
Mesh-RFT: Enhancing Mesh Generation via Fine-Grained Reinforcement Fine-Tuning

Jian Liu^{1,2*} Jing Xu^{2*} Song Guo^{1†} Jing Li^{2,3} Jingfeng Guo^{2,4} Jiaao Yu²
Haohan Weng^{2,4} Biwen Lei² Xianghui Yang² Zhuo Chen² Fangqi Zhu¹
Tao Han¹ Chunchao Guo^{2†}

¹ Hong Kong University of Science and Technology ² Tencent Hunyuan
³ University of Science and Technology of China ⁴ South China University of Technology

<https://hitcslj.github.io/mesh-rft/>



Figure 1: **Representative High-Fidelity Mesh Generation by Mesh-RFT.** Gallery of meshes generated from point clouds, demonstrating intricate geometric detail and artist-like aesthetic quality.

Abstract

Existing pretrained models for 3D mesh generation often suffer from data biases and produce low-quality results, while global reinforcement learning (RL) methods rely on object-level rewards that struggle to capture local structure details. To address these challenges, we present **Mesh-RFT**, a novel fine-grained reinforcement fine-tuning framework that employs Masked Direct Preference Optimization (M-DPO) to enable localized refinement via quality-aware face masking. To facilitate efficient quality evaluation, we introduce an objective topology-aware scoring system to evaluate geometric integrity and topological regularity at both object and face levels through two metrics: Boundary Edge Ratio (BER) and Topology Score (TS). By integrating these metrics into a fine-grained RL strategy, Mesh-RFT becomes the first method to optimize mesh quality at the granularity of individual

*Equal Contribution.

†Corresponding Author.

faces, resolving localized errors while preserving global coherence. Experiment results show that our M-DPO approach reduces Hausdorff Distance (HD) by 24.6% and improves Topology Score (TS) by 3.8% over pre-trained models, while outperforming global DPO methods with a 17.4% HD reduction and 4.9% TS gain. These results demonstrate Mesh-RFT’s ability to improve geometric integrity and topological regularity, achieving new state-of-the-art performance in production-ready mesh generation.

1 Introduction

3D polygonal meshes serve as the foundational representation for digital assets in industries such as gaming, film, and product design. Despite their ubiquity, high-quality, topologically optimized meshes—essential for downstream tasks like editing, rigging, and animation—are still predominantly handcrafted by skilled artists. Recent advances in generative models have enabled automated mesh synthesis, significantly reducing the time and expertise required to produce production-ready 3D assets. This democratization of mesh generation broadens access to 3D content creation, empowering non-experts to produce geometrically precise and artistically viable models for applications ranging from immersive media to industrial design.

Existing 3D generative models often use intermediate representations like voxels [1, 2], point clouds [3, 4, 5], latent space [6, 7] or implicit fields [8, 9]. While these avoid direct mesh generation complexities, post-processing (e.g., Marching Cubes [10]) often introduces topological issues and smoothing. Native mesh generation [11] is more direct, with recent work using autoregressive models and neural compression (e.g., VQ-VAE [12, 13, 14]) or geometric serialization tokenizers (e.g., [15, 16, 17, 18, 19]) for sequence-based generation. However, long sequences for high-resolution meshes can cause structural ambiguities and hallucinations (inconsistent edges, non-manifold vertices, distortions, holes), deviating from geometric constraints or artistic intent, ultimately leading to results that may not align with human aesthetic preferences or intended design. Though truncated training [20] helps, autoregressive methods still lack stable generation and high fidelity.

Recently, reinforcement learning [21, 22] has emerged as a compelling approach for aligning mesh generation more closely with human preferences. For example, DeepMesh [23] leverages Direct Preference Optimization (DPO) [24], a simple yet effective preference alignment technique that has also found utility in various other domains [25, 26, 27]. Nevertheless, directly applying reinforcement fine-tuning to mesh generation using this method encounters two primary challenges. Firstly, objectively quantifying mesh quality is difficult. DeepMesh relies on manual annotation of preference pairs, which is expensive, time-consuming, introduces subjective bias, and limits the training data to only 5,000 samples, hindering generalization. Secondly, its use of global reward signals fails to capture the local topological variations inherent in 3D meshes. As illustrated in Figure 2, high-quality and low-quality structures often coexist within a single mesh, leading to training noise due to this mismatch in supervision.

To overcome these limitations, we introduce **Mesh-RFT**, a novel framework that combines **Masked Direct Preference Optimization (M-DPO)** with fine-grained mesh quality evaluation for both global and localized refinement. Unlike prior work using subjective global rewards as supervision signals [23], we employ a topology-aware scoring system with automated metrics—Boundary Edge Ratio (BER) and Topology Score (TS)—to objectively evaluate mesh quality at both object and face levels, circumventing the laborious manual annotation efforts. Mesh-RFT further employs a localized optimization mechanism utilizing M-DPO and quality-aware masks to specifically refine defective regions, thereby addressing the coarse supervision of global rewards. Extensive experiments across diverse meshes demonstrate Mesh-RFT’s superior performance, achieving significant improvements over both the pretrain baseline (24.6% HD

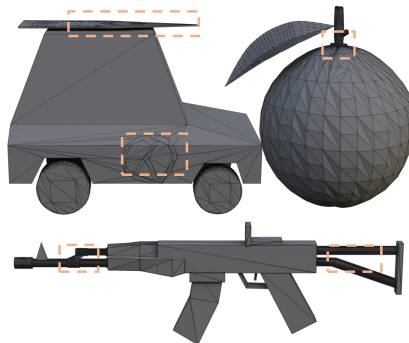


Figure 2: High-quality, artist-like structures often coexist with messy, low-quality regions within the same mesh.

reduction, 3.8% TS improvement) and global DPO (17.4% HD reduction, 4.9% TS improvement), establishing a new benchmark for accuracy and fidelity in generative mesh modeling.

In summary, our contributions are as follows:

- We introduce the first **fine-grained** reinforcement fine-tuning framework, that integrates Masked Direct Preference Optimization (M-DPO) with fine-grained mesh quality evaluation.
- We devise an objective topology-aware scoring system for evaluating mesh quality, eliminating dependency on manual annotation and addressing subjectivity and scalability limitations.
- We propose a novel localized alignment mechanism that optimizes deficient regions geometrically and topologically via quality-aware masks, bridging the gap between global and local supervision.
- Experiments demonstrate that our method achieves state-of-the-art performance in high-fidelity 3D mesh generation.

2 Related work

2.1 3D Generation via Alternative Representations

Many 3D generative models avoid direct mesh modeling by using intermediate representations like voxels, point clouds, or implicit fields. Early voxel methods [1, 2] using grids faced memory issues. Point cloud methods [3, 4, 28, 29] with networks like PointNet [5, 30] struggle with consistency and detail. Implicit fields, especially neural fields [8, 9, 31, 32], offer efficient representations. These include score distillation with 2D diffusion models [33, 34, 35, 36, 37, 38] and 3D Transformer models like LRM [39, 40, 41, 42, 43], alongside recent latent diffusion methods [44, 45, 46, 47, 48, 49, 50, 51] that have demonstrated good scalability and performance. However, these approaches often rely on post-processing via Marching Cubes [10], which can cause topological issues, smoothing, and artifacts.

2.2 Native Mesh Generation

While neural shape representations such as implicit fields have been extensively studied, native mesh generation is an emerging area of research. Early approaches leveraging surface patches [52] or mesh graphs [53] often suffered from quality limitations. Diffusion-based methods [54, 55] have seen limited exploration in this domain, potentially due to inherent difficulties in directly processing meshes. PolyGen [11] demonstrated promise by autoregressively generating mesh vertices and faces. MeshGPT [12] encoded meshes into quantized tokens using VQ-VAE [56] for autoregressive generation. Subsequently, MeshXL [15] proposed a one-stage autoregressive model operating on coordinate-level mesh sequences. Various tokenization techniques [16, 17, 19, 57] and efficient training strategies [20, 58] have been explored to address the challenges of long sequences in high-resolution generation; however, achieving stable and high-fidelity results remains a significant hurdle.

2.3 Reinforcement Learning for Mesh Generation

Reinforcement Learning (RL) [59] has gained traction for 3D generation [60] using human feedback. Reinforcement Learning from Human Feedback (RLHF) aligns models with preferences by training a reward model, then fine-tuning with RL. However, RLHF is costly and unstable for 3D tasks. Direct Preference Optimization (DPO) [24] offers a more efficient, stable alternative by removing the reward model. Despite success in language and image domains [25, 26], DPO’s application to 3D meshes is limited. Closely related, DeepMesh [23] uses global rewards for alignment but struggles with 3D mesh heterogeneity, over-optimizing some regions and under-optimizing others. Thus, RL methods addressing local mesh structures are crucial for better 3D mesh quality and consistency.

3 Method

This section details the Mesh-RFT framework. As illustrated in Figure 3, our pipeline consists of three stages: First, supervised pretraining is performed by feeding point clouds and ground truth

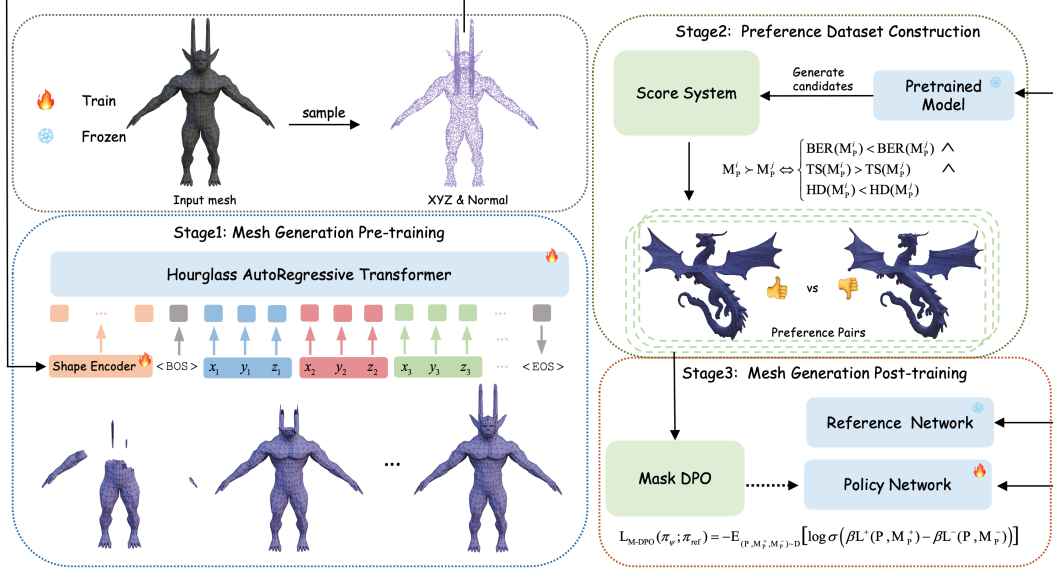


Figure 3: **Mesh-RFT Framework Overview.** The pipeline comprises three stages: **1) Mesh Generation Pre-training** using an Hourglass AutoRegressive Transformer and a Shape Encoder; **2) Preference Dataset Construction** where a pretrained model generates candidate meshes, and a topology-aware score system establishes preference pairs; and **3) Mesh Generation Post-training** which employs Mask DPO with reference and policy networks for subsequent refinement.

mesh sequences into the model. Second, the pretrained model generates candidates, and a topology-aware score system builds a preference dataset. Third, topology-aware Masked Direct Preference Optimization is applied to post-train the model using this preference dataset to refine its performance.

3.1 Mesh Generation Pre-training

Firstly, we discuss mesh tokenization. Prior works [16, 17, 19] compress mesh sequences to manage sequence growth with increasing faces, but such techniques embed excessive geometric information per token, causing cascading face errors when a single token is incorrect (e.g., BPT [19] often introduces patch-level holes). To avoid these issues, we adopt the uncompressed mesh sequence method introduced from MeshXL [15]. Specifically, for a given mesh $\mathcal{M}$, we first quantize the vertex coordinates of each face, and then flatten them in XYZ order to construct a complete token sequence.

Model Architecture. To better capture the structure of the mesh, rather than framing mesh generation as a generic sequence task, we utilize Hourglass Transformer architecture [20, 61]. Our model processes inputs hierarchically and incorporates two shorten and two upsample operations. The shorten operations reduce the token sequence length using techniques such as linear or attention-based pooling, while the upsample operations expand the sequence back to its original length through linear or attention-based methods. This design enables the model to efficiently capture both high-level patterns and fine-grained details. In point-cloud conditioned mesh generation, achieving fine-grained and complex structures requires not only a powerful decoder but also high-quality point cloud features. To this end, we adopt the point cloud encoder pretrained in Hunyuan3D 2.0 [48] to do this. These features are injected into our autoregressive decoder as keys and values via cross-attention [62].

Truncated Training and Sliding-Window Inference. To reduce memory and computational costs, we employ truncated training with fixed-length segments. This approach involves extracting smaller, fixed-length segments from the mesh sequence for training, rather than using the entire sequence. When a segment does not contain the start-of-sequence (SOS) token, we pad a small prefix portion to avoid misleading the model. During inference, we use a sliding window approach to enhance both speed and generation quality. The sliding process begins once 40% of the training window size is covered, and only the most recent 30% of tokens are retained. This method reduces computational load by focusing on the most relevant tokens, as distant tokens typically have less influence on each



Figure 4: **Examples of collected preference pairs.** Meshes are annotated as preferred using our scoring system. For certain pairs, the selected "good" meshes may exhibit inferior local performance in specific regions compared to the rejected "bad" meshes.

other. Additionally, it helps mitigate high perplexity at the tail of each window, leading to more accurate and efficient generation.

3.2 Preference Dataset Construction

We establish a systematic pipeline for constructing the preference dataset, which is used for RLHF fine-tuning in the second stage. This pipeline consists of three key components: candidate generation, multi-metric evaluation, and preference ranking. The process is described as follows.

Candidate Generation. For each input point cloud $\mathcal{P}$, we generate eight candidate meshes $\{\mathcal{M}_{\mathcal{P}}^1, \mathcal{M}_{\mathcal{P}}^2, \dots, \mathcal{M}_{\mathcal{P}}^8\}$ using the pre-trained model G_{θ}^{pre} .

Multi-Metric Evaluation. We evaluate each candidate mesh using a comprehensive set of criteria to assess both geometric consistency and topological quality. In addition to measuring the geometric alignment with the input data, we introduce two topology-oriented metrics that specifically aim to capture the structural integrity and coherence of the generated meshes. These three metrics are: Boundary Edge Ratio (BER) and Topology Score (TS) for evaluating topology, and Hausdorff Distance (HD) for evaluating geometric consistency.

- **Boundary Edge Ratio (BER):** This metric, defined as $BER(\mathcal{M}) = \frac{E_{\partial\mathcal{M}}}{E_{\mathcal{M}}}$, quantifies the integrity of the mesh by calculating the proportion of its boundary edges ($E_{\partial\mathcal{M}}$) to the total number of edges ($E_{\mathcal{M}}$). Boundary edges are those connected to only one face, and a high BER value (typically above 0.002 in our dataset, which consists mostly of closed meshes) suggests potential issues like surface discontinuities, holes, or mesh damage. Ideally, a closed, manifold mesh should have a BER of 0.
- **Topology Score (TS):** The Topology Score, $TS(\mathcal{M}) = \sum_{i=1}^4 w_i s_i(Q(\mathcal{M}))$, assesses the structural quality of a mesh $\mathcal{M}$ by analyzing a derived quadrilateral mesh $Q(\mathcal{M})$, obtained through standard triangle-to-quad merging. The score is a weighted sum of four sub-metrics: Quad Ratio ($w_1 = 0.4$), which measures the efficiency of the conversion; Angle Quality ($w_2 = 0.2$), quantifying the deviation of quadrilateral angles from 90° ; Aspect Ratio ($w_3 = 0.3$), evaluating the regularity of quadrilateral shapes; and Adjacent Consistency ($w_4 = 0.1$), encouraging uniform aspect ratios between neighboring quadrilaterals. This quadrilateral-based evaluation is used because quad meshes are preferred in industrial applications, making the quality of the quadrangulation a practical indicator of the topological soundness of the original triangular mesh. Further details are in the supplementary material A.3.
- **Hausdorff Distance (HD):** This standard metric measures the maximum distance from a point in one set to the closest point in the other set. Here, it quantifies the geometric alignment between the

reconstructed mesh $\mathcal{M}_{\mathcal{P}}^i$ and the input point cloud $\mathcal{P}$ by measuring the maximum distance between their respective point samples. A lower HD value indicates a better geometric reconstruction.

Preference Ranking. To construct the preference dataset, we generate pairwise comparisons through exhaustive combinations of the eight candidate meshes for each input point cloud $\mathcal{P}$, resulting in a total of $\binom{8}{2} = 28$ pairs. For each pair $(\mathcal{M}_{\mathcal{P}}^i, \mathcal{M}_{\mathcal{P}}^j)$, we define a preference relation $\mathcal{M}_{\mathcal{P}}^i \succ \mathcal{M}_{\mathcal{P}}^j$ if and only if $\mathcal{M}_{\mathcal{P}}^i$ outperforms $\mathcal{M}_{\mathcal{P}}^j$ across all three evaluation metrics:

$$\begin{aligned} \mathcal{M}_{\mathcal{P}}^i \succ \mathcal{M}_{\mathcal{P}}^j \iff & \quad \begin{aligned} & BER(\mathcal{M}_{\mathcal{P}}^i) < BER(\mathcal{M}_{\mathcal{P}}^j) \quad \wedge \\ & TS(\mathcal{M}_{\mathcal{P}}^i) > TS(\mathcal{M}_{\mathcal{P}}^j) \quad \wedge \\ & HD(\mathcal{M}_{\mathcal{P}}^i) < HD(\mathcal{M}_{\mathcal{P}}^j) \end{aligned} \end{aligned} \quad (1)$$

We refer to $\mathcal{M}_{\mathcal{P}}^i$ as the positive sample (denoted $\mathcal{M}_{\mathcal{P}}^+$) and $\mathcal{M}_{\mathcal{P}}^j$ as the negative sample (denoted $\mathcal{M}_{\mathcal{P}}^-$) for the pair. Using this rule, we construct a set of preference triplets of the form $(\mathcal{P}, \mathcal{M}_{\mathcal{P}}^+, \mathcal{M}_{\mathcal{P}}^-)$, which constitutes our preference dataset for reinforcement learning with human feedback.

3.3 Mesh Generation Post-training

While our pre-trained model produces topologically valid meshes, two persistent challenges remain: (1) localized geometric imperfections in high-curvature regions, and (2) inconsistent face density distribution causing aesthetic artifacts. Although DeepMesh [23] adopts RLHF for mesh refinement, its reward function is primarily based on global mesh structure, making it insufficient for fine-grained control over local mesh quality. To address these limitations, we propose Masked Direct Preference Optimization (M-DPO)—a spatially aware extension of DPO [24]. M-DPO introduces quality localization masks to guide learning toward problematic regions, enabling more targeted and effective mesh refinement.

Quality-Aware Local Masking. The goal of local masking is to differentiate high-quality regions of a mesh from those of lower quality. Given a triangular mesh $\mathcal{M}$, we assess each triangle face individually. A face is labeled as *good* if it satisfies the following two conditions: (1) it can be successfully merged into a quadrilateral, and (2) the resulting quad has a quality score above a predefined threshold. The quad quality is evaluated using a weighted combination of three metrics introduced in Section 3.2: Angle Quality, Aspect Ratio, and Adjacent Consistency. For each triangle face labeled as *good*, we assign a value of 1 to all corresponding token positions in the mesh sequence (typically 9 tokens per face). Conversely, faces that do not meet the criteria are considered *bad*, and their associated tokens are assigned a value of 0. We define the local masking function as ϕ , such that $\phi(\mathcal{M}) \in \{0, 1\}^{|\mathcal{M}|}$, where $|\mathcal{M}|$ denotes the length of the token sequence representing mesh $\mathcal{M}$.

Masked Direct Preference Optimization. Standard DPO tends to optimize global reward signals uniformly across the entire mesh sequence, which can lead to over-smoothed results and the loss of fine-grained geometric details. In contrast, our Masked Direct Preference Optimization (M-DPO) addresses this limitation by applying element-wise importance weighting guided by local quality masks, allowing the model to focus refinement specifically on low-quality regions. As illustrated in Figure 3, we designate the pretrained model from the first stage as the reference model, denoted as $G_{\text{ref}} := G_{\theta}^{\text{pre}}$, whose parameters are frozen during training. A trainable policy model G_{ψ} is then initialized with the parameters of G_{θ}^{pre} , and subsequently fine-tuned to better align with human preferences by encouraging it to generate outputs closer to the positive examples in our preference dataset. The objective of M-DPO is to maximize the likelihood of preferred (positive) samples over less-preferred (negative) ones, with a focus on quality-critical regions identified via local masks:

$$\mathcal{L}_{\text{M-DPO}}(\pi_{\psi}; \pi_{\text{ref}}) = -\mathbb{E}_{(\mathcal{P}, \mathcal{M}_{\mathcal{P}}^+, \mathcal{M}_{\mathcal{P}}^-) \sim \mathcal{D}} \left[\log \sigma \left(\beta \mathcal{L}^+(\mathcal{P}, \mathcal{M}_{\mathcal{P}}^+) - \beta \mathcal{L}^-(\mathcal{P}, \mathcal{M}_{\mathcal{P}}^-) \right) \right] \quad (2)$$

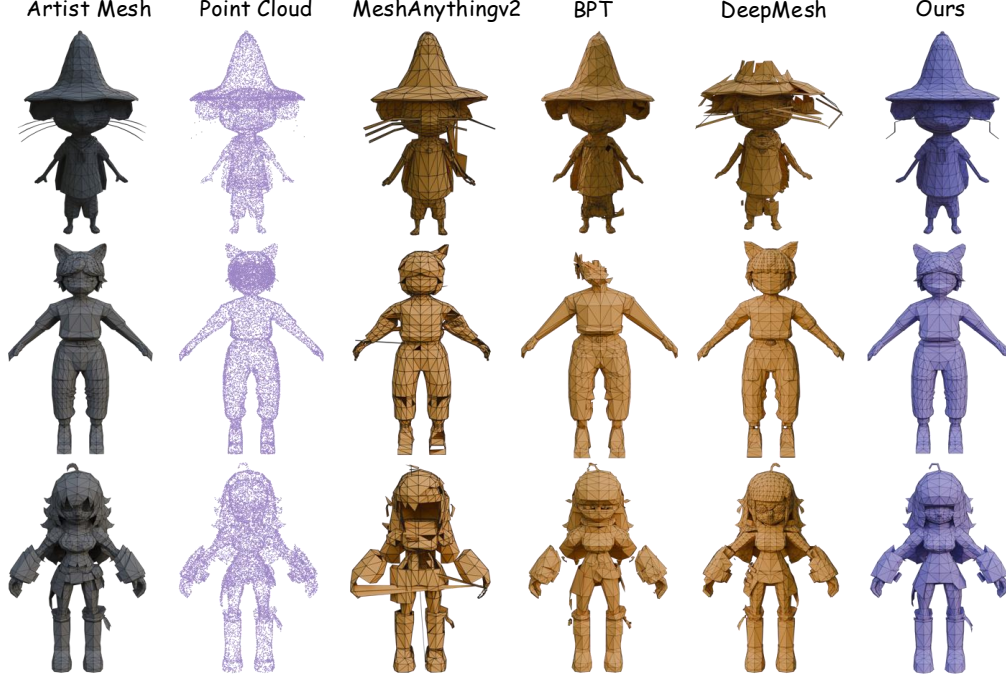


Figure 5: **Qualitative comparison on artist-designed meshes.** Our method generates more coherent and visually plausible surfaces with finer structural details and fewer topological artifacts compared to baseline approaches.

where the positive and negative log-ratio terms are computed as:

$$\begin{aligned}\mathcal{L}^+(\mathcal{P}, \mathcal{M}_{\mathcal{P}}^+) &= \log \frac{\|\pi_{\psi}(\mathcal{M}_{\mathcal{P}}^+|\mathcal{P}) \odot \phi(\mathcal{M}_{\mathcal{P}}^+)\|_1}{\|\pi_{\text{ref}}(\mathcal{M}_{\mathcal{P}}^+|\mathcal{P}) \odot \phi(\mathcal{M}_{\mathcal{P}}^+)\|_1} \\ \mathcal{L}^-(\mathcal{P}, \mathcal{M}_{\mathcal{P}}^-) &= \log \frac{\|\pi_{\psi}(\mathcal{M}_{\mathcal{P}}^-|\mathcal{P}) \odot (1 - \phi(\mathcal{M}_{\mathcal{P}}^-))\|_1}{\|\pi_{\text{ref}}(\mathcal{M}_{\mathcal{P}}^-|\mathcal{P}) \odot (1 - \phi(\mathcal{M}_{\mathcal{P}}^-))\|_1}\end{aligned}\tag{3}$$

Here, $\mathcal{D}$ denotes the preference dataset, and π is the token-level probability distribution produced by the model. The operator $\odot$ indicates element-wise (Hadamard) multiplication, and $\|\cdot\|_1$ denotes the ℓ_1 norm over the token sequence. The hyperparameter β controls the sharpness of preference separation, and σ is the standard sigmoid function. M-DPO effectively preserves satisfactory regions while actively refining low-quality areas identified by the local quality mask. This targeted optimization strategy not only maintains the global structure but also enhances local geometric fidelity, offering a finer control over mesh generation quality compared to standard DPO.

4 Experiments

4.1 Experiment Settings

Datasets Our model is pretrained on 2M meshes from large-scale datasets including ShapeNetV2 [63], 3D-FUTURE [64], Objaverse [65], Objaverse-XL [66], and licensed assets. After filtering low-quality scans and poorly topologized CAD models, 800K meshes form the fine-tuning subset. For preference alignment, we construct a specialized dataset of 10,000 generated meshes, each paired with 8 topological variations derived from the same input point cloud. To enhance geometric generalization, meshes are perturbed at the vertex level and subsampled from an initial 50K-point cloud to 16,384 points, without enforcing watertightness. For evaluation, we employ two test sets: (1) 100 high-quality, artist-designed meshes for qualitative analysis, and (2) 100 dense, out-of-distribution meshes generated by Hunyuan2.5 [48], providing rigorous real-world validation. More data details can be seen in Supplementary A.1.

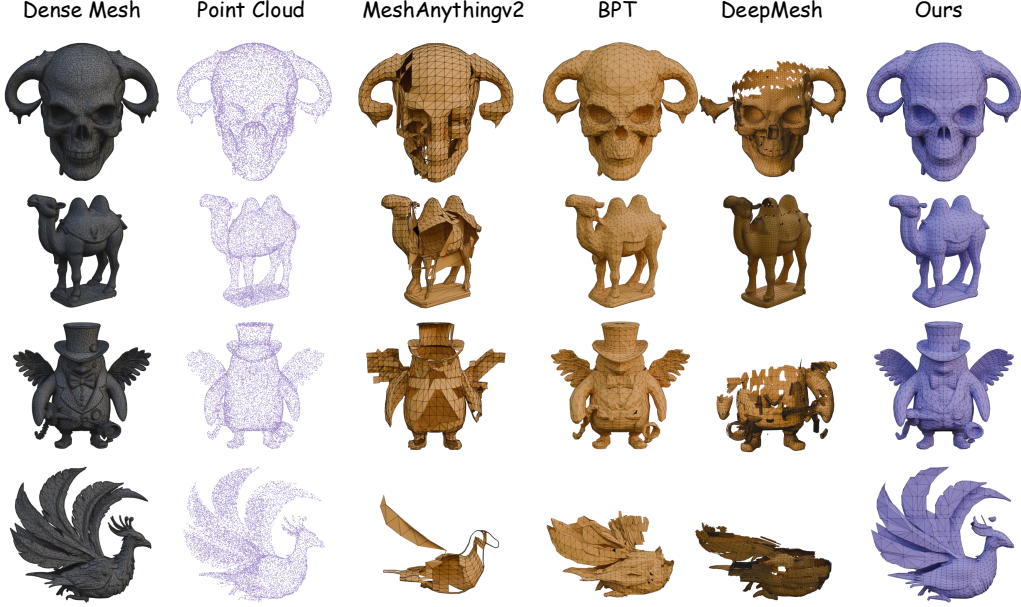


Figure 6: **Generalization results on dense, out-of-distribution meshes.** Our model demonstrates superior geometric fidelity and surface continuity, maintaining high-quality reconstruction even under complex and unseen input conditions.

Implementation Details We pretrained on 256 NVIDIA H20 GPUs (2/GPU) for 10 days with AdamW [67] ($\beta_1 = 0.9$, $\beta_2 = 0.99$) and Flash Attention, following a 100-step linear warm-up. M-DPO post-training took 8 hours on 64 GPUs with a $5e - 7$ learning rate. See supplementary material A.2 for full details.

Baselines. We benchmark our approach against leading mesh generation methods, including **MeshAnythingV2** [16], **BPT** [19], and **DeepMesh** [23]. Since DeepMesh only publicly provides inference code and a 512M parameter version, we use this configuration for comparison.

4.2 Qualitative Results

We qualitatively compare our method with existing baselines. As shown in Figure 5, our model produces meshes that are significantly more coherent, artistically plausible, and faithful to the input geometry, particularly in challenging regions such as fine-grained structures and curved surfaces. These results highlight our model’s ability to preserve detail and maintain topological regularity. In contrast, baseline methods often exhibit structural artifacts such as incomplete regions, broken connectivity, or excessive smoothing, especially in geometrically intricate areas. To further evaluate generalization beyond the training distribution, we conduct experiments on a set of dense, high-resolution meshes not seen during training. As illustrated in Figure 6, our method consistently outperforms prior approaches in reconstructing complex geometry and maintaining surface continuity under high-resolution inputs. These results demonstrate that our model not only performs well on curated artistic data but also generalizes effectively to challenging, real-world examples.

4.3 Quantitative Results

Table 1 presents a quantitative comparison of our method against baselines on artist-designed meshes and dense meshes derived from AI-generated representations. We report both geometric and topological metrics, including Hausdorff Distance (HD), Topology Score (TS), and Boundary Error Rate (BER). Our method consistently outperforms competing approaches across all metrics, demonstrating superior geometric fidelity and topological coherence. To further validate perceptual quality, we conducted a user study (US) in which participants were asked to compare mesh outputs based on visual plausibility and structural integrity. The results indicate a strong preference for our

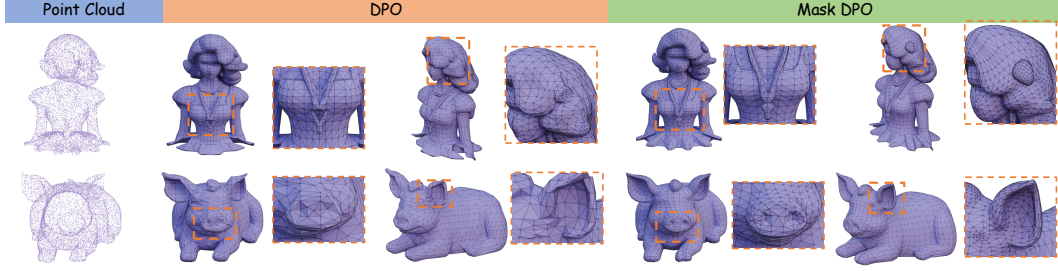


Figure 7: **The effectiveness of Mask DPO.** The addition of Mask DPO enhances the visual fidelity of the generated meshes, despite similar geometric performance across methods.

Table 1: **Quantitative comparison with other baselines in Artist and Dense Meshes.** Our approach achieves superior performance in both geometric accuracy and visual fidelity compared to existing baselines. DeepMesh* were tested using their 0.5 B version.

Data Type	Artist Meshes					Dense Meshes				
Metrics	CD ↓	HD ↓	TS ↑	BER ↓	US ↑	CD ↓	HD ↓	TS ↑	BER ↓	US ↑
MeshAnythingv2 [16]	0.2143	0.4197	68.3	0.0749	9%	0.2265	0.4760	72.0	0.0913	8%
BPT [19]	0.1275	0.2735	72.7	0.0280	20%	0.1615	0.3347	73.7	0.0113	18%
DeepMesh* [23]	0.1331	0.2866	74.9	0.0296	22%	0.1760	0.3570	75.8	0.0044	20%
Ours	0.0973	0.1826	77.5	0.0182	45%	0.1286	0.2411	79.4	0.0015	40%

method, confirming that its advantages are not only quantitatively measurable but also perceptually significant.

4.4 Ablation Study

4.4.1 Score System

We evaluate the efficacy of our score-based preference system within the domain of dense mesh generation. As demonstrated in Table 2, employing only Hausdorff Distance to differentiate between high- and low-quality meshes (denoted as N-DPO) yields marginal improvements in geometric consistency over the pretrained model (Pretrain) and exhibits a decrease in the TS score. Conversely, leveraging our proposed composite scoring system (denoted as S-DPO) for the construction of preference data facilitates a substantial performance gain.

Table 2: **Quantitative Evaluation of Score System and Mask DPO Methods.**

Method	CD ↓	HD ↓	TS ↑	BER ↓	US ↑
Pretrain	0.1588	0.3196	76.5	0.0033	30%
N-DPO	0.1455	0.2919	75.7	0.0028	32%
S-DPO	0.1348	0.2625	77.9	0.0023	35%
M-DPO	0.1286	0.2411	79.4	0.0015	40%

4.4.2 Mask DPO

Figure 4 illustrates that standard global DPO often fails to capture local variations in mesh quality. Our proposed topology-aware local mask mechanism effectively addresses this limitation by enabling the model to learn from spatially localized preference signals. Built on the preference dataset derived from our scoring system, the Mask-DPO model (denoted as M-DPO) demonstrates a clear advantage over the global score-based DPO baseline (S-DPO), as shown in Figure 7. This localized learning strategy leads to significant improvements in both quantitative metrics and human preference, as confirmed in Table 2. Notably, M-DPO produces outputs that are not only closer to the ground truth but also more consistently favored by human evaluators, providing strong empirical support for localized preference learning.

5 Conclusion

Generating high-quality 3D meshes remains a significant challenge. We introduced Mesh-RFT, a novel framework employing topology-aware scoring and Masked Direct Preference Optimization (M-DPO) for fine-grained refinement. By leveraging objective metrics and localized optimization, Mesh-RFT advances the state-of-the-art in automated mesh generation. Our approach significantly improves both the geometric accuracy and topological fidelity of generated meshes compared to previous methods. This work offers a substantial step forward in creating production-ready 3D assets for a wide range of applications. Limitations and future work are discussed in appendix C.

Acknowledgements

This research was supported by fundings from the Hong Kong RGC General Research Fund (152244/21E, 152169/22E, 152228/23E, 162161/24E), Research Impact Fund (No. R5011-23F, No. R5060-19), Collaborative Research Fund (No. C1042-23GF), NSFC/RGC Collaborative Research Scheme (No. CRS_HKUST602/24), Areas of Excellence Scheme (No. AoE/E-601/22-R), and the InnoHK (HKGAI).

References

- [1] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016.
- [2] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions On Graphics (TOG)*, 36(4):1–11, 2017.
- [3] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2837–2845, 2021.
- [4] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.
- [5] 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*, pages 652–660, 2017.
- [6] 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.
- [7] Junliang Ye, Zhengyi Wang, Ruowen Zhao, Shenghao Xie, and Jun Zhu. Shapellm-omni: A native multimodal llm for 3d generation and understanding. *arXiv preprint arXiv:2506.01853*, 2025.
- [8] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5939–5948, 2019.
- [9] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.
- [10] 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*, pages 347–353. 1998.
- [11] Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*, pages 7220–7229. PMLR, 2020.
- [12] 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*, pages 19615–19625, 2024.

- [13] Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, et al. Meshanything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*, 2024.
- [14] 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.
- [15] Sijin Chen, Xin Chen, Anqi Pang, Xianfang Zeng, Wei Cheng, Yijun Fu, Fukun Yin, Billzb 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, 2025.
- [16] 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. *arXiv preprint arXiv:2408.02555*, 2024.
- [17] Jiaxiang Tang, Zhaoshuo Li, Zekun Hao, Xian Liu, Gang Zeng, Ming-Yu Liu, and Qinsheng Zhang. Edgerunner: Auto-regressive auto-encoder for artistic mesh generation. *arXiv preprint arXiv:2409.18114*, 2024.
- [18] Stefan Lionar, Jiabin Liang, and Gim Hee Lee. Treemeshgpt: Artistic mesh generation with autoregressive tree sequencing. *arXiv preprint arXiv:2503.11629*, 2025.
- [19] 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. *arXiv preprint arXiv:2411.07025*, 2024.
- [20] 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.
- [21] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [22] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- [23] Ruowen Zhao, Junliang Ye, Zhengyi Wang, Guangce Liu, Yiwen Chen, Yikai Wang, and Jun Zhu. Deepmesh: Auto-regressive artist-mesh creation with reinforcement learning. *arXiv preprint arXiv:2503.15265*, 2025.
- [24] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.
- [25] Shuaijie She, Wei Zou, Shujian Huang, Wenhao Zhu, Xiang Liu, Xiang Geng, and Jiajun Chen. Mapo: Advancing multilingual reasoning through multilingual alignment-as-preference optimization. *arXiv preprint arXiv:2401.06838*, 2024.
- [26] Zixuan Liu, Xiaolin Sun, and Zizhan Zheng. Enhancing llm safety via constrained direct preference optimization. *arXiv preprint arXiv:2403.02475*, 2024.
- [27] Yiyang Zhou, Chenhang Cui, Rafael Rafailov, Chelsea Finn, and Huaxiu Yao. Aligning modalities in vision large language models via preference fine-tuning. *arXiv preprint arXiv:2402.11411*, 2024.
- [28] Qianjiang Hu, Zhimin Zhang, and Wei Hu. Rangeldm: Fast realistic lidar point cloud generation. In *European Conference on Computer Vision*, pages 115–135. Springer, 2024.
- [29] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pages 131–147. Springer, 2024.
- [30] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.
- [31] Xin-Yang Zheng, Hao Pan, Peng-Shuai Wang, Xin Tong, Yang Liu, and Heung-Yeung Shum. Locally attentional sdf diffusion for controllable 3d shape generation. *ACM Transactions on Graphics (ToG)*, 42(4):1–13, 2023.

- [32] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4456–4465, 2023.
- [33] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022.
- [34] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *arXiv preprint arXiv:2303.13873*, 2023.
- [35] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [36] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
- [37] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023.
- [38] 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*, pages 6796–6807, 2024.
- [39] 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. *arXiv preprint arXiv:2311.04400*, 2023.
- [40] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation, 2023.
- [41] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. Crm: Single image to 3d textured mesh with convolutional reconstruction model. *arXiv preprint arXiv:2403.05034*, 2024.
- [42] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024.
- [43] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024.
- [44] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in neural information processing systems*, 36:73969–73982, 2023.
- [45] 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.
- [46] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. *arXiv preprint arXiv:2405.14832*, 2024.
- [47] Weiyu Li, Jiarui Liu, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman: High-fidelity mesh generation with 3d native generation and interactive geometry refiner. *arXiv preprint arXiv:2405.14979*, 2024.
- [48] 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*, 2025.
- [49] 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. *arXiv preprint arXiv:2503.21732*, 2025.

- [50] 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. *arXiv preprint arXiv:2502.06608*, 2025.
- [51] Weiyu Li, Xuanyang Zhang, Zheng Sun, Di Qi, Hao Li, Wei Cheng, Weiwei Cai, Shihao Wu, Jiarui Liu, Zihao Wang, et al. Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. *arXiv preprint arXiv:2505.07747*, 2025.
- [52] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. In *European Conference on Computer Vision*, pages 1–20. Springer, 2024.
- [53] Angela Dai and Matthias Nießner. Scan2mesh: From unstructured range scans to 3d meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5574–5583, 2019.
- [54] Antonio Alliegro, Yawar Siddiqui, Tatiana Tommasi, and Matthias Nießner. Polydiff: Generating 3d polygonal meshes with diffusion models. *arXiv preprint arXiv:2312.11417*, 2023.
- [55] 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.
- [56] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.
- [57] 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. *arXiv preprint arXiv:2501.14317*, 2025.
- [58] Hanxiao Wang, Biao Zhang, Weize Quan, Dong-Ming Yan, and Peter Wonka. iflame: Interleaving full and linear attention for efficient mesh generation. *arXiv preprint arXiv:2503.16653*, 2025.
- [59] Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback. *Advances in Neural Information Processing Systems*, 36:10935–10950, 2023.
- [60] Junliang Ye, Fangfu Liu, Qixiu Li, Zhengyi Wang, Yikai Wang, Xinzhou Wang, Yueqi Duan, and Jun Zhu. Dreamreward: Text-to-3d generation with human preference. In *European Conference on Computer Vision*, pages 259–276. Springer, 2024.
- [61] Piotr Nawrot, Szymon Tworowski, Michał Tyrolski, Łukasz Kaiser, Yuhuai Wu, Christian Szegedy, and Henryk Michalewski. Hierarchical transformers are more efficient language models. *arXiv preprint arXiv:2110.13711*, 2021.
- [62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [63] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.
- [64] 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*, 129:3313–3337, 2021.
- [65] 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*, pages 13142–13153, 2023.
- [66] 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, 2023.
- [67] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

- [68] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.
- [69] Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.
- [70] Yufeng Yuan, Qiyang Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”,**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: We have clearly stated the claims made in the abstract and introduction, accurately reflecting the paper’s contributions and scope.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We have discussed the limitations of our work in the appendix, which include the inability to generalize well to snake-like data due to the lack of such samples in the training dataset.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: Our paper does not include theoretical results, and therefore, this question is not applicable to our work.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide a detailed description of the model and experimental settings in our paper, ensuring that readers have the necessary information to reproduce the main experimental results. Additionally, we plan to release the code to further enhance reproducibility and facilitate verification of our results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: While we currently do not provide open access to the data and code, we plan to release the code along with sufficient instructions to reproduce the main experimental results after the paper has been accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: We have provided all the necessary details regarding the training and testing process, including data splits, network structure, hyperparameters, and the type of optimizer used.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[No\]](#)

Justification: We conducted our experiments and baseline experiments on the same training and testing datasets to ensure a fair comparison.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provided sufficient information on the computer resources needed to reproduce the experiments in the implementation details section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: Our research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discussed the social impact of this work in the introduction and conclusion sections.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: No such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We will release the code, data, and models publicly upon the acceptance of the paper.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We did not submit any new assets at the time of submission. However, we plan to release well-documented code after the paper's acceptance.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our research does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our research does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in our research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendix

A More Implementation Details

A.1 Data Details

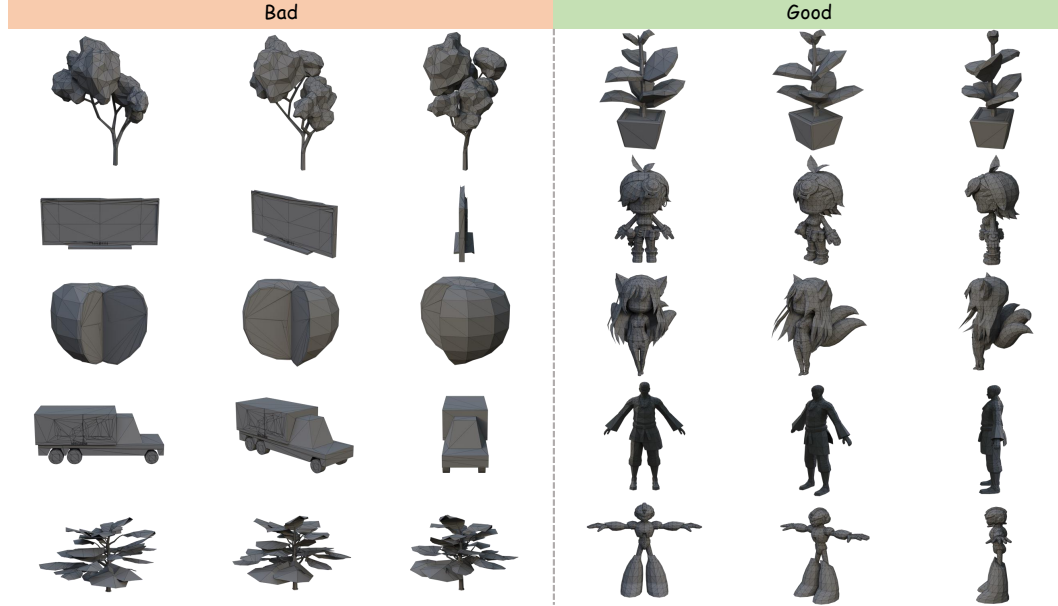


Figure 8: **Examples of Wiring Complexity in the Dataset.** The dataset contains cases with high-quality surface triangulations alongside instances where local regions exhibit lower quality.

After filtering low-quality scans and poorly topologized CAD models, our dataset size was reduced from 2 million to approximately 800,000 samples, with an average face count of 5,000. The distribution of face counts in this refined dataset is illustrated in Figure 9. Despite this initial filtering, as demonstrated in Figure 8, the dataset still includes instances where local surface triangulation quality is suboptimal. These instances are challenging to entirely eliminate due to the fact that even within lower-quality cases, regions with good topology often exist.

A.2 More Training and Inference Details

Our model consists of 24 Transformer layers (1.1B parameters) arranged in a three-stage hour-glass structure(2-4-12-4-2). It features a hidden dimension of 1536 and 16 attention heads. The vocabulary size for vertex coordinate quantization is 1024. The architecture supports a 36,864-token context window during inference and generates meshes through temperature-controlled sampling ($T = 0.5$), balancing output diversity and stability. For the pretraining phase, we initially trained on 2M meshes for 6 days, followed by an additional 4 days of training on a filtered set of 800k meshes. A 5k-face mesh from the preference dataset requires approximately 45k tokens. Generating 80,000 meshes from 10,000 dense meshes took about 2 days, with processing handled by 64 GPUs at a batch size of 8 per GPU, resulting in a speed of around 40 tokens/s.

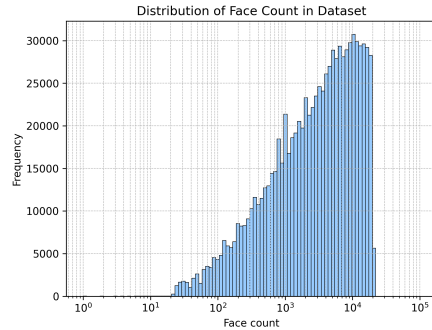


Figure 9: **Face Count Distribution in the Fine-tuning Dataset.** This figure presents the distribution of face counts within our fine-tuning dataset, which comprises approximately 800k samples with an average of 5k faces per model.

In contrast, calculating the EDR, TS, HD, and local mask for each mesh was completed in under 1 hour on a single machine. Furthermore, we utilize ZeRO-2 to minimize GPU memory consumption.

A.3 Metrics Details

The Topology Score $TS(\mathcal{M})$ provides a quantitative measure of the structural quality of a mesh $\mathcal{M}$. It is computed based on the properties of a derived quadrilateral mesh $\mathcal{Q}(\mathcal{M})$ and is defined as a weighted linear combination of four sub-metrics:

$$TS(\mathcal{M}) = w_1 \cdot s_1(\mathcal{Q}(\mathcal{M})) + w_2 \cdot s_2(\mathcal{Q}(\mathcal{M})) + w_3 \cdot s_3(\mathcal{Q}(\mathcal{M})) + w_4 \cdot s_4(\mathcal{Q}(\mathcal{M})) \quad (4)$$

where the weights are empirically set to $w_1 = 0.4$ (Quad Ratio), $w_2 = 0.2$ (Angle Quality), $w_3 = 0.3$ (Aspect Ratio), and $w_4 = 0.1$ (Adjacent Consistency), satisfying $\sum_{i=1}^4 w_i = 1$. The sub-metrics are formally defined as follows:

- **Quad Ratio** (s_1): This metric assesses the efficiency of the triangle-to-quad conversion. Let $\mathcal{F}_\mathcal{Q}$ be the set of quadrilateral faces and $\mathcal{F}_\mathcal{T}$ be the set of triangular faces in $\mathcal{Q}(\mathcal{M})$. The Quad Ratio is given by:

$$s_1(\mathcal{Q}(\mathcal{M})) = \frac{|\mathcal{F}_\mathcal{Q}|}{|\mathcal{F}_\mathcal{T}| + |\mathcal{F}_\mathcal{Q}|} \quad (5)$$

where $|\cdot|$ denotes the cardinality of the set.

- **Angle Quality** (s_2): This metric quantifies the deviation of quadrilateral angles from the ideal 90° . For each quadrilateral $q \in \mathcal{Q}(\mathcal{M})$, let $A(q) = \{\alpha_1^q, \alpha_2^q, \alpha_3^q, \alpha_4^q\}$ be the set of its internal angles. The Angle Quality is defined as the average normalized deviation:

$$s_2(\mathcal{Q}(\mathcal{M})) = 1 - \frac{1}{|\mathcal{Q}(\mathcal{M})|} \sum_{q \in \mathcal{Q}(\mathcal{M})} \frac{\sum_{\alpha \in A(q)} |\alpha - 90^\circ|}{360^\circ} \quad (6)$$

- **Aspect Ratio** (s_3): This metric evaluates the regularity of the quadrilateral shapes. For a quadrilateral $q \in \mathcal{Q}(\mathcal{M})$ with side lengths $l_{q,1}, l_{q,2}, l_{q,3}, l_{q,4}$, the aspect ratio r_q is defined as:

$$r_q = \max \left(\frac{\max(l_{q,1}, l_{q,3})}{\min(l_{q,1}, l_{q,3})}, \frac{\max(l_{q,2}, l_{q,4})}{\min(l_{q,2}, l_{q,4})} \right) \quad (7)$$

An additional edge ratio e_q for each quadrilateral is computed as the average of its side lengths normalized by the maximum side length:

$$e_q = \frac{1}{4} \sum_{i=1}^4 \frac{l_{q,i}}{\max_{j=1}^4 l_{q,j}} \quad (8)$$

The Aspect Ratio sub-metric s_3 is then a combination of these measures:

$$s_3(\mathcal{Q}(\mathcal{M})) = 0.5 \cdot \left(\frac{1}{|\mathcal{Q}(\mathcal{M})|} \sum_{q \in \mathcal{Q}(\mathcal{M})} r_q \right) + 0.5 \cdot \left(\frac{1}{|\mathcal{Q}(\mathcal{M})|} \sum_{q \in \mathcal{Q}(\mathcal{M})} e_q \right) \quad (9)$$

- **Adjacent Consistency** (s_4): This metric encourages smooth variations in the aspect ratios of neighboring quadrilaterals. For a quadrilateral $q_i \in \mathcal{Q}(\mathcal{M})$, let $\mathcal{N}(q_i)$ be the set of its adjacent quadrilaterals, and let r_{q_j} be the aspect ratio of a neighboring quadrilateral $q_j \in \mathcal{N}(q_i)$ (calculated as in Equation 7). The average aspect ratio difference for q_i is:

$$d_{q_i} = \frac{1}{|\mathcal{N}(q_i)|} \sum_{q_j \in \mathcal{N}(q_i)} |r_{q_i} - r_{q_j}| \quad (10)$$

The Adjacent Consistency sub-metric s_4 is then defined as the average of a consistency score based on this difference over all quadrilaterals:

$$s_4(\mathcal{Q}(\mathcal{M})) = \frac{1}{|\mathcal{Q}(\mathcal{M})|} \sum_{q \in \mathcal{Q}(\mathcal{M})} \frac{1}{1 + d_q} \quad (11)$$

B More Results

We present further comparative results in Figure 10 and Figure 11, respectively. **MeshAnythingV2** [16], due to its Adjacent tokenizer, frequently exhibits line-shaped discontinuities. **BPT** [19], employing a block patch-based tokenizer, is prone to generating patch-level holes. **DeepMesh** [23] 512M version demonstrates significant instability. While exhibiting better topological visual quality, likely due to the use of truncated training and global-reward DPO, it generates excessively dense meshes lacking the adaptive tessellation characteristic of artist-designed meshes. Our method, which incorporates M-DPO, achieves superior visual quality and mesh tessellation.

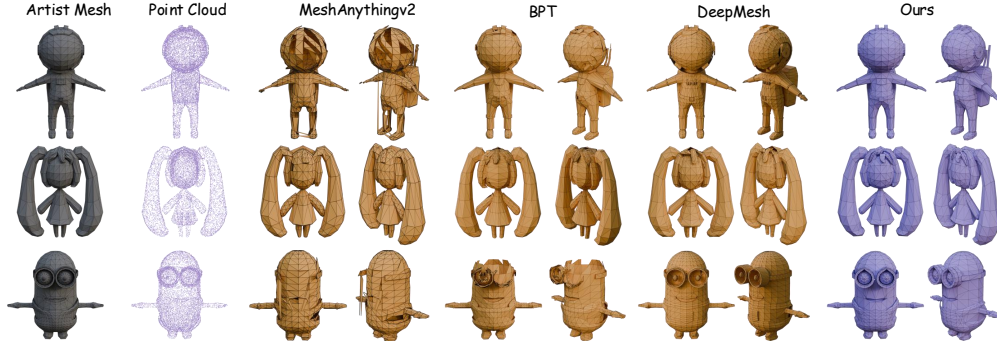


Figure 10: Comparative Results for Mesh-RFT and Baseline Methods on Artist-Designed Meshes.

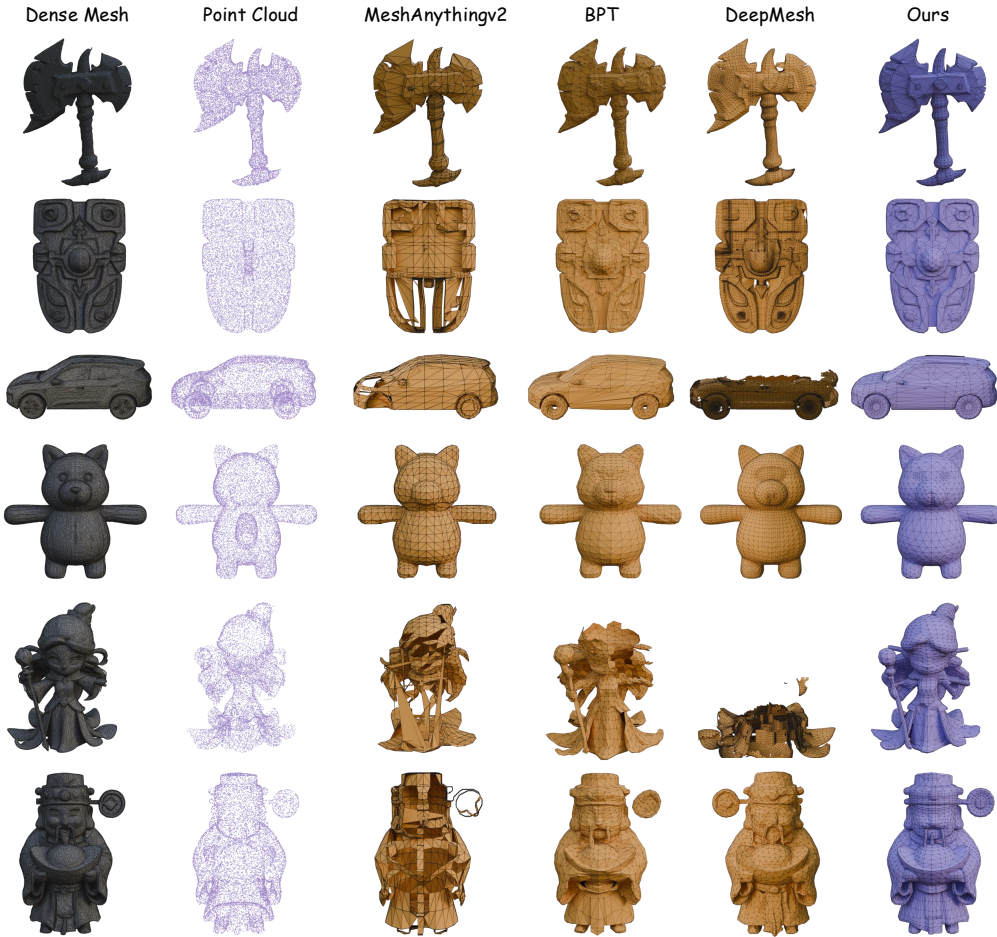


Figure 11: Comparative Results for Mesh-RFT and Baseline Methods on AI-Generated Dense Meshes.

C Limitations and Future Work

Computational Efficiency While Mesh-RFT demonstrates significant advancements in mesh generation, its computational efficiency warrants further investigation. Exploring engineering optimizations, potentially drawing inspiration from efficient inference techniques such as vLLM [68] employed in large language models, could lead to substantial accelerations.

Topological Correctness in Complex Geometries Ensuring robust topological correctness, particularly for intricate object geometries, necessitates continued research. As depicted in Figure 12, our model can exhibit topological defects such as holes in complex geometric scenarios. This may stem from limitations in the representational capacity of the point cloud encoder to capture fine-grained details within these complex structures. Future directions could involve leveraging more powerful, pre-trained point cloud encoders, increasing the number of tokens utilized, and scaling the decoder parameters to enhance the model’s ability to discern intricate geometric features.

Conditioning Modality As illustrated in Figure 12, dense meshes generated by Hunyuan2.0 [48] can sometimes exhibit a loss of fine details. Furthermore, conditioning on point clouds sampled from watertight dense meshes may exacerbate this information loss. Future work could explore alternative conditioning strategies, potentially bypassing the intermediate dense mesh representation and directly generating artist-quality meshes from image inputs (image-to-mesh generation).

Topology Reward Refinement The current reward function is relatively basic. Future research could focus on exploring more generalized and sophisticated topology rewards, as well as integrating real-time, state-of-the-art reinforcement learning strategies [69, 70].

Addressing these limitations will be crucial for broadening the applicability and enhancing the robustness of Mesh-RFT across a wider range of diverse and challenging 3D modeling tasks.



Figure 12: **Limitations of Mesh-RFT.** Examples showcasing potential topological defects (holes) in complex geometries and loss of fine details in generated meshes.