FACEFLOW: NATIVE ARTIST-LIKE MESH GENERATION WITH FACE FLOW DIFFUSION

Anonymous authors
Paper under double-blind review



Figure 1: We present **FaceFlow**, diffusion-based framework for generating faces directly. Unlike conventional sorting strategies, we introduce explicit spatial alignment to resolve the assignment ambiguity in face diffusion arising from the variable cardinality, permutation invariance, and non-uniform spatial distribution of mesh faces.

ABSTRACT

Diffusion-based artist-like mesh generation has recently emerged as an important direction in 3D content creation. However, the discrete nature of mesh topology poses a fundamental challenge for continuous diffusion models. In this paper, we further investigate the parallel generation of complete discrete mesh sequences. To this end, we propose **FaceFlow**, a diffusion-based framework for artist-like mesh generation. FaceFlow employs a Face VAE to encode mesh faces into compact face latents and decode them into complete mesh sequences in a single step. However, conventional sorting strategies cannot resolve the assignment ambiguity caused by the variable cardinality and non-uniform spatial distribution of face latents. Consequently, the face diffusion model cannot reliably determine which target face latent each noisy latent should represent. To address this issue, we introduce a Point Sink Flow model that transports a dense noisy point cloud into a sparse set of face centers. These centers specify the 3D locations of the corresponding face latents and are injected into the Face DiT as positional embeddings, explicitly establishing spatial correspondences between noisy and target latents. This spatial guidance enables the Face DiT to learn a more accurate transformation toward the face-latent distribution, thereby improving the quality of the generated mesh sequences. Experiments on Toys4K demonstrate that FaceFlow achieves state-of-the-art performance in both mesh reconstruction and generation.

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

Unlike point clouds, voxel grids, and implicit representations, meshes encode both geometric information and explicit topological relationships. This combination makes them indispensable in applications such as games, film production, and 3D printing. However, current production pipelines still rely heavily on professional artists to construct 3D models manually, a process that requires specialized expertise and software and is both time-consuming and labor-intensive. Consequently, developing generative models capable of creating high-quality 3D assets has become an important research problem.

Generating artist-like meshes presents a central challenge: their discrete topological structure is difficult to model directly. Recent methods Hunyuan3D et al. (2025); Zhang et al. (2024); Li et al. (2025b); Xiang et al. (2025) address this challenge by converting meshes into continuous representations, such as signed distance fields (SDFs) or FlexiCubes Shen et al. (2023), and learning diffusion models over the resulting continuous latent spaces. However, this conversion discards the explicit topology of the original mesh. Meshes extracted from generated implicit representations using Marching Cubes Lorensen & Cline (1998) often exhibit irregular topology and therefore cannot be used directly in game or film production. Substantial retopology is required to transform these outputs into artist-like meshes suitable for downstream applications.

To model discrete topology, several works adopt the autoregressive (AR) paradigm from large language models (LLMs) to generate mesh sequences. Specifically, these methods Chen et al. (2025a;b); Weng et al. (2025); Hao et al. (2024); Weng et al. (2024); Siddiqui et al. (2024) discretize mesh vertices and faces, represent each face using nine discrete tokens, and arrange the resulting tokens into a complete mesh sequence according to a predefined ordering. A Transformer decoder then reconstructs the mesh sequence conditioned on an input signal. However, these methods face the fundamental limitations of token-by-token inference, including slow generation, error accumulation, and premature termination. To overcome these limitations, recent works Alliegro et al. (2023); Li et al. (2026c); Sun et al. (2026); Wang et al. (2026); Zhao et al. (2026); Long et al. (2026) have explored diffusion-based generation of both geometry and topology. Some of these methods decouple vertex generation from connectivity generation. However, the resulting vertex-connectivity matrices incur $O(N^2)$ complexity, posing a challenge for scaling up, while inferring triangular faces from predicted edges is both complex and ambiguous. In a different direction, MeshCraft He et al. (2025a) and MeshFlow Li et al. (2026c) attempt to use face diffusion model to parallelize the generation of mesh sequences. They leverage a predefined ordering strategy to solve the correspondence between face noises and target faces.

In this paper, we further investigate diffusion-based generation of face sequences for artist-like mesh synthesis. We observe that face-based mesh sequences exhibit variable cardinality, permutation invariance, and a highly non-uniform spatial distribution. A predefined ordering is therefore insufficient to resolve the ambiguities associated with variable cardinality and spatial arrangement. Consequently, the DiT struggles to align noisy face latents with their corresponding target latents, making it difficult to recover accurate face locations and limiting the quality of the generated artist-like meshes. To address this challenge, we propose **FaceFlow**, a novel diffusion framework for artist-like mesh generation that uses a face flow-matching model to generate complete mesh sequences in parallel. Specifically, we construct a Face VAE that encodes mesh faces into compact face latents and decodes each latent into the discrete and continuous coordinates of its three associated vertices. During diffusion, we establish explicit 3D correspondences between noisy and target face latents to resolve the ambiguities arising from variable cardinality, permutation invariance, and non-uniform spatial distribution. To this end, we use face centers, which characterize the spatial layout of the mesh faces, as positional embeddings for the corresponding face latents and inject them into the DiT as guidance. With this spatial guidance, the diffusion model can accurately determine the face location associated with each noisy latent, thereby reducing assignment ambiguity, accelerating convergence, and improving the accuracy of the generated face latents.

To generate face centers, we introduce a preliminary stage called the *Point Sink Flow* model, which consists of a VAE and a DiT. This model infers dense, uniformly distributed point latents from the conditioning input and transports points belonging to the same face toward its center. Specifically, the VAE encodes a dense, uniformly sampled point cloud together with its attributes, including surface normals and the coordinates of the corresponding face centers, into latent representations. It

then directly decodes the face center associated with each input point. To avoid relying on an explicit clustering algorithm, we formulate the predictions in a discrete space and obtain the final sparse set of face centers through deduplication. The DiT in this stage is a standard flow-matching model that extracts the corresponding point latents from image or point-cloud conditions. Experiments on the Toys4K dataset demonstrate that FaceFlow substantially outperforms state-of-the-art methods and provides an effective solution for the parallel generation of complete mesh sequences.

In summary, our contributions are as follows:

- We propose FaceFlow, a mesh-native framework for artist-like mesh generation based on flow matching. FaceFlow comprises a Point Sink Flow model and a Face Flow model, which jointly resolve the assignment ambiguity in face diffusion and enable the direct generation of high-quality mesh face sequences.
- We propose the Point Sink Flow model, a latent flow-matching model that transforms dense, uniformly distributed noise into a sparse, non-uniform set of face centers corresponding to the target face latents, thereby providing the face diffusion model with explicit length, order, and spatial-distribution information.
- We demonstrate that our method achieves state-of-the-art performance in the task of artist-like mesh generation and show that high-quality mesh sequences can be generated efficiently in a diffusion model.

2 Related work

Artist-like mesh generation has been explored primarily along two directions: autoregressive models and diffusion-based models. We review prior work from these two perspectives.

AR-based models. AR-based methods represent meshes as ordered face sequences and learn to generate them token by token. The pioneering work PolyGen Nash et al. (2020) employs two separate decoder-only Transformers for vertex generation and connectivity prediction. Following this paradigm, subsequent methods have improved mesh generation by designing new traversal orders and adjacency structures Chen et al. (2025b); Tang et al. (2025); Song et al. (2025), developing more effective tokenization strategies Wang et al. (2025b); Liu et al. (2025); Lionar et al. (2025); Weng et al. (2025), and introducing architectural modifications that support longer face-token sequences and faster inference. For example, MeshAnythingV2 Chen et al. (2025b) proposes Adjacent Mesh Tokenization (AMT), achieving a 46% reduction in sequence length. BPT Weng et al. (2025) further reduces mesh sequence length by approximately 75% through blocked and patchified tokenization. Meshtron Hao et al. (2024) adopts an hourglass Transformer that allocates most computation to face-level tokens, thereby enabling the modeling of substantially longer mesh sequences. In a different line of work, MeshGPT Siddiqui et al. (2024), MeshAnything Chen et al. (2025a), and PivotMesh Weng et al. (2024) use VQ-VAE to learn a discrete mesh vocabulary, which is then modeled by a decoder-only Transformer. Despite these advances, AR-based methods inherently generate face tokens sequentially, which leads to fragmented long-range dependencies and slow inference. This motivates diffusion-based approaches that can generate all face tokens in parallel.

Diffusion-based models. PolyDiff Alliegro et al. (2023) represents 3D meshes using a quantized triangle-soup structure and models the joint distribution of mesh vertices and faces in a single stage. SpaceMesh Shen et al. (2024) employs a point diffusion model to generate mesh vertices together with corresponding embeddings, which are subsequently used to construct vertex connectivity. Similarly, MeshFlow Sun et al. (2026) adopts equivariant optimal-transport flow matching to preserve key symmetries within the triangle-soup representation. However, these methods formulate diffusion in point space, which is difficult to scale and lacks compactness, making the diffusion learning problem challenging. Recent methods Li et al. (2026b) begin to expand the 3D native paradigms Zhang et al. (2023); Ren et al. (2024); Li et al. (2025a); Zhang et al. (2024); He et al. (2025b); Li et al. (2026a); Wu et al. (2026) toward native artist-like mesh generation. Inspired by SpaceMesh, MeshFlow Li et al. (2026c), LATO Zhao et al. (2026), LATO2 Long et al. (2026), and Nexus Wang et al. (2026) employ latent diffusion models to generate vertices and connectivity relationships in a decoupled manner. In contrast, MeshCraft He et al. (2025a) parallelizes the token-by-token generation process used in AR-based methods Lionar et al. (2025); Song et al. (2025); Wei et al. (2025); Zhao et al. (2025). It constructs a face VAE to encode face latents and directly decodes all face tokens from

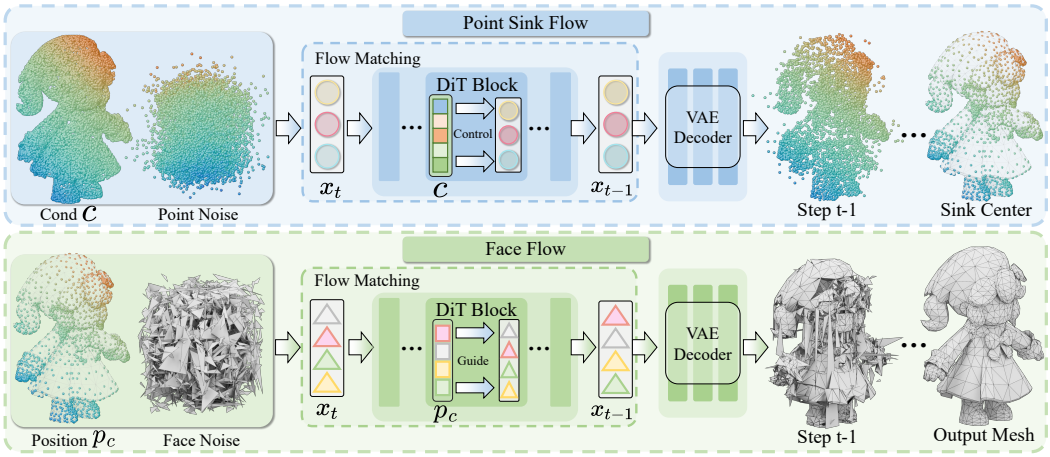

Figure 2: **Pipeline of FaceFlow**. Given a conditioning input (e.g., a point cloud), the Point Sink Flow model first predicts a sparse set of face centers. These centers are then injected into the Face Flow model to establish spatial correspondences between noisy and target face latents. Finally, the Face VAE decoder directly decodes the denoised face latents into the complete mesh.

them. However, MeshCraft establishes the correspondence between noise latents and face latents only through sorting, which makes it difficult for the DiT model to recover accurate face latents and consequently limits the quality of the generated artist-like meshes. In this paper, we observe that unordered and non-uniformly distributed face latents require precise 3D positional guidance for a DiT model to transform noise latents into their corresponding face latents. Based on this observation, we propose a Point Sink Flow model to estimate face positions and construct positional embeddings, which are injected into the DiT model to guide the transformation from noise latents to face latents.

3 METHOD

3.1 OVERVIEW

As illustrated in Fig. 2, we first employ the Point Sink Flow model to predict point latents from the conditioned point cloud c and decode them into sink centers, i.e., face centers. These predicted spatial locations are then injected into the FaceFlow model to guide the denoising of noisy face latents. Given the resulting clean face latents, the Face VAE decoder reconstructs the 3D coordinates of the three vertices associated with each latent, enabling parallel inference of the complete mesh sequence. We first describe in Sec. 3.2 how the Point Sink Flow model generates sparse face centers from a dense, uniformly sampled point cloud. We then present the architecture of the Face VAE and explain how face-center positional information is incorporated into the Face DiT in Sec. 3.3.

3.2 POINT SINK FLOW MODEL

The Point Sink Flow model aims to generate the face centers associated with a given conditioning input. Its output can therefore be viewed as a sparse, irregular, and variable-cardinality point set. Recovering such a point set from a dense noisy point cloud is challenging. A straightforward approach, similar to PDT Wang et al. (2025a), is to apply a point diffusion model to achieve clusters of the noisy input points. However, this strategy requires an additional clustering procedure. Moreover, prior generative studies Nichol et al. (2022); Zhang et al. (2023) have shown that performing diffusion directly in unordered and irregular point space often fails to produce high-quality results. We therefore adopt a latent flow-matching formulation and design a Point Sink VAE together with a DiT model to transform dense noisy point representations into a sparse set of face centers.

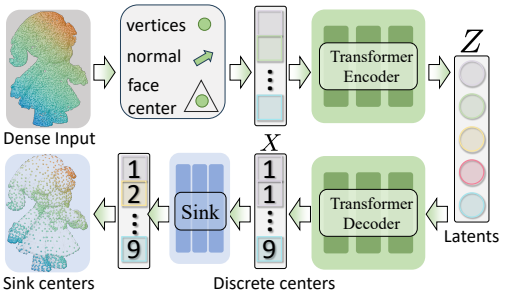

Figure 3: Detailed structure of Point Sink VAE. We aggregate the dense input point set into sparse face centers by predicting a discrete face-center location for each input point.

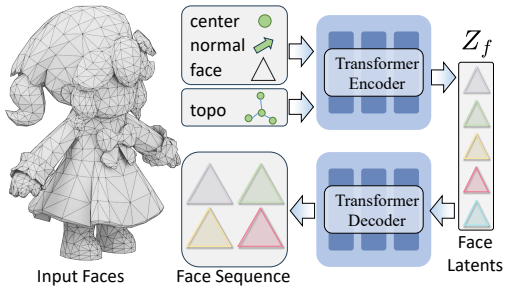

Figure 4: Detailed structure of Face VAE. We encode the center position and local topology of each face into face latents, which are then directly decoded into a discrete face sequence.

3.2.1 Point Sink VAE

The Point Sink VAE first encodes a uniformly sampled point cloud into a continuous and compact latent representation z. It then decodes the target face center associated with each input point. We adopt an architecture inspired by VecSet-based methods Zhang et al. (2023; 2024); Li et al. (2025a). Specifically, as shown in Fig. 3, the encoder captures the position and surface normal of each surface point, together with the position of its corresponding face center. We represent these inputs as $X = \mathrm{Concat}\big(\mathrm{PE}(V), V_n, \mathrm{PE}(V_c)\big)$, where V, V_n, and V_c denote the point positions, surface normals, and corresponding face-center positions, respectively, and $\mathrm{PE}(\cdot)$ denotes positional encoding. We then apply a multilayer perceptron (MLP) to obtain high-dimensional point features. Following MeshFlow Li et al. (2026c), we use point shuffling to construct the query set q. This operation performs implicit downsampling and thereby reduces the latent sequence length. Finally, a cross-attention (CA) layer followed by eight self-attention layers is applied to obtain the latent representation. The overall encoding process is formulated as:

$$Z = \mathcal{E}(\mathrm{X}) = SelfAttn^{8}(CrossAttn(q, X)) \tag{1}$$

The decoder adopts a hierarchical architecture to produce point-level features, followed by separate discrete and continuous prediction heads for estimating the face center associated with each input point. It first applies 16 self-attention layers to obtain query-level features. These features are then re-shuffled and processed by four additional self-attention layers to produce the point-level representation $\hat{\mathbf{X}}$. The re-shuffling operation reduces the feature dimensionality, thereby lowering the computational cost of the subsequent self-attention layers. The complete decoding process is formulated as:

$$\hat{\mathbf{X}} = \mathcal{D}(Z) = SelfAttn^{4}(\mathcal{S}(SelfAttn^{16}(Z))) \tag{2}$$

where $\mathcal{S}(\cdot)$ denotes the re-shuffling operation. The resulting features $\hat{\mathbf{X}}$ are fed into the discrete and continuous prediction heads to estimate the corresponding face centers. Further discussion of these two prediction heads are provided in Sec. A.2.1 of the appendix.

The Point Sink VAE is trained with three loss terms: a KL-regularization loss $\mathcal{L}\mathrm{kl}$, an ℓ_1 loss $\mathcal{L}\ell_1$ for supervising the continuous face-center predictions, and a cross-entropy (CE) loss $\mathcal{L}\mathrm{ce}$ for supervising the discrete face-center predictions. The overall reconstruction objective is defined as:

$$\mathcal{L}_{rec} = \lambda_{l1}\mathcal{L}_{l1} + \lambda_{ce}\mathcal{L}_{ce} + \lambda_{kl}\mathcal{L}_{kl} \tag{3}$$

3.2.2 Point Sink DiT

The Point Sink DiT adopts the widely used flow-matching framework Liu et al. (2022) with velocity prediction for conditional generation. Its backbone consists of a stack of diffusion Transformer (DiT) blocks. For point-cloud conditioning, we voxelize the input point cloud at a resolution of 64^3 and use the resulting voxel coordinates to construct 3D rotary positional embeddings (3D RoPE), which accelerates convergence during training. In this diffusion model, the velocity v can be approximated with a neural network v_θ by minimizing the conditional flow matching objective:

$$\mathbb{E}_{t,x_0,x_1,c} \left\| v_\theta(x, t, c) - (x_1 - x_0) \right\|_2^2 . \tag{4}$$

where c is the condition input, e.g., an image or a point cloud.

3.3 FACE FLOW MODEL

The FaceFlow model aims to generate a complete mesh sequence from a given conditioning input, thereby producing an artist-like mesh. Unlike approaches that decouple vertex and connectivity generation, directly predicting the complete mesh sequence avoids the ambiguous process of inferring triangular faces from predicted connectivity. Instead, FaceFlow directly generates all triangular faces that constitute the final mesh, providing a more mesh-native formulation of the generation process. Unlike PolyDiff and MeshFlow, which apply point diffusion directly to triangle-soup representations, we introduce a Face VAE that encodes mesh faces into compact face latents, substantially reducing the token sequence length. Furthermore, we replace conventional sequence-sorting strategies with explicit 3D positional embeddings to resolve the assignment ambiguity between noisy face latents and their target counterparts.

3.3.1 FACE VAE

A triangular mesh $\mathcal{M} = (V, F)$ with N faces can be represented as a mesh sequence $\mathcal{M} := (f_1, f_2, \ldots, f_N)$, where each face $f_i \in \mathbb{R}^{3\times3}$ comprises the 3D coordinates of its three vertices. The resulting mesh sequence therefore has a length of $9N$. The Face VAE aims to encode each face feature into compact face latents and reconstruct the complete mesh sequence from them.

As illustrated in Fig. 4, we adopt an architecture similar to that of Point Sink VAE. Specifically, we first construct a query $q = \mathrm{Concat}\left(\mathrm{PE}(P_c), P_n, F_{\mathrm{pos}}\right)$, where P_c and P_n denote the face centers and face normals, respectively. The per-face positional feature F_{pos} is constructed from the three vertices of each face: $F_{\mathrm{pos}} = \mathrm{Concat}\left(\mathrm{PE}(v_1), \mathrm{PE}(v_2), \mathrm{PE}(v_3)\right)$. We then apply four self-attention layers to F_{pos} to capture interactions among faces and obtain the topology-aware features F_{topo}. Similar to the Point Sink VAE, a cross-attention layer followed by eight self-attention layers produces the final face latents. The encoding process is formulated as:

$$Z_f = \mathrm{SelfAttn}^8\left(\mathrm{CrossAttn}\left(\phi(q), F_{\mathrm{topo}}\right)\right) \tag{5}$$

where $\phi(\cdot)$ denotes a learnable feature projection.

The Face VAE decoder adopts the same hierarchical architecture as the Point Sink VAE decoder. The key difference is that the Face VAE decoder first extracts face-level features and subsequently expands them into vertex-level features. The complete decoding process can be formulated as:

$$F_V = \mathcal{D}(Z_f) = SelfAttn^4(\mathcal{S}(SelfAttn^{16}(Z_f))) \tag{6}$$

Finally, a discrete prediction head and a continuous head map the vertex-level features to quantized vertex coordinates and continuous vertex coordinates. The quantized vertex coordinates are used to yield the complete discrete mesh sequence. The loss function is similar to equation 3 used in the Point Sink VAE.

3.3.2 FACE DIT

Face latents exhibit variable cardinality, permutation invariance, and a highly non-uniform spatial distribution. Consequently, directly applying a conventional diffusion model, such as the Point Sink DiT, makes it difficult to infer accurate face latents from the conditioning input. Without explicit correspondence information, the model cannot determine which noisy latent should be assigned to each target face latent. This assignment ambiguity greatly expands the space of possible solutions and hinders convergence. Although imposing a predefined ordering can remove permutation ambiguity and reduce the search space, it does not adequately capture the variable cardinality or non-uniform spatial distribution of the face-latent set.

We therefore introduce explicit spatial constraints into the Face DiT to align noisy and target face latents. Specifically, each noisy latent is associated with the spatial location of its corresponding face, represented by the face center, allowing the network to distinguish otherwise exchangeable noise tokens. The set of face centers specifies both the number and spatial arrangement of the target faces, thereby capturing the unordered, variable-cardinality, and non-uniform nature of the face-latent set. These spatial locations can be encoded using either absolute positional embeddings

Table 1: Quantitative comparison for VAE reconstruction.

Method	CD-L1 ($\times 10^3$) ↓	CD-L2 ($\times 10^4$) ↓	HD ($\times 10^2$) ↓	NC (%) ↑
MeshGPT Siddiqui et al. (2024)	10.79	3.68	4.90	0.899
EdgeRunner Tang et al. (2025)	16.79	13.82	7.35	0.868
LATO Zhao et al. (2026)	7.71	0.90	3.23	0.923
LATO 2 Long et al. (2026)*	7.87	1.08	3.38	0.915
MeshFlow Li et al. (2026c)	8.61	1.02	2.99	0.917
Triflow Li et al. (2026b)	7.39	0.79	2.58	0.935
FaceFlow (Ours)	**5.58**	**0.55**	**2.46**	**0.948**

Table 2: Quantitative comparison of point-conditioned artist-mesh generation on the Toys4K dataset.

Method	CD-L1 ($\times 10^3$) ↓	CD-L2 ($\times 10^3$) ↓	HD ($\times 10^2$) ↓	NC (%) ↑
MeshAnythingV2 Chen et al. (2025b)	25.741	2.570	12.198	80.48
TreeMeshGPT Lionar et al. (2025)	52.734	12.339	16.444	78.83
BPT Weng et al. (2025)	9.456	0.330	3.867	90.44
FastMesh-V1K Kim et al. (2026)	9.355	0.147	4.369	88.44
FastMesh-V4K Kim et al. (2026)	9.568	0.145	4.622	88.13
MeshFlow Li et al. (2026c)	9.812	0.126	3.371	88.13
FaceFlow (Ours)	**7.735**	**0.089**	**2.740**	**92.85**

or 3D rotary positional embeddings (RoPE). Accordingly, given the face-center condition $\mathbf{p}_c$, the flow-matching objective of the Face DiT is formulated as:

$$\mathbb{E}_{t,x0,x_1,c,p_c} \left| v\theta(PE(x, p_c), t, c) - (x_1 - x_0) \right|_2^2 . \tag{7}$$

4 Experiments

4.1 Implementation Details and Metrics

We curate a dataset of approximately 230K high-quality, artist-like meshes from Objaverse Deitke et al. (2023b) and Objaverse-XL Deitke et al. (2023a), retaining only meshes with fewer than 5K faces. Both the Point Sink Flow and FaceFlow models employ DiT backbones comprising 24 Transformer blocks with a hidden dimension of 1,536, each containing approximately 960M parameters. For point cloud condition, we use the pretrained VAE encoder from PoseMaster Yan et al. (2026) to extract geometry features of size 4096×1024. This VAE encoder is jointly fine-tuned with the DiT models during training. All VAEs and DiT models are optimized using AdamW with an initial learning rate of 1×10^{-4} and a cosine-annealing schedule. For a fair and reliable evaluation, we construct the test set from Toys4K Stojanov et al. (2021). We retain meshes with fewer than 5K faces and randomly select two objects from each of the 105 categories, yielding 210 test samples in total. Following prior work Li et al. (2026c); Zhao et al. (2026); Long et al. (2026), we evaluate reconstruction and generation quality using Chamfer Distance (CD), Hausdorff Distance (HD), and Normal Consistency (NC) by normalizing both the generated and ground-truth shapes and sampling 5,000 points for metric computation.

4.2 Evaluation on Mesh Reconstruction

We first evaluate the mesh reconstruction performance of our Face VAE against methods spanning several representation paradigms. These baselines include MeshGPT Siddiqui et al. (2024), which employs a VQ-VAE; EdgeRunner Tang et al. (2025), which adopts an autoregressive VAE; LATO Zhao et al. (2026), LATO 2 Long et al. (2026), and MeshFlow Li et al. (2026c), which reconstruct vertices and connectivity separately; and TriFlow Li et al. (2026b), which represents meshes using a nearest-point field. Because MeshCraft He et al. (2025a) has not released its source code or pretrained weights, we are unable to include it in the direct comparison.

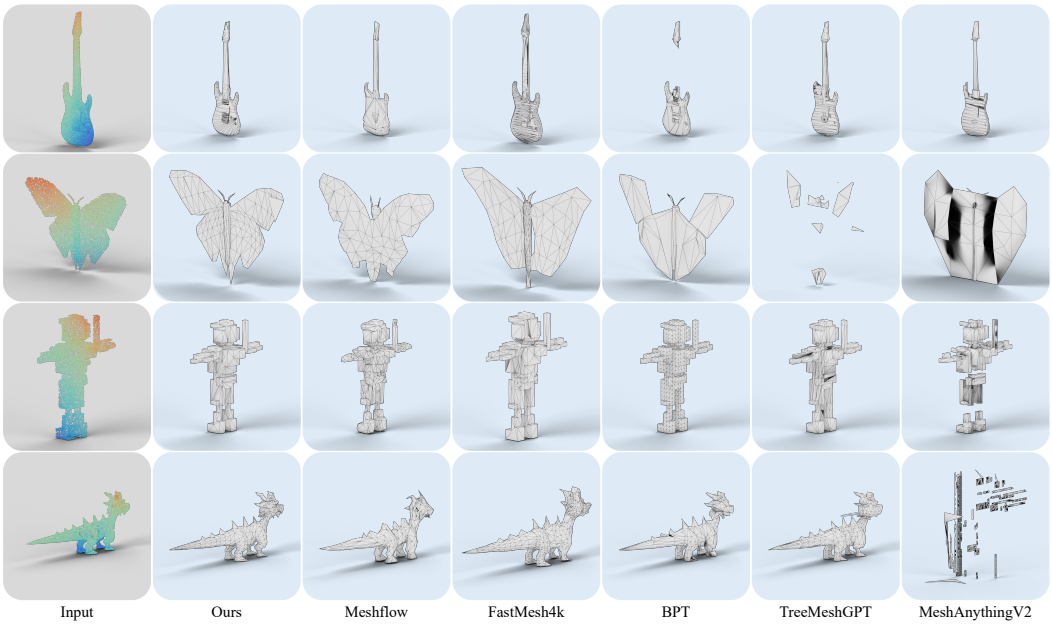

Figure 5: The qualitative comparison of point-conditioned artist-mesh generation.

As shown in Table 1, our method substantially outperforms the baselines in both Chamfer Distance (CD) and Hausdorff Distance (HD), demonstrating its superior geometric reconstruction accuracy. The improvement in Normal Consistency (NC) further indicates that our reconstructions better preserve local surface orientations and structures. We attribute these gains to the mesh-native design of our Face VAE, which directly encodes face-level features and reconstructs the three vertices associated with each face. This formulation models mesh geometry and topology more directly than approaches that reconstruct vertices and connectivity separately or rely on topology-aware implicit fields.

4.3 Evaluation on Mesh Generation

Following the evaluation protocol of MeshFlow Li et al. (2026c), we evaluate shape-conditioned artist-like mesh generation against several state-of-the-art methods. The baselines include the AR-based methods MeshAnythingV2 Chen et al. (2025b), TreeMeshGPT Lionar et al. (2025), and BPT Weng et al. (2025), as well as the diffusion-based methods FastMesh Kim et al. (2026) and MeshFlow Li et al. (2026c).

As shown in Table 2, FaceFlow achieves the best performance across all evaluation metrics. Its superior CD and HD scores demonstrate that the generated meshes accurately recover the input geometry, while the improvement in Normal Consistency (NC) indicates better preservation of local surface orientations and mesh structure. Unlike decoupled approaches such as FastMesh and MeshFlow, which predict vertices and connectivity separately and require explicit normal prediction to determine face orientation, FaceFlow directly generates complete mesh sequences. It therefore recovers consistently oriented faces without explicitly predicting surface normals.

We provide qualitative comparisons in Fig. 5 to further demonstrate the topological advantages of FaceFlow. Compared with AR-based methods, FaceFlow generates more complete meshes with fewer missing regions. Compared with FastMesh and MeshFlow, it produces cleaner connectivity and more regular edge flow that more closely resemble artist-designed topology. We attribute this advantage to the Point Sink Flow model, which aggregates surface points belonging to the same face according to their geometry before reconstructing the corresponding triangular face. In contrast, the formulations of FastMesh and MeshFlow may generate points that do not correspond to actual mesh vertices, leading to ambiguous connectivity and irregular edge flow.

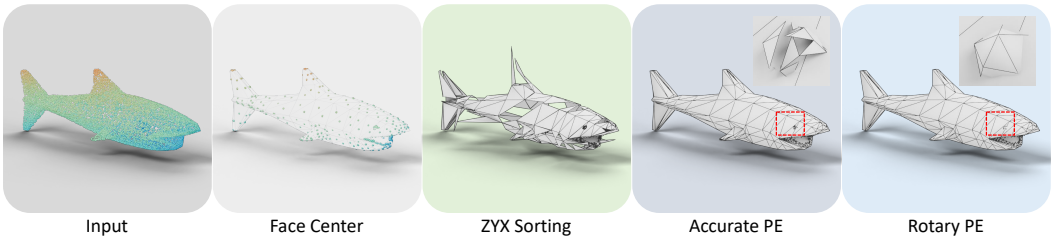

Figure 6: Qualitative results of different latent alignment strategies for face diffusion.

Table 3: Quantitative comparison of different latent alignment strategies for face diffusion.

Alignment	CD-L1 ($\times 10^3$) ↓	CD-L2 ($\times 10^3$) ↓	HD ($\times 10^2$) ↓	NC (%) ↑
ZYX Sorting	11.8	0.337	8.24	88.13
Absolute PE	7.77	0.101	3.38	92.10
Rotary PE	7.73	0.089	2.740	92.85

4.4 Ablation Study

Impact of latent alignment.

We first compare a zyx-sorting strategy used in MeshCraft He et al. (2025a) with our explicit position alignment for Face DiT training. We further evaluate two mechanisms for incorporating spatial information: absolute positional embeddings and rotary positional embeddings (RoPE). For a controlled comparison, we train the sorting-based baseline using the same DiT architecture and provide the ground-truth number of faces at inference to determine the length of the initial noise sequence.

As shown in Table 3, explicit positional conditioning substantially outperforms the sorting-based strategy, demonstrating its effectiveness in resolving the assignment ambiguity caused by the variable cardinality and non-uniform spatial distribution of face latents. Absolute positional embeddings and RoPE achieve comparable CD scores, whereas RoPE yields a more pronounced improvement in HD. This lower HD suggests that modeling relative spatial relationships improves robustness to localized geometric errors.

The qualitative results in Fig. 6 visualize the face centers predicted by the Point Sink Flow model together with the meshes generated using the three positional strategies. The sorting-based model frequently produces holes in regions with non-uniform face distributions, even when provided with the ground-truth face count. In contrast, explicit positional conditioning uses the predicted face centers to convey the cardinality and spatial arrangement of the target face latents, enabling more accurate recovery of face locations. The zoomed-in views further show that RoPE produces more plausible local geometry and is more robust to inaccuracies in the predicted face centers. These results also validate the effectiveness of the Point Sink Flow model in predicting face centers that provide reliable spatial guidance for subsequent mesh generation.

5 Conclusion

In this paper, we presented FaceFlow, a diffusion-based framework for generating discrete mesh sequences in parallel. We observed that existing sorting-based strategies cannot fully resolve the assignment ambiguity caused by the variable cardinality and non-uniform spatial distribution of mesh faces. To address this challenge, we introduced the Point Sink Flow model, which predicts face centers from the conditioned input to specify the cardinality, token-wise correspondence, and spatial arrangement of the target face latents. These face centers are injected into the Face DiT as positional embeddings, explicitly establishing spatial correspondences between noisy and target face latents. This guidance enables the diffusion model to determine the face location associated with each noisy latent, leading to more accurate and complete mesh generation. Extensive experiments demonstrate that FaceFlow outperforms existing artist-like mesh generation methods in both reconstruction fidelity and conditional generation quality.

AI USE STATEMENT

Generative AI tools, primarily ChatGPT and Codex, were used to assist with language editing and writing refinement. We also used generative AI to support the qualitative assessment of geometric topology and the development of rendering scripts. All AI-assisted outputs were reviewed and verified by the authors. The authors are solely responsible for the methodology, equations, experimental results, analysis, and conclusions presented in this work.

REFERENCES

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

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

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.

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*, 2025a.

Xianglong He, Zi-Xin Zou, Chia-Hao Chen, Yuan-Chen Guo, Ding Liang, Chun Yuan, Wanli Ouyang, Yan-Pei Cao, and Yangguang Li. Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14822–14833. IEEE, 2025b.

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.

Dong-Yang Li, Wang Zhao, Yuxin Chen, Wenbo Hu, Meng-Hao Guo, Fang-Lue Zhang, Ying Shan, and Shi-Min Hu. Pixal3d: Pixel-aligned 3d generation from images. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12, 2026a.

Haoxuan Li, Ziya Erkoç, Daniele Sirigatti, Vladislav Rosov, Lei Li, Angela Dai, and Matthias Nießner. Triflow: Generating artist-like 3d mesh topology via nearest-vertex vector fields. *arXiv preprint arXiv:2606.20131*, 2026b.

Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman3d: High-fidelity mesh generation with 3d native diffusion and interactive geometry refiner. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5307–5317. IEEE, 2025a.

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. *arXiv preprint arXiv:2606.04621*, 2026c.

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*, 2025b.

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.

Jian Liu, Haohan Weng, Biwen Lei, Xianghui Yang, Zibo Zhao, Zhuo Chen, Song Guo, Tao Han, and Chunchao Guo. Freemesh: Boosting mesh generation with coordinates merging. *arXiv preprint arXiv:2505.13573*, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

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.

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.

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.

Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4209–4219. IEEE, 2024.

Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Transactions on Graphics (ToG)*, 42(4):1–16, 2023.

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, Zibo Zhao, Haohan Weng, Jingbo Zeng, Rongfei Jia, and Shenghua Gao. Mesh silksong: Auto-regressive mesh generation as weaving silk. *arXiv e-prints*, pp. arXiv–2507, 2025.

Stefan Stojanov, Anh Thai, and James M Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1798–1808. IEEE, 2021.

Qi Sun, Kiyohiro Nakayama, Jing Nathan Yan, Qixing Huang, Alexander Rush, Leonidas Guibas, Gordon Wetzstein, Jing Liao, and Guandao Yang. Meshflow: Mesh generation with equivariant flow matching. *arXiv preprint arXiv:2606.23489*, 2026.

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.

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.

Jionghao Wang, Cheng Lin, Yuan Liu, Rui Xu, Zhiyang Dou, Xiaoxiao Long, Haoxiang Guo, Taku Komura, Wenping Wang, and Xin Li. Pdt: Point distribution transformation with diffusion models. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2025a.

Yuxuan Wang, Xuanyu Yi, Haohan Weng, Qingshan Xu, Xiaokang Wei, Xianghui Yang, Chunchao Guo, Long Chen, and Hanwang Zhang. Nautilus: Locality-aware autoencoder for scalable mesh generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–10. IEEE, 2025b.

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, 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480. IEEE, 2025.

Hongyu Yan, Kunming Luo, Weiyu Li, Kaiyi Zhang, Yixun Liang, Jingwei Huang, Chunchao Guo, and Ping Tan. Posemaster: A unified 3d native framework for stylized pose generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34292–34302, 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.

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, 2025.

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.

Table 4: Quantitative comparison between continuous and discrete face centers generated by the Point Sink Flow model when used as positional embeddings.

Face Center	CD-L1 ($\times 10^3$) ↓	CD-L2 ($\times 10^3$) ↓	HD ($\times 10^2$) ↓	NC (%) ↑
Continuous	8.94	0.154	4.164	91.56
Discrete	7.73	0.089	2.740	92.85

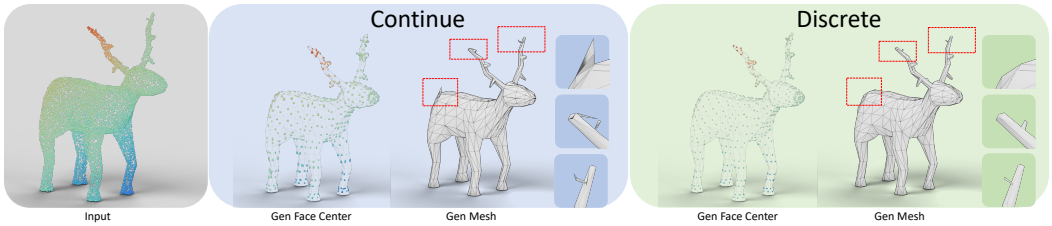

Figure 7: Qualitative comparison between continuous and discrete face centers generated by the Point Sink Flow model when used as positional embeddings. Discrete face centers produce more plausible results.

A Appendix

A.1 Additional implementation details

We normalize all meshes to the coordinate range $(-1, 1)$ and discard those containing fewer than 10 faces. To construct discrete mesh sequences, we quantize the vertex coordinates along each axis into 512 levels. Following prior AR-based methods Weng et al. (2025), we merge vertices that collapse to identical coordinates after quantization and remove degenerate and duplicate faces.

Because the current model is trained on meshes with fewer than 5,000 faces, we sample 10,240 surface points as input to the Point Sink VAE without additional preprocessing-level downsampling. To ensure coverage of every face, we first sample at least one point from each face and then uniformly sample the remaining points from the mesh surface. For point-cloud conditioning, we uniformly sample 81,920 points from each mesh surface together with their corresponding surface normals.

For both the Face VAE and Point Sink VAE, the discrete prediction head outputs 3×512 logits for each predicted 3D coordinate, corresponding to categorical distributions over 512 quantization levels along the x, y, and z axes. The continuous prediction head directly regresses the corresponding three floating-point coordinates. The weights of the KL-regularization, ℓ_1, and cross-entropy losses are set to $\lambda_{\text{kl}} = 10^{-3}$, $\lambda_{\ell_1} = 1$, and $\lambda_{\text{ce}} = 1$, respectively. To accommodate variable-length face sequences, we use a per-GPU batch size of 1 when training both the VAEs and DiTs. All models are trained on 32 NVIDIA A100 GPUs using BF16 mixed precision.

A.2 Additional Experiments

A.2.1 Impact of discrete and continuous supervision

Why discrete outputs?

We adopt discrete outputs in both the Point Sink VAE and the Face VAE to recover the final geometry. For the Point Sink VAE, quantized predictions allow dense sink-center outputs to be converted into a sparse set of face centers through direct deduplication. For the Face VAE, discrete predictions promote consistent coordinates for vertices and edges shared across adjacent faces. Although an alternative is to regress continuous coordinates and quantize them afterward, we find that directly predicting discrete coordinates is more stable. The continuous coordinate space demands substantially higher precision, making small regression errors unavoidable and the subsequent quantization process less reliable.

Table 5: Quantitative comparison of models trained with and without continuous supervision with different noise levels. Continuous supervision improves the spatial awareness of the CE head, leading to discrete predictions that are closer to the ground-truth coordinates.

Continuous	t	Point VAE			Face VAE		
		CD-$\ell_1 \downarrow$	CD-$\ell_2 \downarrow$	HD $\downarrow$	CD-$\ell_1 \downarrow$	CD-$\ell_2 \downarrow$	HD $\downarrow$
✓	0.00	1.72×10^{-2}	3.23×10^{-6}	0.00228	1.81×10^{-5}	1.42×10^{-7}	0.00106
	0.05	1.74×10^{-2}	3.47×10^{-6}	0.00417	4.18×10^{-5}	3.22×10^{-7}	0.00220
	0.10	2.41×10^{-2}	4.44×10^{-5}	0.1579	2.67×10^{-4}	2.50×10^{-6}	0.0120
×	0.00	1.76×10^{-2}	1.23×10^{-5}	0.02189	2.43×10^{-4}	2.50×10^{-6}	0.00373
	0.05	1.81×10^{-2}	2.18×10^{-5}	0.03439	5.73×10^{-4}	6.44×10^{-6}	0.0122
	0.10	2.34×10^{-2}	1.03×10^{-4}	0.1933	4.51×10^{-3}	1.04×10^{-4}	0.0424

We first quantitatively compare the use of continuous and discrete face centers for positional conditioning. As shown in Table 4, discrete face centers consistently achieve better performance across the evaluation metrics. Figure 7 visualizes the discrete and continuous face centers predicted by the Point Sink Flow model. It further compares the meshes generated using RoPE under two face-center extraction schemes: (i) directly deduplicating the discrete predictions and (ii) quantizing and then deduplicating the continuous predictions. The zoomed-in views show that the latter scheme produces additional invalid faces in some local regions. This occurs because small errors in the continuous predictions can map centers that should coincide into different quantization bins. Consequently, a region that should contain a single sink center may retain two or more centers after quantization and deduplication, causing the Face DiT to generate redundant or conflicting faces.

Our positional augmentation alleviates moderate errors of this kind, explaining why the continuous predictions still produce reasonable results in most regions. However, when too many sink centers accumulate within a small neighborhood, the perturbation exceeds the range covered by the augmentation strategy. Directly predicting discrete face centers avoids the additional error introduced by post-quantization and therefore produces cleaner and more reliable mesh geometry.

Why continuous supervision?

In addition to the discrete head supervised by cross-entropy (CE) loss, we introduce a continuous head that regresses 3D coordinates under an ℓ_1 loss. CE treats coordinate quantization as a classification problem and does not encode the spatial relationships among quantization bins. For example, it does not recognize that bin 100 is closer to bin 99 than bin 400 is. Consequently, an incorrect classification may select a bin far from the ground truth, producing isolated floater points. In contrast, the ℓ_1 objective directly penalizes coordinate distance and therefore provides spatially aware supervision. We combine the two objectives so that the continuous head regularizes the shared decoder features used by the discrete head.

To evaluate the effect of continuous supervision, we compare VAEs trained with and without the continuous head. Following the forward noising process of diffusion models, we perturb the latent representations at different timesteps before decoding them. This experiment evaluates the robustness of the VAE decoder to the latent perturbations encountered during generation. We conduct the study on both the Point Sink VAE and the Face VAE and report the Chamfer Distance (CD) and Hausdorff Distance (HD) between the decoded coordinates and their ground-truth counterparts in Table 5. As the noise level increases, both CD and HD deteriorate substantially faster without the continuous head. These results demonstrate that continuous supervision regularizes the shared decoder features, improves discrete coordinate prediction, and increases robustness to latent perturbations.

A.2.2 Impact of position augmentation

During Face DiT training, we extract face centers from the ground-truth meshes and use them as positional inputs to RoPE. At inference time, however, the face centers predicted by the Point Sink Flow model inevitably deviate from their ground-truth counterparts, creating a distribution gap between the spatial conditions observed during training and inference. We identify three primary forms

Table 6: Quantitative comparison of position augmentation. Using position augmentation can enhance the model's robustness.

Augmentation	CD-L1 ($\times 10^3$) ↓	CD-L2 ($\times 10^3$) ↓	HD ($\times 10^2$) ↓	NC (%) ↑
×	10.85	0.218	6.601	89.20
✓	7.73	0.089	2.740	92.85

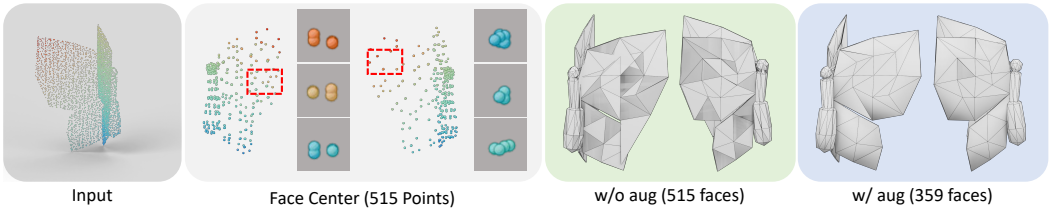

Figure 8: Qualitative comparison of position augmentation. Positional augmentation improves the model's robustness to the face centers predicted by the Point Sink Flow model.

of prediction error. First, a predicted point may lie elsewhere on the corresponding face rather than precisely at its center. Second, multiple predicted centers may correspond to the same face. Third, a predicted center may be displaced from the face surface along either direction of its normal. The zoomed-in visualizations in Fig. 8 show the local distributions of the sink centers generated by the Point Sink Flow model. In challenging regions, multiple predicted centers may be concentrated within a small neighborhood, collectively exhibiting the three error patterns described above.

To improve robustness to these errors, we introduce targeted positional augmentations during Face DiT training. Specifically, we perturb the face centers supplied to RoPE according to the three observed error patterns while preserving the correspondence between each augmented center and the latent of its original face. This strategy exposes the Face DiT to the positional variations encountered at inference time while maintaining consistent supervision for the target face latents.

As shown in Table 6, the proposed augmentations substantially improve generation quality, with the most pronounced gain observed in HD. This improvement indicates that positional augmentation effectively suppresses spurious faces and localized geometric outliers. The qualitative comparisons in Fig. 8 further support this observation. Without augmentation, the model generates one face for each input face center. Consequently, multiple distinct faces may be generated within the same local region, leading to overlapping geometry or faces that are inconsistent with the input shape. With augmentation, both the number of generated faces and the rendered results indicate that the model learns to determine whether nearby centers belong to the same target face. As a result, centers associated with the same face are mapped to consistent face geometry, substantially reducing conflicting and overlapping faces.

A.2.3 Extra Visualized comparison

In this section, we compare against methods that use different modalities as conditioning inputs, including TriFlow Li et al. (2026b) and LATO2 Long et al. (2026). Because the conditioning modalities differ, we perform a qualitative comparison of topological quality based on visual results. LATO2 uses images as conditioning; therefore, we render the 210 Toys4K test objects as grayscale images and use them as inputs. TriFlow uses an SDF as conditioning, so we directly provide each complete mesh object and extract its SDF values as the input. Both methods require the target number of faces or vertices during inference. We therefore evaluate two settings: one using the ground-truth vertex and face counts, and the other using the default values specified in their inference scripts. As shown in Fig. 9, even when provided with the ground-truth vertex and face counts, the topologies generated by TriFlow and LATO2 remain irregular and cluttered, falling short of the quality expected of artist-like meshes. The artifacts become more pronounced when the specified face or vertex counts differ from those of the ground-truth mesh. In contrast, FaceFlow produces substantially cleaner and more coherent topology, with results that more closely resemble artist-like meshes.

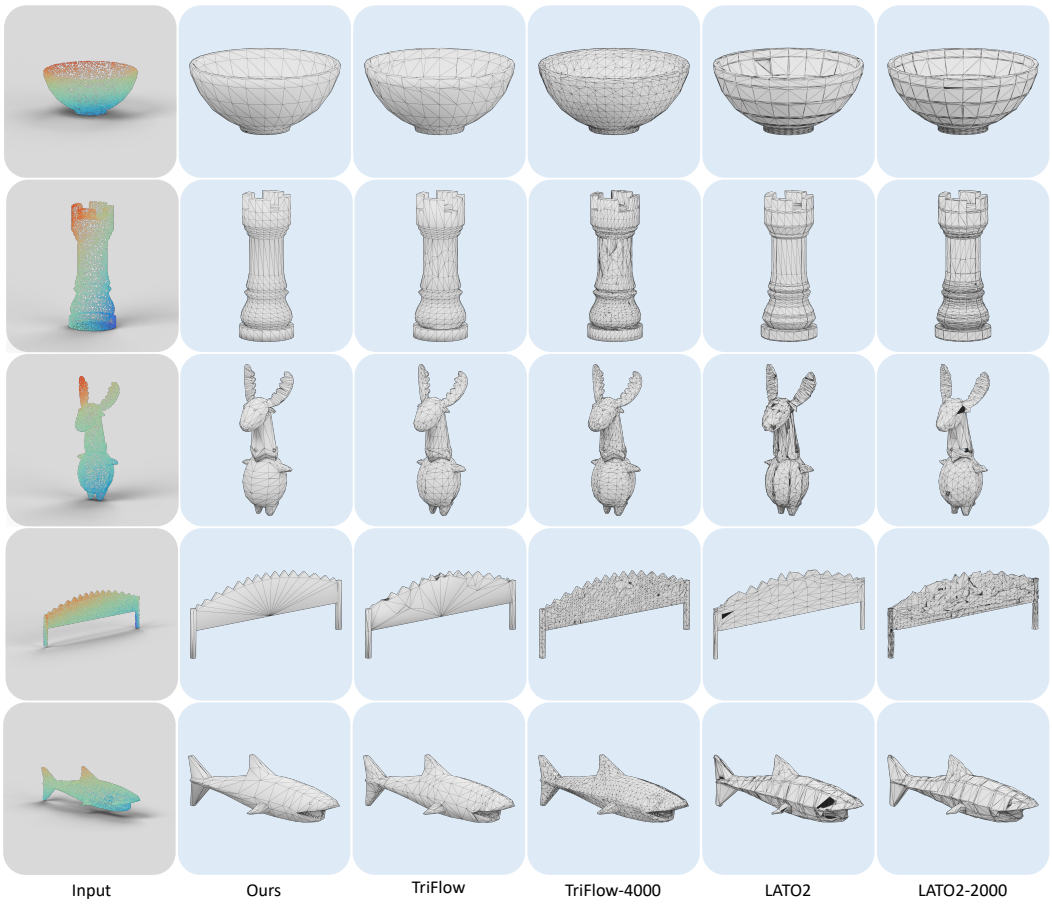

Figure 9: Additional qualitative comparison with TriFlow Li et al. (2026b) and LATO2 Long et al. (2026). FaceFlow produces substantially cleaner and more coherent topology, with results that more closely resemble artist-like meshes.

A.2.4 Topology Quality Comparison

Our evaluation metrics provide only a partial assessment of topological quality. Meshes representing the same geometry may follow different edge-flow and topology conventions depending on the artist, making a fair quantitative comparison of generated topology difficult. As complementary quantitative references, we measure the relative difference in the number of boundary edges and the similarity between the triangle-area distributions of the generated and ground-truth meshes. We also use a visual score produced by a vision-language model (VLM) from wireframe renderings.

A higher boundary score indicates that the generated mesh has a boundary structure similar to that of the ground-truth mesh, whereas a lower score indicates a substantial excess or deficit of boundary edges. Such discrepancies may reflect missing surfaces, unintended holes, incomplete components, or excessive open boundaries. The similarity between triangle-area distributions provides an additional indication of whether the generated mesh is over-tessellated or overly simplified. Wireframe renderings expose the generated connectivity directly; therefore, a higher VLM score indicates more plausible edge flow and topology that more closely resembles artist-authored meshes. We use GPT-5.6-Sol as the VLM evaluator. Consistent with the preceding experiments, we use Toys4K as the test set. We compute the boundary and triangle-area scores between the mesh generated by each method and its ground-truth counterpart. We also render each generated mesh in wireframe mode and use a VLM to assess the plausibility of its topology. All scores are normalized to the range $(0, 100)$. These metrics should be regarded as complementary evidence rather than definitive measures of

Table 7: Quantitative comparison of topology quality. Boundary measures the similarity of the boundary-edge structure to the ground-truth mesh. Triangle Area measures the similarity of the normalized triangle-area distributions. Visual score measures whether the edge flow and topology of generated meshes conform to common artist-authored modeling practices.

Method	Boundary (↑)	Triangle Area (↑)	Visual (↑)
MeshAnythingV2 Chen et al. (2025b)	25.61	35.45	69
TreeMeshGPT Lionar et al. (2025)	16.31	40.94	23
BPT Weng et al. (2025)	15.13	46.80	60
FastMesh-V4K Kim et al. (2026)	9.56	40.33	42
MeshFlow Li et al. (2026c)	10.51	32.93	85
Lato2 Long et al. (2026)	21.14	28.03	78
Lato2-2000 Long et al. (2026)	30.02	20.29	74
TriFlow Li et al. (2026b)	22.96	45.46	80
TriFlow-4000 Li et al. (2026b)	23.75	21.61	83
Ours	**41.49**	**61.34**	**89**

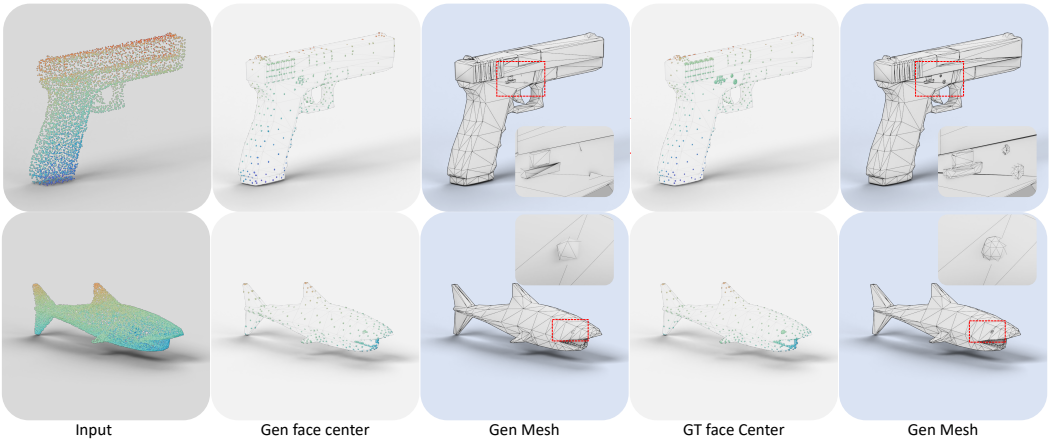

Figure 10: Failure cases.

topology quality, and we continue to recommend qualitative visualizations as the primary basis for comparison.

As shown in Table 7, FaceFlow achieves the best performance in both boundary-edge and triangle-area distribution metrics. These results indicate that the generated meshes better preserve surface completeness and exhibit a more appropriate level of tessellation. Together with the qualitative comparisons in Figs. 5 and 9, they further demonstrate that FaceFlow produces more regular edge flows and coherent topology that more closely resemble artist-authored meshes. The VLM-based evaluation provides additional evidence of the topological advantages of FaceFlow.

A.3 Limitation and Discussion

As shown in Fig. 10, we compare the meshes generated using the face centers predicted by the Point Sink Flow model with those generated using ground-truth face centers. The current Point Sink Flow model occasionally fails to recover face centers in small, fine-scale regions, preventing the subsequent FaceFlow stage from reconstructing the corresponding geometric details and local topology. When ground-truth face centers are provided, however, the FaceFlow model successfully recovers these structures, indicating that the primary bottleneck lies in face-center prediction rather than face generation. We attribute this limitation mainly to the quantization resolution of the Point Sink Flow model and the scale of the training data. Increasing the spatial resolution using a sparse-voxel representation and expanding the training dataset may alleviate this issue.

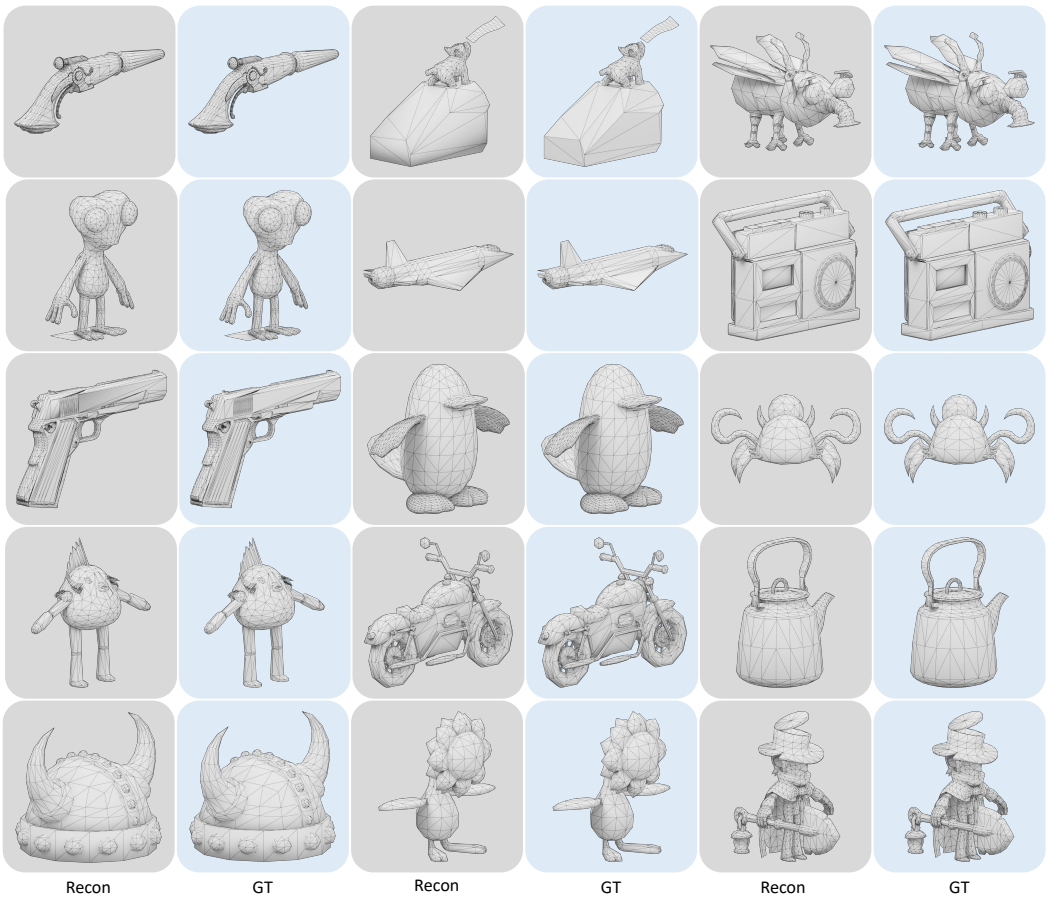

Figure 11: Visualized reconstruction results for our Face VAE.

Moreover, although discrete vertex prediction reduces the inconsistencies between shared vertices that commonly arise from continuous coordinate regression, errors in the predicted discrete tokens can still cause nominally shared vertices to become disconnected, as shown in Fig. 10. While such artifacts can be mitigated through post-processing, explicitly incorporating shared-vertex consistency constraints into model training may offer a more effective solution. In future work, we plan to investigate such constraints and develop higher-resolution representations that better preserve small faces and fine-scale topology.

A.3.1 Extra results on Reconstruction and Generation

We provide additional qualitative reconstruction results in Fig. 11, demonstrating that our Face VAE supports high-fidelity mesh reconstruction. We also present further qualitative generation results produced by FaceFlow in Figs. 12, 13, and 14.

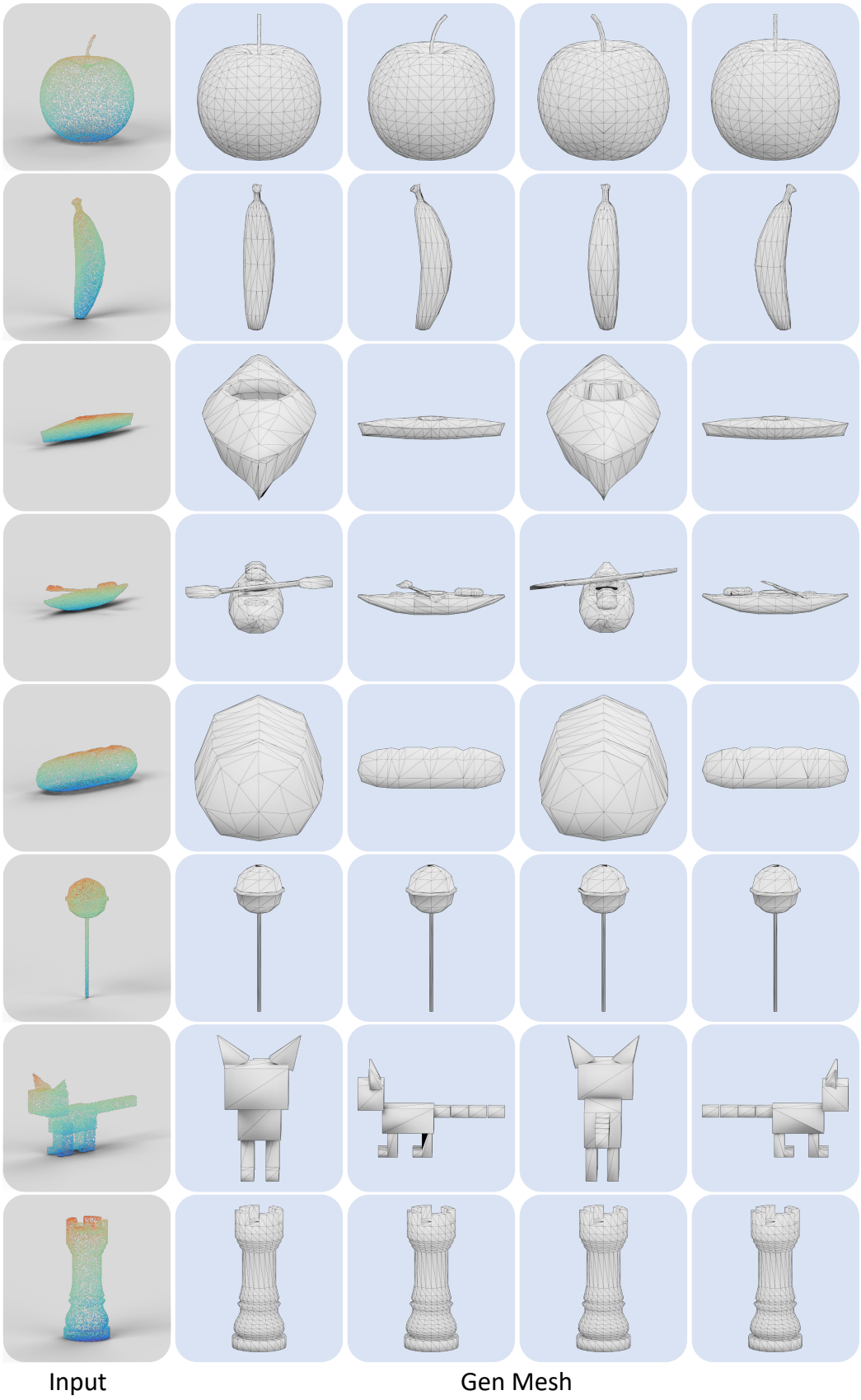

Figure 12: Extra generation results for our FaceFlow.

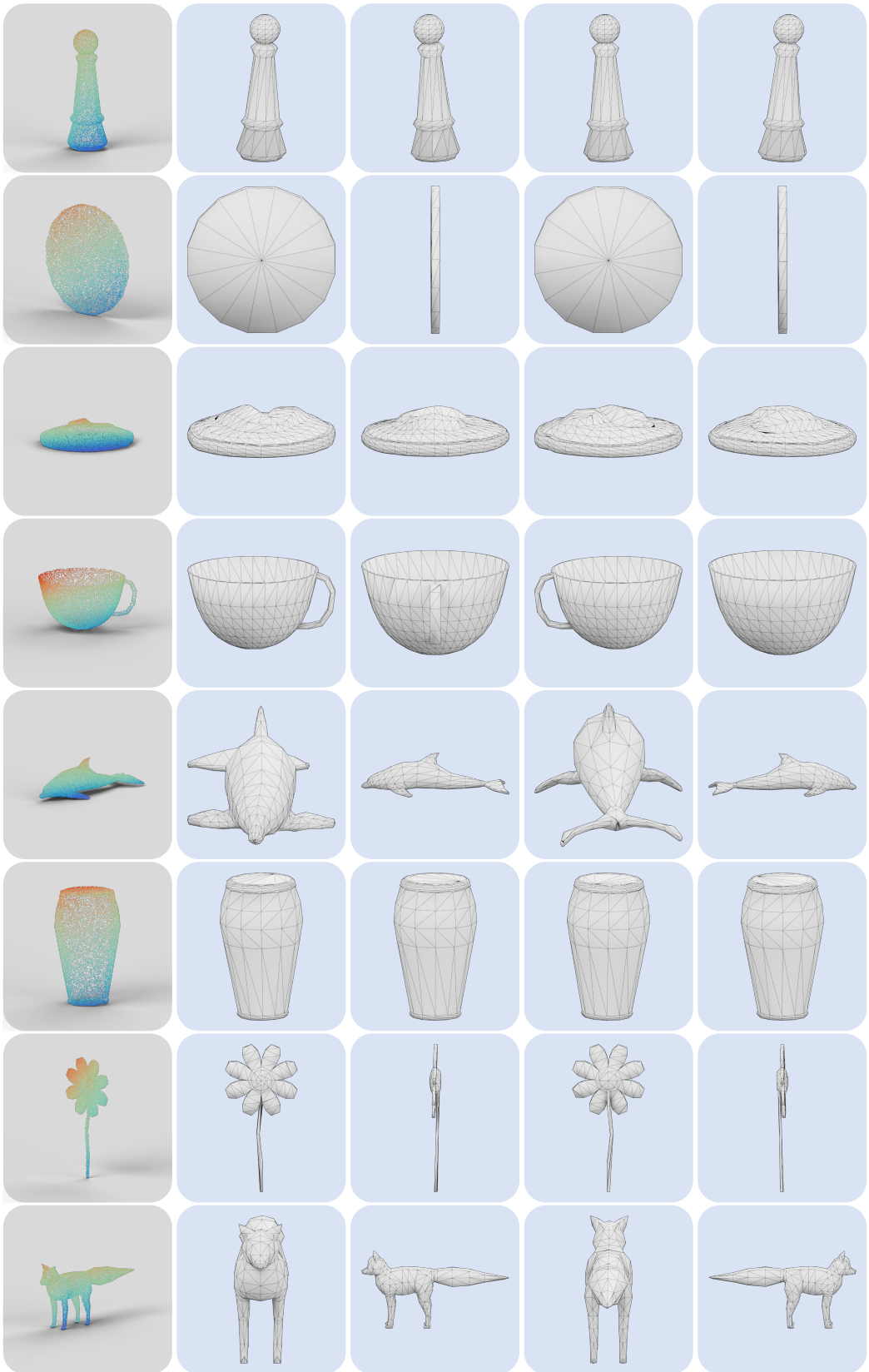

Figure 13: Extra generation results for our FaceFlow.

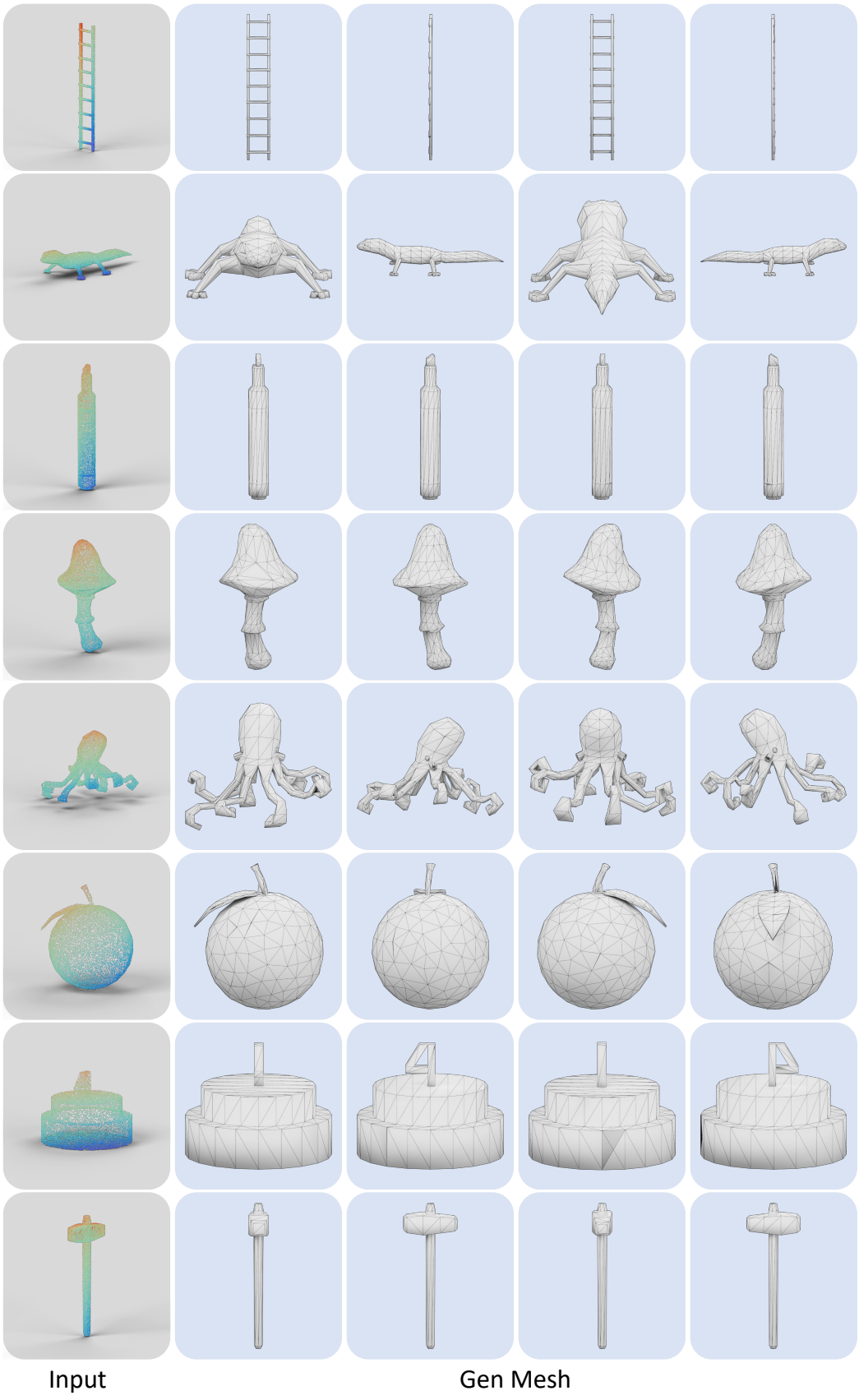

Figure 14: Extra generation results for our FaceFlow.