

Delaunay Canopy: Building Wireframe Reconstruction from Airborne LiDAR Point Clouds via Delaunay Graph

Donghyun Kim^{1,2}, Chanyoung Kim^{1,2}, Youngjoong Kwon^{2*}, and Seong Jae Hwang^{1*}

¹ Yonsei University, Seoul, Republic of Korea

seongjae@yonsei.ac.kr

² Emory University, Atlanta, GA, USA

{donghyun.kim, chanyoung.kim, youngjoong.kwon}@emory.edu

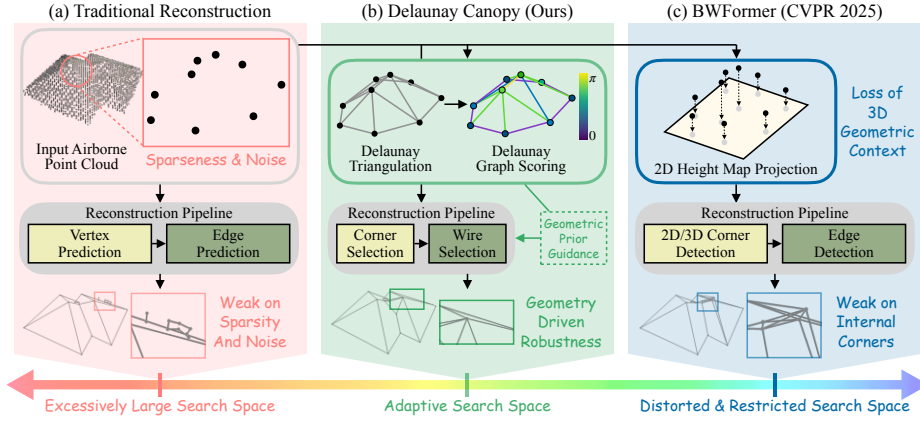


Fig. 1: We present Delaunay Canopy for building wireframe reconstruction. Compared to traditional methods with overly large search space (left) and 2D height map projection that loses 3D context (right), our approach scores a Delaunay graph with geometric priors to select corners and wires, yielding an adaptive (*i.e.*, geometrically context-aware) search space that enables precise wireframe reconstruction.

Abstract. Reconstructing building wireframe from airborne LiDAR point clouds yields a compact, topology-centric representation that enables structural understanding beyond dense meshes. Yet a key limitation persists: conventional methods have failed to achieve accurate wireframe reconstruction in regions afflicted by significant noise, sparsity, or internal corners. This failure stems from the inability to establish an adaptive search space to effectively leverage the rich 3D geometry of large, sparse building point clouds. In this work, we address this challenge with Delaunay Canopy, which utilizes the Delaunay graph as a geometric prior to define a geometrically adaptive search space. Central to our approach is Delaunay Graph Scoring, which not only reconstructs the underlying

* Corresponding authors.

geometric manifold but also yields region-wise curvature signatures to robustly guide the reconstruction. Built on this foundation, our corner and wire selection modules leverage the Delaunay-induced prior to focus on highly probable elements, thereby shaping the search space and enabling accurate prediction even in previously intractable regions. Extensive experiments on the Building3D Tallinn city and entry-level datasets demonstrate state-of-the-art wireframe reconstruction, delivering accurate predictions across diverse and complex building geometries.

Keywords: 3D reconstruction · LiDAR point clouds · Delaunay graph

1 Introduction

3D building data serves as a cornerstone across diverse domains such as the metaverse, XR (Extended Reality) [5, 24, 29], autonomous driving [7, 11, 33], gaming [4, 6], and robotics [1, 23, 28]. Acquiring real-world 3D building data typically begins by scanning buildings with LiDAR sensors to obtain a 3D point cloud. Although point clouds are highly accurate representations [34–37, 39], their inherent sparsity and irregularity, coupled with data size that scales with object size, may pose challenges for their use as lightweight representations for the aforementioned applications [22]. Consequently, there is a need for techniques that reconstruct 3D building point clouds into precise and lightweight representations, and the *wireframe* has increasingly become the format that best satisfies these requirements. Nevertheless, robustly dealing with the point cloud’s sparsity and irregularity remains a key challenge, as wireframe reconstruction hinges on accurately recovering sharp edges, corners, and topological connectivity, which are highly sensitive to noise and missing data.

Despite the glaring challenge, studies on reconstructing 3D building point clouds into wireframes have advanced rapidly. Beginning with the first network-based Point2Roof [13], subsequent studies [16, 32] in wireframe reconstruction have adopted a pipeline that first predicts corners of the wireframe and then predicts wires. This pipeline simplifies the overall problem by decomposing a complex task into smaller modules; however, it fails to account for the enormous scale, sparsity, and noise of building point clouds (Fig. 1(a)), resulting in an *excessively large search space* that challenges accurate reconstruction.

To address this, BWFormer [18] introduced a 2D-to-3D corner detection approach in which the point cloud is projected onto a 2D height map (Fig. 1(c)). In the 2D dimension, corners are detected first, and these detections are then used to infer the 3D corners. While this dimensionality reduction constrains the search space, such constraint becomes *overly restrictive*, leading to critical limitations due to the loss of intricate 3D contextual information. Most critically, BWFormer struggles to detect interior corners, particularly because it restricts the search space to a 2D BEV (Bird’s Eye View) perspective, thereby failing to fully leverage given 3D information (Fig. 2). A related drawback is that projecting the continuous 3D point cloud into a 2D grid representation distorts the inherent 3D search space, as the 3D coordinates must be quantized into integer

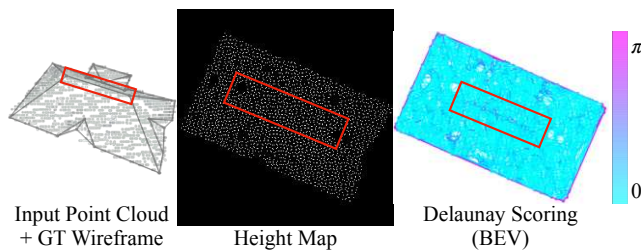


Fig. 2: Comparison of geometric context between 2D heightmap and Delaunay scoring. While the 2D heightmap (center) collapses 3D structural details and obscures internal corners (red box), our Delaunay scoring (right) effectively preserves the underlying geometric context.

indices. Thus, in 3D building wireframe reconstruction, substantial effort must be paid to establish an *adaptive search space* over point clouds that is informed by the underlying 3D context, ensuring compactness to suppress the detrimental effects of sparsity and noise while preserving adequate scope to encompass all geometrically valid structures.

Our 3D building wireframe reconstruction framework, *Delaunay Canopy*, aims to ensure an adaptive search space by leveraging the geometric prior of the Delaunay graph (Fig. 1(b)). The Delaunay graph, constructed via Delaunay triangulation of the input point cloud, captures 3D geometric information by approximating the underlying surface of a building roof in 3D space, irrespective of sparsity and scale. Motivated by the fact that critical points (*e.g.*, roof corners) exhibit high local curvature, we propose *Delaunay graph scoring*, the core approach of our framework designed to extract curvature signatures from the graph for guiding reconstruction (Fig. 3). Specifically, by computing the *dihedral angle* along an edge (shared by two triangular faces), one can infer the local curvature of the underlying surface at that location. These dihedral angles are then aggregated for each vertex to calculate a *corner score*, representing the degree of local curvature associated with that vertex.

We leverage the curvature signatures from Delaunay graph scoring to predict wireframe corners and wires. The resulting Delaunay-induced prior concentrates the corner and wire selection on highly probable elements (*i.e.*, candidates in high-curvature regions), scoping the search space and enabling accurate prediction in challenging regions (*e.g.*, internal corners). With this prior in place, we adaptively narrow the search space by first sampling and classifying input points with high corner scores as corner candidates, and then train a corner selection module to predict the final corners. For wire selection, we generate wire candidates from predicted corners and assign a *path score* by summing dihedral angles along the shortest Delaunay graph path between a candidate’s endpoints. Subsequently, we scale the query corresponding to each candidate based on the path score, and the wire selection module is trained to amplify information about promising candidates, thereby selecting valid wires. Through a carefully designed

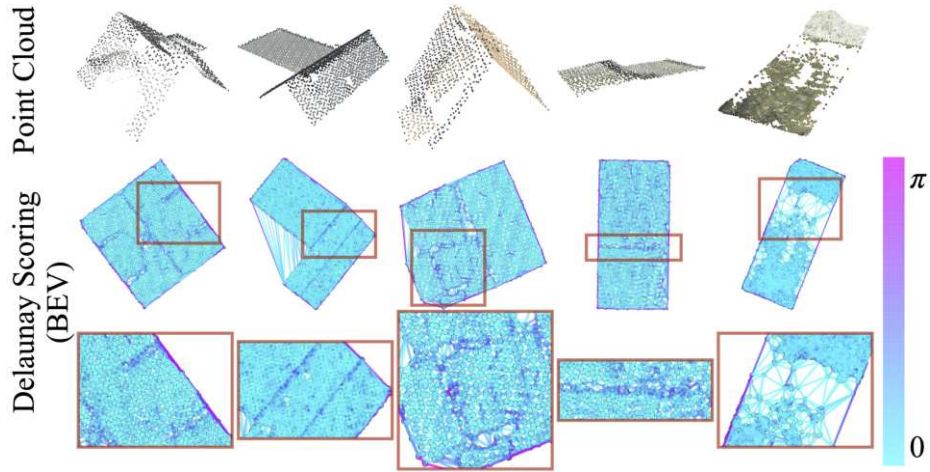


Fig. 3: Visualization of the edge-wise dihedral angle and vertex-wise corner score, derived through Delaunay graph scoring across input point clouds, projected onto the Bird’s-Eye View (BEV). The results demonstrate a robust capability to accurately capture the underlying surface and the curvature of the corresponding region. Notably, the scoring adeptly represents even the most minute degrees of surface curvature (fourth column) and maintains efficacy even when the point cloud is highly sparse (fifth column).

pipeline, our Delaunay Canopy adaptively calibrates the search space using Delaunay graph scoring, a strong geometric cue, which in turn yields accurate final wireframe through a corner and wire selection module.

The main contributions of this work are:

- We propose Delaunay graph scoring, a method that applies Delaunay triangulation to the point cloud to quantify local curvature, transforming the underlying surface geometry into essential curvature signatures.
- We introduce a corner and wire selection pipeline that leverages a geometric cue to adaptively adjust the search space, enabling the model to focus on probable candidates and achieve precise results in challenging regions.
- We demonstrate the effectiveness of Delaunay Canopy on the Building3D dataset and achieve state-of-the-art results, supported by a comprehensive experimental study.

2 Related Work

Building Reconstruction. Building reconstruction plays a crucial role in vision applications [10,13,14,17,18,21,30]. Recently, some methods such as City3D [9] and Kinetic Shape Reconstruction [2] have been proposed to directly output surface meshes by enforcing global plane primitives and topological correctness. While these approaches guarantee watertightness, they often over-close

sparse topology by forcing global consistency. In contrast, wireframe representations offer a more flexible alternative by prioritizing structural correctness and superior editability over forced watertightness, focusing on concise, topology-centric descriptions [10, 13, 18]. Pioneering work [13] detects corners and predicts wires to assemble wireframes, but it often misses corners due to the sparsity of point clouds. PBWR [10] outputs edges in a direct fashion, yet it still requires post-processing and falls short of being a unified pipeline. More recently, BW-Former [18] introduced a 2D-to-3D corner detection approach that projects the point cloud to a height map, detects 2D corners, and then infers 3D corners. However, this projection quantizes 3D coordinates on a 2D grid, introduces geometric errors, requires corner detection in both 2D and 3D with doubled computational cost, and overly constrains the search space that it underuses 3D cues and struggles with interior corners in BEV. In contrast, our model operates explicitly in 3D with a Delaunay graph representation, while leveraging geometric cues for adaptive search space guidance.

Delaunay Graph for 3D Reconstruction. Delaunay triangulation is a classical geometric tool for organizing unstructured 3D point clouds into surface and has been widely used in reconstruction tasks [19, 27, 31]. Its graph structure provides a powerful geometric prior, as edges naturally encode local neighborhood relationships and capture nearest-neighbor connections [40]. This property confines the search space to spatially meaningful candidate edges on the underlying surface. Classical methods like the Ball-Pivoting Algorithm [3] have successfully leveraged this connectivity to guide surface growth and ensure manifold consistency. In our approach, we exploit the Delaunay graph to define the search space for wireframe reconstruction. Each triangulation edge represents a potential roof edge, and its dihedral angle is used to estimate local curvature. Aggregating these angles yields a corner score, enabling the selection of high-curvature corner candidates. This Delaunay-based formulation ensures a geometrically informed and computationally efficient search for structural elements grounded in the input geometry.

3 Method

In this section, we introduce our 3D building reconstruction framework, *Delaunay Canopy*. We first describe the *Delaunay graph scoring* (Sec. 3.1), which serves as a foundational module providing a 3D geometric guidance within our framework. Subsequently, we present the corner and wire selection pipeline (Sec. 3.2), guided by the geometric prior derived from Delaunay graph scoring.

3.1 Delaunay Graph Scoring

Key features in 3D building (*e.g.*, corners) exhibit pronounced local curvature. This observation raises the question: *Can a curvature metric be leveraged as a strong prior to inform the topology and guide wireframe reconstruction?* To investigate, we first introduce *Delaunay graph scoring*, which approximates the

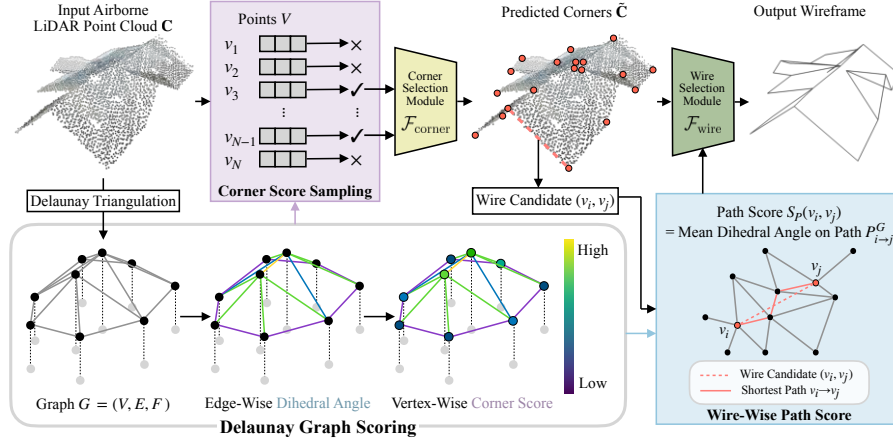


Fig. 4: Overall pipeline of Delaunay Canopy. The pipeline begins by utilizing Delaunay triangulation to construct the point cloud into a Delaunay graph. The method then sequentially computes metrics that represent the local curvature around each element, proceeding in the order of face, edge, and vertex. This process encapsulates rich 3D information and introduces a judicious search space into the pipeline.

underlying surface of the point cloud and computes region-wise curvature metrics.

Unlike many conventional 3D object point clouds [8, 38], airborne LiDAR scans of building roofs inherently represent surfaces with open topologies: for a given (x, y) coordinate, the z value is mapped without occlusion. This property enables us to estimate an implicit representation of the point cloud’s underlying surface. Building upon this, we apply *Delaunay Triangulation* to the airborne LiDAR building point cloud to approximate its underlying surface. This process generates a Delaunay graph $G = (V, E, F)$, which holds both mesh and graph properties and comprises vertices V , edges E , and faces F . To estimate the curvature of a specific region, we derive geometric priors specifically for curvature characterization upon the Delaunay graph G . These priors are assigned to the graph’s components as follows: (i) faces F via *Face Normal Computation*, (ii) edges E via *Edge-Wise Dihedral Angle*, and (iii) vertices V via *Vertex-Wise Corner Score* (see Fig. 4). We describe each step in a detail below.

(I) Face Normal Computation. For every face $f = (v_a, v_b, v_c)$ within a Delaunay graph G , let the 3D coordinates of its three constituent vertices be $\mathbf{v}_a$, $\mathbf{v}_b$, and $\mathbf{v}_c$. The normal vector $\mathbf{n}_f$ for each face f is computed as $\mathbf{n}_f = (\mathbf{v}_b - \mathbf{v}_a) \times (\mathbf{v}_c - \mathbf{v}_a) \in \mathbb{R}^3$. To ensure consistent normal orientation, if the calculated normal vector points downward (*i.e.*, the z -component of $\mathbf{n}_f$ is less than zero), its direction is inverted to $-\mathbf{n}_f$.

(II) Edge-Wise Dihedral Angle. With the face normals in hand, we now compute curvature-related geometric information at the edge level by computing the *dihedral angle* for each edge. For every edge e , we define the set of two

adjacent faces as follows: $F(e) = \{f \in F | e \subset f\}$. Given an edge e with its adjacent faces $F(e) = \{f^1, f^2\}$, and their respective normal vectors $\mathbf{n}_1$ and $\mathbf{n}_2$, the dihedral angle θ_e is computed as

$$\theta_e = \arccos \left(\frac{\mathbf{n}_1 \cdot \mathbf{n}_2}{\|\mathbf{n}_1\| \|\mathbf{n}_2\|} \right) \in [0, \pi]. \quad (1)$$

A large dihedral angle at an edge signifies a sharp fold or corner where the two adjacent faces meet, whereas a small angle indicates that the region is relatively flat. For edges situated on the periphery and thus abutting only a single surface, the dihedral angle θ_e is configured to π .

(III) Vertex-Wise Corner Score. At the most granular level of the graph, the vertex, we compute a *corner score* using curvature information from its incident edges. For a vertex v , its corner score $S_c(v)$ is determined by aggregating the dihedral angles of all incident edges $E(v) = \{e \in E | v \in e\}$, defined as

$$S_c(v) = \frac{1}{|E(v)|} \sum_{e \in E(v)} \theta_e. \quad (2)$$

A high corner score implies the presence of numerous edges with high dihedral angles, suggesting that the local region around that vertex has significant curvature and the vertex is highly likely to be an actual corner of the building.

3.2 Corner and Wire Selection

Delaunay graph scoring, described in Sec. 3.1, captures local 3D geometric information and provides region-wise curvature metrics. These metrics enable adaptive scoping of the search space for predicting the final corners and wires of the wireframe. Let us describe how the wireframe is reconstructed (see Fig. 4) by utilizing Delaunay-informed priors through the following stages: corner selection and wire selection.

Corner Selection. Corners define the key geometric structure of a building, marking where planes meet or change direction. However, airborne point clouds contain dense planar samples, most of which are not structurally meaningful. The goal of the corner selection stage is therefore to (i) isolate a small, geometry-aware subset of points that are likely to represent true corners, and then (ii) refine them into the final set of corners.

(I) Corner Score Sampling. We first narrow the search space by sampling vertices that exhibit high corner scores, computed using the *vertex-wise corner score* (see Sec. 3.1). Specifically, from the input point cloud $\mathbf{C} \in \mathbb{R}^{N \times 3}$, we select the top K points with the highest corner scores, forming the subset $\mathbf{C}' \in \mathbb{R}^{K \times 3}$. As illustrated in Fig. 5, decreasing K yields sampled points that are progressively more densely located along the roof’s boundaries and junctions, indicating a higher likelihood of representing actual building corners. This step effectively removes redundant planar points, and yields an *adaptive and focused search space* guided by Delaunay priors.

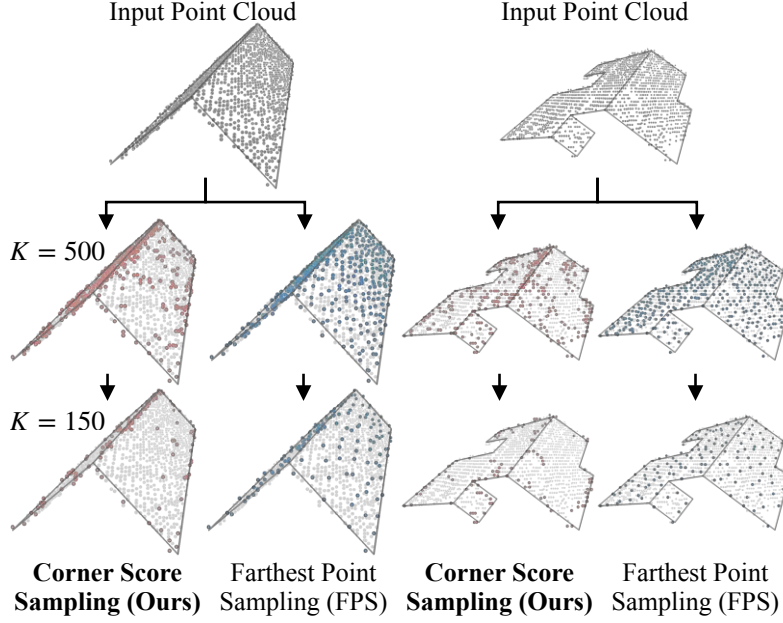


Fig. 5: Comparison between Corner Score Sampling (ours) and Farthest Point Sampling (FPS). Resultant point sets after sequentially sampling 500 points ($K = 500$) and then 150 points ($K = 150$) with each method are shown. For enhanced clarity, the ground truth wireframe is visualized. While FPS performs a uniform sampling, our corner score sampling method retains points located in areas of pronounced curvature (e.g., roof corners and edges), making them highly probable corner candidates.

(II) Corner Selection Module. Then, the subset $\mathbf{C}'$ is subsequently passed into the corner selection module, expressed as a learnable function $\mathcal{F}_{\text{corner}}(\cdot)$, which is trained to refine $\mathbf{C}'$ to the final set of predicted corners $\tilde{\mathbf{C}} \in \mathbb{R}^{M \times 3}$, formally as $\tilde{\mathbf{C}} = \mathcal{F}_{\text{corner}}(\mathbf{C}')$. Details of the training process are provided in the supplementary.

Wire Selection. Once the corners are identified, the next step is to determine which pair of corners form valid wires using a differentiable (i) *wire selection module*. However, treating wire candidates that possess distinct geometric properties in the same manner can potentially lead to distorted information processing. Thus, we introduce (ii) *wire-wise path score*, which measures geometric coherence, and use it within (iii) *prior-based query scaling* to dynamically weight each wire query. Let us first describe the wire selection module.

(I) Wire Selection Module. The wire selection module $\mathcal{F}_{\text{wire}}$ takes all $\binom{M}{2}$ wire candidates $\mathcal{W} = \{w_{ij} = (v_i, v_j) \mid 1 \leq i < j \leq M\}$ as input, which are constructed from every possible pair of the M predicted corners, and determines

whether each candidate w_{ij} constitutes a valid wire. The module is built upon a multi-layer Transformer decoder [42] that classifies each wire candidate using its spatial configuration. Each candidate w_{ij} is represented as a query and, for layer l , is encoded as a wire-specific query vector Q_l^w , which is initialized from the 3D coordinates of its endpoints v_i and v_j at the first decoder layer. Within each decoder layer, the model performs self-attention among all queries and edge attention [18], where features are uniformly sampled along the edge and aggregated via max-pooling. After all layers have been processed, the final query embeddings are passed through an MLP to produce a validity score for each wire. Additional details are included in the supplementary. Although this effectively models relationships among wire candidates, it treats all queries as equally important. In reality, wire candidates vary substantially in geometric plausibility (*e.g.*, some follow clear structural ridges, while others cross unstructured regions). Giving all queries equal weight may therefore divert model capacity toward implausible wires. To address this gap, we incorporate the Delaunay-derived *path score* as a dynamic weighting factor.

(II) Wire-Wise Path Score. For a wire candidate w_{ij} connecting vertices v_i and v_j , we first find the shortest path $P_{i \rightarrow j}^G$ between these two vertices on the Delaunay graph G . The edges along this path, denoted as $L(P_{i \rightarrow j}^G)$, share the same local region as the wire candidate. By averaging the dihedral angles of these edges, we can compute the path score $S_P(v_i, v_j)$, which represents the wire curvature:

$$S_P(v_i, v_j) = \frac{1}{|L(P_{i \rightarrow j}^G)|} \sum_{e \in L(P_{i \rightarrow j}^G)} \theta_e. \quad (3)$$

The path score subsequently serves as the primary value used by the wire selection module to determine valid wires.

(III) Prior-based Query Scaling. At each decoder layer l , the query vector Q_l^w of a wire candidate w is modulated by its path score S_P^w . The score is mapped through a sigmoid activation $s^w = \sigma(S_P^w)$ to produce a scale factor that is applied element-wise: $Q_l'^w = Q_l^w \odot s^w$. This modulation rescales query magnitudes, damping low-score wires and emphasizing high-confidence ones. Consequently, the model’s attention is focused on geometrically plausible structures, leading to stable and accurate wire predictions.

4 Experiments

In this section, we first outline the experimental setup, including the datasets, evaluation metrics, and implementation details. We then present both quantitative and qualitative results, followed by analyses, including ablation studies and a detailed examination of each proposed component. Additional experiments are provided in the supplementary.

4.1 Experimental Setup

Datasets. To evaluate the wireframe reconstruction performance of the Delaunay Canopy, we utilize the Building3D dataset [32], which encompasses both the Tallinn city and entry-level datasets. Crucially, the complete complement of input point cloud and ground truth (GT) wireframe pairs for both datasets is not available in an entirely open-source format; only a subset has been publicly disclosed. Consequently, from the Tallinn city dataset, we employ the 32,618 publicly available input-GT pairs, which are randomly partitioned into a training set of 30,000 and a test set of 2,618. For the entry-level dataset, we employ the total of 5,698 disclosed pairs, splitting them randomly into 5,000 for the training set and the remaining 698 for the test set. To ensure a fair comparison with established benchmarks, we also evaluate our model following the Building3D official leaderboard protocols as detailed in the supplementary material.

Metrics. We evaluate our method using eight common metrics for building wireframe reconstruction. Distance metrics include Wireframe Edit Distance (WED) and Average Corner Offset (ACO). We also incorporate corner-centric metrics (Corner Precision (CP), Corner Recall (CR), and Corner F1 (CF1)) and wire-level metrics (Edge Precision (EP), Edge Recall (ER), and Edge F1 (EF1)). In the tables, the corner and wire metrics are presented as percentages (%).

Baselines. The representative baselines selected for comparison are PC2WF [16], Point2Roof [13], and BWFormer [18]. Additionally, a baseline setting utilizing PointNet [25], as introduced in Building3D [32], is also implemented and evaluated. Consistent with prior works [10, 18], evaluation of [16] utilizes a model pretrained on ABC dataset [12]. Notably, to resolve potential performance discrepancies, we train and evaluate Point2Roof [13] on the identical datasets used for all other baselines, whereas prior works [10, 18, 32] often evaluated it using models pretrained on synthetic data. Other prior works, such as PBWR [10] and various other methods (*e.g.*, linear self-supervised) reported in Building3D [32], can not be utilized for performance assessment due to the unavailability of their code. To establish a more diverse set of baselines, we measure performance by substituting diverse point cloud processing backbones (PointTransformer [41], PointMLP [20], PointNeXt [26], and PointMeta [15]) for the feature extraction component within the Point2Roof [13] pipeline, analogous to the methodology employed in Building3D [32].

Implementation Details. All airborne LiDAR point clouds utilized as input are scaled to a range between 0 and 256. Our model is trained for 300 epochs using AdamW with an initial learning rate of 0.0001. Within our framework, the corner selection module and the wire selection module both use a Transformer-based architecture with 6 layers each. In the corner selection process, the number of points sampled via *corner score sampling* is set to 150. Additionally, within the wire selection module, the *prior-based query scaling* is only implemented across the first three transformer layers.

Table 1: Quantitative comparisons on the Building3D Tallinn city and entry-level datasets. The notation MODEL* signifies a model acting as the feature extractor within the Point2Roof [13] pipeline. **Best**, **second best**, and **third best** are highlighted.

Method	Tallinn City									Entry-Level								
	Distance			Corner			Wire			Distance			Corner			Wire		
	WED ↓	ACO ↓	CP ↓	CP ↑	CR ↑	CF1 ↑	EP ↑	ER ↑	EF1 ↑	WED ↓	ACO ↓	CP ↓	CP ↑	CR ↑	CF1 ↑	EP ↑	ER ↑	EF1 ↑
Building3D (PointNet) [25]	0.264	0.241	88.2	77.6	82.6	80.4	71.9	75.9		0.262	0.239	89.1	77.8	83.1	80.7	72.7	76.5	
PC2WF [16]	0.554	0.508	21.4	50.5	30.1	2.8	16.7	4.8		0.490	0.422	28.5	16.3	20.7	3.1	19.1	5.3	
Point2Roof [13]	0.260	0.236	89.3	78.5	83.6	81.2	72.2	76.4		0.257	0.238	89.5	78.1	83.4	81.4	73.4	77.2	
PointTransformer* [41]	0.257	0.238	89.7	79.1	84.1	81.3	72.4	76.6		0.259	0.233	89.7	79.4	84.2	81.5	73.8	77.5	
PointMLP* [20]	0.255	0.233	90.6	79.0	84.4	81.6	72.8	76.9		0.255	0.229	89.8	79.7	84.4	81.4	74.1	77.6	
PointNeXt* [26]	0.256	0.229	90.4	79.4	84.5	82.0	73.9	78.0		0.255	0.231	91.2	79.9	85.2	82.7	74.5	78.4	
PointMeta* [15]	0.251	0.225	90.6	79.8	84.9	82.4	73.3	77.6		0.253	0.222	90.9	80.9	85.6	83.0	75.2	78.9	
BWFormer [18]	0.245	0.213	91.4	80.1	85.4	84.2	74.6	79.1		0.242	0.201	92.4	82.8	87.3	84.9	77.6	81.1	
Delaunay Canopy	0.232	0.194	94.5	82.8	88.3	87.1	74.0	80.0		0.230	0.190	95.1	85.7	90.2	87.5	77.9	82.4	

4.2 Evaluation Results

Quantitative Evaluation. We conduct experiments measuring the wireframe reconstruction performance of the Delaunay Canopy against other methods [13, 18] on both the Tallinn city and entry-level datasets from Building3D [32]. As evident in Table 1, our framework performs favorably against other methods on the Tallinn city dataset. Table 1 further confirms that our framework exhibits superior performance across all metrics on the entry-level dataset. This indicates that our pipeline, by establishing an adaptive search space and precluding unnecessary candidates, possesses an enhanced capacity for selecting valid corners and wires. Furthermore, to ensure the robustness of our findings against potential dataset biases, we evaluate our framework from scratch on the Tokyo LoD2 dataset [32], where it consistently outperformed recent state-of-the-art methods. Detailed results on this dataset, along with further experimental results including K -fold cross-validation, are provided in the supplementary material.

Qualitative Analysis. Fig. 6 shows the qualitative comparison between our framework and recent state-of-the-art methods [13, 18]. It is particularly noteworthy that the most challenging aspects of reconstruction—namely, areas that are severely sparse, highly noisy, or where subtle curvature is imperceptible—are notoriously difficult to render faithfully. Our framework, however, successfully reconstructs these complex, high-difficulty sections. Furthermore, Fig. 7 presents a comparison with BWFormer [18] on several challenging cases featuring complicated roof structures. While BWFormer utilizes a height map as input and consequently forfeits 3D geometric context, making it vulnerable in detecting internal corners (those not on the boundary), our framework selects these interior corners with superior efficacy by fully leveraging the available 3D information. Finally, Fig. 8 demonstrates the robustness of our approach on highly complex real-world buildings. As visually evidenced, our framework successfully reconstructs accurate wireframes even in challenging cases afflicted by surrounding clutter (Fig. 8(a)), severe occlusion and missing data (Fig. 8(b)), and intricate,

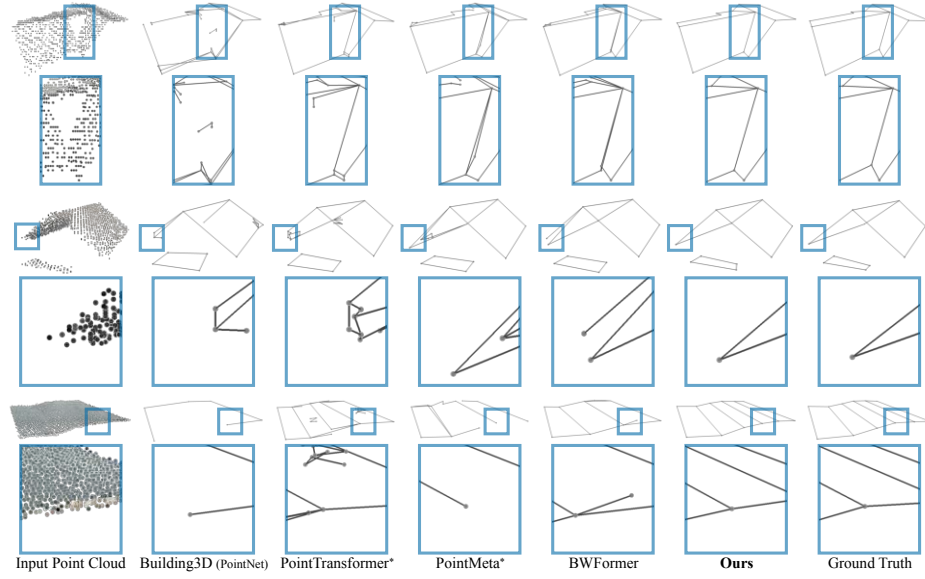


Fig. 6: Qualitative comparison with other methods. The notation MODEL* signifies a model acting as the feature extractor within the Point2Roof [13] pipeline. The blue boxes illustrate the robustness of our framework, demonstrating its capability to reconstruct the wireframe even in regions that are severely sparse (first row), highly noisy (second row), or exhibit minute degrees of curvature (third row).

complex structures (Fig. 8(c)). Extended qualitative comparisons and a detailed robustness analysis under diverse challenging scenarios are available in the supplementary.

4.3 Additional Analyses

Ablation Study. We conduct an ablation study on the Tallinn city dataset to demonstrate the efficacy of our proposed method, which guides the overall pipeline with a geometric prior via Delaunay graph scoring. We measure how performance shifts based on the intervention of two components: the *corner score sampling* performed during the corner selection process and the *prior-based query scaling* within the wire selection module. In the setting where corner score sampling is not employed, 150 points were selected as input via Farthest Point Sampling (FPS). Conversely, when prior-based query scaling is not utilized, the process followed the conventional flow of a standard Transformer. As shown in Table 2, performance conclusively improves as each method is incrementally introduced, with the setting utilizing all of our proposed methods ultimately achieving the superior performance.

Analysis of Corner Score Sampling. Table 3 illustrates the performance variation of the *corner score sampling*, which specifically depends on the value

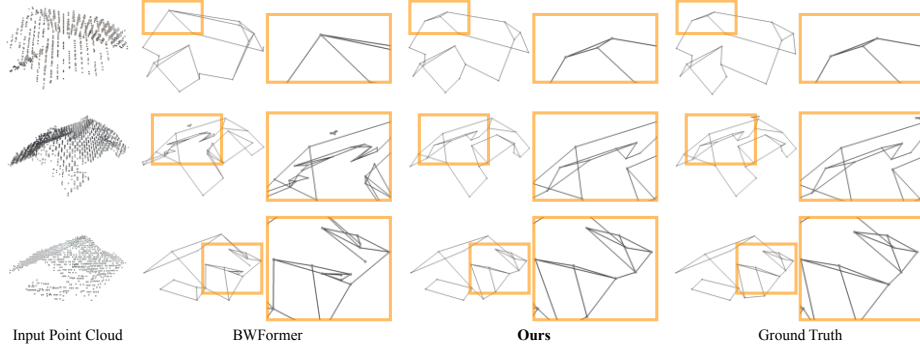


Fig. 7: Qualitative comparison with the strongest baseline BWFormer [18] on challenging cases with complicated roof structures. The orange boxes show our method’s ability to fully leverage 3D geometric context, a capability often absent when only 2D information is employed, thus facilitating superior detection of interior corners in addition to those on the periphery.

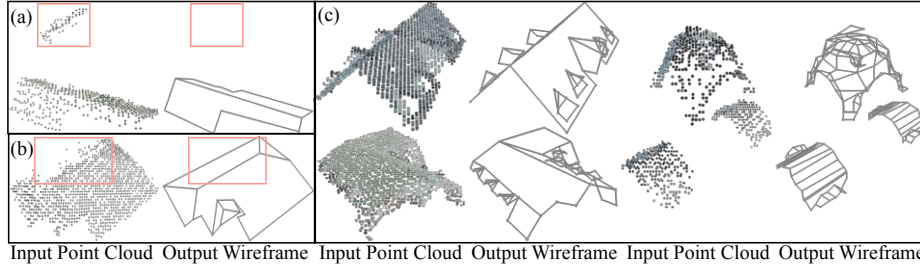


Fig. 8: Qualitative results on complex buildings. Our framework exhibits strong robustness across challenging real-world scenarios, successfully handling (a) cluttered, (b) occluded, and (c) highly complex geometrical cases.

of K , representing the number of vertices with a high corner score selected from the input point cloud. If an insufficient number of points is filtered, the search space becomes unduly constrained, potentially excluding valid corners from the candidate pool. Conversely, filtering too many points enlarges the search space, a limitation observed in prior methods, and reduces the benefit of the geometric prior. The experiment conclusively demonstrates that the setting filtering 150 vertices yields the most favorable overall performance.

Analysis of Prior-based Query Scaling. Table 4 delineates the shift in performance contingent upon the number of layers to which the prior-based query scaling is applied within the wire selection module. Performance progressively improves as the number of layers receiving the scaling is incrementally increased from the initial layer. However, applying query scaling to the later layers risks diluting the semantically rich features that the model has independently generated. Consequently, based on the setting that yields the most excellent performance

Table 2: Ablation study on corner score sampling and prior-based query scaling. **Best** and **second best** are highlighted.

Score Sampling	Query Scaling	Distance		Corner			Wire		
		WED ↓	ACO ↓	CP ↑	CR ↑	CF1 ↑	EP ↑	ER ↑	EF1 ↑
-	-	0.257	0.234	88.9	78.2	83.2	81.1	72.1	76.3
✓	-	0.239	0.210	93.1	81.1	86.7	85.6	73.7	79.2
-	✓	0.242	0.223	92.4	80.3	85.9	84.9	72.5	78.2
✓	✓	0.232	0.194	94.5	82.8	88.3	87.1	74.0	80.0

Table 3: Evaluation based on the number of sampled points in corner score sampling. **Best** and **second best** are highlighted.

# of Points	Distance		Corner			Wire		
	WED ↓	ACO ↓	CP ↑	CR ↑	CF1 ↑	EP ↑	ER ↑	EF1 ↑
100	0.253	0.235	89.3	80.2	84.5	84.1	70.4	76.7
125	0.249	0.219	91.4	82.9	87.0	86.2	73.9	79.6
150	0.232	0.194	94.5	82.8	88.3	87.1	74.0	80.0
175	0.243	0.223	92.1	80.7	86.1	85.9	74.5	79.8
200	0.260	0.237	90.6	79.2	84.5	84.7	72.2	77.9

Table 4: Performance variation based on the number of initial layers to which prior-based query scaling is applied. **Best** and **second best** are highlighted.

# of Layers	Distance		Corner			Wire		
	WED ↓	ACO ↓	CP ↑	CR ↑	CF1 ↑	EP ↑	ER ↑	EF1 ↑
0	0.239	0.210	93.1	81.1	86.8	85.6	73.7	79.2
1	0.233	0.205	93.5	81.6	87.1	85.8	73.6	79.2
2	0.230	0.197	94.2	82.2	87.7	86.7	74.2	80.0
3	0.232	0.194	94.5	82.8	88.3	87.1	74.0	80.0
4	0.237	0.204	93.9	82.3	87.7	87.3	73.5	79.8
5	0.246	0.211	93.2	81.9	87.3	86.5	73.1	79.2
6	0.255	0.213	92.8	80.8	86.4	85.2	72.4	78.3

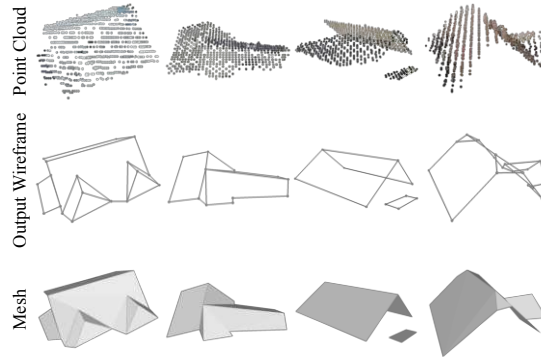
in Table 4, our framework implements query scaling only across the first three layers.

Effect of Geometric Prior vs. Architecture. To verify that the performance gains stem from our geometric prior rather than the Transformer architecture, we conduct a comparative analysis by substituting our wire selection module. As shown in Table 5, replacing our module (Exp.#2) with that of Point2Roof [13] (Exp.#1) results in only marginal performance shifts. In contrast, the introduction of the Delaunay-derived prior via prior-based query scaling (Exp.#3) yields significant improvements. This confirms that the primary contribution lies in the geometric guidance of the Delaunay Canopy rather than mere architectural modifications.

Extensions and Limitations. To further demonstrate the structural fidelity of our reconstructed wireframes, we convert them into a mesh format (Fig. 9). While the sparsity and noise of airborne LiDAR data inherently make precise comprehension of building shape challenging, our Delaunay graph scoring method extracts the underlying surface and curvature of the actual building. This capability ensures the creation of a geometrically consistent output form. However, our framework can encounter challenges in cases of data absence, geometric ambiguity, or topologically closed structures. A comprehensive analysis and visualization of these failure cases are provided in the supplementary.

Table 5: Analysis of geometric prior vs. architecture.

Exp. #	Wire Selection Module	Prior-based Query Scaling	Distance		Corner			Wire		
			WED ↓	ACO ↓	CP ↑	CR ↑	F1 ↑	EP ↑	ER ↑	F1 ↑
1	Point2Roof	-	0.248	0.225	92.4	80.5	86.0	85.8	72.8	78.8
2	Ours	-	0.239	0.210	93.1	81.1	86.7	85.6	73.7	79.2
3	Ours	✓	0.232	0.194	94.5	82.8	88.3	87.1	74.0	80.0

**Fig. 9:** The resulting mesh, reconstructed from the wireframe output generated by the Delaunay Canopy. The wireframe demonstrably encapsulates sufficient geometric information, allowing the resultant mesh to exhibit a highly faithful form.

5 Conclusion

In this work, we present Delaunay Canopy, a 3D geometry-driven framework that redefines building wireframe reconstruction from airborne LiDAR point clouds. By introducing the Delaunay Graph as a structural prior, our method transforms the raw, irregular point cloud into a coherent geometric manifold through Delaunay Graph Scoring, yielding rich curvature-aware cues that guide precise corner and wire inference. This principled design not only reduces the ambiguity of sparse 3D search but also overcomes the distortions inherent in 2D projections, enabling the model to reason directly within the true 3D space. Extensive evaluations on the Building3D Tallinn city and Building3D entry-level datasets reveal that Delaunay Canopy achieves striking improvements in structural fidelity and prediction stability, marking a decisive step toward topology-centric, geometry-grounded urban reconstruction.

Acknowledgments

This work was supported in part by the IITP RS-2024-00457882 (AI Research Hub Project), IITP 2020-II201361, NRF RS-2024-00345806, and NRF RS-2023-002620, while the authors are affiliated with the Department of Artificial Intelligence (C.K., S.J.H.) and the Department of Computer Science (D.K.).

References

1. Baruch, G., Chen, Z., Dehghan, A., Dimry, T., Feigin, Y., Fu, P., Gebauer, T., Joffe, B., Kurz, D., Schwartz, A., Shulman, E.: ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In: *NeurIPS Datasets and Benchmarks Track* (2021)
2. Bauchet, J.P., Lafarge, F.: Kinetic shape reconstruction. *ACM Transactions on Graphics (TOG)* **39**(5), 1–14 (2020)
3. Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., Taubin, G.: The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics* (2002)
4. Buyuksalih, I., Bayburt, S., Buyuksalih, G., Baskaraca, A., Karim, H., Rahman, A.A.: 3d modelling and visualization based on the unity game engine—advantages and challenges. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **4**, 161–166 (2017)
5. Dou, M., Khamis, S., Degtyarev, Y., Davidson, P., Fanello, S.R., Kowdle, A., Escolano, S.O., Rhemann, C., Kim, D., Taylor, J., et al.: Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (ToG)* **35**(4), 1–13 (2016)
6. Forberg, A., Mayer, H.: Generalization of 3d building data based on scale-spaces. *INTERNATIONAL ARCHIVES OF PHOTOGRAMMETRY REMOTE SENSING AND SPATIAL INFORMATION SCIENCES* **34**(4), 225–230 (2002)
7. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: *CVPR* (2012)
8. Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K., Pollefeys, M.: Semantic3d. net: A new large-scale point cloud classification benchmark. *arXiv preprint arXiv:1704.03847* (2017)
9. Huang, J., Stoter, J., Peters, R., Nan, L.: City3d: Large-scale building reconstruction from airborne lidar point clouds. *Remote Sensing* **14**(9), 2254 (2022)
10. Huang, S., Wang, R., Guo, B., Yang, H.: Pbwr: Parametric-building-wireframe reconstruction from aerial lidar point clouds. In: *CVPR* (2024)
11. Huang, X., Cheng, X., Geng, Q., Cao, B., Zhou, D., Wang, P., Lin, Y., Yang, R.: The apolloscape dataset for autonomous driving. In: *CVPR* (2018)
12. Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D., Panozzo, D.: Abc: A big cad model dataset for geometric deep learning. In: *CVPR* (2019)
13. Li, L., Song, N., Sun, F., Liu, X., Wang, R., Yao, J., Cao, S.: Point2roof: End-to-end 3d building roof modeling from airborne lidar point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* (2022)
14. Li, W., Yang, H., Hu, Z., Zheng, J., Xia, G.S., He, C.: 3d building reconstruction from monocular remote sensing images with multi-level supervisions. In: *CVPR* (2024)
15. Lin, H., Zheng, X., Li, L., Chao, F., Wang, S., Wang, Y., Tian, Y., Ji, R.: Meta architecture for point cloud analysis. In: *CVPR* (2023)
16. Liu, Y., D’Aronco, S., Schindler, K., Wegner, J.D.: Pc2wf: 3d wireframe reconstruction from raw point clouds. *arXiv preprint arXiv:2103.02766* (2021)
17. Liu, Y., Obukhov, A., Wegner, J.D., Schindler, K.: Point2building: Reconstructing buildings from airborne lidar point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* (2024)

18. Liu, Y., Zhu, L., Ye, H., Huang, S., Gao, X., Zheng, X., Shen, S.: Bwformer: Building wireframe reconstruction from airborne lidar point cloud with transformer. In: CVPR (2025)
19. Luo, Y., Mi, Z., Tao, W.: Deepdt: Learning geometry from delaunay triangulation for surface reconstruction. In: AAAI (2021)
20. Ma, X., Qin, C., You, H., Ran, H., Fu, Y.: Rethinking network design and local geometry in point cloud: A simple residual mlp framework. arXiv preprint arXiv:2202.07123 (2022)
21. Mahmud, J., Price, T., Bapat, A., Frahm, J.M.: Boundary-aware 3d building reconstruction from a single overhead image. In: CVPR (2020)
22. Mao, J., Wang, X., Li, H.: Interpolated convolutional networks for 3d point cloud understanding. In: CVPR (2019)
23. Mateo, C.M., Corrales, J.A., Mezouar, Y.: Hierarchical, dense and dynamic 3d reconstruction based on vdb data structure for robotic manipulation tasks. *Frontiers in Robotics and AI* **7**, 600387 (2021)
24. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: 2011 10th IEEE international symposium on mixed and augmented reality. pp. 127–136. Ieee (2011)
25. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR (2017)
26. Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., Ghanem, B.: Pointnext: Revisiting pointnet++ with improved training and scaling strategies (2022)
27. Rakotosaona, M.J., Guerrero, P., Aigerman, N., Mitra, N.J., Ovsjanikov, M.: Learning delaunay surface elements for mesh reconstruction. In: CVPR (2021)
28. Ramakrishnan, S.K., Gokaslan, A., Wijmans, E., Maksymets, O., Clegg, A., Turner, J.M., Undersander, E., Galuba, W., Westbury, A., Chang, A.X., Savva, M., Zhao, Y., Batra, D.: Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In: NeurIPS Datasets and Benchmarks Track (2021)
29. Sarlin, P.E., Dusmanu, M., Schönberger, J.L., Speciale, P., Gruber, L., Larsson, V., Miksik, O., Pollefeys, M.: Lamar: Benchmarking localization and mapping for augmented reality. In: ECCV (2022)
30. Selvaraju, P., Nabail, M., Loizou, M., Maslioukova, M., Averkiou, M., Andreou, A., Chaudhuri, S., Kalogerakis, E.: Buildingnet: Learning to label 3d buildings. In: ICCV (2021)
31. Son, S., Gadelha, M., Zhou, Y., Xu, Z., Lin, M., Zhou, Y.: DMesh: A differentiable mesh representation (2024)
32. Wang, R., Huang, S., Yang, H.: Building3d: A urban-scale dataset and benchmarks for learning roof structures from point clouds. In: CVPR (2023)
33. Wang, S., Bai, M., Mattyus, G., Chu, H., Luo, W., Yang, B., Liang, J., Cheverie, J., Fidler, S., Urtasun, R.: Torontocity: Seeing the world with a million eyes. arXiv preprint arXiv:1612.00423 (2016)
34. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)* **38**(5), 1–12 (2019)
35. Wu, W., Qi, Z., Fuxin, L.: Pointconv: Deep convolutional networks on 3d point clouds. In: CVPR (2019)
36. Wu, X., Jiang, L., Wang, P.S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., Zhao, H.: Point transformer v3: Simpler faster stronger. In: CVPR (2024)

37. Wu, X., Lao, Y., Jiang, L., Liu, X., Zhao, H.: Point transformer v2: Grouped vector attention and partition-based pooling (2022)
38. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: CVPR (2015)
39. Xiang, T., Zhang, C., Song, Y., Yu, J., Cai, W.: Walk in the cloud: Learning curves for point clouds shape analysis. In: ICCV (2021)
40. Zhang, C., Yuan, G., Tao, W.: Dmnet: Delaunay meshing network for 3d shape representation. In: ICCV (2023)
41. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: ICCV (2021)
42. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: ICLR (2021)