MeshWeaver: Sparse-Voxel-Guided Surface Weaving for Autoregressive Mesh Generation

Supplementary Material

A. Implementation Details

Point Cloud Conditioner. Similar to prior works, we adopt a point cloud encoder based on the architecture of 3DShape2VecSet [3] to encode the input point cloud into fix-length conditional tokens, which are prepended to the mesh sequence to provide global generation context. The conditional tokens are attended to each other via bidirectional attention, while the subsequent vertex tokens attend to their predecessors via causal attention. This attention mechanism can be implemented efficiently with PyTorch's FlexAttention.

Training Loss. Our framework adheres to a fully causal generation scheme: each vertex is conditioned on all preceding vertices, and within a vertex, each level's token is conditioned on predictions from coarser levels. Training therefore reduces to a sequence-modeling problem, optimized using the standard cross-entropy loss commonly employed in causal language models:

$$\mathcal{L} \;=\; -\sum_{j=1}^{N}\sum_{l=0}^{L-1} \log p\big(v_j^l \,|\, \boldsymbol{v}_{<j}, v_j^{<l}\big)\,, \tag{1}$$

where N denotes the number of vertices, L the number of levels, v_j^l the level-l token of vertex j, $\boldsymbol{v}_{<j}$ all previously generated vertices, and $v_j^{<l}$ the already predicted levels of the j-th vertex.

Mesh Tokenization Benchmarking. The mesh tokenization results in Table 1 of the main paper are evaluated on the Toys4K [2] dataset. For each mesh in the dataset, we perform 7-bit quantization, and tokenize it into sequences using different tokenization algorithms. The compression ratio is computed as $L/(9N)$, where L is the tokenized sequence length and $9N$ is the vanilla sequence length of the quantized N-face mesh. Finally, we compute the average compression ratio over all meshes of each method for benchmarking.

Mesh Quantization in Data Preprocessing. Vertex merging caused by mesh quantization can introduce topology artifacts. While increasing the resolution helps, training cost will increase too. Instead, we mitigate this problem in data preprocessing: we notice that the main failure is unintended bridging between nearby connected components, creating non-manifold vertices/faces; merges within a component mostly remove high-frequency detail and rarely break topology unless the component is too tiny. We therefore split meshes into connected components and quantize each component independently. On 300 Toys4K meshes quantized at 128^3, this reduces the non-manifold ratio from 24% to 5.7%, addressing the dominant source of topology errors.

Subvolume Sampling Details. During training, each level-l token ($l > 0$) only cross-attends to sparse-voxel features inside its parent subvolume at level $l-1$. To reduce cross-attention overhead, we apply subvolume pruning: we randomly sample a set of level- $l-1$ subvolumes and compute the loss only on the level- l tokens that attend to them. Concretely, we sample a fixed number L of level- 0 subvolumes; when sampling level-1 subvolumes (for a level-2 model), we use $D_1^2 \cdot L$, where D_1 is the spatial subdivision factor of level 1 relative to level 0 . This scaling keeps an approximately constant sampling ratio across levels (since refining resolution increases the number of active sparse subvolumes by $\sim D_1^2$), preventing under-training at finer levels. Empirically, this pruning significantly improves training efficiency without degrading performance.

B. Additional Results

Training Curve Ablation on Sparse-Voxel Encoder. We further visualize the training dynamics in Figure 1. Compared to the ablated variant without VF and CA, our full model exhibits markedly faster convergence. Without the sparse-voxel encoder, the model effectively reduces to pure language modeling over vertices, lacking geometric priors and therefore requiring substantially longer optimization. In contrast, the inclusion of the sparse-voxel encoder provides an explicit surface-aware context, reducing the difficulty of next-vertex prediction and accelerating training.

Additional Quantitative Results. We augment our evaluation with 3 topology measures: (i) Boundary Edge Ratio (BER), the fraction of boundary edges among all edges (ideally 0 for a closed manifold mesh; higher BER indicates discontinuities/holes/damage); (ii) Topology Score (TS), which evaluates structural quality via the quality of a derived quad mesh obtained by standard triangle-to-quad merging (a weighted sum of four sub-metrics; details follow Mesh-RFT [1]), motivated by the fact that quad layouts are preferred in production and thus serve as a practical proxy for topological soundness; and (iii) Manifoldness Ratio (MR), the percentage of generated meshes that are manifold among the test set. As shown in Table 1, MeshWeaver achieves the best performance across all 3 topology metrics compared to baselines, supporting our claim of higher-quality meshes beyond geometric proximity.

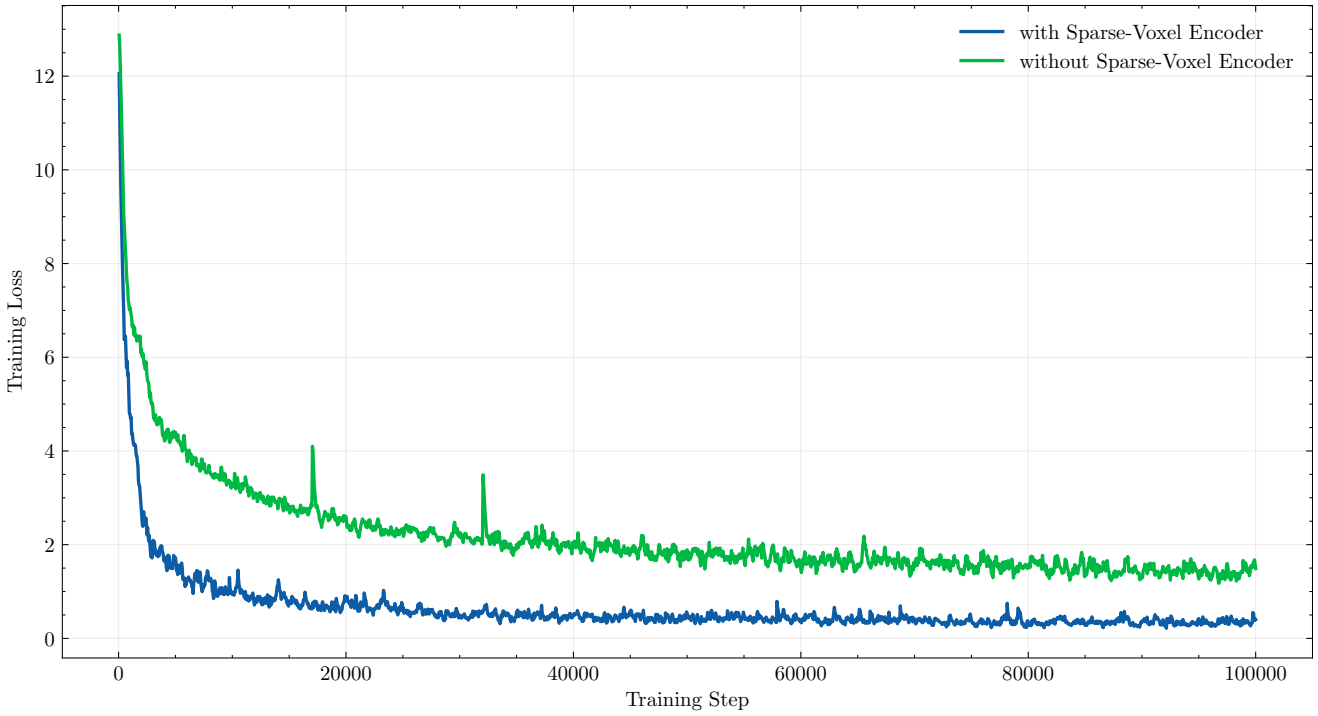

Figure 1. **Comparision on Training Loss with or without Sparse-Voxel Encoder.**

Table 1. Quantitative results on topology quality.

Method	BER ↓	TS ↑	MR ↑
MeshAnythingV2	0.0852	0.678	0.311
EdgeRunner	0.0794	0.691	0.417
BPT	0.0308	0.726	0.764
TreeMeshGPT	0.0340	0.703	0.726
Mesh-Silksong	0.0295	0.727	0.786
Ours	**0.0271**	**0.742**	**0.827**

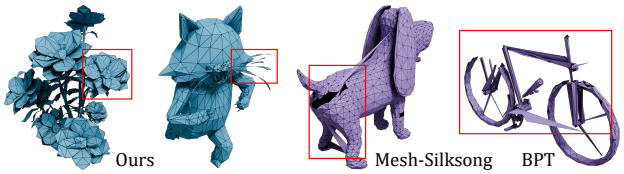

Figure 2. Failure cases of different methods.

Additional Qualitative Results. We present more generated results in Figure 3 and Figure 4.

Failure Cases. As Figure 2 shows, MeshWeaver's failures are typically local: (i) on very complex topology it may produce small self-intersections; (ii) with insufficient voxel resolution it may miss thin parts. These errors usually stay localized, while global fidelity remains strong; in contrast, baselines more often exhibit more global incoherence (e.g., "flying faces", large holes, severe self-intersections).

C. Discussions

Relationship to "Artist-created Mesh Generation". Artist-created mesh generation aims to produce structured, clean, editable meshes, and it does not prescribe the conditioning modality (it can be text-, image-conditioned, or even unconditional). We view surface-conditioned retopology as a core component of this broader goal: given a target surface—often obtained from text/image-to-3D generators or implicit+Marching-Cubes pipelines that yield topologically messy meshes—MeshWeaver constructs artist-like topology aligned to that surface. We believe the autoregressive paradigm is particularly well-suited here, since it can devote its capacity to topology construction under strong geometric guidance, rather than spending tokens on rediscovering the global shape.

Complexity and Scalability of Sparse Voxels. Sparse voxels add engineering complexity but provide local, explicit geometric cues, mirroring a broader 3D-generation shift from global implicit representations to local explicit representations that greatly improves fidelity. Their cost scales with occupied cells (not dense N^3), and our sparse-voxel encoder is lightweight ($<$100M params). However,

scaling the sparse-voxel representation to very high resolutions (e.g., 1024^3) is indeed challenging and we leave it as future work.

Limitations. While MeshWeaver advances the state of automatic mesh generation, several challenges remain. First, real-world assets often contain tens to hundreds of thousands of faces, which are still beyond the capacity our framework can reliably produce. Second, the sparse-voxel encoder, though effective for structural guidance, introduces additional computational overhead, making it difficult to scale to very high resolutions (e.g., 1024^3). Finally, performance is bounded by the scale and quality of available training data; we expect that larger and more diverse curated datasets will further improve both fidelity and robustness.

References

[1] 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. 1

[2] Stefan Stojanov, Anh Thai, and James M Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1798–1808, 2021. 1

[3] 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. 1

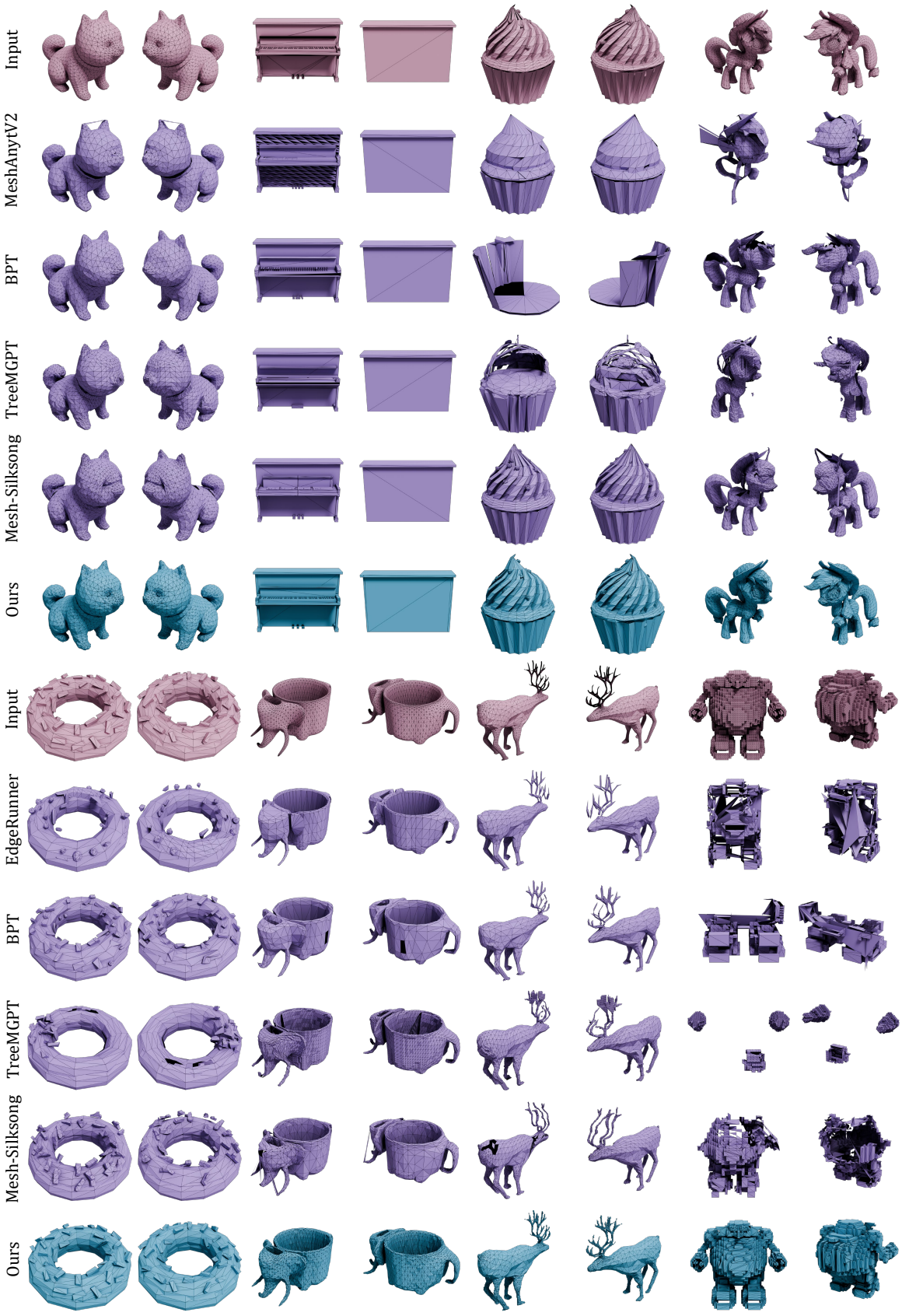

Figure 3. **Additional Results on Point-Cloud Conditioned Mesh Generation.**

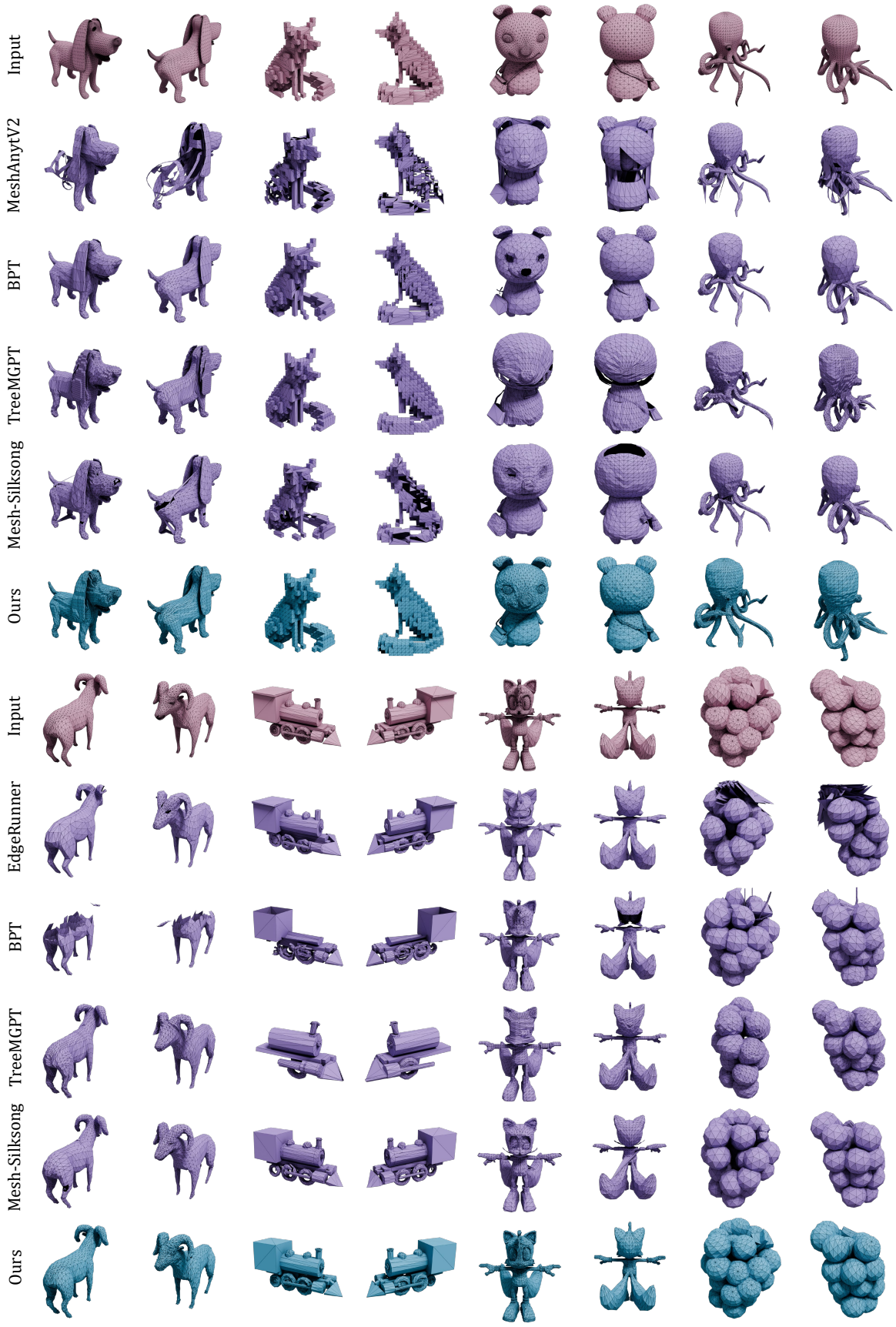

Figure 4. **Additional Results on Point-Cloud Conditioned Mesh Generation.**