilities from final-layer vertex features $\mathbf{h}_i$:

$$\mathbf{h} = \mathcal{D}_\mathbf{t}(\mathbf{z_t}), \qquad \hat{\mathbf{e}}_{ij} = \sigma\big[\mathrm{MLP}(\mathbf{h}_i \oplus \mathbf{h}_j) + \mathrm{MLP}(\mathbf{h}_j \oplus \mathbf{h}_i)\big], \tag{3}$$

where $\oplus$ denotes concatenation, and combining logits from both input orders enforces symmetry. To avoid scoring all N^2 pairs during training, the candidate set $\mathcal{C}$ includes all true edges, top-K nearest-neighbor pairs to provide hard local negatives, and uniformly sampled random pairs to provide easy negatives. At inference, faces are recovered from predicted edges via loop detection.

Training objectives. Both VAEs use asymmetric focal loss $\mathcal{L}_{\text{asy}}$ (Ridnik et al., 2021) to address class imbalance and KL regularization of their latents. The V-VAE supervises pruning at each subdivision level r and regresses offsets with MSE $\mathcal{L}_{\text{off}}$; the T-VAE supervises edge predictions over $\mathcal{C}$. Omitting loss weights, the objectives are

$$\mathcal{L}_v = \sum_r \mathcal{L}_{\text{asy}}^r + \mathcal{L}_{\text{off}} + \mathcal{L}_{\text{KL}}, \quad \mathcal{L}_t = \frac{1}{|\mathcal{C}|} \sum_{(i,j)\in\mathcal{C}} \mathcal{L}_{\text{asy}}\left(\hat{\mathbf{e}}_{ij}, \mathbf{e}_{ij}\right) + \mathcal{L}_{\text{KL}}. \tag{4}$$

3.2 Factorized Mesh Generation

With both autoencoders trained, we learn flow models over their latent spaces. Given a task condition, such as an image or a point cloud, LATO.2 generates a mesh in three steps: it first produces a coarse sparse structure, then samples the vertex latent on its active voxels and decodes high-resolution vertices, and finally, conditioned on the realized vertices, samples the topology latent and decodes pairwise connectivity.

Vertex Flow Model. Following TRELLIS (Xiang et al., 2025), we train a flow model to generate a coarse sparse structure $\hat{\mathcal{S}}$ from an image or a point cloud. The sparse V-Flow transformer samples the vertex latent $\mathbf{z_v}$ on its active voxels, and $\mathcal{D}_\mathbf{v}$ decodes the realized vertices $\hat{\mathbf{V}}$ via progressive sparse refinement. V-Flow receives DINOv2 (Oquab et al., 2023) features $\mathbf{c}_{\text{img}}$ from a random rendering of the training mesh through cross-attention, and a vertex-count condition $\mathbf{c}_{\text{vn}} = \log_2 N$ embedded by an MLP and injected via adaLN together with the timestep embedding. Here, N is the ground-truth count during training and user-specified at inference.

Topology Flow Model. The T-Flow transformer samples the topology latent $\mathbf{z_t}$ over per-vertex tokens aligned with $\hat{\mathbf{V}}$, incorporating vertex positions through 3D positional encoding. Coarse geometric context $\mathbf{c}_{\text{g}} = \text{3DCNN}(\hat{\mathcal{S}})$ supplies global shape information beyond local neighborhoods. We randomly drop $\mathbf{c}_{\text{g}}$ during training to support sampling without this context, as needed for editing existing meshes, while retaining vertex conditioning. RMS normalization of attention queries and keys (Zhang & Sennrich, 2019) stabilizes attention across vertex sets of varying size.

Training and Inference. Both flows are trained separately with the same rectified-flow objective. For branch $b \in \{\mathbf{v}, \mathbf{t}\}$,

$$\mathcal{L}_{\text{flow}}^b = \mathbb{E}_{\mathbf{z}_b,\, \tau,\, \epsilon,\, \mathbf{c}_b}\left[\left\| v_{\theta_b}(\mathbf{z}_b^\tau, \tau, \mathbf{c}_b) - (\epsilon - \mathbf{z}_b)\right\|_2^2\right], \tag{5}$$

where $\mathbf{z}_b^\tau = (1-\tau)\mathbf{z}_b + \tau\epsilon$, $\epsilon \sim \mathcal{N}(0, I)$, and $\tau \in [0,1]$. The branch-specific conditions are $\mathbf{c_v} = (\mathbf{c}_{\text{img}}, \mathbf{c}_{\text{vn}})$ and $\mathbf{c_t} = (\hat{\mathbf{V}}, \mathbf{c}_{\text{g}})$, with $\hat{\mathbf{V}}$ supplied through token positional encodings. At inference, $\mathcal{D}_\mathbf{t}$ scores all vertex pairs using a lightweight MLP over precomputed vertex features, an $O(N^2)$ operation, and faces are recovered via loop detection.

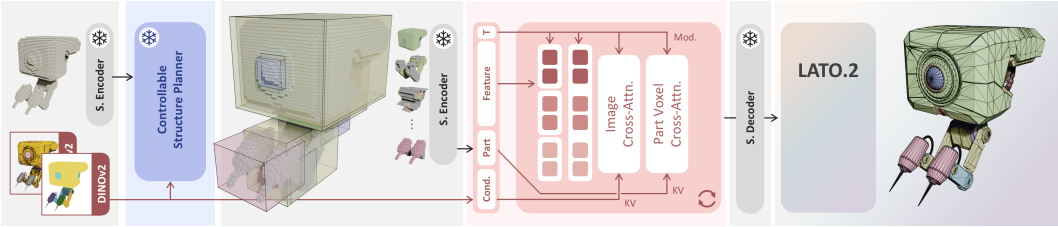

Figure 3: **Part-wise generation pipeline.** A controllable structure planner (Yang et al., 2025) augments the sparse structure with part-level bounding boxes. Its encoded part features are fused with image features via cross-attention between image tokens and part voxels. The structure decoder predicts a part-aware sparse structure for LATO.2 to generate per-part vertices. T-Flow predicts connectivity jointly over all vertices or per part followed by stitching.

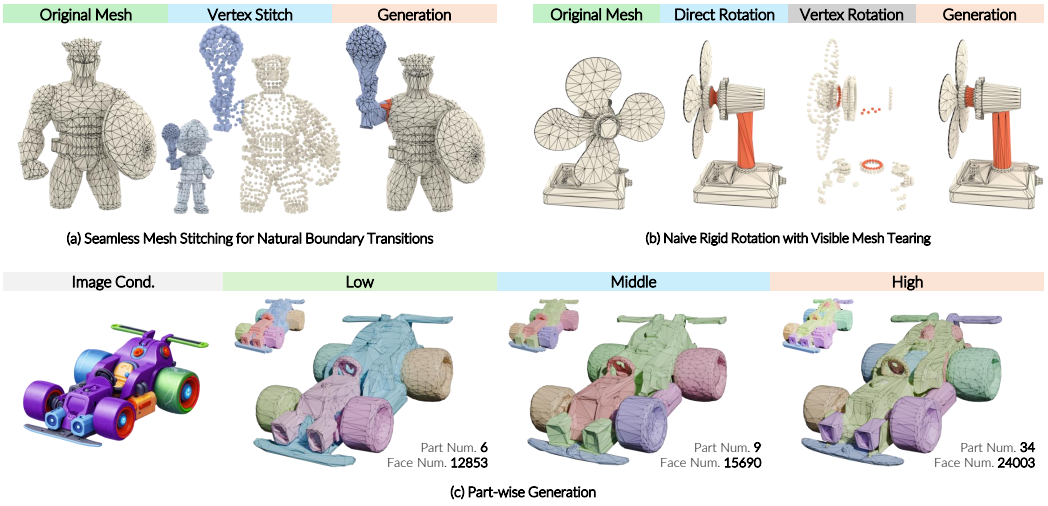

Figure 4: **Topology-adaptive editing and part-wise generation.** (a) Mesh stitching: T-Flow connects vertices from different meshes; new faces are red. (b) Part transformation: regenerating connectivity repairs rotation-induced artifacts (red: artifacts on the left, regenerated faces on the right). (c) Part-wise generation: full latent capacity per part yields denser meshes with finer detail as the part count increases.

3.3 Vertex-Induced Adaptive Topology

The factorized design carries a practical consequence beyond simple generation: since the topology flow is conditioned on realized vertices, *any* modification to the vertex set, whether user edits or programmatic recomposition, can be propagated to connectivity simply by re-running the T-Flow. We demonstrate two applications of this property: topology-adaptive mesh editing and part-wise refined generation.

Topology-Adaptive Mesh Editing. Users can modify vertices from generated or existing meshes and re-sample the topology latent conditioned on the edited set. Figure 4(a,b) demonstrates mesh stitching and part transformation, where the T-Flow reconstructs connectivity across combined or transformed regions. Implementation details are provided in Appendix A.4.2.

Part-wise Generation and Refinement. To increase mesh resolution beyond the capacity of a single latent volume, we partition the coarse structure $\hat{\mathcal{S}}$ into parts following a strategy similar to OmniPart (Yang et al., 2025). Each part is rescaled to fill the latent volume and generated independently with the V-Flow. The resulting vertices are transformed back to the original coordinate frame, and connectivity is generated using one or multiple T-Flow passes. This allocates full latent capacity to each part, enabling higher vertex density and finer detail at the same latent grid resolution, shown in Figure 4(c); the pipeline is illustrated in Figure 3. The same procedure supports local refinement by regenerating a selected region with increased vertex capacity and stitching it back into the mesh.

Table 1: Quantitative comparison with topology-aware mesh generation methods. Our method achieves the strongest performance across all metrics.

Method	Mesh Any.V2	Fast Mesh	Mesh Silk.	BPT	Deep Mesh	Mesh Ripple	Mesh Flow	LATO	**Ours**
Type	AR	AR	AR	AR	AR	AR	Flow	Flow	Flow
CD($L2$) ↓	0.1083	0.0822	0.0654	0.0603	0.0529	0.0458	0.0455	<u>0.0421</u>	**0.0407**
CD($L1$) ↓	0.1505	0.1163	0.0937	0.0862	0.0765	0.0668	0.0668	<u>0.0617</u>	**0.0596**
HD ↓	0.2318	0.1397	0.1455	0.1067	0.0946	0.0941	0.0772	<u>0.0738</u>	**0.0657**
|NC| ↑	0.6946	0.6939	0.7759	0.8111	0.8218	0.8174	0.8227	<u>0.8262</u>	**0.8341**

4 EXPERIMENTS

We evaluate LATO.2 from four perspectives: mesh reconstruction quality, conditional mesh generation quality, topology fidelity, and the effectiveness of the proposed factorized formulation.

Datasets. We train on approximately 450K 3D assets from TRELLIS500K (Xiang et al., 2025), Objaverse (Deitke et al., 2023b), Objaverse-XL (Deitke et al., 2023a), 3D-FUTURE (Fu et al., 2021), Toys4K (Stojanov et al., 2021), and ABO (Collins et al., 2022). We additionally construct 100K synthetic meshes, with the construction procedure detailed in Appendix A.1. We evaluate on dense meshes and artist meshes; details are provided in Appendix A.2.

Implementation Details. Our V-VAE compresses vertices into a 64^3 sparse latent grid and reconstructs them at 1024^3 resolution with continuous sub-voxel offsets. V-Flow is conditioned on DINOv2 image features and the target vertex count, while T-Flow generates per-vertex topology latents conditioned on vertex positions and coarse geometric structure. All models are trained on 8 NVIDIA H100 GPUs using AdamW with a learning rate of 10^{-4}. Training takes approximately 4, 1, 7, and 2 days for V-VAE, T-VAE, V-Flow, and T-Flow, respectively. Detailed architectures and training configurations are provided in the Appendix A.1.

Baselines. We compare LATO.2 with two categories of explicit mesh generation approaches. The first category is autoregressive mesh generators, including MeshAnythingV2 (Chen et al., 2025b), BPT (Weng et al., 2025), FastMesh (Kim et al., 2026), DeepMesh (Zhao et al., 2025a), MeshSilkSong (Song et al., 2025), and MeshRipple (Lin et al., 2026). These methods generate explicit meshes by autoregressively predicting mesh tokens or by factorizing mesh components into separately modeled representations. The second category consists of latent-flow-based topology-aware mesh generators, including LATO (Zhao et al., 2026) and MeshFlow (Li et al., 2026a). For a fair comparison, we evaluate all methods under identical input conditions on the same mixed test set whenever official checkpoints or publicly released generation results are available. For methods that generate meshes with different resolutions, we follow their default inference configurations and directly evaluate the resulting meshes without additional post-processing.

Metrics. We evaluate both surface reconstruction fidelity and vertex reconstruction accuracy. For mesh level reconstruction and conditional generation, we uniformly sample points from the surfaces of the predicted and reference meshes, and compute Chamfer Distance (CD), Hausdorff Distance (HD), and Normal Consistency (NC). CD measures the average bidirectional geometric discrepancy between the two surfaces, HD evaluates the maximum surface deviation, and NC quantifies the consistency of local surface orientations. For vertex-level reconstruction, we evaluate the predicted sparse occupancy representation at the finest resolution using precision, recall, F1 score, and Intersection over Union (IoU) between the predicted and ground-truth occupancy grids.

4.1 QUANTITATIVE ANALYSIS

V-VAE Performance. On artist meshes, our V-VAE achieves lower CD(L1) and HD than the evaluated mesh autoencoding baselines (Table 5; details in Appendix A.2). These results validate the importance of sub voxel offset modeling for high-fidelity vertex reconstruction.

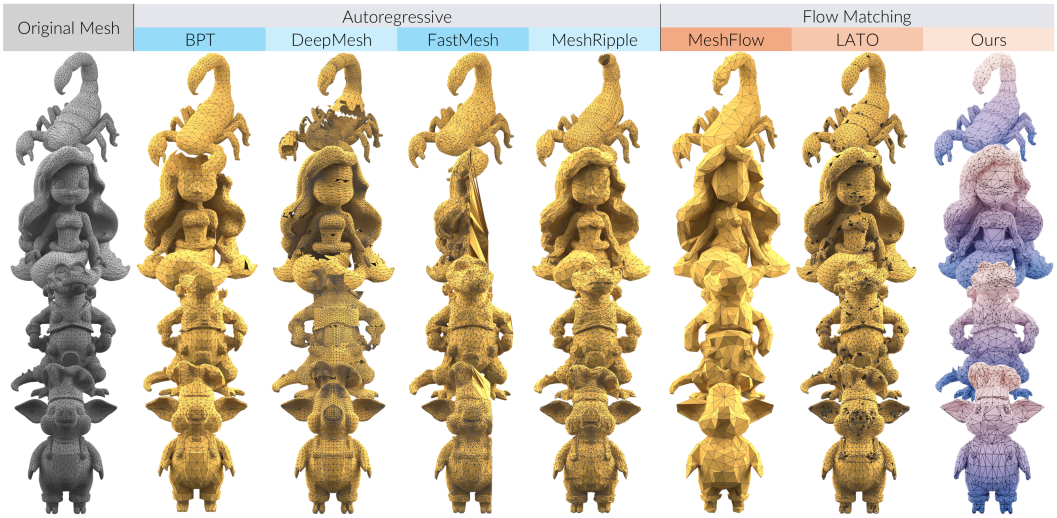

Figure 5: **Qualitative comparison on geometry-conditioned mesh generation.** Given point cloud or voxel conditions, baseline methods exhibit missing or broken surface regions, whereas our method produces complete shapes with more coherent topology.

Geometry Conditioned Mesh Generation. We evaluate geometry conditioned mesh generation on mixed test set using point cloud conditions, details in Appendix A.2, comparing against both autoregressive mesh generators and latent flow based approaches. As shown in Tab. 1, LATO.2 achieves the strongest overall performance across both geometric and topology related metrics. Compared with autoregressive approaches, our sparse latent generation avoids the need for excessively long mesh token sequences, enabling more stable generation of meshes with higher vertex counts. Compared with existing latent flow methods, our explicit factorization of vertex geometry and topology provides more accurate geometric reconstruction while maintaining more coherent connectivity.

Topology Generation. To isolate the effect of topology modeling, we compare different combinations of vertex and topology sources on artist meshes in Tab. 2 (details in Appendix A.2). Using ground-truth vertices with the topology autoencoder provides an upper bound for connectivity reconstruction. Replacing the topology autoencoder with the topology flow measures the generative error of topology sampling, while replacing ground-truth vertices with generated vertices evaluates the full pipeline. The results show that the topology flow preserves most of the reconstruction quality and that the factorized design localizes errors to the generative stages of vertex placement and connectivity prediction, rather than the autoencoders themselves.

4.2 Qualitative Analysis

Qualitative Comparison. Fig. 5 shows qualitative comparisons with autoregressive baselines and latent flow based topology-aware baselines. Autoregressive methods often produce broken surfaces, missing thin structures, or irregular local connectivity when the target mesh contains complex geometry. Latent flow baselines generate more coherent global shapes, but may still lose fine geometric details or produce less faithful connectivity. In contrast, LATO.2 preserves fine structures while maintaining cleaner mesh connectivity, benefiting from the separation of high resolution vertex generation and topology generation conditioned on the resulting vertex positions.

Vertex Number Controllability. Fig. 6 visualizes vertex count controllability. Given the same input condition, increasing the target vertex budget produces progressively denser meshes while preserving the object level geometry. This demonstrates that the explicit sparse vertex generation process can serve as a practical control mechanism for mesh resolution. Additional examples of meshes generated with different target vertex budgets are shown in Appendix A.5.

4.3 Ablation Study

On V-VAE. Tab. 3 studies the vertex-branch design on artist meshes (details in Appendix A.2). Training without procedural synthetic meshes also reduces sparse occupancy accuracy, suggesting

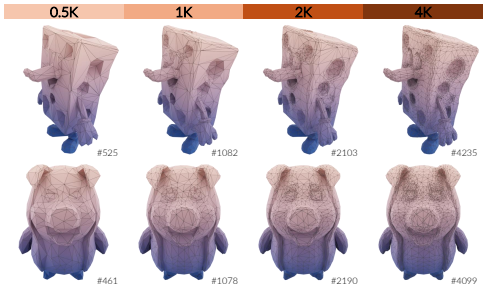

Figure 6: **Vertex number controlled generation.** Given the same structure voxels, increasing the target budget from 0.5K to 4K produces denser meshes while preserving the underlying shape structure.

Table 2: **Topology generation under different vertex inputs.** GT-Verts denotes ground-truth vertices. GT-Verts + T-VAE reconstructs topology from the ground-truth mesh, providing the performance upper bound. The three T-Flow columns use ground-truth, V-VAE reconstructed, and V-Flow generated vertices, respectively, isolating the error introduced by each stage.

Vtx. / **Topo.**	GT-Verts T-VAE	GT-Verts T-Flow	V-VAE T-Flow	V-Flow T-Flow
CD($L2$) ↓	**0.0351**	0.0399	<u>0.0393</u>	0.0407
CD($L1$) ↓	**0.0478**	0.0543	<u>0.0538</u>	0.0570
HD ↓	**0.0859**	0.1047	<u>0.0993</u>	0.1029
|NC| ↑	**0.8615**	0.8199	<u>0.8201</u>	0.8051

Table 3: **V-VAE ablation.** All metrics are computed between the predicted vertices and ground-truth vertices. Removing the offset head only affects continuous sub-voxel localization, degrading CD and HD but leaving voxel-level occupancy metrics unchanged.

Method	w/o Training on Synthetic Meshes	w/o Offset Head	w/o VDF DownSample	**Full V-VAE**
CD($L1$) ↓	0.0005	0.0011	0.0034	**0.0003**
HD ↓	0.0071	0.0082	0.0145	**0.0069**
ACC ↑	0.9134	**0.9752**	0.8981	**0.9752**
F1 ↑	0.9137	**0.9735**	0.9025	**0.9735**
Recall ↑	0.9140	**0.9730**	0.9069	**0.9730**
IOU ↑	0.9034	**0.9603**	0.8873	**0.9603**

Table 4: **T-Flow ablation.** Conditioning T-Flow on the 64-resolution sparse structure improves final mesh quality, indicating that coarse geometric support helps generate more consistent topology.

Method	w/o Geometric Condition $\mathbf{c}_g$	**Full T-Flow**
CD($L2$) ↓	0.0411	**0.0407**
CD($L1$) ↓	0.0601	**0.0596**
HD ↓	0.0694	**0.0657**
|NC| ↑	0.8238	**0.8341**

that the synthetic set improves coverage of diverse vertex structures. Removing the sub-voxel offset head leaves the sparse occupancy prediction nearly unchanged but noticeably degrades surface reconstruction quality, showing that the offset head specifically improves continuous vertex precision rather than sparse voxel classification. Removing the high-resolution point-to-voxel aggregation and downsampling path causes a larger drop in occupancy metrics and geometric quality, indicating that fine local evidence should be captured before compression to the coarse latent support.

On Topology Generation. Tab. 4 evaluates topology generation on the mixed test set (details in Appendix A.2). Removing the sparse structure condition from the topology flow degrades both geometric and topology-related metrics, showing that the shared coarse support provides useful context for generating consistent connectivity. We also compare against a deterministic topology predictor that directly completes edges from vertices without sampling topology latents. The topology latent flow performs better, supporting our claim that mesh connectivity should be modeled as a vertex conditioned latent distribution rather than a feed forward completion step.

5 Conclusion

We presented LATO.2, a sparse latent framework that factorizes explicit mesh generation into vertex and topology flow conditioned on the resulting vertex positions. This design improves mesh quality while enabling vertex count control, topology adaptive editing, and scalable part-wise generation beyond the capacity of a single latent representation. The current two-stage factorization also introduces several limitations, including geometric inaccuracies from the vertex stage may propagate to topology synthesis, and topology decoding requires quadratic complexity. Future work will explore tighter interactions between the vertex and topology stages through joint or iterative refinement, enabling topology feedback to improve vertex placement.

AI USE STATEMENT

During the preparation of this manuscript, we used ChatGPT solely to improve language, grammar, and readability. We did not use generative AI tools to generate or process datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims or proofs, propose or refine hypotheses, design research methodology or experiments, implement methods, translate content, conduct qualitative or thematic analyses, or interpret results. All AI-assisted edits were reviewed and revised by the authors to ensure accuracy and preserve the intended meaning. The authors take full responsibility for the final content of this paper.

REFERENCES

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

Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 22246–22256, 2023.

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, Zhongang Cai, Lei Yang, Gang Yu, Guosheng Lin, et al. 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13922–13931, 2025b.

Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21126–21136, 2022.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36:35799–35813, 2023a.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13142–13153, 2023b.

Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129 (12):3313–3337, 2021.

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

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.

Xianglong He, Junyi Chen, Di Huang, Zexiang Liu, Xiaoshui Huang, Wanli Ouyang, Chun Yuan, and Yangguang Li. Meshcraft: Exploring efficient and controllable mesh generation with flow-based dits. *arXiv preprint arXiv:2503.23022*, 2025.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. In *International Conference on Learning Representations*, volume 2024, pp. 50678–50702, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

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.

Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In *International conference on learning representations*, volume 2024, pp. 21896–21920, 2024.

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*, pp. 5849–5858, 2026a.

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

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 300–309, 2023.

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.

Stefan Lionar, Jiabin Liang, and Gim Hee Lee. Treemeshgpt: Artistic mesh generation with autoregressive tree sequencing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26608–26617, 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. *arXiv preprint arXiv:2509.21420*, 2025.

Jian Liu, Jing Xu, Song Guo, Jing Li, Jiaao Yu, Haohan Weng, Biwen Lei, Xianghui Yang, Zhuo Chen, Fangqi Zhu, et al. Mesh-rft: Enhancing mesh generation via fine-grained reinforcement fine-tuning. *Advances in Neural Information Processing Systems*, 38:156033–156058, 2026.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9298–9309, 2023a.

Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. In *International conference on learning representations*, volume 2024, pp. 27676–27697, 2024.

Zhen Liu, Yao Feng, Michael J Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. Meshdiffusion: Score-based generative 3d mesh modeling. *arXiv preprint arXiv:2303.08133*, 2023b.

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9970–9980, 2024.

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. 1998.

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.

Piotr Nawrot, Szymon Tworkowski, Michał Tyrolski, Łukasz Kaiser, Yuhuai Wu, Christian Szegedy, and Henryk Michalewski. Hierarchical transformers are more efficient language models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 1559–1571, 2022.

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

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

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

Tal Ridnik, Emanuel Ben-Baruch, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 82–91, 2021.

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

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.

Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023.

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *International conference on learning representations*, volume 2024, pp. 39838–39859, 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19615–19625, 2024.

Sanghyun Son, Matheus Gadelha, Yang Zhou, Zexiang Xu, Ming C Lin, and Yi Zhou. Dmesh: A differentiable mesh representation. *arXiv preprint arXiv:2404.13445*, 2024.

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*, pp. 26590–26599, 2025.

Gaochao Song, Zibo Zhao, Haohan Weng, Jingbo Zeng, Rongfei Jia, and Shenghua Gao. Mesh silksong: Auto-regressive mesh generation as weaving silk. *arXiv preprint arXiv:2507.02477*, 2025.

Stefan Stojanov, Anh Thai, and James M Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1798–1808, 2021.

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *International Conference on Learning Representations*, volume 2024, pp. 33879–33896, 2024.

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.

Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In *European Conference on Computer Vision*, pp. 439–457. Springer, 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.

Peng Wang and Yichun Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023.

Zhengyi Wang, Jonathan Lorraine, Yikai Wang, Hang Su, Jun Zhu, Sanja Fidler, and Xiaohui Zeng. Llama-mesh: Unifying 3d mesh generation with language models. *arXiv preprint arXiv:2411.09595*, 2024.

Haohan Weng, Yikai Wang, Tong Zhang, CL Chen, and Jun Zhu. Pivotmesh: Generic 3d mesh generation via pivot vertices guidance. *arXiv preprint arXiv:2405.16890*, 2024.

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 Computer Vision and Pattern Recognition Conference*, pp. 11093–11103, 2025.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, Yao Yao, et al. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *Advances in Neural Information Processing Systems*, 38:170778–170804, 2026.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21469–21480, 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.

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.

Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6796–6807, 2024.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

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.

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*, pp. 10612–10623, 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. *arXiv preprint arXiv:2603.06357*, 2026.

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 Implementation Details

This appendix details synthetic data construction, model architectures, and training configurations for the vertex and topology branches described in Sections 3.1 and 3.2.

Synthetic data construction. We construct a procedural dataset of 100K synthetic meshes to increase geometric and topological diversity. Starting from cubes, cylinders, cones, and spheres, we apply subdivision, geometric deformation, twisting, warping, and primitive composition. These operations produce meshes with varied vertex densities, local curvature distributions, and connectivity patterns.

Vertex branch. For each mesh, we uniformly sample 819,200 surface points with positions, normals, and vertex displacement vectors. The V-VAE encodes point-wise features using PointNet and aggregates them into a 1024^3 sparse voxel grid. Sparse 3D convolutional downsampling and a Sparse Transformer with a hidden dimension of 512 and 8 attention heads compress these features into a 64^3 sparse latent grid with 32 channels. The decoder reconstructs vertices at 1024^3 resolution through occupancy-pruned subdivision and latent cross-attention, followed by sub-voxel offset regression. V-Flow adopts a 12-block conditional flow matching Transformer adapted from TRELLIS (Xiang et al., 2025), conditioned on DINOv2 image features and the target vertex count.

Topology branch. For topology modeling, vertex coordinates are discretized into 1024 bins and embedded using Fourier features with a width of 768. The T-VAE uses adjacency-masked attention in its encoder and unmasked self-attention in its decoder, with a per-vertex latent dimension of 16. A pairwise MLP classifier predicts edge probabilities from the decoded vertex features. T-Flow generates topology latents conditioned on vertex positions and coarse geometric structure, incorporating vertex positional encodings into its tokens.

Model size and training. The V-VAE, V-Flow, T-VAE, and T-Flow contain approximately 320M, 160M, 180M, and 240M trainable parameters, respectively. VAE parameter counts include both encoders and decoders; frozen conditioning encoders are excluded. All models are optimized using AdamW with a learning rate of 10^{-4} on 8 NVIDIA H100 GPUs. The V-VAE and T-VAE are trained for approximately 4 days and 1 day, respectively. After training the autoencoders, we train V-Flow for 7 days with an effective batch size of 256 and T-Flow for 2 days using dynamic batching based on the vertex-token budget.

A.2 Evaluation Protocol

We evaluate on a mixed test set of 400 meshes and its subset of 200 held-out artist meshes.

Mixed test set. The mixed test set combines the 200 held-out artist meshes described below with 200 dense meshes generated by TRELLIS (Xiang et al., 2025) from randomly selected images. We use this 400-mesh set for the geometry-conditioned generation comparison in Table 1 and the T-Flow ablation in Table 4. These meshes provide the reference geometry and the point-cloud conditions for evaluating the generated meshes.

Held-out artist meshes. We select 200 artist meshes in total from Objaverse-XL (Deitke et al., 2023a) and Toys4K (Stojanov et al., 2021) and exclude them from model training. This set is used for the vertex/topology source comparison in Table 2, the V-VAE ablation in Table 3, and the V-VAE reconstruction comparison in Table 5. The original artist meshes provide the reference vertex sets and connectivity. All four combinations of vertex and topology sources in Table 2 are evaluated on this same held-out set.

A.3 VAE Reconstruction

We evaluate the performance of V-VAE on the 200 held-out artist meshes described in Appendix A.2. As reported in Tab. 5, LATO.2 achieves superior reconstruction performance compared with existing mesh autoencoding baselines. Compared with quantized token-based representations and the vertex autoencoder of LATO, our continuous sub-voxel offset prediction significantly reduces geometric reconstruction errors. This indicates that, after sparse voxel pruning, the dominant source

of reconstruction error arises from discretization artifacts, which can be effectively mitigated by predicting continuous offsets within each voxel. These results validate the importance of sub-voxel offset modeling for high-fidelity vertex reconstruction.

Figure 7 visualizes (a) vertex reconstruction by V-VAE, (b) the effect of the offset head on reconstruction quality, and (c) topology reconstruction by T-VAE.

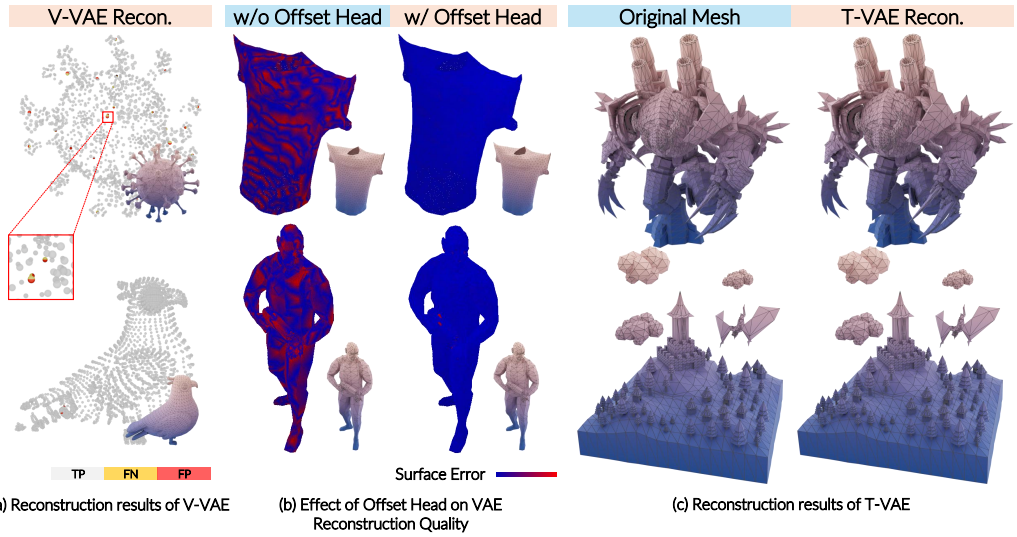

Figure 7: **Visualization on V-VAE and T-VAE reconstruction performances.** a) Visualization of V-VAE vertex reconstruction. Gray vertices denote correctly reconstructed vertices, yellow vertices denote missing ground-truth vertices, and red vertices denote false positive predictions. b) Effect of the offset head on VAE reconstruction quality, significant less error presented with offset head. c) Topology reconstruction by T-VAE. Given the original mesh, T-VAE reconstructs connectivity from the learned topology latent perfectly.

Table 5: Quantitative comparison on V-VAE performance.

Method	Quantization Levels	CD(L2)↓	CD(L1)↓	HD↓
MeshGPT	128	<u>0.0013</u>	0.0185	0.0955
PivotMesh	128	0.0074	0.0395	0.2227
MeshCraft	256	0.0106	0.0524	0.2842
LATO	512	**0.0000**	<u>0.0038</u>	<u>0.0083</u>
Ours (V-VAE)	**1024 + δ (Float)**	**0.0000**	**0.0003**	**0.0069**

A.4 Part-wise Generation and Mesh Editing

Part-wise generation and mesh editing both construct or modify vertices before reconstructing connectivity with T-Flow. We detail the applications illustrated in Figure 4.

A.4.1 Part-wise Generation and Refinement

Part-aware structure. Figure 3 shows the complete pipeline. A controllable structure planner (Yang et al., 2025) augments the sparse structure generated from an image or point cloud with part-level bounding boxes. The resulting box-aware structure is encoded into part features and fused with image features through cross-attention between image tokens and part voxels. The structure decoder then predicts a part-aware sparse structure.

Part generation and integration. Each part is rescaled to fill the latent volume and generated independently by V-Flow. After transforming the vertices back into the original coordinate frame,

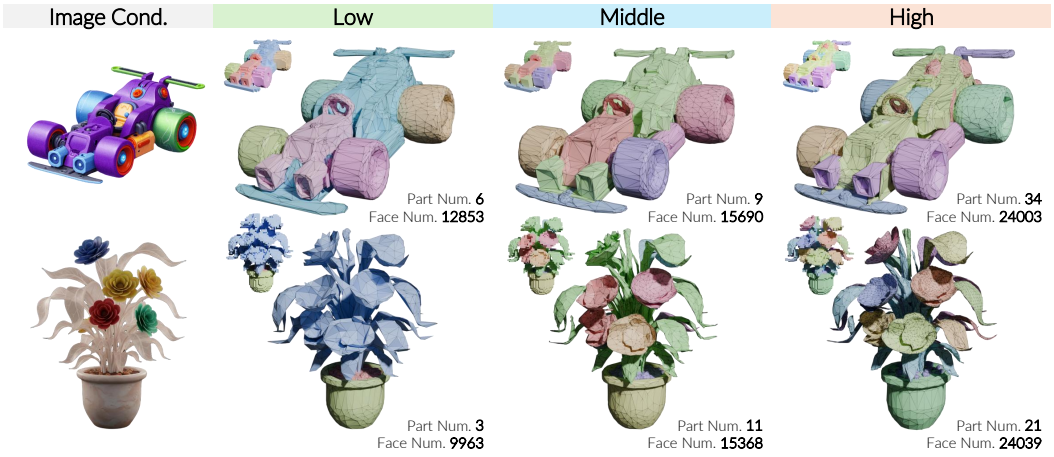

Figure 8: **Part-wise generation and refinement.** For each input image, the low, middle, and high settings use progressively more parts, producing denser meshes with finer geometric detail. Part counts and resulting face counts are reported below each mesh.

T-Flow reconstructs connectivity jointly over their union or per part followed by stitching (Appendix A.4.2). This assigns full latent capacity to each part, supporting higher vertex density and finer detail at the same latent grid resolution. Figure 8 compares different part counts, demonstrating that increasing the number of generated parts progressively improves geometric resolution and mesh quality.

Local refinement. For local refinement, we extract a selected region from an existing or generated mesh, regenerate its vertices with increased vertex capacity, and reintegrate it through topology-adaptive stitching.

A.4.2 Topology-Adaptive Mesh Editing

T-Flow re-samples connectivity conditioned on vertices edited from a generated or existing mesh, separating vertex manipulation from topology reconstruction.

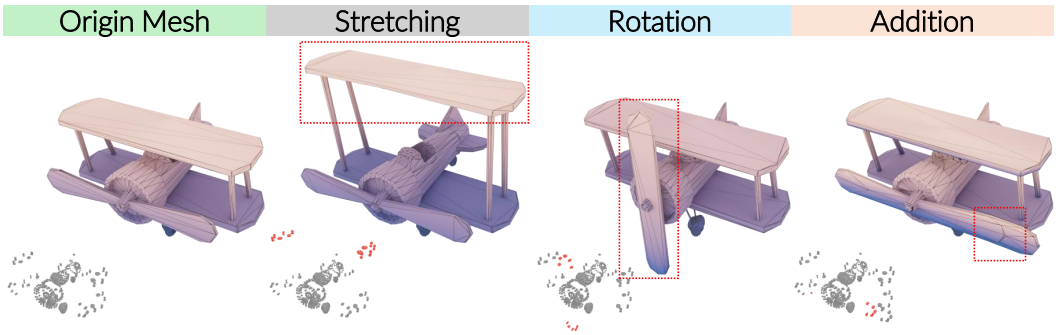

Figure 9: **Topology-Adaptive Mesh Editing.** Starting from the same airplane mesh, users stretch or rotate a selected region, or add vertices to extend the local geometry. T-Flow reconstructs connectivity conditioned on each modified vertex set. Red boxes mark the edited regions in the resulting meshes.

Mesh stitching. To combine multiple meshes or mesh parts, we first transform them into a shared coordinate frame through alignment and rescaling, and then take the union of their vertex sets. The T-Flow subsequently generates connectivity over the combined vertices, automatically producing coherent topology across the newly formed interfaces. As shown in Figure 4(a), the generated topology effectively bridges the junction regions while preserving the geometric structures of individual components.

Part transformation. For local deformation, a subset of vertices can be independently transformed through rotation or translation. Since the internal topology of both transformed and unchanged regions remains valid, only the transition area, where faces connect moved and unmoved vertices, requires topology adaptation. We identify these affected regions and regenerate their connectivity using the T-Flow conditioned on the updated vertex positions. Figure 4(b) illustrates that the originally invalidated connectivity caused by part transformation can be automatically repaired through our topology generation process.

Additional editing examples. Figure 9 shows stretching, rotation, and vertex addition on the same airplane mesh. These operations change vertex positions or vertex count, with T-Flow inferring the faces automatically.

A.5 Additional Generation Results

Vertex Number Controllability. Figure 10 shows the full set of vertex-count control results, extending Figure 6 to target budgets from 0.2K to 4K.

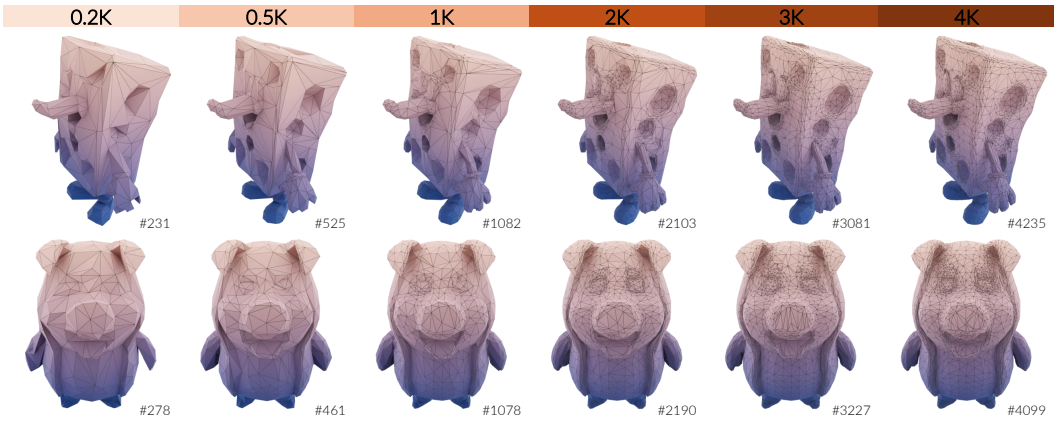

Figure 10: **Vertex number controlled generation across the full range of budgets.** The explicit vertex number condition of V-Flow enables controllable mesh resolution by specifying the target number of vertices. Given the same structure voxels, increasing the vertex budget from 0.2K to 4K produces progressively denser meshes with finer geometric details, while the topology flow automatically adapts connectivity to the generated vertex set and preserves the underlying shape structure.

Gallery. We further present a gallery of generated meshes obtained through both monolithic generation and the multi-part generation process in Fig. 11. The diverse examples, ranging from high-density geometric structures to stylized artistic shapes, demonstrate the effectiveness and generalization capability of our approach across various object categories and geometric complexities. Moreover, multi-part generation consistently produces meshes with finer geometric details and improved surface quality, highlighting the scalability advantage of our factorized representation for high-resolution mesh synthesis.

A.6 Discussion

Limitations. The current two-stage factorization also introduces several limitations. Since topology generation operates on the generated vertex set, geometric inaccuracies from the vertex stage cannot be corrected during topology synthesis. In addition, topology decoding considers vertex-pair relationships, resulting in quadratic complexity that remains manageable at current resolutions but may require efficient sparse candidate selection for substantially larger meshes. Finally, while LATO.2 focuses on geometry and connectivity generation, production-ready assets typically require additional attributes, such as UV coordinates, textures, and material properties.

Future work. Future work will explore tighter interactions between the vertex and topology stages through joint or iterative refinement, enabling topology feedback to improve vertex placement. We



Figure 11: **Generation gallery.** Our method can generate various types of meshes with one pass generation. With multi-part generation, LATO.2 is capable of generating meshes with high face count and extreme fine details.

also aim to extend the factorized formulation toward comprehensive generation of complete 3D assets with richer appearance attributes. We believe that modeling topology as a conditional latent distribution provides a promising foundation for scalable, controllable, and production-oriented explicit mesh generation.