QUADGPT: NATIVE QUADRILATERAL MESH GENERATION WITH AUTOREGRESSIVE MODELS

Anonymous authors
Paper under double-blind review

Figure 1: **QuadGPT** can generate diverse, high-quality quad meshes conditioned on point clouds.

ABSTRACT

The generation of quadrilateral-dominant meshes is a cornerstone of professional 3D content creation. However, existing generative models generate quad meshes by first generating triangle meshes and then merging triangles into quadrilaterals with some specific rules, which typically produces quad meshes with poor topology. In this paper, we introduce QuadGPT, the first autoregressive framework for generating quadrilateral meshes in an end-to-end manner. QuadGPT formulates this as a sequence prediction paradigm, distinguished by two key innovations: a unified tokenization method to handle mixed topologies of triangles and quadrilaterals, and a specialized Reinforcement Learning fine-tuning method tDPO for better generation quality. Extensive experiments demonstrate that QuadGPT significantly surpasses previous triangle-to-quad conversion pipelines in both geometric accuracy and topological quality. Our work establishes a new benchmark for native quad-mesh generation and showcases the power of combining large-scale autoregressive models with topology-aware RL refinement for creating structured 3D assets.

1 INTRODUCTION

In the creation of high-fidelity 3D assets for game development, quadrilateral-dominant meshes play a fundamental role in ensuring modeling efficiency, deformation stability, and animation readiness. The structured topology provided by quad-dominant layouts facilitates smoother subdivision surfaces, more natural articulation, and easier UV unwrapping, all of which are essential in production environments (Lei et al., 2025). Consequently, quad meshes have become the industry standard, especially in character and organic modeling, where clean edge flow and controllable curvature are critical.

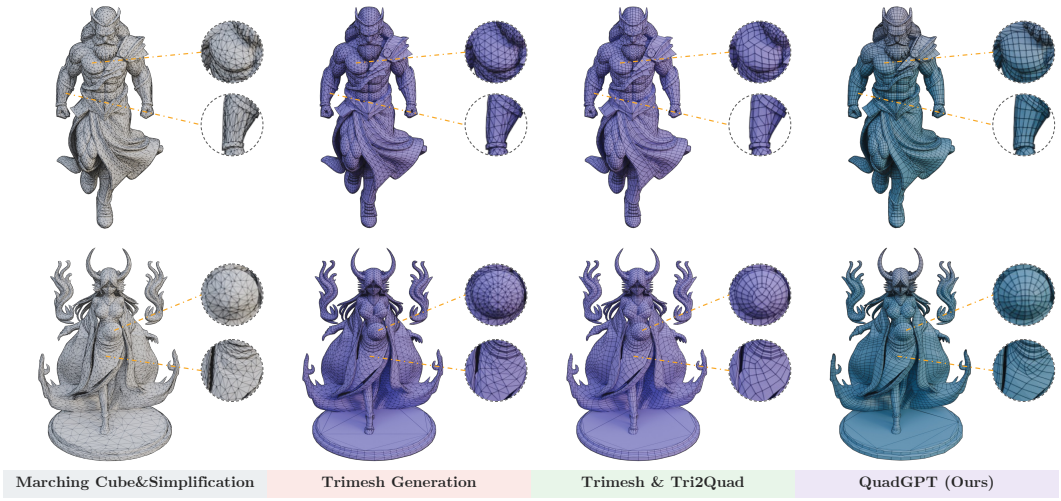

Figure 2: **Comparison of topological quality across different pipelines.** Iso-surfacing methods produce dense, unstructured triangular meshes. Autoregressive triangle generation followed by heuristic conversion fails to create coherent structure. Our QuadGPT directly generates native quadrilateral meshes with clean, artist-friendly edge flow.

In the pursuit of automated generation of artist-ready 3D assets from inputs such as text or images, existing approaches typically decouple the problem into two separate stages: geometry generation and topology generation. For geometry, latent diffusion models (LDMs) have achieved remarkable success in generating 3D shapes (Zhao et al., 2025b; Li et al., 2025d) via implicit representations like SDF (Li et al., 2024). Meshes are then extracted with iso-surface algorithms (Lorensen & Cline, 1998; Shen et al., 2021; 2023), inevitably resulting in unstructured dense meshes. On the other hand, topology-focused methods using cross-field guidance (Huang et al., 2018; Dong et al., 2025a) aim for structure but are often not robust, requiring pristine input meshes and failing to produce the adaptive, artist-like tessellation where polygon density matches geometric complexity.

More recently, autoregressive mesh generation methods such as MeshAnything (Chen et al., 2025b), BPT (Weng et al., 2025) and Mesh-RFT (Liu et al., 2025) have shown promise by modeling mesh sequences with Transformer architectures. While these approaches capture artist-like topology, they remain limited to generating only triangular meshes. Converting these outputs into quadrilateral meshes still requires triangle-merging algorithms that often break natural edge flow and introduce artifacts, as demonstrated in Figure 2. Consequently, even high-quality triangle meshes are hard to translate into production-ready quad layouts, highlighting a fundamental discrepancy between generated 3D assets and industrial applications.

To address these challenges, we propose an end-to-end autoregressive framework for direct generation of native quadrilateral meshes. Our model consumes a point cloud as input and produces a structured face sequence as output. Recognizing that artist-crafted meshes are typically quad-dominant yet usually incorporate a small number of triangles, we design a novel unified representation that explicitly supports mixed-element topologies through a tailored padding strategy for triangular faces. For computation efficiency, we employ an Hourglass Transformer architecture that first condenses the face sequence and subsequently compresses vertex information. The model is trained using a truncated sequence strategy, enabling support for high-poly meshes. To further improve topology quality, we introduce a reinforcement learning fine-tuning phase with truncated direct preference optimization (tDPO) that rewards the formation of coherent edge loops, a characteristic feature of professionally designed assets. Our reinforcement learning framework is specifically designed to evaluate and compare truncated sequences, ensuring effective optimization even for large-scale meshes.

Extensive experiments on the Toys4K dataset (Stojanov et al., 2021) confirm that QuadGPT consistently generates higher-quality 3D meshes than state-of-the-art baselines. By leveraging a large-scale, carefully curated dataset and comprehensive pre-training and post-training protocols, we fur-

ther evaluate QuadGPT on dense meshes generated by Hunyuan3D (Zhao et al., 2025b). The model demonstrates robust performance across both soft-surface models (e.g., human characters) and hard-surface objects (e.g., props). To ensure a rigorous comparison, we trained an triangle-only variant (TriGPT) followed by triangle-to-quad conversion. As shown in Figure 2, QuadGPT's native quadrilateral architecture yields significantly superior topology. This breakthrough establishes QuadGPT as an unequivocally state-of-the-art solution, effectively bridging the gap between text/image inputs and production-ready 3D artist meshes.

The main contributions of this paper are summarized as follows:

- We present QuadGPT, the first autoregressive model that generates native quad-dominant meshes in an end-to-end manner.
- We propose a unified sequence representation for mixed-element meshes with a padding-based serialization, enabling the scalable process of heterogeneous mesh topologies.
- We introduce tDPO, which is designed to optimize global quadrilateral flow through a novel reward mechanism that encourages the formation of structured edge loops.
- QuadGPT achieves state-of-the-art performance, producing game-ready meshes that exceed existing methods in both geometric and topological quality.

2 Related Work

Indirect and Field-Guided Mesh Generation. A dominant paradigm in 3D generation relies on continuous neural representations like vecsets (Zhang et al., 2023; 2024; Li et al., 2024; Zhao et al., 2023; Wu et al., 2024; Li et al., 2025b; Zhao et al., 2025b) or sparse voxels (Xiang et al., 2024; Ye et al., 2025; He et al., 2025; Wu et al., 2025; Li et al., 2025d). A universal limitation of these methods is their reliance on an iso-surfacing step, such as Marching Cubes (Lorensen & Cline, 1998), which invariably yields dense, topologically unstructured **triangular meshes**. Concurrently, traditional approaches to quadrilateral meshing are predominantly guided by cross-field computation. These methods are either optimization-based, requiring slow, per-shape optimization (Bommes et al., 2009; Jakob et al., 2015; Huang et al., 2018; Bommes et al., 2013; Ebke et al., 2016; Campen & Kobbelt, 2014; Jiang et al., 2015; Diamanti et al., 2015), or more recent learning-based techniques that accelerate field prediction (Dielen et al., 2021; Li et al., 2025c; Dong et al., 2025b;a). However, all field-guided methods depend on multi-stage pipelines that are not end-to-end generative frameworks.

Native Triangle Mesh Generation. To address the limitations of indirect methods, a promising direction has been the direct autoregressive generation of mesh sequences. This field, pioneered by MeshGPT (Siddiqui et al., 2024), has seen rapid progress. Subsequent work has largely focused on three key areas: (1) developing more efficient tokenization and compression schemes to manage long sequences (Chen et al., 2024; 2025c; Tang et al., 2025; Weng et al., 2025; Lionar et al., 2025; Song et al., 2025; Kim et al., 2025); (2) achieving scalability to tens of thousands of faces through architectural innovations, most notably with the Hourglass Transformer in Meshtron (Hao et al., 2024); and (3) enhancing output quality and alignment with human preferences via reinforcement learning, as demonstrated by DeepMesh (Zhao et al., 2025a) and Mesh-RFT (Liu et al., 2025). Complementary efforts have focused on areas such as acceleration (Chen et al., 2025a; Wang et al., 2025a;b) and continuous level-of-detail (Zhang et al., 2025). Despite these significant advances, all existing methods in this domain are fundamentally confined to generating **triangular meshes**. This reveals a critical gap between the state-of-the-art in generative modeling and the practical need for industry-standard quadrilateral assets. Unlike the above works, QuadGPT makes the first attempt to bridge this gap, presenting a scalable generative model for native quadrilateral mesh generation.

3 QuadGPT

Our approach, QuadGPT, introduces the first autoregressive framework for the direct generation of native quadrilateral and mixed-element meshes. The methodology consists of three core pillars: (1) a unified serialization scheme to represent mixed-topology meshes as a single token sequence; (2) a powerful autoregressive architecture for generative pre-training; and (3) tDPO stage with a novel

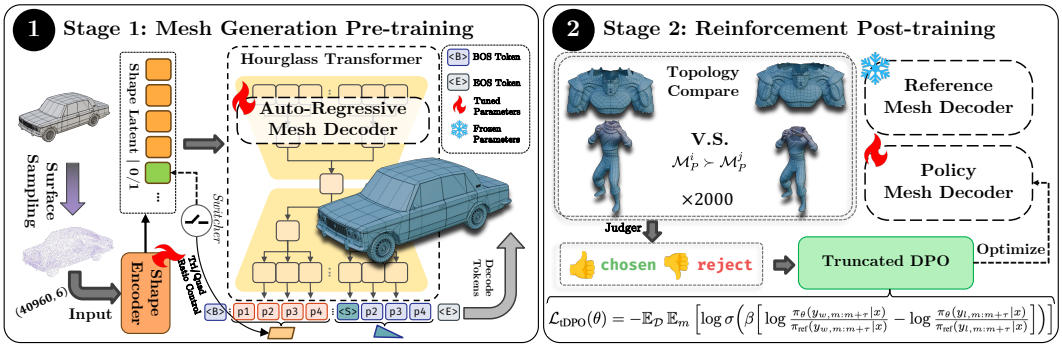

Figure 3: **Pipeline of QuadGPT.** First, an autoregressive Hourglass Transformer is pre-trained to generate mesh sequences conditioned on an input point cloud. Subsequently, the model is fine-tuned using Truncated Direct Preference Optimization (tDPO), where a preference dataset is automatically constructed by comparing truncated sequences via a novel topological reward.

topological reward for direct topological optimization. An overview of our pipeline is presented in Figure 3.

3.1 Unified Serialization for Mixed-Element Meshes

We formulate mixed-element mesh generation as a sequence prediction problem, transforming a mesh $\mathcal{M}$ into a single, linear sequence of discrete integer tokens. Our serialization scheme is designed to be canonical and to uniformly handle both triangular ($n = 3$) and quadrilateral ($n = 4$) faces. A mesh is represented at three hierarchical levels:

$$\begin{aligned} \mathcal{M} &= \{\mathbf{f}^1,\ \mathbf{f}^2,\ \ldots\ \mathbf{f}^{N_f}\} && \texttt{Face Level} \\ &= \{\mathcal{S}(\mathbf{f}^1),\ \mathcal{S}(\mathbf{f}^2),\ \ldots\ \mathcal{S}(\mathbf{f}^{N_f})\} && \texttt{Token Block Level} \\ &= \{\underbrace{\tau_{\text{pad}}, \ldots, \mathbf{c}_1^1, \ldots, \mathbf{c}_3^1}_{\text{e.g., Triangle Block}},\ \underbrace{\mathbf{c}_1^2, \ldots, \mathbf{c}_4^2}_{\text{e.g., Quad Block}}\ , \ldots\} && \texttt{Coordinate} - \texttt{Token Level} \end{aligned} \tag{1}$$

where $\mathcal{S}(\cdot)$ is the face serialization function and τ_{pad} is a special padding token.

Canonical Representation. To ensure a deterministic sequence for any given geometry, we first establish a canonical representation. Vertex coordinates are normalized to a $[-0.95, 0.95]^3$ cube and then quantized. To minimize precision loss, we employ a high-resolution **1024-level (10-bit)** quantization, mapping each coordinate to an integer in $\{0, 1, \ldots, 1023\}$. All unique vertices are then sorted lexicographically by (z, x, y) coordinates, and the face set is sorted accordingly. This multi-step process guarantees a unique mapping from mesh to sequence.

Unified Token Block Structure. The cornerstone of our approach is a **unified fixed-length block representation**. Each face, regardless of its valence, is tokenized into a consistent 12-token block using the padding token τ_{pad} (integer value 1024). A quadrilateral face is tokenized by directly flattening its $4 \times 3 = 12$ vertex coordinate tokens. A triangular face is prepended with three τ_{pad} tokens, followed by its $3 \times 3 = 9$ coordinate tokens, also forming a 12-token block. This design allows the model to implicitly learn the face type from the presence of padding.

This unified structure offers several advantages: it creates a simple, highly parallelizable tokenization process that scales effortlessly; it simplifies the model architecture; and it allows the Transformer to naturally differentiate face types without explicit type tokens.

3.2 Autoregressive Pre-training

The pre-training stage is designed to teach QuadGPT the fundamental distribution of mesh geometry and connectivity by training it to predict the next token in a sequence. The model is optimized using a standard cross-entropy loss objective:

$$\mathcal{L}_{\text{ce}} = \text{CrossEntropy}(\hat{\mathbf{S}}[:-1], \mathbf{S}[1:]), \tag{2}$$

where $\mathbf{S}$ is the ground-truth token sequence and $\hat{\mathbf{S}}$ represents the predicted logits. Our approach incorporates a powerful hierarchical architecture, shape conditioning, and a specialized training strategy tailored for structured geometric data.

Hierarchical Model Architecture. Rather than treating mesh generation as a generic sequence task, we leverage the inherent hierarchical structure of mesh data. To this end, we utilize the **Hourglass Transformer** architecture (Hao et al., 2024; Nawrot et al., 2021), which processes the input sequence at multiple levels of abstraction. Let the input token sequence embeddings be $\mathbf{E}^{(0)} \in \mathbb{R}^{L \times D_0}$. The architecture employs a series of causality-preserving shortening layers to create a computational bottleneck. The sequence is first processed by a Transformer block and then shortened by a factor of 4, and subsequently by a factor of 3:

$$\mathbf{E}^{(1)} = \texttt{Shorten}_4(\texttt{TransformerBlock}_1(\mathbf{E}^{(0)})) \in \mathbb{R}^{(L/4) \times D_1} \tag{3}$$

$$\mathbf{E}^{(2)} = \texttt{Shorten}_3(\texttt{TransformerBlock}_2(\mathbf{E}^{(1)})) \in \mathbb{R}^{(L/12) \times D_2} \tag{4}$$

This hierarchical processing enables the model to efficiently capture high-level global context in its bottleneck layers and fine-grained local details in its outer layers before upsampling back to the original sequence length for prediction.

Shape and Topological Conditioning. QuadGPT's generation is guided by two primary conditions. First, a point cloud with normals, $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^6\}_{i=1}^{N_p}$, is encoded into a global shape embedding $\mathbf{E}_{\text{shape}}$ by a pre-trained Michelangelo encoder (Zhao et al., 2023). To ensure this geometric context remains influential throughout the generation of long sequences, the embedding is supplied to the decoder via cross-attention:

$$\mathbf{H}' = \texttt{CrossAttn}(\mathbf{H}, \mathbf{E}_{\text{shape}}, \mathbf{E}_{\text{shape}}), \tag{5}$$

where $\mathbf{H}$ represents the decoder's hidden states. Second, to enable our training curriculum, we introduce a learnable embedding conditioned on a quad-dominance parameter $r \in [0, 1]$. This parameter explicitly controls the target ratio of face types, from purely triangular ($r = 0$) to mixed quadrilateral ($r = 1$), providing the mechanism for our curriculum learning strategy.

Training Strategy. Our training strategy combines truncated sequence training for efficiency with a novel curriculum for stability. To manage the long sequences of high-resolution meshes, we employ **truncated training**, using fixed-length segments (e.g., 36,864 tokens) to enable efficient, large-batch processing. Furthermore, to ensure stable learning of complex quadrilateral topology, we introduce a **curriculum learning** strategy. We first initialize QuadGPT with weights from a model pre-trained exclusively on triangular meshes. We then progressively finetune this model, using our quad-dominance condition r to gradually anneal the training data distribution from purely triangular ($r = 0$) to quad-dominant ($r \to 1$). This graduated exposure allows the model to master basic geometric syntax before tackling the more complex rules of quadrilateral topology, significantly improving stability and convergence speed.

Data Strategy. The scarcity of dedicated quad-mesh datasets is addressed by a novel curation pipeline. Starting from diverse 3D sources, we apply automated triangle-to-quad conversion and multi-stage quality filtering to select 1.3 million high-quality models. This dataset is pivotal for training QuadGPT. Further details on our data curation can be found in Appendix A.

3.3 Topological Refinement with Reinforcement Learning

While pre-training teaches syntactic validity, the cross-entropy loss is a local objective that cannot optimize for global, emergent properties like clean topology. To address this, we introduce a reinforcement learning (RL) stage using Direct Preference Optimization (DPO) (Rafailov et al., 2023) to explicitly align our model with the topological structures preferred in professional 3D workflows.

Topological Scoring Standard. Our alignment is guided by a specialized scoring standard designed to evaluate the quality of *generated mesh subsequences*. The reward function quantifies topological integrity by primarily rewarding the formation of long, continuous edge loops (L_{avg}) and penalizing generation fractures (R_{frac}). These metrics are computed automatically, providing a scalable signal for topological quality. The detailed formulation of these metrics is provided in Appendix B.

Truncated DPO(tDPO)-based Post-Training. We finetune the pretrained QuadGPT policy by forming preference pairs using our topological rewards and optimizing a DPO-style objective on the collected pairs. Let π_θ denote the trainable policy being fine-tuned, π_{ref} the frozen reference model, and $\beta > 0$ a parameter controlling the deviation from π_{ref}. The current policy π_θ produces candidates for input x, which are ranked by topological rewards to yield preference pairs $\big(y_w, y_l\big)$ with

$$L_{\text{avg}}(y_w) > L_{\text{avg}}(y_l), \tag{6}$$

$$R_{\text{frac}}(y_w) < R_{\text{frac}}(y_l). \tag{7}$$

We assume pairwise preferences follow a Bradley–Terry (BT) likelihood parameterized by an implicit reward $r_\theta(y|x)$. In KL-regularized policy optimization with reference π_{ref}, the optimal policy satisfies

$$\pi_\theta(y|x) \;\propto\; \pi_{\text{ref}}(y|x)\,\exp\big(r_\theta(y|x)/\beta\big), \tag{8}$$

which implies the implicit reward is (up to an x-only constant $c(x)$):

$$r_\theta(y|x) \;=\; \beta[\log \pi_\theta(y|x) - \log \pi_{\text{ref}}(y|x)] + c(x). \tag{9}$$

Substituting equation 9 into the BT model cancels $c(x)$ and yields

$$\mathbb{P}_\theta(y_w \succ y_l \,|\, x) = \sigma\left(\beta\Big[\log \tfrac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log \tfrac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\Big]\right). \tag{10}$$

where $\sigma(z) = 1/(1 + e^{-z})$.

To make this process computationally tractable for long sequences, we train on random prefixes of length $m \sim \mathcal{U}\{1, \ldots, L\}$, where L is the total length of the mesh. Let $y_{m:m+\tau}$ denote the sequence from the prefix to the truncation window, where τ is the window length (e.g., 36,864 tokens), then maximizing the BT likelihood equation 10 gives the tDPO loss

$$\mathcal{L}_{\text{tDPO}}(\theta) = -\mathbb{E}_{\mathcal{D}}\,\mathbb{E}_m\left[\log \sigma\Big(\beta\Big[\log \tfrac{\pi_\theta(y_{w,m:m+\tau}|x)}{\pi_{\text{ref}}(y_{w,m:m+\tau}|x)} - \log \tfrac{\pi_\theta(y_{l,m:m+\tau}|x)}{\pi_{\text{ref}}(y_{l,m:m+\tau}|x)}\Big]\Big)\right]. \tag{11}$$

tDPO optimizes each face sequence block $\mathcal{S}(\mathbf{f}^j)$ within the truncation window. This teaches QuadGPT to make locally optimal decisions that lead to globally superior topology. After tDPO optimization, QuadGPT presents a higher quality quad-mesh with more structured edge loops and reduced fractures.

4 Experiments

4.1 Experimental Settings

Datasets QuadGPT is pretrained on a curated dataset of 1.3 million quad-dominant models, assembled through an extensive collection, conversion, and filtering pipeline from sources including ShapeNetV2 (Chang et al., 2015), 3D-FUTURE (Fu et al., 2021), Objaverse (Deitke et al., 2023b), Objaverse-XL (Deitke et al., 2023a), and proprietary licensed assets. For preference alignment, we construct a specialized post-training dataset built upon 500 diverse, high-quality dense meshes (including both hard-surface and organic models) generated by Hunyuan3D 2.5 (Lai et al., 2025). After an initial quality filter via rejection sampling, we generate multiple candidates from these source meshes to construct an initial set of approximately 2,000 preference pairs for fine-tuning. This core set of 500 meshes serves as the foundation for DPO. To enhance the model's robustness against noisy, dense inputs, point clouds during pre-training are densely sampled at 40,960 points and augmented with random perturbations.

Implementation Details We pretrain QuadGPT on a cluster of 64 NVIDIA A100 GPUs for 7 days using the AdamW optimizer (Loshchilov & Hutter, 2019) ($\beta_1 = 0.9, \beta_2 = 0.95$) with a learning rate of 1e-4 and a linear warmup schedule. The **Decoder** is a 1.1B parameter model featuring 24 Transformer layers arranged in a three-stage hourglass architecture. This architecture employs linear downsample layers with factors of 4 and 3 to efficiently process long sequences. The subsequent RL fine-tuning stage is performed for 4 hours on the same hardware setup, using a reduced learning rate

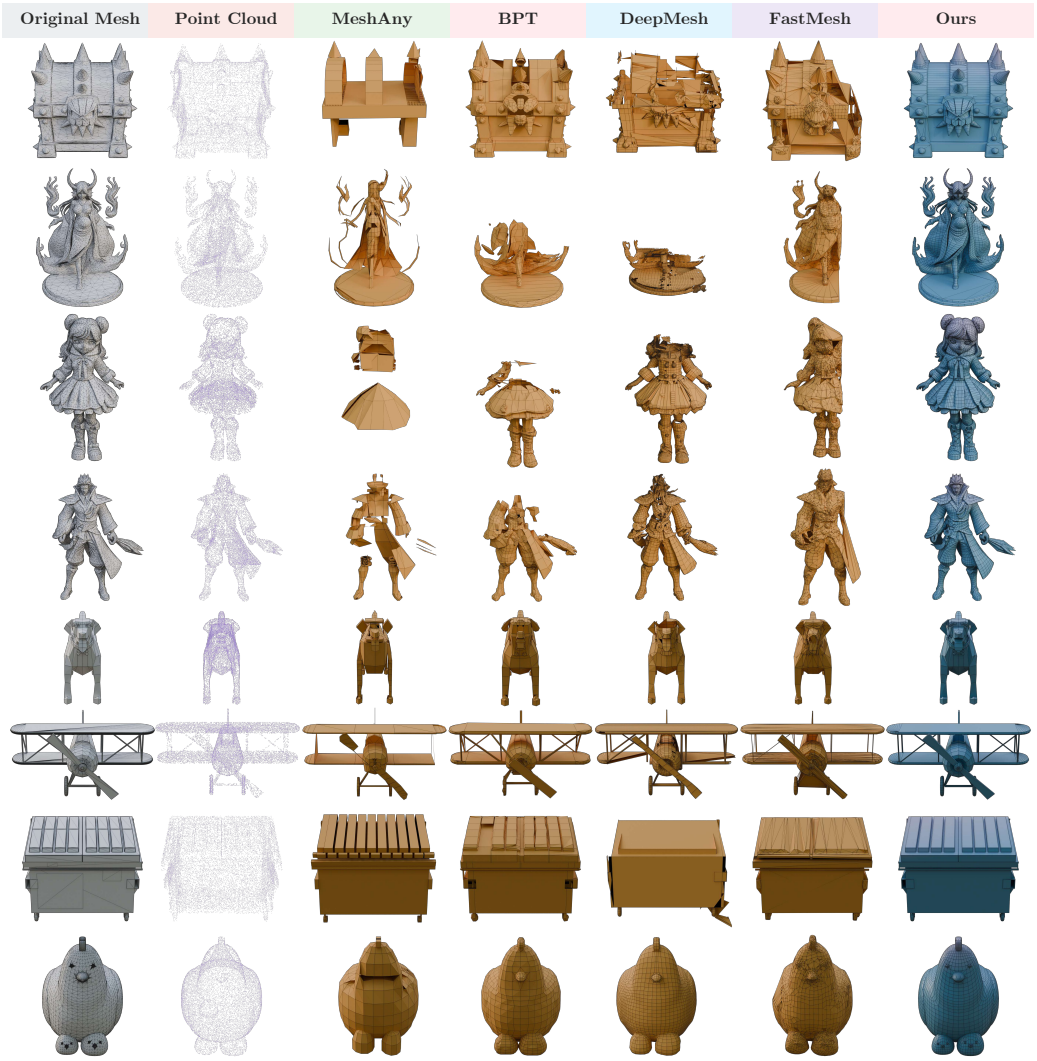

Figure 4: **Qualitative Comparison against Indirect Autoregressive Pipelines.** The top four rows show results on out-of-distribution dense meshes generated by Hunyuan3D (Lai et al., 2025), while the bottom four rows showcase performance on artist-designed meshes. Baseline methods followed by tri-to-quad conversion often produce topological artifacts and lose geometric detail. QuadGPT consistently generates meshes with superior topological coherence and fidelity across both domains.

of 1e-7. Both pre-training and fine-tuning leverage a truncated sequence strategy to manage the long contexts of high-resolution meshes. During inference, the architecture supports a context window of 36,864 tokens. We generate meshes using a combination of top-k and nucleus (top-p) sampling with $k = 10$ and $p = 0.95$, along with a temperature of $T = 0.5$, to balance output diversity and stability. To optimize performance, inference is accelerated through a custom implementation of KV caching and CUDA graphs, specifically tailored for the hourglass architecture. This achieves a generation speed of approximately 230 tokens per second on a single A100 GPU.

Baselines. We evaluate QuadGPT against two primary categories of state-of-the-art mesh generation methods. The first category comprises leading autoregressive models that generate triangular meshes, including **MeshAnythingV2** (Chen et al., 2025c), **BPT** (Weng et al., 2025), **DeepMesh** (Zhao et al., 2025a), and **FastMesh** (Kim et al., 2025). Since these methods are fundamentally designed to produce triangular outputs, we apply a robust triangle-to-quadrilateral conversion algorithm like (Muntoni & Cignoni, 2021) as a post-processing step to facilitate a fair com-

Figure 5: **Qualitative Comparison against a Field-Guided Method.** Field-guided methods like QuadriFlow can be unstable on meshes with complex topology or sharp features.

Table 1: **Quantitative comparison with other baselines in Artist and Dense Meshes.** Our approach achieves superior performance compared to existing baselines. Quadriflow* are computed only on the subset of inputs for which it successfully generated a mesh; its user study score incorporates a score of 0 for all failure cases.

Data Type	Dense Meshes				Artist Meshes			
Metrics	CD ↓	HD ↓	QR ↑	US ↑	CD ↓	HD ↓	QR ↑	US ↑
Quadriflow* (Huang et al., 2018)	**0.045**	**0.099**	**100%**	1.6	0.281	0.531	**100%**	0.3
MeshAnythingv2 (Chen et al., 2025c)	0.153	0.394	53%	1.4	0.096	0.251	60%	2.1
BPT (Weng et al., 2025)	0.115	0.283	43%	2.7	0.051	0.125	49%	3.1
DeepMesh (Zhao et al., 2025a)	0.246	0.435	64%	3.3	0.236	0.417	66%	2.8
FastMesh (Kim et al., 2025)	0.105	0.257	3%	1.1	0.052	0.141	17%	1.9
Ours	0.057	0.147	80%	**4.9**	**0.043**	**0.095**	78%	**4.8**

parison. The second category represents specialized quad-meshing techniques, for which we include the well-established field-guided method **QuadriFlow** (Huang et al., 2018). This selection provides a comprehensive benchmark against both the latest in generative modeling and classic, topology-focused approaches.

4.2 Qualitative Results

We first present a qualitative comparison of QuadGPT against existing baselines across two distinct domains: challenging, out-of-distribution dense meshes from other AI models, and in-distribution, high-quality artist-designed meshes. Visual inspection is crucial, as it reveals the subtle yet critical differences in topological quality that metrics alone cannot fully capture. As illustrated in Figure 4, the indirect pipeline of converting triangular meshes from autoregressive baselines often struggles. These methods frequently produce meshes with significant topological artifacts, missing geometric details, or overly simplified structures that fail to capture the original shape's nuance. In contrast, QuadGPT consistently generates meshes that are significantly more coherent and artistically plausible, producing the clean edge flow characteristic of professional work. Our model demonstrates strong robustness on challenging AI-generated assets and achieves near-perfect topological reconstruction on artist-designed meshes. Figure 5 compares our method against the field-guided approach of QuadriFlow. The baseline exhibits significant instability on meshes with complex topology or sharp features, often resulting in severe geometric degradation or catastrophic failures. QuadGPT, in contrast, demonstrates exceptional robustness, faithfully reconstructing geometry while maintaining high-quality, structured topology in all examples.

4.3 Quantitative Results

Table 1 summarizes the quantitative comparison of QuadGPT against all baselines across both artist-designed and AI-generated dense meshes. We evaluate geometric fidelity using Chamfer Distance (CD) and Hausdorff Distance (HD), topological quality via Quad Ratio (QR), and perceptual quality

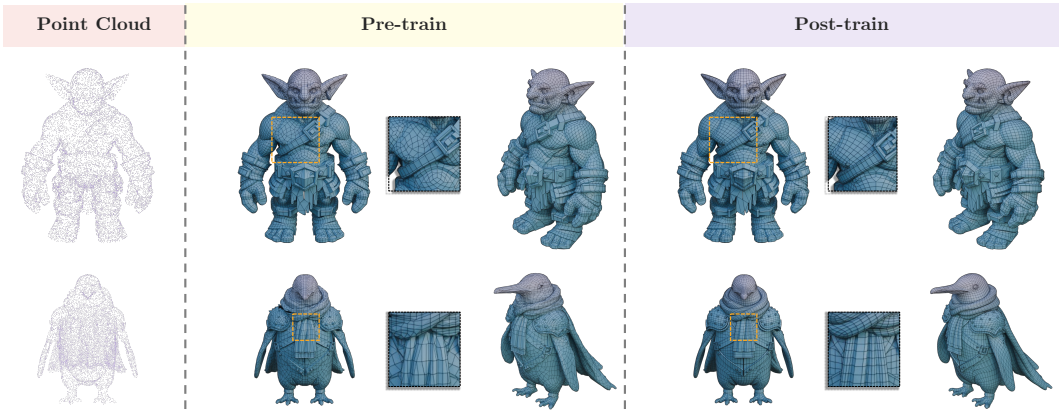

Figure 6: **Effectiveness of tDPO-Pro.** Our comprehensive training strategy significantly enhances both the geometric quality and structural integrity of the generated quad-meshes.

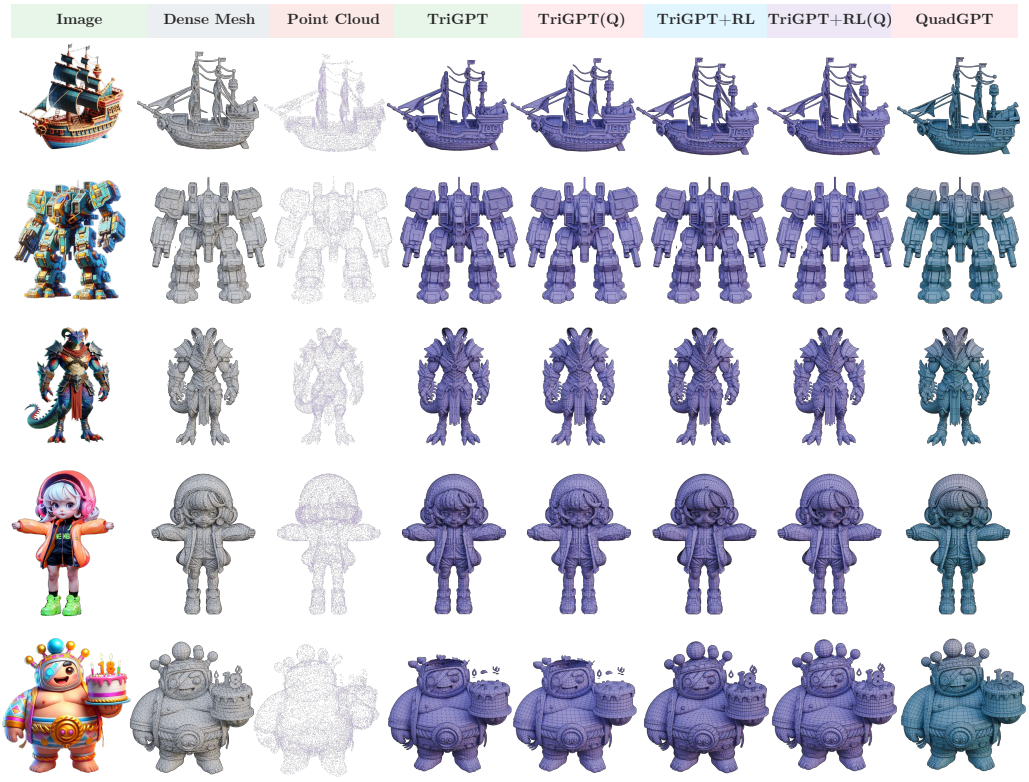

Figure 7: **Native Generation vs. Conversion Pipeline.** (Q) denotes the use of a triangle-to-quad conversion step. Although TriGPT employs the same RL fine-tuning to mitigate fractures, its topological quality is inherently constrained by the post-processing conversion, yielding significantly inferior edge flow compared to our native QuadGPT.

through a comprehensive user study (US). The results demonstrate that QuadGPT consistently and significantly outperforms competing approaches in both quality and robustness. The user study, where experts ranked the outputs of all six methods from best (5 points) to worst (0 points), reveals a decisive preference for our method. This confirms that its advantages in producing production-ready assets are not only quantitatively measurable but also perceptually significant.

4.4 Ablation Studies

4.4.1 Effectiveness of tDPO

We analyze the impact of our DPO components by comparing three variants: standard DPO (fine-tuning on full, low-face-count meshes), tDPO (truncated training with a basic fracture penalty), and our full tDPO-Pro model (truncated training with the complete topological reward). Table 2 shows that standard DPO fails to generalize to complex meshes. In contrast, tDPO dramatically improves performance, while tDPO-Pro achieves the best results across all metrics. This improvement, visualized in Figure 6, provides strong empirical support for our comprehensive, topology-aware, truncated DPO framework.

4.4.2 Effectiveness of Curriculum-based Pre-training

We validate our curriculum learning strategy by comparing it against a model trained directly on quad-dominant meshes from scratch ("From Scratch"). As shown in Table 2, the "From Scratch" model struggles to converge, resulting in poor geometric fidelity. This difficulty arises because predicting a quadrilateral face is inherently more complex than predicting a triangle, as it is topologically equivalent to predicting two correlated triangles simultaneously. Initializing weights from a converged triangle-generation model ("Finetune") yields significantly superior results. This confirms that our curriculum strategy leveraging the simpler task of triangle generation as a warmup is essential for establishing the stable geometric foundation required to master the more challenging patterns of quadrilateral topology.

Table 2: **Quantitative Evaluation of Training Strategy.** User study scores reflect expert rankings of the five methods, ranging from 0 (worst) to 4 (best).

Method	CD ↓	HD ↓	QR ↑	US ↑
From Scratch	0.081	0.203	75%	0.6
Finetune	0.065	0.167	72%	1.3
DPO	0.073	0.188	74%	1.1
tDPO	0.061	0.156	78%	3.3
tDPO-Pro	**0.057**	**0.147**	**80%**	**3.7**

4.4.3 Native Generation vs. Conversion Pipeline

To isolate the benefits of our native approach, we introduce a strong baseline, **TriGPT**, which generates triangles that are then converted to quads. To ensure a fair comparison, TriGPT shares the **identical architecture, 1.3 million mesh training dataset, and tDPO reinforcement learning strategy** as QuadGPT. This setup controls for all confounding variables, testing only the efficacy of the end-to-end native pipeline versus the generation-then-conversion paradigm.

The results in Table 3 and Figure 7 confirm our hypothesis. While the highly optimized triangle baseline (TriGPT+RL) achieves slightly better geometric scores (CD/HD), it cannot match the topological quality of our native approach. QuadGPT demonstrates a substantially higher Quad Ratio (QR) and, crucially, a **user preference score 2.6x higher** than the strongest baseline. This decisive gap in perceptual quality confirms that while post-hoc conversion struggles to create coherent global structures, our end-to-end native framework excels at learning the artist-preferred topologies, validating its superior practical utility.

Table 3: **Native vs. Conversion Pipeline.** QuadGPT is compared against a strong triangle-generation baseline (TriGPT), both with and without RL. US scores are expert rankings (0–2).

Method	CD ↓	HD ↓	QR ↑	US ↑
TriGPT(Q)	0.062	0.160	70%	0.2
TriGPT+RL(Q)	**0.051**	**0.138**	72%	0.5
QuadGPT (Ours)	0.057	0.147	**80%**	**1.3**

5 Conclusion

We present QuadGPT, the first autoregressive framework that directly generates native quadrilateral and mixed-element meshes. It achieves state-of-the-art results in generative meshing, setting a new standard for both geometric fidelity and topological quality. Our scalable, neural-first approach departs from previous triangle-based methods and conversion pipelines, which rely on heuristic post-processing to approximate quad topology. By leveraging a unified serialization scheme and a novel topology-aware fine-tuning stage (tDPO), QuadGPT directly optimizes for global structure, making it well-suited for the automated creation of production-ready 3D assets.

6 Ethics Statement

This work introduces QuadGPT, an autoregressive framework for direct generation of production-ready, quad-dominant meshes. The model was trained on a blend of public and professionally sourced data, subjected to rigorous quality filtering. Our primary goal is to demonstrate that a scalable, end-to-end framework is effective for industrial-quality mesh generation, and we encourage the community to explore further along this scalable paradigm. To support practical adoption, a public API and online interface will be provided. The authors declare no conflicts of interest.

7 Reproducibility Statement

This paper presents a scalable autoregressive framework for native quad-mesh generation. Although full model release is not feasible at this stage, the method is comprehensively detailed in terms of data representation, network architecture, and training protocols to facilitate replication and extension. We emphasize that high-quality data curation, including rigorous collection, processing, and filtering of open-source data, is essential for achieving comparable performance. To support validation and downstream use, we will provide a public API and Code.

References

Blender Foundation. Blender. URL https://www.blender.org.

D. Bommes, M. Campen, H.-C. Ebke, P. Alliez, and L. Kobbelt. Integer-grid maps for reliable quad meshing. *ACM Transactions on Graphics (TOG)*, 32(4):98, 2013.

David Bommes, Henrik Zimmer, and Leif Kobbelt. Mixed-integer quadrangulation. *ACM Transactions on Graphics*, 28(3):1–10, 2009.

Marcel Campen and Leif Kobbelt. Dual strip weaving: Interactive design of quad layouts using elastica strips. *ACM Transactions on Graphics*, 33(6):183:1–183:10, 2014.

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.

Dian Chen, Yansong Qu, Xinyang Li, Ming Li, and Shengchuan Zhang. Xspecmesh: Quality-preserving auto-regressive mesh generation acceleration via multi-head speculative decoding. *arXiv preprint arXiv:2507.23777*, 2025a.

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, 2024.

Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Zhongang Cai, Lei Yang, Gang Yu, Guosheng Lin, and Chi Zhang. Meshanything: Artist-created mesh generation with autoregressive transformers. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=KGZAs8VcOM.

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. *IEEE International Conference on Computer Vision*, 2025c.

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.

O. Diamanti, A. Vaxman, D. Panozzo, and O. Sorkine-Hornung. Integrable polyvector fields. *ACM Transactions on Graphics (TOG)*, 34(4):38, 2015.

Alexander Dielen, Isaak Lim, Max Lyon, and Leif Kobbelt. Learning direction fields for quad mesh generation. In *Computer Graphics Forum*, volume 40, pp. 181–191. Wiley Online Library, 2021.

Qiujie Dong, Jiepeng Wang, Rui Xu, Cheng Lin, Yuan Liu, Shiqing Xin, Zichun Zhong, Xin Li, Changhe Tu, Taku Komura, et al. Crossgen: Learning and generating cross fields for quad meshing. *arXiv preprint arXiv:2506.07020*, 2025a.

Qiujie Dong, Huibiao Wen, Rui Xu, Shuangmin Chen, Jiaran Zhou, Shiqing Xin, Changhe Tu, Taku Komura, and Wenping Wang. Neurcross: A neural approach to computing cross fields for quad mesh generation. *ACM Trans. Graph.*, 44(4), 2025b.

Hans-Christian Ebke, Patrick Schmidt, Marcel Campen, and Leif Kobbelt. Interactively controlled quad remeshing of high resolution 3d models. *ACM Transactions on Graphics*, 35(6):218:1–218:13, November 2016.

Exoside. Exoside quad remesher: Automatic quad remeshing plugin, 2019. URL `https://exoside.com/`.

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.

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, 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.

Jingwei Huang, Yichao Zhou, Matthias Niessner, et al. Quadriflow: A scalable and robust method for quadrangulation. *Computer Graphics Forum*, 2018. ISSN 1467-8659.

Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. Instant field-aligned meshes. *ACM Transactions on Graphics*, 34(6):189, 2015.

Tengfei Jiang, Xianzhong Fang, Jin Huang, et al. Frame field generation through metric customization. *ACM Transactions on Graphics*, 34(4):1–11, 2015.

Jeonghwan Kim, Yushi Lan, Armando Fortes, Yongwei Chen, and Xingang Pan. Fastmesh: Efficient artistic mesh generation via component decoupling. *arXiv preprint arXiv:2508.19188*, 2025.

Zeqiang Lai, Yunfei Zhao, Haolin Liu, Zibo Zhao, Qingxiang Lin, Huiwen Shi, Xianghui Yang, Mingxin Yang, Shuhui Yang, Yifei Feng, et al. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. *arXiv preprint arXiv:2506.16504*, 2025.

Biwen Lei, Yang Li, Xinhai Liu, Shuhui Yang, Lixin Xu, and Chunchao Guo. Hunyuan3d studio: End-to-end ai pipeline for game-ready 3d asset generation, 2025.

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.

Yang Li, Victor Cheung, Xinhai Liu, Yuguang Chen, Zhongjin Luo, Biwen Lei, Haohan Weng, Zibo Zhao, Jingwei Huang, Zhuo Chen, et al. Auto-regressive surface cutting. *arXiv preprint arXiv:2506.18017*, 2025a.

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*, 2025b.

Zezeng Li, Zhihui Qi, Weimin Wang, Ziliang Wang, Junyi Duan, and Na Lei. Point2quad: Generating quad meshes from point clouds via face prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025c.

Zhihao Li, Yufei Wang, Heliang Zheng, Yihao Luo, and Bihan Wen. Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. *arXiv preprint arXiv:2505.14521*, 2025d.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

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

Jian Liu, Jing Xu, Song Guo, Jing Li, Jingfeng Guo, Jiaao Yu, Haohan Weng, Biwen Lei, Xianghui Yang, Zhuo Chen, et al. Mesh-rft: Enhancing mesh generation via fine-grained reinforcement fine-tuning. *arXiv preprint arXiv:2505.16761*, 2025.

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.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Alessandro Muntoni and Paolo Cignoni. PyMeshLab, January 2021.

Piotr Nawrot, Szymon Tworkowski, Michał Tyrolski, Łukasz Kaiser, Yuhuai Wu, Christian Szegedy, and Henryk Michalewski. Hierarchical transformers are more efficient language models. *arXiv preprint arXiv:2110.13711*, 2021.

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.

Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, 34:6087–6101, 2021.

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 Trans. Graph.*, 42(4):37–1, 2023.

Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19615–19625, 2024.

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

Stefan Stojanov, Anh Thai, and James M. Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *CVPR*, pp. 1798–1808, June 2021.

Jiaxiang Tang, Zhaoshuo Li, Zekun Hao, Xian Liu, Gang Zeng, Ming-Yu Liu, and Qinsheng Zhang. Edgerunner: Auto-regressive auto-encoder for artistic mesh generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=81cta3WQVI.

Marco Tarini, Nico Pietroni, Paolo Cignoni, Daniele Panozzo, and Enrico Puppo. Practical quad mesh implification. *Computer Graphics Forum (Special Issue of Eurographics 2010 Conference)*, 29(2), 200.

Tripo AI Team. Tripo ai: Intelligent remesher, 2025. URL `https://studio.tripo3d.ai/workspace/remesh`.

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*, 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. *IEEE International Conference on Computer Vision*, 2025b.

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

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, 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.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, et al. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *arXiv preprint arXiv:2505.17412*, 2025.

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. *arXiv preprint arXiv:2412.01506*, 2024.

Xinhao Yan, Jiachen Xu, Yang Li, Changfeng Ma, Yunhan Yang, Chunshi Wang, Zibo Zhao, Zeqiang Lai, Yunfei Zhao, Zhuo Chen, et al. X-part: high fidelity and structure coherent shape decomposition. *arXiv preprint arXiv:2509.08643*, 2025.

Chongjie Ye, Yushuang Wu, Ziteng Lu, Jiahao Chang, Xiaoyang Guo, Jiaqing Zhou, Hao Zhao, and Xiaoguang Han. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. *arXiv preprint arXiv:2503.22236*, 2025.

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

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

Xiang Zhang, Yawar Siddiqui, Armen Avetisyan, Chris Xie, Jakob Engel, and Henry Howard-Jenkins. Vertexregen: Mesh generation with continuous level of detail. *IEEE International Conference on Computer Vision*, 2025.

Ruowen Zhao, Junliang Ye, Zhengyi Wang, Guangce Liu, Yiwen Chen, Yikai Wang, and Jun Zhu. Deepmesh: Auto-regressive artist-mesh creation with reinforcement learning. *IEEE International Conference on Computer Vision*, 2025a.

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

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

APPENDIX

A DATASET CURATION PIPELINE

The foundation of QuadGPT is a large-scale, high-quality dataset. This section details our three-stage pipeline for its construction: data sourcing and augmentation, multi-stage quality filtering, and final compilation.

A.1 DATA SOURCING AND AUGMENTATION

The foundation of QuadGPT is a large-scale, high-quality dataset. This section details our three-stage pipeline for its construction: data sourcing and augmentation, multi-stage quality filtering, and final compilation. We aggregate 3D models from diverse sources, including Objaverse-XL Deitke et al. (2023a) and professional modeling repositories. To substantially augment our training data with quad-dominant examples, we developed a **Triangle-to-Quadrilateral Conversion Operator**. Given a triangle mesh $\mathcal{M}_T$, the goal is to find an optimal set of internal edges $E_{\text{int}} \subset \mathcal{M}_T$ to dissolve. We formulate this as an Integer Linear Programming (ILP) problem. For each edge $e \in E_{\text{int}}$, we define a binary decision variable $x_e \in \{0, 1\}$. The optimization objective is:

$$\text{maximize} \quad \sum_{e \in E_{\text{int}}} w_e \cdot x_e \tag{12}$$

subject to the constraint that for any triangle $t \in \mathcal{M}_T$,

$$\sum_{e \in \text{edges}(t)} x_e \leq 1 \tag{13}$$

The weight w_e is a quality score that favors dissolving edges that form well-conditioned quadrilaterals. Constraint 13 ensures each triangle participates in at most one merge operation, preserving manifold integrity. Following this conversion, we perform a crucial **geometric validation** step: any newly formed quadrilateral exhibiting a maximum interior angle greater than 150 degrees is deemed geometrically unstable and is split back into its two original constituent triangles. This ensures our automated pipeline produces only high-quality, well-shaped quadrilaterals.

A.2 MULTI-STAGE QUALITY FILTERING

The initial dataset, while extensive, contains numerous low-quality models. We employ a rigorous, two-stage filtering pipeline to ensure data fidelity. First, we apply a suite of **rule-based operators** to discard models with specific flaws, including high-aspect-ratio faces and patterns characteristic of poor automated decimation. A critical component is our **Fractured Geometry Detector** (Algorithm 1), which identifies open seams by performing a "test weld" and checking for a significant reduction in edge count without a loss of faces. Second, to filter for aesthetic quality, we trained a **vision-based quality assessment model** on a manually annotated corpus of 100,000 models, allowing us to automate the removal of assets with poor, albeit technically valid, edge flow.

A.3 FINAL DATASET COMPILATION

Following this pipeline, we selected models with face counts between 500 and 20,000 to form our final training dataset, comprising **1.3 million** high-quality, production-ready 3D models.

Algorithm 1: Fractured Geometry Detection

Input : Mesh Object $\mathcal{M}$
Output: Boolean 'has_fracture'

$\mathcal{M}'' \leftarrow$ Preprocess($\mathcal{M}$) /* Remove duplicates and loose geometry */
Components $\leftarrow$ SeparateIntoConnectedComponents($\mathcal{M}''$);
for *component in Components* **do**
  if *num_vertices(component)* ¡ τ_{vtx_min} **then**
    continue;
    /* Ignore small fragments */
  end
  if *not IsManifold(component)* **then**
    $E_{\text{before}}, F_{\text{before}} \leftarrow$ GetEdgeAndFaceCount(component);
    component' $\leftarrow$ MergeVerticesByDistance(component, threshold $= \tau_{\text{weld}}$);
    $E_{\text{after}}, F_{\text{after}} \leftarrow$ GetEdgeAndFaceCount(component');
    if $E_{before} - E_{after} > \tau_{edge\_delta}$ ***and*** $F_{before} = F_{after}$ **then**
      return true;
      /* Fracture detected */
    end
  end
end
return false;

Algorithm 2: Fracture Count Calculation

Input : Partial Mesh $\mathcal{M}_k = (\mathcal{V}_k, \mathcal{F}_k)$
Output: Fracture Count C_{frac}

/* Define generation frontier from the last generated face. */
$\mathbf{f}_{\text{last}} \leftarrow$ last face in the sequence of $\mathcal{F}_k$;
if $\mathbf{f}_{last}$ *is null* **then**
  return 0;
end
$y_{\text{frontier}} \leftarrow \min_{v \in \mathbf{f}_{\text{last}}}(v.y)$;
/* Count boundary faces at or below the frontier. */
$\mathcal{F}_{\text{boundary}} \leftarrow$ set of faces in $\mathcal{F}_k$ on the mesh boundary;
$C_{\text{frac}} \leftarrow 0$;
for *face* f *in* $\mathcal{F}_{boundary}$ **do**
  if $\forall v \in f, v.y \leq y_{frontier}$ **then**
    $C_{\text{frac}} \leftarrow C_{\text{frac}} + 1$;
  end
end
return C_{frac}

B Topological Quality Metrics for Truncated Sequences

This section provides a formal definition of the metrics used to evaluate the topological quality of partially generated quadrilateral meshes. These metrics are specifically designed to operate on truncated sequences, forming the basis of the reward signal for our Direct Preference Optimization (DPO) fine-tuning stage.

B.1 Fracture Detection

A primary failure mode in autoregressive generation is fractures. Since our serialization is canonically ordered bottom-to-top, we detect fractures by identifying boundary faces at the current generation frontier. Algorithm 2 formalizes this by defining the frontier based on the lowest Y-coordinate of the last generated face. A non-zero count indicates a failure to generate an adjacent face in the expected upward direction, signaling a topological break.

B.2 Quantifying Edge Flow: Quad Rings and Lines

The hallmark of high-quality quad topology is structured edge flow, which we quantify by identifying two key structures: **Quad Rings** (closed loops of faces) and **Quad Lines** (open strips), as illustrated in Figure 8. Algorithm 3 formalizes an edge-based traversal that "walks" across adjacent quadrilateral faces. If a path terminates, it is classified as a Quad Line; if it returns to its starting edge, it forms a Quad Ring. From the discovered sets of rings ($\mathbb{R}$) and lines ($\mathbb{L}$), we compute the final reward components: the ratio of faces participating in closed rings and the average length of the open lines, both of which are maximized during DPO.

Algorithm 3: Quad Ring and Line Discovery

Input : Partial Mesh $\mathcal{M}_k$
Output: Set of Quad Rings $\mathbb{R}$ (face lists), Set of Quad Lines $\mathbb{L}$ (face lists)

```
ℝ, 𝕃 ← ∅, ∅;
E_processed ← ∅;
for edge e_start in M_k.edges do
    if e_start ∉ E_processed then
        path_faces ← ∅; path_edges ← ∅;
        current_edge ← e_start;
        while current_edge ≠ null and current_edge ∉ path_edges do
            add current_edge to path_edges;
            adj_quad ← a quadrilateral face adjacent to current_edge;
            if adj_quad exists then
                add adj_quad to path_faces;
                current_edge ← edge opposite to current_edge in adj_quad;
            end
            else
                current_edge ← null;
            end
        end
        E_processed.add(path_edges);
        /* Classify the discovered path based on closure.        */
        if current_edge is null or current_edge ≠ e_start then
            add path_faces to 𝕃;
        end
        else
            add path_faces to ℝ;
        end
    end
end
return ℝ, 𝕃
```

C More Results

To further demonstrate the superiority of our direct generation approach, we provide extensive qualitative comparisons in Figure 9 and Figure 10. These results highlight the robustness and high fidelity of QuadGPT across a wide variety of complex geometries. Additionally, to illustrate the structural integrity of our generated models, we present multi-view renderings of several samples in Figure 11. Finally, we compare our method with closed-source, commercial quad mesh generation techniques, including Tripo Team (2025) and Quad Remesher Exoside (2019) in Figure 12.

D Additional Experimental Analysis

In this section, we provide further analysis to complement the experiments in the main paper.

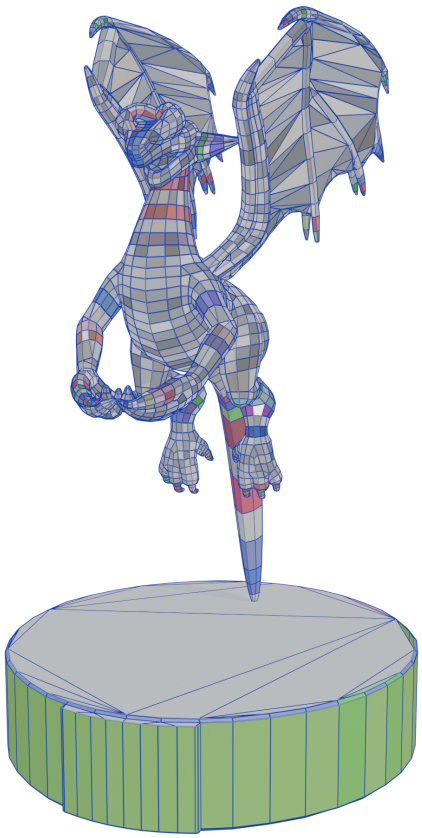

Figure 8: An illustration of a Quad Ring which is closed loop of quadrilateral faces. Our tDPO reward is designed to encourage the formation of such desirable, structured patterns.

D.1 Impact of Curriculum Learning and Topology-Aware Rewards

Figure 13(a) compares the next token prediction loss curves between from-scratch quadrilateral mesh training and fine-tuning from pre-trained triangle mesh weights. Fine-tuning achieves faster convergence and lower loss, confirming that inheriting triangle mesh knowledge provides better initialization for quadrilateral generation. Figure 13(b) presents the reward curves during DPO training, where our tDPO-Pro approach with truncated edge loop optimization consistently achieves the highest reward, validating the effectiveness of our complete topological reward design. The progressive improvement from DPO to tDPO and finally tDPO-Pro demonstrates the cumulative benefits of our methodological contributions.

D.2 The Inherent Limitations of Triangle-to-Quad Conversion

To illustrate the limitations of any post-hoc conversion pipeline, we conduct a best-case scenario experiment. We begin with a high-quality artist-created quad mesh and triangulate it. This process represents an irreversible information loss: the artist's original topological intent becomes ambiguous, creating a complex combinatorial problem that heuristic algorithms struggle to solve.

As shown in Figure 14, we compare our own **ILP-based operator** against the default methods in MeshLab Tarini et al. (200) and Blender Blender Foundation. While our optimized operator performs best, none of the methods can perfectly recover the original edge flow, leaving behind topological artifacts.

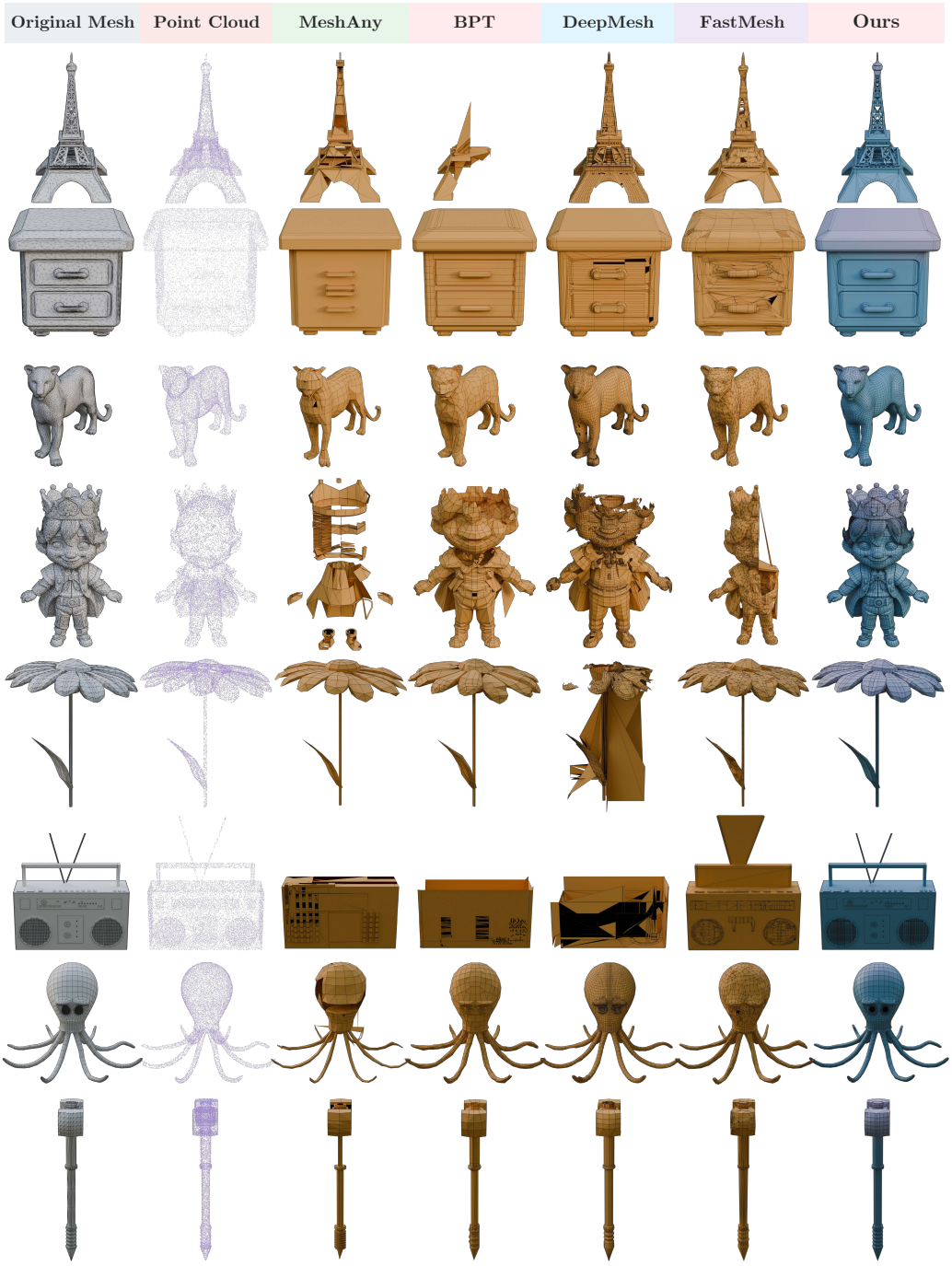

Figure 9: **Extended Qualitative Comparison against triangle generation.**

This challenge is significantly amplified when the input is an AI-generated triangle mesh, which often has an edge flow unsuitable for conversion, as seen in our TriGPT experiments (Figure 7). This compounding effect of a suboptimal input and an imperfect algorithm reinforces the need for our native generation approach.

Furthermore, algorithms from adjacent fields like Blossom-Quad (**?**), designed for 2D finite element analysis, are too restrictive for complex 3D surfaces. Its perfect-matching requirement caused it to fail on all but our simplest test cases, underscoring the need for a domain-specific, learning-based solution like QuadGPT.

Figure 10: A qualitative comparison against the field-guided method QuadriFlow. This example highlights the superior robustness and topological quality of our direct generative approach.

D.3 Analysis of Compressed vs. Direct Coordinate Representation

A critical design choice in autoregressive mesh generation is the tokenization strategy. Methods like DeepMesh (Zhao et al., 2025a) and BPT (Weng et al., 2025) employ compressed representations. Our experiments with these methods on triangle meshes confirmed a similar finding: compressed tokenizers lead to faster initial convergence due to shorter sequence lengths. However, we argue that faster convergence does not necessarily lead to better final performance. As shown in Table 4, our extensive experiments at the 1B-parameter scale show that a simple, **direct coordinate representation ultimately achieves superior results** in both geometric fidelity and topological quality, a finding strongly corroborated by our user study. We hypothesize that overly complex, "over-designed" tokenizers can introduce an unintended inductive bias, constraining the model's expressive capacity and limiting its performance ceiling, especially when trained on large, diverse datasets. This observation aligns with a broader trend in deep learning, demonstrated by models like DepthAnything V3 (Lin et al., 2025), where simple, highly scalable architectures often outperform more complex designs when sufficient data and compute are available. For QuadGPT, we therefore prioritized the higher performance ceiling of the direct representation, confident that techniques like truncated training effectively manage the longer sequence lengths.

D.4 Ablation on Training Data

To disentangle the impact of our methodology from our curated dataset, we conduct a data ablation study. We train a variant, **QuadGPT-OS**, using only the publicly available portions of our dataset (e.g., ShapeNet Chang et al. (2015), Objaverse-XL Deitke et al. (2023a)), subjected to the same rigorous filtering pipeline described in Section A. This experiment serves to answer a critical question: can a robust, scalable methodology achieve state-of-the-art results even when limited to open-source data?

Table 4: **Quantitative Comparison: Tokenization on Triangle Meshes.** Direct coordinate representation (TriGPT) outperforms a compressed BPT-style tokenizer (TriGPT+BPT). User Study scores reflect expert rankings from 0 (worst) to 1 (best). (Q) denotes outputs were converted to quads for evaluation.

Method	CD ↓	HD ↓	QR ↑	US ↑
TriGPT+BPT (Compressed)	0.078	0.198	68%	0.3
TriGPT (Direct Coord.)	**0.062**	**0.160**	**70%**	**0.7**

The results, shown in Figure 15, are twofold and unequivocal. First, **QuadGPT-OS still significantly outperforms all prior autoregressive baselines**. This confirms that our core contributions—the scalable architecture, curriculum learning strategy, and tDPO refinement—are the primary drivers of our model's high performance. It proves that our method is not merely a product of better data, but a fundamentally more solid approach to the problem.

Second, our full model, trained on the complete 1.3 million mesh dataset, demonstrates a further substantial improvement in quality over QuadGPT-OS. This highlights a key insight: while a superior methodology can establish a new state of the art on public data, the full potential of a truly scalable approach is unlocked through careful, large-scale data curation. Our work demonstrates the

Figure 11: **Multi-View Renderings of Generated Meshes.**

synergistic impact of advancing both the model and the data to push the frontier of production-ready 3D generation.

E Downstream Applications

This section demonstrates the advantages of high-quality topology in downstream applications, such as UV mapping and animation. Thanks to our superior quad mesh structure, the models facilitate

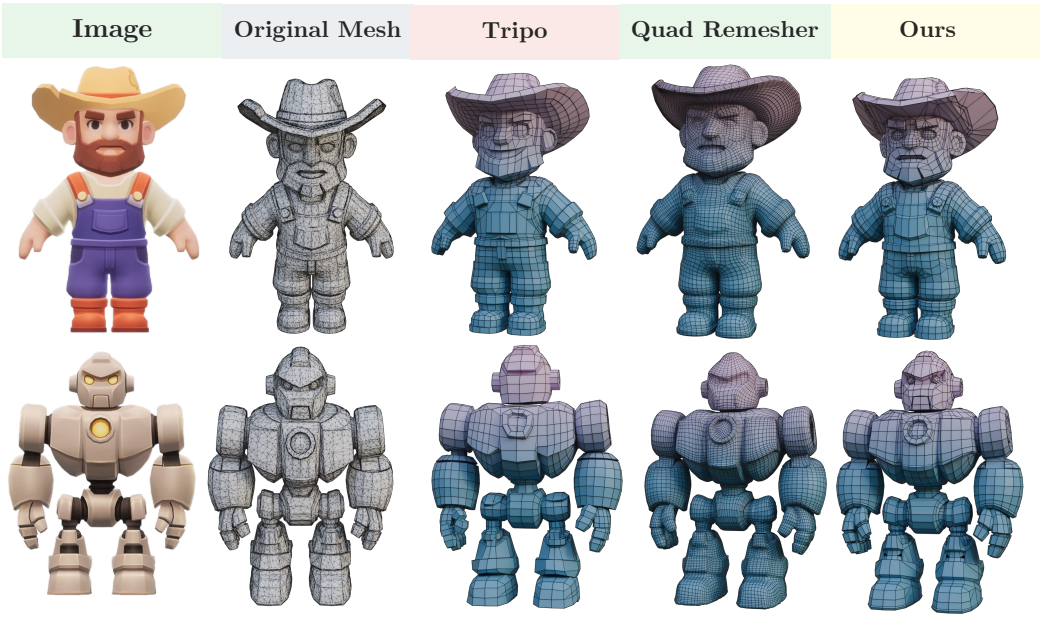

Figure 12: **Comparison with closed-source commercial quad mesh generation methods.** Our approach preserves more geometric details, especially edge loops, which are crucial for UV segmentation and animation. Compared to Blender's remeshing plugin, our method produces compact faces.

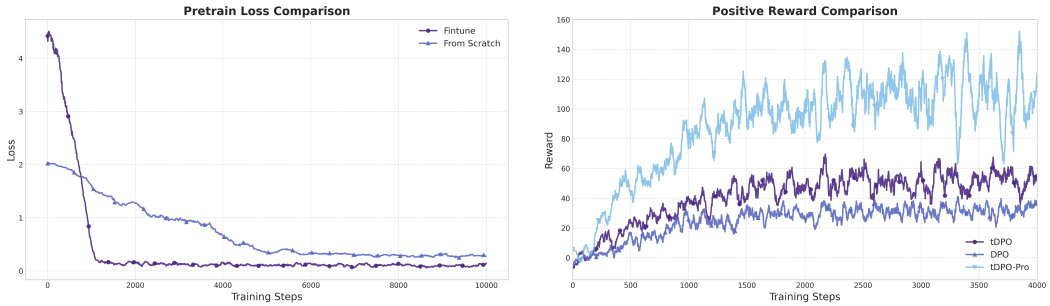

Figure 13: **Training Dynamics Analysis.** (a) Next token prediction loss comparison between from-scratch quadrilateral mesh training and fine-tuning from pre-trained triangle mesh weights. Fine-tuning demonstrates faster convergence and lower loss. (b) DPO reward curves for different variants, showing that our tDPO-Pro approach achieves the highest reward with truncated edge loop optimization.

high-quality UV unwrapping Li et al. (2025a). Furthermore, the abundance of edge loops enables more natural mesh deformation.

F Limitation and Future Work

Despite the promising results of QuadGPT, several limitations present important avenues for future research.

Domain Gap and Part-based Generation. Our framework operates within the prevailing two-stage paradigm, where topology generation follows geometry generation. This creates a critical domain gap: the model is trained on point clouds from clean, artist-crafted assets but is often deployed on noisy point clouds derived from AI-generated implicit fields. This mismatch can lead to fractures and other failures on complex, out-of-distribution shapes, as shown in Figure 17. A

Figure 14: **The Irreversibility of Triangulation and Limitations of Conversion Algorithms.** (a) A high-quality artist-created quad mesh with clean edge flow. (b) The same mesh after triangulation—the ground truth topological information is now ambiguous. (c-e) The results of applying different tri-to-quad conversion algorithms to (b). While our ILP-based operator (e) produces a better result than the default methods in MeshLab (c) and Blender (d), none can perfectly recover the original topology, resulting in broken edge loops and artifacts. This demonstrates the fundamental limitations of post-hoc conversion.

promising direction is to develop an end-to-end model that co-generates geometry and topology, bypassing the intermediate dense mesh entirely. Furthermore, since professional assets are inherently part-based, integrating part-aware generation, potentially building on recent advances in part generation (Yan et al., 2025), could significantly improve robustness and fidelity.

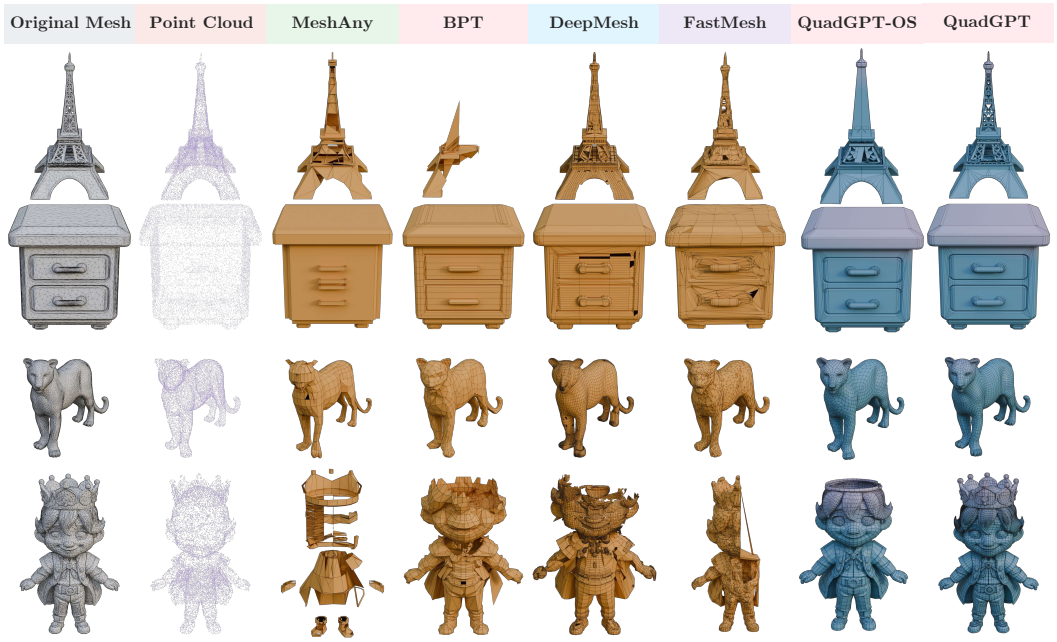

Figure 15: **Qualitative results of our data ablation study.** "QuadGPT-OS" is trained exclusively on filtered public datasets, while "QuadGPT" uses our complete curated dataset. "QuadGPT-OS" already demonstrates a significant improvement in topological coherence and geometric detail over prior baselines. The full model shows further refinement, particularly in achieving cleaner, more professional edge flow, validating the scalability of our approach with high-quality data.

Lack of Controllable Polygon Number. QuadGPT currently cannot explicitly control the final polygon count; attempts to add a simple number condition were ineffective. This limits its utility in production, where generating assets with specific polygon budgets for LODs is essential. Future work must develop a robust control mechanism.

Advancing the Reinforcement Learning Framework. Our current implementation uses off-policy DPO for its simplicity and stability. However, our programmatic reward, while effective at enforcing structural rules like edge loops, cannot capture the full nuance of artistic preference. Future work could explore two key directions. First, developing a reward model trained on human preference data from professional artists could provide a richer, more aligned optimization signal. Second, investigating more advanced, potentially online RL algorithms could further elevate the quality and complexity of the generated topology.

G LLM USAGE STATEMENT

The authors used Gemini 2.5 Pro exclusively for grammar checking and language polishing of the manuscript text. All technical content, experimental design, data analysis, and scientific conclusions are the original work of the authors. The LLM was not involved in generating scientific ideas, conducting experiments, or interpreting results.

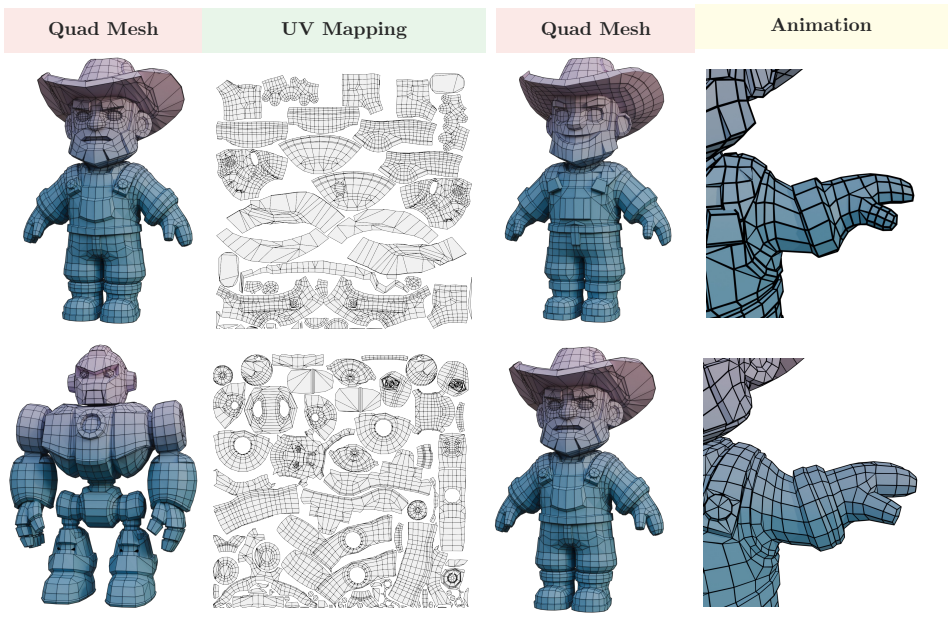

Figure 16: **UV mapping and animation deformation comparison.** (a) UV unwrapping result on a quad mesh generated by our method. (b) Animation deformation comparison: the top row shows results on a mesh from Tripo, while the bottom row presents ours. Our method captures geometric details more effectively and produces more edge loops, leading to noticeably more realistic deformation.

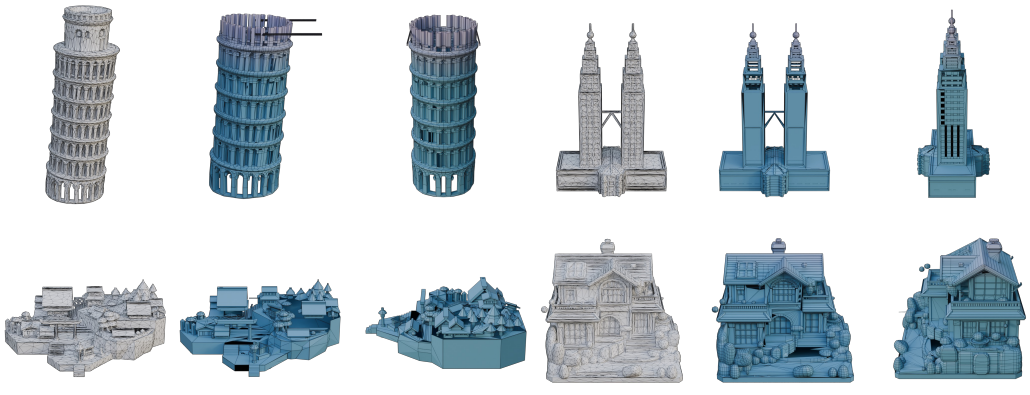

Figure 17: **Failure Cases and Limitations.** Our model can sometimes struggle to generalize to out-of-distribution point clouds sampled from AI-generated assets, particularly for architectural models with sharp features, which can result in fractures or incomplete geometry.