PixARMesh: Autoregressive Mesh-Native Single-View Scene Reconstruction

Abstract

We introduce PixARMesh, a method to autoregressively reconstruct complete 3D indoor scene meshes directly from a single RGB image. Unlike prior methods that rely on implicit signed distance fields and post-hoc layout optimization, PixARMesh jointly predicts object layout and geometry within a unified model, producing coherent and artist-ready meshes in a single forward pass. Building on recent advances in mesh generative modeling, we enrich a point-cloud encoder with pixel-aligned image features and global scene context via cross-attention, enabling accurate spatial reasoning from a single image. Scenes are generated autoregressively from a unified token stream of context, pose, and mesh tokens, yielding compact meshes with high-fidelity geometry. Experiments on synthetic and real-world datasets show that PixARMesh achieves state-of-the-art reconstruction quality while producing lightweight, high-quality meshes ready for downstream applications.

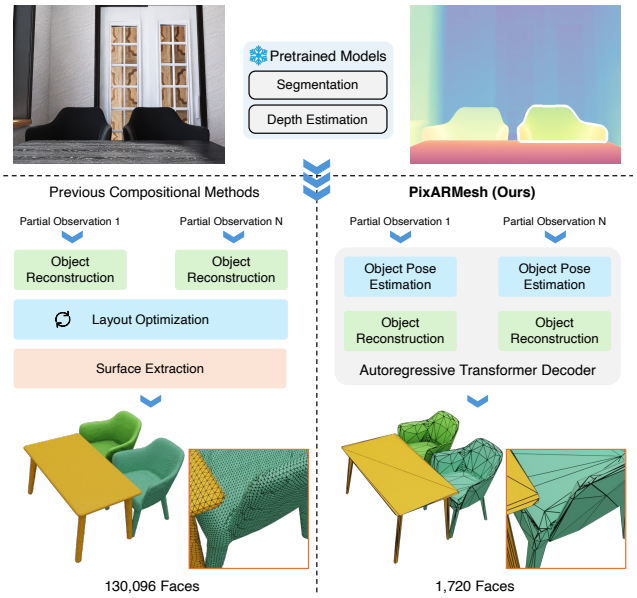

Figure 1. Comparison of PixARMesh with recent compositional scene reconstruction methods. PixARMesh predicts object poses and reconstructs native meshes in a single autoregressive decoding process, without relying on SDF-based surface extraction or layout optimization, producing compact and artist-ready mesh outputs.

1. Introduction

Reconstructing a complete 3D scene from a single RGB image is a long-standing and fundamentally ill-posed problem. A single viewpoint provides only partial, depth-ambiguous observations of objects, while large portions of the scene remain occluded or unobserved. Recovering accurate object shapes and coherent spatial layouts therefore requires strong priors about indoor scenes and plausible object structures.

Earlier methods [7, 8, 41] reconstruct the entire scene holistically by back-projecting image features into 3D volumes and predicting a scene-level signed distance field (SDF) using an encoder–decoder architecture. While these approaches bypass explicit layout estimation, they are fundamentally constrained by the spatial resolution of volumetric grids and the limited expressiveness of feed-forward decoders. As a result, they struggle to produce high-quality geometry and lack the generative flexibility and generalization capability needed for complex real-world scenes.

Recently, the compositional generation paradigm has gained significant attention, driven by advances in large-scale object-level reconstruction models [16, 18, 24–26, 40]. Since these models are typically pre-trained on clean, unoccluded object images, existing pipelines [11, 45] require an inpainting or amodal completion stage to recover occluded regions before passing object crops to the reconstruction network. To assemble the reconstructed instances into a coherent scene, they further rely on optimization-based layout estimation, often formulated as point-cloud matching, which is prone to local minima. Recent works such as DepR [43] mitigate the need for inpainting by conditioning generation directly on partial observations, while MIDI [17] eliminates layout optimization by predicting each instance directly in normalized scene coordinates. Although these methods generally achieve higher reconstruction fidelity, their dependence on SDF-based representa-

tions introduces additional complexity in surface extraction and often yields overly smooth, high-face-count meshes that deviate from artist-ready geometry.

Meanwhile, there is steady progress in object-level mesh generative models [3–5, 20, 34, 36, 39, 42], where artist-like mesh sequences are directly predicted by an autoregressive Transformer decoder, eliminating the need for iso-surface extraction. However, despite these advances, autoregressive mesh generators remain limited to object-level outputs, and no existing scene reconstruction pipeline leverages their native, artist-ready mesh representations. This gap motivates integrating strong partial observations with mesh-level generative priors for scene-level reconstruction.

To bridge this gap, We propose PixARMesh, a framework built on top of pre-trained object-level autoregressive mesh generative models such as EdgeRunner [36] and BPT [39], introducing a new paradigm for single-view scene reconstruction using native, artist-ready mesh representations. To leverage the limited geometric cues available in depth-back-projected point clouds, we fuse pixel-aligned image features into the point-cloud encoder, injecting appearance cues on top of partial geometry. To further enhance scene-level understanding, we incorporate cross-attention between each object's point-cloud features and a global scene point cloud, enabling context-aware reconstruction under heavy occlusion. Finally, we utilize the coordinate vocabulary of existing mesh generative models to tokenize scene composition, allowing PixARMesh to jointly predict object poses and meshes within a single feed-forward autoregressive sequence. We validate PixARMesh on synthetic 3D-FRONT [12] and real-world images, demonstrating that it produces high-quality, artist-ready meshes with coherent layouts and strong reconstruction performance.

Our main contributions are summarized as follows:

- We present the first framework that does single-view scene reconstruction *directly*, *autoregressively* in mesh space, avoiding SDF-based decoding and surface extraction while producing high-quality, artist-ready outputs.
- We repurpose recent object-level mesh generative models by incorporating *pixel-aligned image features* and *global scene context* into the point-cloud encoder, enabling context-aware pose and geometry generation from a single image.
- We jointly predict object poses and meshes in a single feed-forward autoregressive manner, achieving coherent scene composition without post-hoc layout optimization. Extensive experiments demonstrate that PixARMesh achieves state-of-the-art reconstruction performance.

2. Related Work

3D Scene Reconstruction from a Single Image. Single-view reconstruction is inherently ill-posed due to scale ambiguity, occlusions, and incomplete geometric cues, often requiring depth or shape priors from large-scale pre-trained models. Early holistic approaches such as Panoptic3D [27], PanoRe [8], Uni-3D [41], and BUOL [7] reconstruct an entire scene using feed-forward encoder–decoder architectures applied to back-projected feature volumes. While these methods do not require explicit layout estimation, they are constrained by limited spatial resolution and exhibit poor generalization and generative capability.

Recent research has shifted toward compositional generation frameworks, which decompose a scene into individual instances and benefit from advances in object-level generative models. For example, Gen3DSR [11] and DeepPriorAssembly [45] perform image inpainting to complete occluded regions before feeding the recovered object crops into pre-trained object reconstruction models [18, 26, 40]. DepR [43] instead generates shapes conditioned on partial image observations using a depth-guided diffusion model. These methods rely on post-hoc, optimization-based layout estimation to compose reconstructed instances back into a scene, which can be susceptible to local minima and spatial misalignment. MIDI [17] alleviates this limitation by generating all instances within a normalized scene space, thereby avoiding explicit pose estimation. Despite these advances, most existing approaches operate on signed distance fields (SDFs) and require iso-surface extraction via marching cubes [28], often producing densely tessellated and overly smooth meshes that hinder geometry-based applications such as editing. Our work addresses these limitations by predicting object layouts in a feed-forward manner and reconstructing each instance as an artist-like mesh sequence.

Native Mesh Generation. Generating 3D shapes directly in native, artist-like meshes has long been attractive for their compactness, editability, and compatibility with downstream graphics applications. Early methods rely on structured primitives such as surface patches [14], deformable ellipsoids [37], mesh graphs [9], and binary space partitioning [6], but they typically impose strong geometric priors and offer limited topological flexibility. More recently, PolyDiff [1] applies discrete diffusion to synthesize meshes, while PolyGen [29] introduces an autoregressive framework that predicts vertices and faces using two coordinated Transformer decoders.

Subsequent approaches move to a single-sequence formulation. MeshGPT [34] employs a Transformer over VQ-VAE–quantized mesh tokens, and MeshAnything [4] extends it with shape-conditional generation. MeshXL [3] fur-

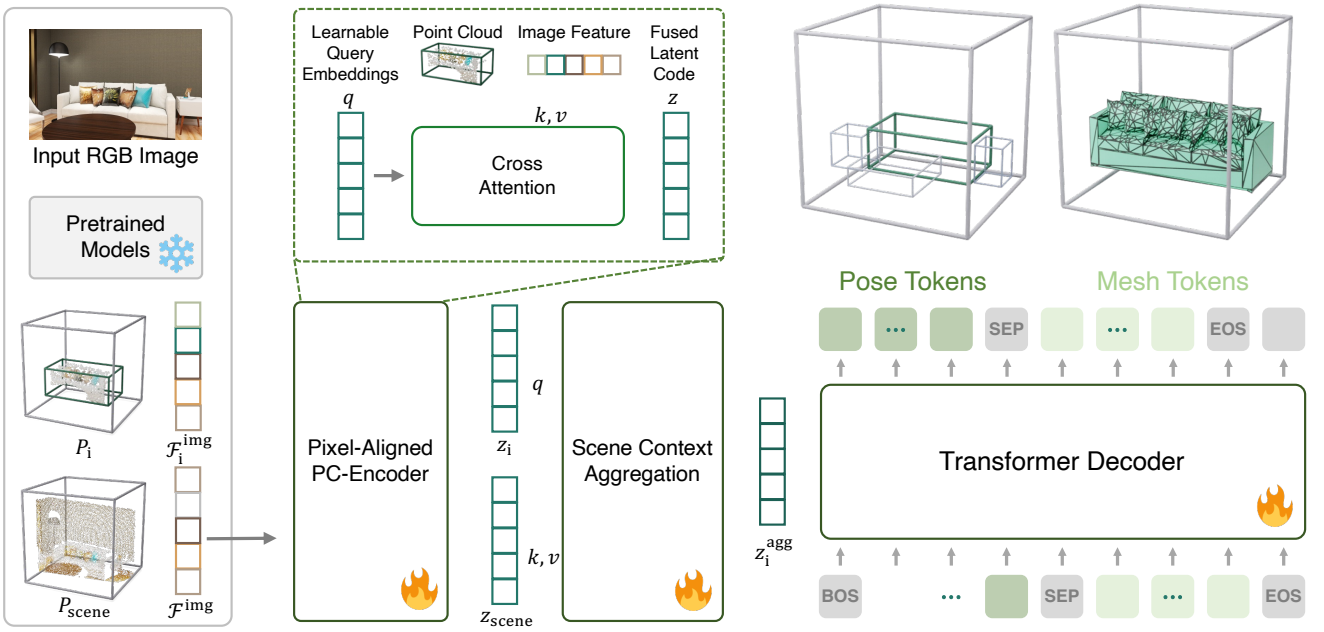

Figure 2. Pipeline overview. Given an RGB image, we use pretrained models to extract the point cloud and image features for both the target object i and the global scene. These local and global cues are fed into the Pixel-Aligned PC-Encoder to produce the fused latent code, which is then aggregated into a single latent vector via cross-attention. This latent vector conditions the Autoregressive Decoder, which predicts the object's pose followed by its mesh token sequence.

ther simplifies the process by operating directly in quantized coordinate space, removing the need for a VQ-VAE but at the cost of longer token sequences. To improve scalability, recent studies propose compressive tokenization strategies that exploit face adjacency [5, 21, 36, 39]. Meshtron [15] follows MeshXL tokenization but introduces an Hourglass Transformer [30] to internally compress long sequences.

Others explore complementary directions for improving mesh generation quality and controllability. DeepMesh [44] and Mesh-RFT [23] incorporate reinforcement learning to align mesh generation with aesthetic or human preferences. PivotMesh [38] generates pivot vertices as coarse structural guidance for subsequent mesh generation, while VertexRegen [42] and ARMesh [20] advance the coarse-to-fine generation paradigm by progressively increasing geometric detail. Building on mesh generative models with strong compression and scalability, such as EdgeRunner [36] and BPT [39], our work extends these advances to scene-level reconstruction with artist-like meshes.

3. Method

We provide an overview of our framework in Fig. 2, which consumes depth-derived point clouds from off-the-shelf perception models and performs autoregressive scene reconstruction. We first introduce the problem setup in Sec. 3.1, then describe how we adapt point-cloud encoders from object-level mesh generative models to operate at the scene level. Finally, we detail our tokenization scheme in Sec. 3.3 and our training strategy in Sec. 3.4.

3.1. Preliminary

The goal of single-view scene reconstruction is to recover the 3D geometry and spatial configuration of a scene from a single RGB image. Following the compositional paradigm used in prior work such as DepR [43] and DeepPriorAssembly [45], we focus on reconstructing only foreground object instances (*e.g.* furniture in indoor scenes) and disregard large planar background structures such as walls and floors.

We introduce PixARMesh, an end-to-end framework that jointly predicts the shape and scene-level pose of each object instance, producing a complete scene where all objects are represented using native, artist-ready meshes rather than implicit SDFs.

Given an input RGB image $I \in \mathbb{R}^{H\times W\times 3}$, we first extract depth D, instance segmentation masks $\mathcal{M} = \{M_i\}_{i=1}^N$, and image features $\mathcal{F}_{\text{img}}$ using off-the-shelf models. The depth map is back-projected using the camera intrinsics K to obtain a raw scene point cloud P_{scene}. Applying the instance masks yields per-object point clouds $\mathcal{P} = \{P_i\}_{i=1}^N$ where $P_i = P_{\text{scene}} \odot M_i$, which capture only the visible portions of each object in global camera coordinates.

Unlike previous compositional methods that reconstruct object shapes first and resolve their spatial layout afterward,

we unify both tasks in a single autoregressive feed-forward architecture. For each instance i, the model F_{AR} predicts both its scene-level pose T_i and its canonical-shape mesh O_i:

$$(T_i, O_i) = F_{\text{AR}}(P_i, M_i, \mathcal{F}_{\text{img}}, P_{\text{scene}}) \quad (1)$$

After processing all instances, the final scene reconstruction is obtained by transforming each canonical mesh into the scene coordinate frame $\mathcal{S} = \{T_i O_i\}_{i=1}^{N}$.

We adopt EdgeRunner [36] and BPT [39] as our base models, both of which are autoregressive mesh generators designed for object-level, shape-conditioned generation. In their original formulations, a point-cloud encoder processes *complete object point clouds* and produces conditioning tokens for the Transformer decoder to autoregressively generate mesh sequences. However, in single-view scene reconstruction, objects are only partially observed due to occlusions, and their global poses within the scene must also be inferred. In the following sections, we describe how we repurpose them for the single-view setting by (1) adapting the point-cloud encoder to incorporate appearance features from an image encoder, (2) injecting global scene context to compensate for missing geometry, and (3) predicting object poses within the same autoregressive framework.

3.2. Repurposing the Point-Cloud Encoder

Injecting Pixel-Aligned Image Features. The original point-cloud encoder used in EdgeRunner and BPT operates solely on point coordinates, without leveraging the rich appearance cues present in image features. To support single-view reconstruction, where objects are often partially observed, we augment the encoder with direct multi-modal fusion between geometry and pixel-aligned image features.

Given an instance point cloud P_i and camera intrinsics K, each 3D point p is projected onto the image plane to obtain its corresponding pixel $\text{Proj}(K, p) = (u, v)$ on the global feature map $\mathcal{F}_{\text{img}}$, establishing a point-pixel correspondence. For each such pair, the encoder $\mathcal{E}_{\text{pc}}$ concatenates the geometric feature $\mathbf{f}_p^{\text{pc}}$ with the aligned image feature $\mathbf{f}_p^{\text{img}} = \mathcal{F}_{\text{img}}(u, v)$ to form the key-value inputs to a Transformer-based fusion block. A set of learnable query embeddings then aggregates these fused features into a compact latent code:

$$\mathbf{z}_i = \mathcal{E}_{\text{pc}}\left(\mathbf{f}_p^{\text{pc}}, \mathbf{f}_p^{\text{img}}\right) \ \forall p \in P_i. \quad (2)$$

This pixel-aligned design enables the autoregressive mesh generator to incorporate per-point appearance cues, enhancing robustness to occlusion and improving the completeness and global consistency of the reconstructed geometry.

Scene Context Aggregation. Instead of normalizing each instance independently in its own canonical space, which discards global spatial relations, we first normalize the entire global point cloud P_{scene} and all instance point clouds $\{P_i\}_{i=1}^{N}$ into a unified scene coordinate frame. This preserves consistent spatial reference among all objects. The normalized instance point clouds are then fed into the pixel-aligned point cloud encoder, ensuring that all encoded features share a coherent spatial frame for subsequent context aggregation. From this encoder, we obtain a scene-level latent $\mathbf{z}_{\text{scene}}$ and per-instance latent codes $\mathbf{z}_i$.

To incorporate global scene context, *e.g.*, cues from nearby objects of similar category or geometry, and to further improve reconstruction quality, each object latent $\mathbf{z}_i$ attends to the scene-level latent via a cross-attention layer:

$$\mathbf{z}_i^{\text{agg}} = \text{CrossAttn}(q = \mathbf{z}_i,\, k = \mathbf{z}_{\text{scene}},\, v = \mathbf{z}_{\text{scene}}), \quad (3)$$

The resulting aggregated feature $\mathbf{z}_i^{\text{agg}}$ enriches the instance representation with holistic scene cues, enabling more accurate pose estimation and geometry prediction for each object.

3.3. Tokenization

As an autoregressive framework, our model represents both object poses and meshes as discrete token sequences. We uniformly quantize the canonical unit cube $[-1, 1]^3$ into N bins along each axis. For EdgeRunner, each vertex is represented by three integer tokens <x>, <y>, <z>, while BPT replaces these with a <block_id> and <offset_id> pair through block-wise decomposition of the N^3 quantized grid.

Object Pose Tokenization. Following standard conventions in 3D detection [19], we represent each object pose using a gravity-aligned 7-DoF bounding box (center, scale, yaw). Rather than introducing a dedicated vocabulary for pose parameters, especially for the yaw angle, we reuse the vertex tokenization scheme by encoding the 8 corner points of the bounding box (normalized with respect to the global normalization in Sec. 3.2). This yields lightweight pose sequences (24 tokens for EdgeRunner and 16 tokens for BPT), negligible compared to mesh sequences. Importantly, this vertex-based formulation enables complete vocabulary sharing with mesh tokenization, avoiding new token types while maintaining expressiveness.

At inference time, the pose sequence is first decoded into the 8 bounding-box corners directly in the normalized scene coordinate frame. The subsequent mesh sequence is decoded in the local canonical space, where each object is normalized to a unit cube. To bridge these two spaces, we recover a local-to-global transformation using the decoded global-space corners as targets. Let $\mathbf{X}_{\text{local}} \in \mathbb{R}^{8\times3}$ denote the canonical box corners and $\mathbf{X}_{\text{global}} \in \mathbb{R}^{8\times3}$ denote the decoded global-space corners. We estimate the best-fit affine transformation $\mathbf{T} \in \mathbb{R}^{3\times4}$ by solving the linear least-

squares problem:

$$\mathbf{T}^{\star} = \arg\min_{\mathbf{T}} \left\| \mathbf{X}_{\text{global}} - [\mathbf{X}_{\text{local}}\ \mathbf{1}]\, \mathbf{T}^{\top} \right\|_2^2. \quad (4)$$

The resulting transformation $\mathbf{T}^{\star}$ is interpreted as a gravity-aligned transform, and is applied to all vertices of the decoded canonical mesh, yielding the final object geometry in the global scene frame.

Object Mesh Tokenization. For mesh sequences, we simply adopt the native tokenization strategy of each base model.

BPT uses a *Blocked and Patchified Tokenization* scheme that partitions the 3D coordinate grid into blocks and aggregates spatially adjacent faces into compact patches. This achieves strong compression (ratio $\approx$ 0.26 at resolution 128) with a large but structured vocabulary of 40,960 tokens.

EdgeRunner employs a *Compact Mesh Tokenization* derived from the EdgeBreaker algorithm [33], traversing triangles via a half-edge structure to maximize vertex reuse. It attains a moderate compression ratio ($\approx$ 0.46 at resolution 512) with a smaller vocabulary of 518 tokens, while preserving high geometric fidelity.

These two tokenization paradigms are complementary: BPT prioritizes aggressive sequence compression with a high-capacity vocabulary, whereas EdgeRunner emphasizes resolution and geometric detail with a more compact vocabulary and moderate compression. In all cases, meshes are normalized to a unit cube and vertex coordinates are discretized according to the respective quantization resolution. Our framework supports both without modification, demonstrating robustness to widely different tokenization designs.

Final Token Sequence. For each object, the final autoregressive sequence is constructed as:

`<bos>, [pose_seq], <sep>, [mesh_seq], <eos>`

where `[pose_seq]` and `[mesh_seq]` denote the tokenized pose and mesh sequences, respectively.

3.4. Training

Our autoregressive decoder is trained using a single next-token prediction objective. Given a token sequence $S = (s_1, \ldots, s_T)$ and the aggregated latent $\mathbf{z}_{\text{agg}}$, the training loss is

$$\mathcal{L}_{\text{ce}} = -\sum_{t=1}^{T} \log p_\theta(s_t \mid s_{<t}, \mathbf{z}_{\text{agg}}), \quad (5)$$

where the model predicts each token conditioned on all preceding tokens and the fused point-cloud latent enriched with pixel-aligned image features and global scene context.

As illustrated in Fig. 2, the model autoregressively generates both the pose tokens and the mesh tokens within a single unified sequence. This joint formulation allows the decoder to learn instance geometry and global layout estimation simultaneously, enabling pose reasoning to benefit from geometry cues and vice versa.

4. Experiments

4.1. Settings

Datasets. We conduct experiments on both synthetic and real-world datasets. For training, we use the synthetic indoor dataset 3D-FRONT [12], adopting the preprocessed version provided by InstPIFu [22]. Since the raw 3D-FRONT meshes are extremely high-poly, we apply planar mesh decimation to all object assets to obtain lightweight, artist-compatible meshes suitable for autoregressive generation. Additional preprocessing details are provided in the supplementary material. 3D-FRONT contains over 16K object meshes sourced from 3D-FUTURE [13], along with scene layouts, RGB images, depth maps, and instance segmentation masks.

Following the standard protocol, our training split consists of 22,673 scene images. For evaluation on synthetic data, we use the test subset curated in DepR [43], which includes 100 scenes for object-level evaluation and 156 scenes for scene-level evaluation.

To assess generalization to real-world imagery, we additionally evaluate our trained model on real images from Pix3D [35].

Implementation Details. For 2D visual priors, we follow DepR [43] and employ off-the-shelf models: Grounded-SAM [32] for instance segmentation, DepthPro [2] for monocular depth estimation, and DINOv2 with register tokens [10, 31] as our image feature encoder.

For back-projected point clouds, we adopt the native sampling densities of each mesh generative base model: BPT-based models use 4,096 points per object, whereas EdgeRunner-based models use 8,192 points. For the global scene representation, we uniformly sample 16,384 points.

All point clouds (partial object-level and full scene-level) and object poses are normalized to a unit cube. We apply random augmentation during training, including a vertical-axis rotation in the range $[-45°, 45°]$, scaling in $[0.75, 1]$, and a translation shift in $[0, 0.2]$. We additionally jitter depth values by up to 0.02 to account for inaccuracies in monocular depth estimation. Object meshes are normalized to a unit cube in their respective canonical space.

We train all models on 8 NVIDIA A100 GPUs using AdamW with a learning rate of 1×10^{-4}, 500 warm-up iterations, and cosine decay. The BPT-based variant converges in roughly 10 hours, while the EdgeRunner-based variant

Method	Scene-level			Object-level	
	CD ($\times 10^{-3}$, ↓)	CD-S ($\times 10^{-3}$, ↓)	F-Score (%, ↑)	CD ($\times 10^{-3}$, ↓)	F-Score (%, ↑)
SDF-based					
InstPIFu [22]	213.4	124.9	13.72	44.74	29.63
Uni-3D [41]	218.3	113.3	12.99	—	—
Gen3DSR [11]	222.4	137.5	13.52	9.74	31.42
DeepPriorAssembly [45]	191.8	76.2	16.72	20.13	27.83
MIDI [17]	213.2	155.6	16.02	11.31	64.15
DepR [43]	153.2	56.4	25.00	**2.57**	**89.66**
Mesh-based					
PixARMesh-EdgeRunner (Ours)	107.95	53.33	**28.79**	5.46	<u>76.91</u>
PixARMesh-BPT (Ours)	**100.81**	**49.68**	27.54	<u>5.27</u>	76.63

Table 1. Qualitative comparison with state-of-the-art methods on the 3D-FRONT [12] dataset. Following DepR [43] and DeepPriorAssembly [45], we report object- and scene-level Chamfer Distance (CD; lower is better) and F-Score (higher is better). We additionally include the single-direction Chamfer Distance (CD-S) to account for missing instances.

requires around 30 hours due to its substantially longer token sequence length.

Evaluation Metrics. We evaluate our method using Chamfer Distance (CD) and F-Score, following standard practice in single-view reconstruction [22, 43, 45]. Unless otherwise noted, we use an F-Score threshold of 0.002. Each reconstructed mesh is uniformly sampled into 10k points prior to metric computation.

At the object level, we normalize predicted and ground-truth meshes to a unit cube and compute CD and F-Score to measure the geometric fidelity of individual objects.

At the scene level, we first assemble all predicted instances using their estimated poses. The composed scene, formed by placing each generated mesh into its predicted bounding box, remains in the normalized scene space described in Sec. 3.2. For fair comparison, we apply a global scale and translation to align the predicted scene with the ground-truth scene, which preserves its original metric scale and coordinate frame. Following DeepPriorAssembly [45], we additionally report the single-direction Chamfer Distance (CD-S), which emphasizes reconstruction completeness while ignoring empty background regions.

4.2. Main Results

Quantitative Results. Tab. 1 reports quantitative comparisons on the synthetic 3D-FRONT [12] dataset. We benchmark PixARMesh against representative single-view scene reconstruction approaches, including diffusion-based methods such as DepR [43] and MIDI [17], feed-forward reconstruction frameworks such as InstPIFu [22], and holistic scene methods such as Uni-3D [41]. Because holistic models do not explicitly generate individual object meshes, object-level metrics are not applicable.

Our method achieves highly competitive performance at both the object and scene levels. At the object level, PixARMesh achieves the second-best performance among all approaches, with F-Score comparable to diffusion-based SDF models. Unlike SDF-based pipelines that require Marching Cubes to extract dense iso-surfaces, our approach directly produces compact, artist-ready meshes with only a few thousand faces per instance while maintaining comparable geometric precision. (Further statistics on face counts are provided in the supplementary material.) At the scene level, our method achieves state-of-the-art performance across all reported metrics. We attribute this to our unified autoregressive framework that jointly predicts object geometry and pose, leveraging our pixel-aligned point cloud encoder and scene-level context aggregation for coherent full-scene reconstruction. We also observe that the EdgeRunner-based variant delivers stronger reconstruction performance than the BPT-based variant.

Qualitative Results. We present qualitative comparisons on the synthetic 3D-FRONT [12] dataset in Fig. 3 and on real-world images from Pix3D [35] in Fig. 4.

Across both synthetic and real settings, PixARMesh produces geometrically coherent scene reconstructions, capturing object shapes and spatial arrangements that generally correspond to the input images. Owing to the native artist-created mesh representation, PixARMesh yields meshes with clear edges and well-defined structural boundaries while maintaining smooth surface continuity, leading to cleaner shapes compared to prior approaches.

On real-world images, PixARMesh shows reasonable generalization and can reconstruct indoor environments with practical and interpretable geometry, despite being trained primarily on synthetic data.

Figure 3. Qualitative comparisons on the 3D-FRONT [12] dataset. For PixARMesh, we also show the mesh wireframe to highlight geometric quality.

Figure 4. Qualitative results on real images from the Pix3D [35] dataset.

4.3. Ablation Studies

We conduct ablation experiments on the 3D-FRONT [12] dataset to analyze the effectiveness of key components in PixARMesh. Our study focuses on two aspects: (1) pipeline design - the contribution of each proposed component, and (2) error analysis — the impact of upstream perception errors on overall scene reconstruction.

Method	Scene-level CD ($\times 10^{-3}, \downarrow$)	Scene-level CD-S ($\times 10^{-3}, \downarrow$)	Scene-level F-Score ($\%, \uparrow$)	Object-level CD ($\times 10^{-3}, \downarrow$)	Object-level F-Score ($\%, \uparrow$)
Baseline	61.07	21.42	40.20	5.04	77.54
w/o Pixel-Aligned Feat.	61.00	24.78	41.47	5.11	77.39
w/o Ctx. Aggregation	45.03	**15.35**	42.02	5.02	78.31
Full model	**43.12**	15.64	**43.48**	**4.85**	**79.41**

Table 2. Ablation studies on our point-cloud encoder design. The baseline encoder receives only the partial object point cloud normalized in the global scene coordinate frame.

Point-cloud Encoder Design. To validate our repurposed point-cloud encoder, we evaluate the performance degradation when removing each module individuall, as shown in Tab. 2. We report results using the EdgeRunner-based variant with ground-truth depth and masks; additional results for the BPT-based model are provided in the supplementary.

Removing the pixel-aligned image features causes the largest performance drop, particularly in scene-level Chamfer Distance. Using scene context aggregation alone, without image features, yields only marginal improvement over the baseline and slightly worsens object-level performance. However, when global context aggregation is combined with pixel-aligned image features, the model achieves consistent improvements across both object- and scene-level metrics. This highlights the importance of image appearance cues when incorporating scene context: under heavy occlusions, geometry-only conditioning becomes ambiguous and can mislead the model without complementary visual features.

GT Depth	GT Layout	CD ($\times 10^{-3}, \downarrow$)	F-Score ($\%, \uparrow$)
		6.67	73.06
✓		5.66	76.32
	✓	5.46	76.91
✓	✓	**4.85**	**79.41**

Table 3. Effects of depth and layout in object-level metrics.

Object-Level Error Analysis. Depth estimated from external models can introduce errors that propagate throughout the reconstruction pipeline, while inaccurate layout estimation may misguide subsequent mesh generation. To assess the full potential of our approach, we evaluate ablations using ground-truth depth and ground-truth layout in object-level reconstruction. For this analysis, we use our full EdgeRunner-based model equipped with pixel-aligned features and scene-level context aggregation. As shown in Tab. 3, object reconstruction quality improves when ground-truth depth is provided. Moreover, supplying the ground-truth layout leads to further performance gains, indicating that accurate pose and scale estimation offers essential guidance for generating high-quality mesh sequences.

GT Inputs Depth	GT Inputs Segm	GT Inputs Layout	CD ($\times 10^{-3}, \downarrow$)	CD-S ($\times 10^{-3}, \downarrow$)	F-Score ($\%, \uparrow$)
			107.95	53.33	28.79
✓			105.58	55.36	31.71
	✓		54.51	21.73	37.50
	✓	✓	21.58	6.52	49.84
✓	✓		43.12	15.64	43.48
✓	✓	✓	**21.19**	**6.18**	**50.65**

Table 4. Effects of upstream (depth, segmentation, and layout) errors in scene-level metrics. Note that ground-truth layout implies ground-truth segmentation.

Scene-Level Error Analysis. Our pipeline begins by constructing a raw point cloud using depth maps and segmentation masks predicted by off-the-shelf models. To isolate the impact of upstream perception errors, we report results using ground-truth inputs in Tab. 4. As in the object-level analysis, we use our full EdgeRunner-based model. Following the evaluation protocol in DepR [43], providing ground-truth layout implies using ground-truth segmentation.

The results show that ground-truth segmentation yields the largest improvement in Chamfer Distance, followed by layout and then depth. This sensitivity to segmentation quality is primarily due to missing objects or corrupted point clouds produced by inaccurate instance masks. Interestingly, we observe relatively strong robustness to depth estimation errors, suggesting that the model can still capture sufficient global context even when the depth input is imperfect.

5. Conclusion

We presented PixARMesh, an autoregressive framework for single-view indoor scene reconstruction. By repurposing object-level mesh generative models with pixel-aligned point-cloud encoding and scene-level context aggregation, PixARMesh jointly predicts object pose and geometry, producing coherent full-scene reconstructions without relying on SDFs or post-hoc layout optimization. Our method achieves competitive object-level accuracy and state-of-the-art scene-level performance, while generating compact, artist-ready meshes. Extensive experiments and ablations highlight the effectiveness of our design and its applicability to real-world inputs, demonstrating the promise of autoregressive mesh generation as a viable alternative to SDF-based pipelines.

References

[1] Antonio Alliegro, Yawar Siddiqui, Tatiana Tommasi, and Matthias Nießner. Polydiff: Generating 3d polygonal meshes with diffusion models. *arXiv preprint arXiv:2312.11417*, 2023. 2

[2] Alexey Bochkovskiy, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *The Thirteenth International Conference on Learning Representations*, 2025. 5

[3] 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. 2

[4] 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. 2

[5] Yiwen Chen, Yikai Wang, Yihao Luo, Zhengyi Wang, Zilong Chen, Jun Zhu, Chi Zhang, and Guosheng Lin. Meshanything v2: Artist-created mesh generation with adjacent mesh tokenization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13922–13931, 2025. 2, 3

[6] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 45–54, 2020. 2

[7] Tao Chu, Pan Zhang, Qiong Liu, and Jiaqi Wang. Buol: A bottom-up framework with occupancy-aware lifting for panoptic 3d scene reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4937–4946, 2023. 1, 2

[8] Manuel Dahnert, Ji Hou, Matthias Nießner, and Angela Dai. Panoptic 3d scene reconstruction from a single rgb image. *Advances in Neural Information Processing Systems*, 34: 8282–8293, 2021. 1, 2

[9] 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. 2

[10] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*. 5

[11] Andreea Dogaru, Mert Özer, and Bernhard Egger. Gen3DSR: Generalizable 3d scene reconstruction via divide and conquer from a single view. In *International Conference on 3D Vision 2025*, 2025. 1, 2, 6

[12] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, and Hao Zhang. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10933–10942, 2021. 2, 5, 6, 7, 8

[13] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129(12):3313–3337, 2021. 5

[14] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018. 2

[15] 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. 3

[16] 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. 1

[17] Zehuan Huang, Yuan-Chen Guo, Xingqiao An, Yunhan Yang, Yangguang Li, Zi-Xin Zou, Ding Liang, Xihui Liu, Yan-Pei Cao, and Lu Sheng. Midi: Multi-instance diffusion for single image to 3d scene generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23646–23657, 2025. 1, 2, 6

[18] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 1, 2

[19] Justin Lazarow, David Griffiths, Gefen Kohavi, Francisco Crespo, and Afshin Dehghan. Cubify anything: Scaling indoor 3d object detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22225–22233, 2025. 4

[20] Jiabao Lei, Kewei Shi, Zhihao Liang, and Kui Jia. ARMesh: Autoregressive mesh generation via next-level-of-detail prediction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 2, 3

[21] 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*, pages 26608–26617, 2025. 3

[22] Haolin Liu, Yujian Zheng, Guanying Chen, Shuguang Cui, and Xiaoguang Han. Towards high-fidelity single-view holistic reconstruction of indoor scenes. In *European Conference on Computer Vision*, pages 429–446. Springer, 2022. 5, 6

[23] 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. 3

[24] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36:22226–22246, 2023. 1

[25] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to

3d objects with consistent multi-view generation and 3d diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10072–10083, 2024.

[26] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023. 1, 2

[27] Sainan Liu, Vincent Nguyen, Yuan Gao, Subarna Tripathi, and Zhuowen Tu. Towards panoptic 3d parsing for single image in the wild. *arXiv preprint arXiv:2111.03039*, 2021. 2

[28] 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. 2

[29] 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. 2

[30] Piotr Nawrot, Szymon Tworkowski, Michał Tyrolski, Łukasz Kaiser, Yuhuai Wu, Christian Szegedy, and Henryk Michalewski. Hierarchical transformers are more efficient language models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1559–1571, 2022. 3

[31] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. Featured Certification. 5

[32] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 5

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

[34] 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. 2

[35] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2974–2983, 2018. 5, 6, 7

[36] 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. 2, 3, 4

[37] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018. 2

[38] Haohan Weng, Yikai Wang, Tong Zhang, C. L. Philip Chen, and Jun Zhu. Pivotmesh: Generic 3d mesh generation via pivot vertices guidance. In *The Thirteenth International Conference on Learning Representations*, 2025. 3

[39] 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*, pages 11093–11103, 2025. 2, 3, 4

[40] 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. 1, 2

[41] Xiang Zhang, Zeyuan Chen, Fangyin Wei, and Zhuowen Tu. Uni-3d: A universal model for panoptic 3d scene reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9256–9266, 2023. 1, 2, 6

[42] Xiang Zhang, Yawar Siddiqui, Armen Avetisyan, Chris Xie, Jakob Engel, and Henry Howard-Jenkins. Vertexregen: Mesh generation with continuous level of detail. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12570–12580, 2025. 2, 3

[43] Qingcheng Zhao, Xiang Zhang, Haiyang Xu, Zeyuan Chen, Jianwen Xie, Yuan Gao, and Zhuowen Tu. Depr: Depth guided single-view scene reconstruction with instance-level diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5722–5733, 2025. 1, 2, 3, 5, 6, 8

[44] Ruowen Zhao, Junliang Ye, Zhengyi Wang, Guangce Liu, Yiwen Chen, Yikai Wang, and Jun Zhu. Deepmesh: Auto-regressive artist-mesh creation with reinforcement learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10612–10623, 2025. 3

[45] Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Zero-shot scene reconstruction from single images with deep prior assembly. *Advances in Neural Information Processing Systems*, 37:39104–39127, 2024. 1, 2, 3, 6