

# Loops2Roofs: Diffusion-based 3D Roof Generation using a Loop Representation

JIANWEI GUO, School of Artificial Intelligence, Beijing Normal University, China

PU LI, MAIS, Institute of Automation, CAS and School of Artificial Intelligence, UCAS, China

QI ZENG, MAIS, Institute of Automation, CAS and School of Artificial Intelligence, UCAS, China

WENHAO ZHANG, MAIS, Institute of Automation, CAS, China

PENGXU LI, MAIS, Institute of Automation, CAS and School of Artificial Intelligence, UCAS, China

LIANGLIANG NAN, Delft University of Technology, Netherlands

BEDRICH BENES, Purdue University, USA

DONGMING YAN, MAIS, Institute of Automation, CAS and School of Artificial Intelligence, UCAS, China

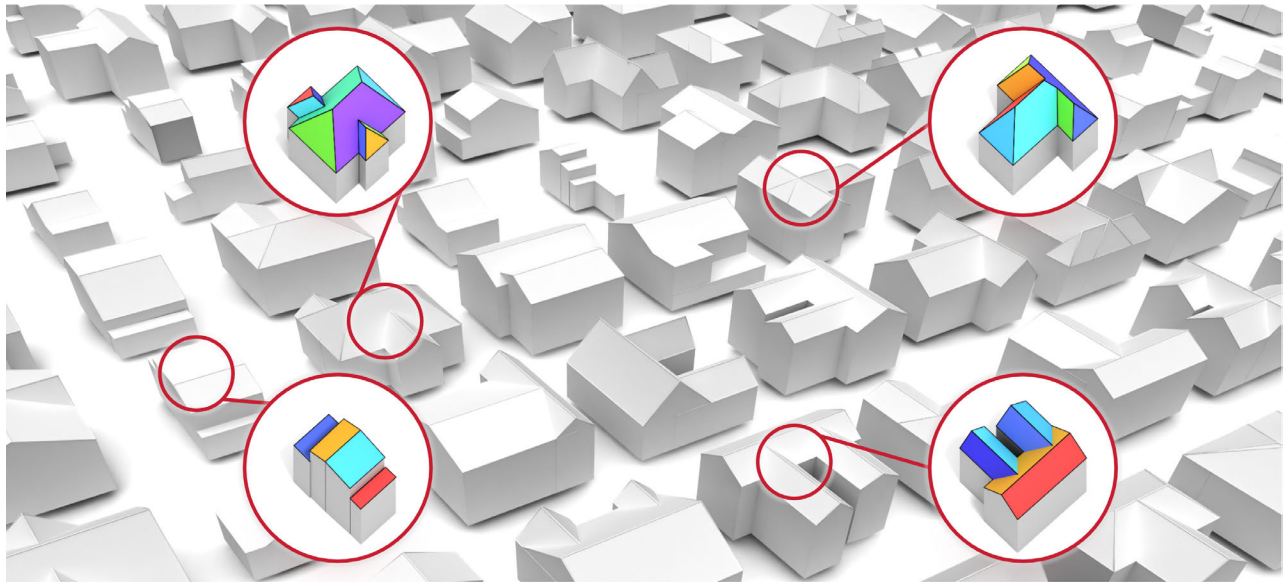


Fig. 1. We introduce *Loops2Roofs*, a novel and versatile generative approach for creating structured 3D roof models. At its core, *Loops2Roofs* employs a flexible loop-based representation that captures intricate roof geometries and forms the foundation for diffusion-based generative modeling.

We thank the reviewers for their constructive comments. This work was supported in part by the National Natural Science Foundation of China (62572059, U22B2034), the Key Project of the National Natural Science Foundation of China (12494550 and 12494553), Guangdong Basic and Applied Basic Research Foundation (2023B1515120026), and the Fundamental Research Funds for the Central Universities (2253200009). This work was supported in part by the U.S. National Science Foundation under awards No. 2412928 and 2417510. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation. This work is based upon efforts supported by the USDA-NIFA grants 2024-67021-42879 and 2023-08563. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government or NRCS. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes, notwithstanding any copyright annotation therein.

Authors' Contact Information: Jianwei Guo, School of Artificial Intelligence, Beijing Normal University, China; e-mail: gjianwei.000@gmail.com; Pu Li, MAIS, Institute of Automation, CAS and School of Artificial Intelligence, UCAS, China; e-mail: 1774136304@qq.com; Qi Zeng, MAIS, Institute of Automation, CAS and School of Artificial Intelligence, UCAS, China; e-mail: zengqi23@mails.ucas.ac.cn; Wenhao Zhang, MAIS, Institute of Automation, CAS, China; e-mail: wenhao.zhang@ia.ac.cn; Pengxu Li, MAIS, Institute of Automation, CAS and School of Artificial Intelligence, UCAS, China;

Effectively representing and generating 3D roof structures remains a challenging task in urban modeling. This article introduces a loop-based roof representation (LFR) and *Loops2Roofs* algorithm, a novel diffusion-based generative approach that directly outputs 3D polygonal roof meshes using the LFR. Our key novelty lies in representing the roofs as a set of polygonal faces bounded by loops. This flexible representation improves generalization to

e-mail: lipengxu24@mails.ucas.ac.cn; Liangliang Nan, Delft University of Technology, Netherlands; e-mail: liangliang.nan@gmail.com; Bedrich Benes, Purdue University, USA; e-mail: bbenes@purdue.edu; Dongming Yan (corresponding author), MAIS, Institute of Automation, CAS and School of Artificial Intelligence, UCAS, China; e-mail: yangdongming@gmail.com.



This work is licensed under a Creative Commons Attribution 4.0 International License.

© 2026 Copyright held by the owner/author(s).  
ACM 0730-0301/2026/10-ART170  
<https://doi.org/10.1145/3807955>

complex roof structures across different architectural styles and can be effectively learned and generated by machine learning models. Roof generation proceeds in two stages. First, a Transformer-based diffusion model directly denoises the 3D coordinates of roof face vertices. This model is conditioned on structural priors, specifically the number of roof faces and vertices per face, which are pre-generated using an auto-regressive model. Second, a neural stitching module enforces roof topology and recovers the geometric incident relationships between edges of different 3D loops. Optionally, our algorithm can incorporate 2D building footprints as input for image-conditioned roof generation. The final output is a compact, structured roof mesh encoded in the LFR format. We demonstrate the effectiveness and efficiency of our approach by generating diverse, realistic roof models, ranging from synthetic to real-world buildings. Experiments show that *Loops2Roofs* significantly advances structured roof generation, outperforming existing methods on three benchmarks.

CCS Concepts: • **Computing methodologies** → *Mesh geometry models*;

Additional Key Words and Phrases: Roof Representation, 3D Generative Modeling, Urban Generation, Diffusion Models

#### ACM Reference Format:

Jianwei Guo, Pu Li, Qi Zeng, Wenhao Zhang, Pengxu Li, Liangliang Nan, Bedrich Benes, and Dongming Yan. 2026. *Loops2Roofs: Diffusion-based 3D Roof Generation using a Loop Representation*. *ACM Trans. Graph.* 45, 5, Article 170 (October 2026), 16 pages. <https://doi.org/10.1145/3807955>

## 1 Introduction

Building roofs exhibit intricate structural details and play a crucial role in urban modeling. Their shapes vary depending on architectural style, era of construction, climate, culture, and other contextual factors. Automatic generation of building shapes is a fundamental challenge in urban modeling, which benefits various applications, such as urban planning, environmental simulation, architectural engineering and construction, and metaverse [Biljecki et al. 2015].

Despite extensive research on urban reconstruction techniques for creating polygonal building or roof meshes from point clouds or aerial images [Wang et al. 2018], automatic roof generation or synthesis is still under-explored. Manual roof modeling with commercial software (e.g., 3ds Max, SketchUp) remains prevalent in industrial workflows, yielding high visual quality. However, such interactive tools are labor-intensive and time-consuming. Procedural urban modeling systems [Müller et al. 2006; Schwarz and Müller 2015] are popular for roof synthesis, but they require in-depth knowledge of the intricate rules and parameters to achieve and fine-tune the user intent. Moreover, they often fail to capture the full range of real-world roof configurations, and creating ad hoc rules is tedious. To achieve fully automatic roof generation, straight skeleton-based approaches [Aichholzer and Aurenhammer 1996; Aichholzer et al. 1996; Biedl et al. 2015; Kelly and Wonka 2011] model the roof structure by assuming that the roof outline determines its topology. Although these approaches are suitable for generating roofs, the results are limited to hip roofs and, with some postprocessing, gable roofs. Furthermore, relying solely on roof outlines introduced ambiguity, as roofs with the same outline may have different topologies [Ren et al. 2021].

Deep generative models have emerged as a promising avenue for generating diverse architectural structures, such as urban layouts [He and Aliaga 2023], buildings [Chang et al. 2021; Kelly et al. 2018], and floorplans [Nauata et al. 2020, 2021; Shabani et al. 2023].

Generating roofs directly using neural networks is challenging due to their structural complexity, e.g., diverse roof types, and region-specific shape variations. Ren et al. [2021] propose a generative model to synthesize new roof meshes, which combines a transformer for roof outline generation with a graph neural network for roof face adjacency prediction. This approach only generates a roof graph structure, and a pre-defined roof optimization step obtains the planar 3D roof. The mixed topological and geometric optimization complicates the generation process. Roof-GAN [Qian et al. 2021] introduced a generative adversarial network that produces a structured roof model as a graph by generating a set of roof primitives and their relationships. However, graph structure, based on rectangular roof primitives, lacks representational capacity. Specifically, it supports limited roof types (i.e., horizontal and vertical gable/hip roofs) and is difficult to generalize to other datasets.

We introduce *Loops2Roofs*, a novel polygon-based diffusion framework for synthesizing realistic and reusable roof models. Our approach is motivated by the performance of diffusion models in many digital content generation tasks [Yang et al. 2023]. Our method's core is a flexible novel **loop-based roof representation (LFR)** that stores a roof shape as a set of 3D polygonal loops, each of which encodes a roof face primitive. This simple yet flexible representation offers several key advantages compared to previous graph-conditioned generative networks. First, it can capture arbitrarily complex roof shapes with high detail while efficiently controlling complexity. Second, this sequential representation aligns well with current advanced generative models (e.g., the diffusion model with a Transformer network).

*Loops2Roofs* adopts a denoising diffusion process that iteratively adds Gaussian noise to roof samples and learns to reverse this process to recover clean roof loop distributions. We employ a Transformer-based denoising neural network to sequentially denoise the corrupted roof features to solve the loop-constrained generation problem. However, the direct generation/regression of 3D coordinates cannot guarantee precise geometric incident relationships (e.g., edge colinearity, or vertex sharing) between adjacent loops. To address this issue, instead of using an optimization-based interactive snapping algorithm [Arikan et al. 2013], we introduce a neural loop stitching network based on the encoder-decoder architecture to predict whether a pair of edges in any two loops is geometrically correlated. This technique ensures topologically correct roof generation by avoiding gaps or overlaps between roof faces.

Experiments demonstrate the strong performance of our approach in unconditional roof layout generation. Besides, we show the application of *Loops2Roofs* on controllable roof generation by encoding the latent building footprint image as a condition in the denoiser. Our key contributions are: (1) A novel loop-based representation (LFR that encodes 3D roof geometry and local topological relationships, enabling the modeling of complex roof structures and facilitating learning. (2) A novel polygon-based diffusion generative model, *Loops2Roofs*, that directly generates diverse and complex 3D roof structures. (3) An encoder-decoder network architecture, *LoopStitchingNet*, tailored to our LRF, which recovers correct roof topology by predicting adjacency relations and stitching neighboring loops.

## 2 Related Work

Our work relates to roof reconstruction, roof layout generation, and recent methods in 3D generative modeling.

### 2.1 3D Generative Modeling

Deep generative models have shown success in creating high-fidelity 3D content. Many 3D generative models have been proposed to synthesize 3D shapes, such as **Variational Autoencoders (VAEs)** [Gao et al. 2019; Zhang et al. 2022], **Generative Adversarial Networks (GANs)** [Chan et al. 2022; Gao et al. 2022a], and **Diffusion Models (DMs)** [Ho et al. 2020; Nichol et al. 2022; Zhou et al. 2021]. In particular, GANs have been applied to generate building models [Chang et al. 2021; Kelly et al. 2018] and house floorplans [Hu et al. 2020; Nauata et al. 2021] based on specific training data. Our proposed method shares similarities with recent mesh generation approaches that employ auto-regressive models to predict sequences of vertices or triangles directly. PolyGen [Nash et al. 2020] represents a mesh as a set of 3D vertices and polygon faces, and predicts vertices and faces sequentially using Transformer-based architectures. Recent methods tokenize meshes into 1D sequences and leverage auto-regressive transformers for generating triangular meshes, including MeshGPT [Siddiqui et al. 2024], MeshAnything [Chen et al. 2024a], and EdgeRunner [Tang et al. 2025]. MeshXL [hen et al. 2024] represents each coordinate using a learnable embedding and employs an autoregressive model with a predefined ordering strategy for direct mesh generation. This discrete autoregressive generation is a viable alternative to our problem. However, our choice of continuous representation via diffusion is motivated by the unordered nature of roof vertices. Autoregressive methods impose artificial orderings for tokenization, which can propagate errors across long sequences. In contrast, our continuous diffusion approach naturally handles unordered geometric data, reducing this cascading error risk inherent to sequential generation methods.

Recently, diffusion models have emerged as the state-of-the-art of generative models with remarkable results in many applications, such as image synthesis [Dhariwal and Nichol 2021; Ho et al. 2020; Rombach et al. 2022] and video generation [Ho et al. 2022; Xing et al. 2024]. However, unlike 2D images, which can be efficiently represented and processed using discrete pixels, the lack of a unified 3D representation poses a significant challenge for diffusion models. A robust 3D representation is crucial in 3D content generation because there is no consensus on a network architecture that can serve as a reliable and scalable backbone in handling spatial relationships in 3D structure [Po et al. 2024]. Our work focuses on roof modeling and proposes a novel loop-based representation capable of accurately synthesizing complex 3D roof shapes.

Our work also draws inspiration from diffusion models for CAD, which demonstrate a promising capability to generate CAD geometries with [Liu et al. 2024b; Shabani et al. 2023] or without [Alliegro et al. 2023; Xu et al. 2024] topology as input conditioning. Borrowing architecture from VQ-Diffusion [Gu et al. 2022], which combines VQ-VAE [Van Den Oord et al. 2017] and conditional **denoising diffusion probabilistic models (DDPM)**, VQ-CAD [Wang et al. 2024] integrates discrete diffusion models for CAD model generation. Our work aims at tackling the challenge of un-

conditionally using the diffusion model to generate well-structured roof geometries.

### 2.2 Building and Roof Reconstruction

Urban reconstruction from real-world sensor data, such as images or 3D point clouds, is an active research area. Existing algorithms to reconstruct city models can be broadly categorized into simplification-based [Gao et al. 2022b; Garland and Heckbert 1997; Salinas et al. 2015; Verdie et al. 2015], model-driven [Guo et al. 2024; Lin et al. 2013; Nan et al. 2010; Zhou and Neumann 2010], primitive-based [Arikan et al. 2013; Bauchet and Lafarge 2020; Chauve et al. 2010; Guo et al. 2024; Monszpart et al. 2015; Nan and Wonka 2017] and Gaussian-based [Liu et al. 2024a; Peng et al. 2025; Yuan et al. 2025] approaches. Earlier state-of-the-art reports on these topics have been presented in [Musialski et al. 2013; Wang et al. 2018].

Roof reconstruction is a key aspect of urban modeling, and it often relies on traditional optimization algorithms to recover roof structures by detecting geometric primitives (planes, lines, etc). For example, [Vanegas et al. 2010] represents the 3D building with axis-aligned non-uniform polycubes but cannot represent sloped roofs. Verma et al. [2006] applied the roof topology graph to represent the relationships between the planar primitives, and then the parametric roof shapes are recognized by subgraph matching. Zhang et al. [2021] create complex models by using a large set of parameterized primitives, including sloped and flat roofs, while Edelsbrunner et al. [2016] introduce a solid roof primitive that can create complex roofs. Kelly et al. [2017] formulate a binary integer program to produce structured models by leveraging data from multiple sources. **Markov Chain Monte Carlo (MCMC)** and **Support Vector Machine (SVM)** have also been used for roof reconstruction [Henn et al. 2013; Huang et al. 2013]. The release of building and roof datasets (e.g., City3D [Huang et al. 2022], Building3D [Wang et al. 2023], BuildingNet [Selvaraju et al. 2021]) enabled deep learning methods that directly output 3D roof structures from point clouds [Chen et al. 2022a, 2024b; Li et al. 2022; Zeng et al. 2018] or images [Alidoost et al. 2020; Zhang and Aliaga 2022]. However, the above-mentioned methods cannot be used for the generation of novel objects, e.g., for movies or video games. Contrary to these approaches, which aim at accurating roof reconstruction, our approach focuses on generating plausible roof structures from scratch with broad variability.

### 2.3 Roof Synthesis

One of the key issues in creating city models of Level of Detail 2 (LoD-2) (i.e., models that distinguish roofs from buildings and include roof details) is the challenge of generating roof structures precisely. A straightforward approach to modeling roofs involves combining simple roof primitives (e.g., gabled, hipped, or flat roofs) with often convex footprint shapes to generate 3D roofs [Finkenzeller 2008; Müller et al. 2006; Sugihara and Hayashi 2008]. While effective for regular building shapes, these methods struggle to model complex and irregular roof structures accurately. The automatic generation of roofs of buildings based on straight skeletons [Aichholzer and Aurenhammer 1996] of their footprints has received widespread attention in large-scale urban modeling. Early works use the unweighted straight skeleton to model roofs [Laycock and Day 2003], where the resulting skeleton corresponds to

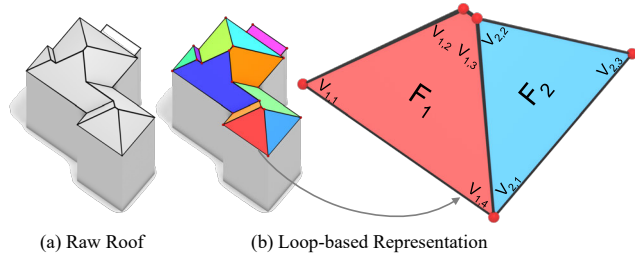


Fig. 2. An illustration of our proposed loop-based representation for roof structures. Given a roof (a), we represent each roof face (color-coded) as a loop  $F_i$  containing a sequence of 3D vertices  $\{V_{i,j}\}$ , as shown in (b).

the edges of a roof on the building. The weighted skeleton [Biedl et al. 2015; Eppstein and Erickson 1998] brings greater flexibility to facilitate the roof faces modeling at different angles specified by users. Kelly and Wonka [Kelly and Wonka 2011] propose an extension to the weighted straight skeleton for the interactive modeling of complete buildings. However, roofs created by straight skeletons are limited to hip roofs and, with some postprocessing, gable roofs. Their ridges tend to be parallel to the long edges of the footprint of the building. These models do not capture ridges perpendicular to the long edges of the footprint [Held and Palfrader 2017].

To overcome the limitations of straight skeleton methods, Ren et al. [2021] propose a new primal-dual roof graph representation and introduce an optimization-based method to reconstruct a 3D planar roof from a primal or dual roof graph. Further, they train neural networks to generate a discrete roof graph for new roof synthesis and then run the roof optimization to enforce the continuous geometric constraints. Drawing inspiration from the floorplan generation backbone House-GAN [Nauata et al. 2020], Qian et al. [2021] propose Roof-GAN, a novel generative adversarial network for generating structured geometry of residential roof structures. The generator produces a structured roof model as a graph by generating primitive geometries at the nodes and incident geometric relationships at the edges. However, relying on rectangular roof primitives limited the representation ability of their graph structure, and consequently, Roof-GAN only supports four roof types (i.e., horizontal and vertical gable/hip roofs) and does not generalize to other datasets. Our loop-based representation is more flexible for modeling complex and irregular roofs than the graph-constrained representation.

### 3 Overview

*Loop-based roof representation.* Generating arbitrarily shaped roofs is challenging due to the complexity and irregularity of roof structures. Graph-based representations, commonly used in layout and roof generation, represent primitives as nodes and encode their adjacency relationships in an adjacency matrix. Although these representations allow flexibility by avoiding a fixed set of geometric primitives, precisely capturing topological relationships in a general graph remains challenging.

To solve this, we represent a roof structure as a set of non-intersecting 3D polygonal loops. As shown in Figure 2, we define a roof structure as a set of polygonal loops  $\mathcal{F} = \{F_1, F_2, \dots, F_N\}$ , where each loop  $F_i$  represents a roof face and is described by a

sequence of 3D vertices arranged in a clockwise order:

$$F_i = \{V_{i,1}, V_{i,2}, \dots, V_{i,N_i} \mid V_{i,j} \in \mathbb{R}^3\}. \quad (1)$$

Each loop  $F_i$  encodes roof geometries (via 3D vertex position) and local topological relationships (via polygonal embedding). A post-processing step or a neural network easily enforces the global relationships between adjacent loops. This flexible representation facilitates the modeling of complex roof structures and serves as the foundation for our diffusion-based learning framework.

While our loop-based representation shares conceptual similarities with floorplan models like HouseDiffusion [Shabani et al. 2023], it is specifically designed to address several challenges unique to roof modeling: (1) Structural complexity. Unlike floorplans, where rooms are naturally independent units, roofs form an interconnected topological graph with intricate geometric dependencies. By representing roofs as a 1D chain of 3D vertices, we enable more effective learning of these complex structures by neural networks. (2) Geometric constraints. Floorplans often conform to Manhattan-world assumptions with axis-aligned, orthogonal geometry. In contrast, roofs feature arbitrary angles and slopes, requiring continuous rather than discrete diffusion to handle their geometric variability. (3) Vertex sharing. Roofs frequently contain shared vertices at which multiple loops intersect in the 3D space. Unlike floorplans with distinct room boundaries, roof modeling must explicitly handle vertex coincidences. We address this through a neural mechanism that restores geometric consistency and ensures valid topological connections. These domain-specific adaptations make our LFR well-suited for roof modeling, despite its conceptual resemblance to floorplan representations.

*Loops2Roofs.* We aim at generating a compact and structured 3D roof model in the LFR format through diffusion. Accurately predicting roof geometry requires addressing three key challenges: (1) synthesizing plausible polygonal loops, (2) enforcing geometric consistency, and (3) recovering topological relationships. To tackle these, we propose a two-stage learning approach. First, a generation module is proposed to synthesize polygonal loops that approximate the roof geometry. Then, we introduce a loop assembly network to ensure precise geometric incident relationships among loops.

First, we train a Transformer-based diffusion model, *Loops2Roofs*. In the forward diffusion process, Gaussian noise is added to ground-truth roof models to produce noisy samples (see Figure 3(a)). The denoising process then reconstructs these samples by sequentially predicting the vertex coordinates of the roof loops, guided by the learned distribution (see Figure 3(b)). The denoiser is trained on our constructed loop-based roof dataset based on the publicly available 3D building models (e.g., [Huang et al. 2022; Luo et al. 2022; Qian et al. 2021]). Note that the diffusion model requires the number of polygonal loops  $N$  and the number of vertices  $N_i$  in each face  $F_i$  as input, making the generation process partially conditional. To enable fully automated generation, we introduce an auto-regressive Transformer decoder to probabilistically predict  $N$  and  $N_i$  (see Section 4.1) as an initial step.

Second, taking the generated polygonal loops as input, we introduce an encoder-decoder network called *LoopStitchingNet*, which recovers the roof topology information by examining the geometric relationship between any pair of edges in the LRF to stitch adjacent loops, as shown in Figure 3(c). Finally, the predicted loop-based

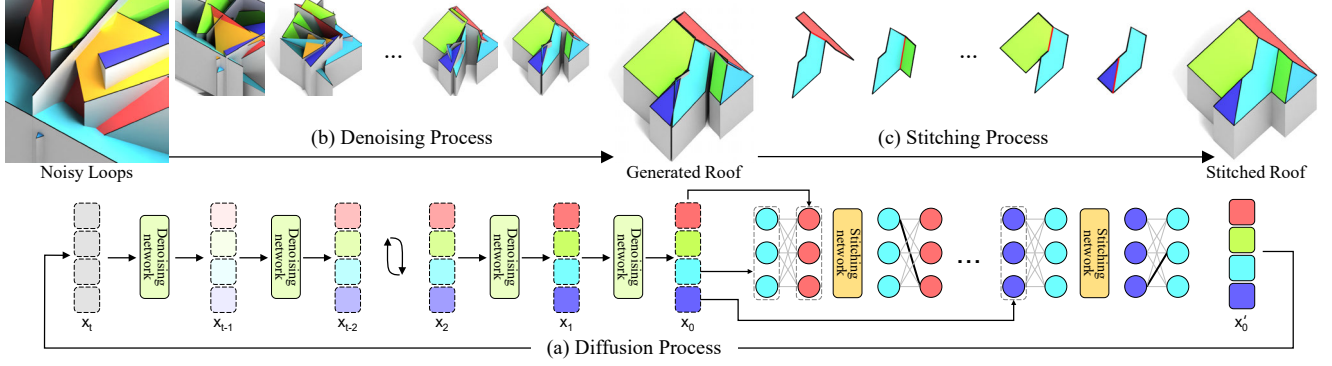


Fig. 3. Overview: *Loops2Roofs* smoothly perturbs data by adding noise in the diffusion process (a), then reverses this process in the denoising stage to generate new data from input noise (b). We then introduce *LoopStitchingNet* to generate a compact roof by stitching adjacent loops (c). The top row visualizes the denoising and stitching processes, and the bottom row illustrates the corresponding data-level pipeline. Each rounded rectangle represents the coordinates of a loop, where gray indicates pure noise, and the opacity of colors indicates the proximity to ground truth loop coordinates - lower opacity suggests closer alignment to the true values. Dashed outlines indicate unstitched states where loop boundaries remain independent, while solid outlines represent loops after stitching. Colored circles represent edges within their corresponding colored loops, with black lines connecting edges that share stitching relationships and gray lines indicating unconnected edges.

roof structure is extruded along the direction of gravity to create volumetric primitives. These primitives are merged using a Boolean union operation to assemble a CAD-level building model.

## 4 Methods

A diffusion model [Ho et al. 2020] consists of two key stages: a forward process (or diffusion process) that perturbs the original data distribution by gradually adding noise and a reverse process (or denoising process) that recovers the target distribution through iterative denoising. *Loops2Roofs* extends diffusion principles to the domain of roof structure generation. We begin by unconditionally generating the number of loops and the number of vertices per loop. Following this, we detail the diffusion-based roof generation and the subsequent loop stitching process.

### 4.1 Loop and Vertex Index Generation

To enable accurate roof generation within our diffusion framework, we represent a 3D roof structure using loop and vertex index sequences. For pure unconditional generation, we introduce an auto-regressive Transformer model that learns the dependencies within these index sequences. It predicts subsequent loop indices based on the previously generated sequence, enabling the incremental construction of full loop index sequences for an entire roof.

Our auto-regressive model is based on a Transformer decoder architecture [Vaswani et al. 2017]. The model input is a sequence of loop indices, beginning with the <BOS> token and terminating with the <EOS> token:

$$\mathcal{Z} = \{\text{<BOS>}, z_1, z_2, \dots, z_K, \text{<EOS>}\}, \quad (2)$$

where  $K$  denotes the number of vertices in a single roof. Each token  $z_i$ , encoded as a 12-dimensional one-hot vector, indicates the loop index of the  $i$ th vertex. The probability distribution over the loop index sequence is defined as follows:

$$P_{\theta}(\mathcal{Z}) = \prod_{k=1}^{|\mathcal{Z}|} p_{\theta}(z_k | z_{<k}), \quad (3)$$

with  $\theta$  denoting the network parameters and  $z_{<k}$  denoting the subsequence of all elements preceding the  $k$ -th element. The model processes each token through input and positional embeddings, followed by Transformer decoder layers. The model outputs a categorical distribution over possible loop index values, with the network trained via maximum log-likelihood estimation using teacher forcing [Williams and Zipser 1989].

A causal attention mask is applied during training to prevent information leakage from future tokens. The model is trained to predict the next token through shift operations and a cross-entropy loss.

In inference, the model generates sequences auto-regressively. Beginning with the start token, <BOS>, the model iteratively predicts the probability distribution of the next loop index. A greedy decoding strategy is employed to select the most probable next token at each step:

$$z_k^* = \operatorname{argmax}_{x_k} P_{\theta}(z_k | z_{<k}). \quad (4)$$

The generation terminates when the end token, <EOS>, is generated or a predefined maximum sequence length is reached.

### 4.2 Diffusion-based Roof Generation

At the training stage, *Loops2Roofs* corrupts a set of training roof examples by gradually adding different levels of Gaussian noise and then learns a denoising process that maps samples of random noise to clean roofs. At the inference stage, *Loops2Roofs* directly generates novel roofs by iteratively transforming a noisy roof into a clean LFR.

*Diffusion process.* The forward step of the diffusion process is a predefined discrete-time Markov chain in the data space spanning all possible roof structures. Every roof structure consists of  $N$  loops that contain the 3D coordinates of the vertices. During training, given a clean roof configuration  $x_0$  from underlying distribution  $q(x_0)$ , we follow Equation (5), obtaining a series of intermediate roof variables  $\{x_1, \dots, x_t\}$  with the same dimensionality as  $x_0$ , according to a predefined standard cosine noise variance schedule  $\beta_t \in (0, 1)$

[Nichol and Dhariwal 2021]:

$$x_t(x_0, \epsilon) = \sqrt{\alpha_t}x_0 + \sqrt{(1 - \alpha_t)}\epsilon, \quad (5)$$

where  $\alpha_t = 1 - \beta_t$ ,  $\tilde{\alpha}_t = \prod_{s=1}^t \alpha_s$ ,  $\epsilon \sim \mathcal{N}(0, I)$ . The data sample gradually loses its distinguishable features as the step  $t$  becomes larger and eventually is equivalent to an isotropic Gaussian distribution after  $T \rightarrow \infty$  time steps. For example, the coordinate vectors of each vertex will be scrambled, and duplicated vertices between incident roof faces will have different features and no longer exhibit the correct topological relations.

**Feature embedding.** To facilitate the generation process, we encode conditional information, such as time steps, loop indices, and vertex indices, into feature vectors. These embeddings are injected into the denoising network to guide the understanding of the loop-based roof representation.

Given a roof structure sample  $\mathcal{F}_t$  at time  $t$ , every 3D vertex position  $V_{i,j}^t$  becomes a node in the Transformer architecture with a  $D(= 512)$  dimensional embedding vector  $\hat{V}_{i,j}^t$ . We initialize the embedding as:

$$\hat{V}_{i,j}^t \leftarrow \text{Linear} \left( \left[ V_{i,j}^t, \mathbf{1}(i), \mathbf{1}(j), t \right] \right), \quad (6)$$

where  $\mathbf{1}(\cdot)$  denotes a 12D one-hot vector for a loop index  $i$  or a vertex index  $j$ , and  $t$  is a scalar indicating time step. A linear layer is applied to convert the embedding into a  $D(= 512)$  dimensional vector.

**Generative process.** The generative process is parameterized as a Markov chain of learnable reverse Gaussian transitions to refine noisy inputs into coherent roof structures. Given a noisy roof from a standard multivariate Gaussian distribution  $x_T \sim \mathcal{N}(0, I)$  as the initial state, it corrects  $x_t$  at each time step by using a learned Gaussian transition  $p_\theta(x_{t-1} | x_t)$  which is parameterized by a learnable network  $\theta$ . By repeating this reverse process until the maximum number of steps  $T$ , we can reach the final state  $x_0$ , the clean roof configuration we aim at obtaining. Specifically, the joint distribution of the generative process  $p_\theta(x_0:T)$  is

$$p_\theta(x_0:T) := p(x_T) \prod_{t=1}^T p_\theta(x_{t-1} | x_t), \quad (7)$$

where the diffusion step at time  $t$  is defined in Equation (8):

$$p_\theta(x_{t-1} | x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \quad (8)$$

where  $\mu_\theta(x_t, t)$  and  $\Sigma_\theta(x_t, t)$  represent the prediction of the mean and variance of the distribution  $q(x_{t-1} | x_t, x_0)$ , respectively. Following the DDPM [Ho et al. 2020] that rather than directly predicting  $\mu_\theta(x_t, t)$ , we synthesize more high-frequent details by estimating the noise  $\epsilon_\theta(x_t, t)$  through a denoising neural network  $\theta$ .

**Denoising network.** Our implementation adopts a Transformer-based neural network [Vaswani et al. 2017] for the generative process, which refines noisy roof representations from time  $t$  to cleaner representations at time  $t - 1$ . Specifically, given the embedding vectors  $\{\hat{V}_{i,j}^t\}$ , we construct a Transformer denoiser with attention blocks to model the noise  $\epsilon_\theta(V_{i,j}, t)$  with time embedding  $t$  added as conditions within every Transformer layer. As is shown in Figure 4, the denoising network is based on  $L$  encoder layers, which contain a self-attention layer and a **feed-forward network (FFN)** with skip

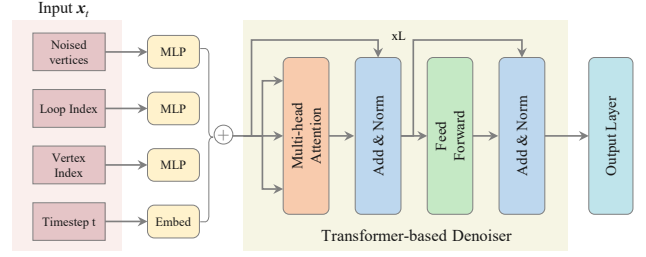


Fig. 4. Illustration of the denoising network architecture, which takes noisy loops  $x_t$  at timestep  $t$  with the conditions of loop index and vertex index as input and embeds them into high-dimensional features. Then we denoise the embedded features using a Transformer-based denoiser with skip connections and attention blocks.

connections, where each encoder layer is interleaved with attention blocks to aggregate the features of different loops, exploiting the inter-corner relationships and capturing the global roof structure context.

**Loss function.** The reverse diffusion process is trained to optimize the denoising network parameters  $\theta$ , which ensures the generation of geometrically consistent and structurally accurate roof structures. Our training objective follows an L2-norm regression loss [Ho et al. 2020], where the estimated noise  $\epsilon_\theta(V_{i,j}, t)$  is compared with the ground-truth noise  $\epsilon$  from the forward process. The loss function is

$$L_{\text{loop}}(\theta) = \mathbb{E}_{x_0, \epsilon, t} \left[ \left\| \epsilon_t - \epsilon_\theta \left( \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon, t \right) \right\|^2 \right] \quad (9)$$

$$= \mathbb{E}_{x_0, \epsilon, t} \left[ \left\| \epsilon_t - \epsilon_\theta(x_t, t) \right\|^2 \right],$$

where  $\epsilon_t$  is the ground-truth Gaussian noise at time step  $t$  in the forward pass and is compared against  $\epsilon_\theta(x_t, t)$ . During sampling, the estimated noise at each time step incrementally refines the data, transforming it from random Gaussian noise into a structured roof representation.

### 4.3 Loop Stitching Network

Ensuring geometric consistency and topological relationships between loops poses significant challenges during denoising. For example, adjacent roof faces typically share identical vertices and may also share joint edges. To ensure consistency, these vertices and edges must align perfectly without gaps and overlaps. Unfortunately, direct regression or generation of 3D coordinates can rarely achieve these relationships. A heuristic-based post-processing approach could merge close vertices/edges using predefined Euclidean distance thresholds. However, this becomes impractical for complex roofs due to its lack of robustness.

To address this problem, we propose a neural approach that learns geometric correlations between coordinates (e.g., covariance, orthogonality, or corner sharing) tailored for our loop-based roof representation. Specifically, we train an encoder-decoder architecture network, called *LoopStitchingNet*, to predict whether two edges are geometrically correlated. The architecture of *LoopStitchingNet* is illustrated in Figure 5, and it consists of two main parts: an Encoder and a Decoder. The Encoder learns the implicit loop features and extracts the high-dimensional hidden state of each loop. The Encoder consists of positional encoding,  $N$  encoder blocks, and

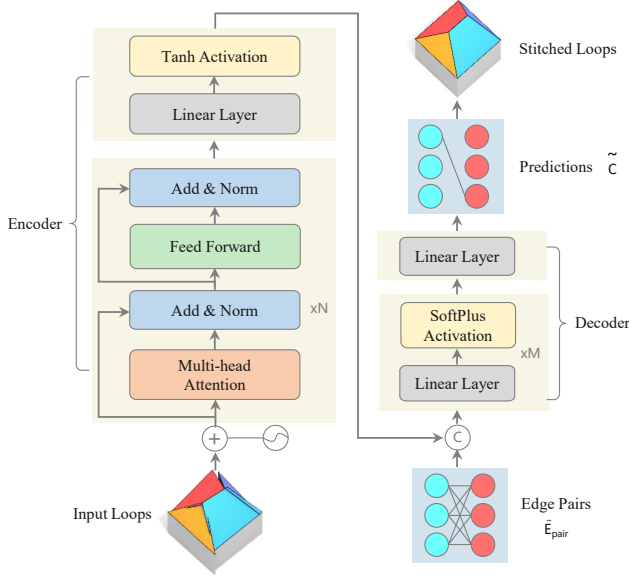


Fig. 5. Illustration of the *LoopStitchingNet* architecture. The generated loops are taken as input and encoded by the Encoder, which is subsequently combined with the edge set information and fed into the Decoder for decoding, and finally outputs the classification prediction (i.e., geometric correlations).

a bottleneck. Each encoder block integrates a multi-head attention mechanism and a fully linked feed-forward network, with both components followed by normalization and residual connections via Add&Norm. The bottleneck contains a linear layer and a Tanh activation function. The Decoder utilizes the implicit features learned by the Encoder and the corresponding edge set information to complete the binary prediction task, i.e., to predict whether two edges are co-edged or not. It contains  $M$  linear layers and a SoftPlus [Nair and Hinton 2010] activation function with an output layer.

**Training dataset.** Since the diffusion model alone cannot fully restore topological relationships between loops during denoising, it is necessary to predict whether collinearity exists between any two edges in a noisy loop-based roof structure. Therefore, to train our *LoopStitchingNet*, we construct a training dataset that meets the above conditions.

From a large set of clean 3D roof examples, we first traverse each loop and construct the ground truth edge relationship  $C \in \mathbb{R}^{M \times M}$  for all edge pairs across loops, where  $M$  denotes the total number of edges.  $C_{i,j} = 1$  if  $e_i$  and  $e_j$  from two loops share a common edge, otherwise  $C_{i,j} = 0$ , where an edge  $e_i$  is represented by its start and end vertices  $e_i = \{[x_i^s, y_i^s, z_i^s], [x_i^e, y_i^e, z_i^e]\}$ . Then, for each roof example  $\mathcal{F}$  containing a set of 3D polygonal loops, we noise each loop by randomly move the vertices (i.e., perturb their location) to simulate the discrete roofs obtained by the above diffusion model, where the noise is randomly sampled from a uniform distribution with noise size of  $[-3, 3]$  to get  $\tilde{\mathcal{F}} = \{\tilde{F}_1, \tilde{F}_2, \dots, \tilde{F}_N\}$ . We then traverse the noisy loops in all roof examples and collect all edges into a set  $\tilde{E} = \{\tilde{e}_1, \tilde{e}_2, \dots, \tilde{e}_M\}$ . Finally, we construct the edge pair data  $\tilde{E}_{pair} \in \mathbb{R}^{M \times M \times 12}$  from the set of edges  $\tilde{E}$ , where  $\tilde{E}_{pair}(i, j) = (\tilde{e}_i, \tilde{e}_j)$ .

The objective of *LoopStitchingNet* is to predict the edge relationship  $\tilde{C}$  that is as close to the true value  $C$  as possible.

**Network training and inference.** With the required inputs  $\tilde{\mathcal{F}}, \tilde{E}_{pair}$  and  $C$  prepared for the training of *LoopStitchingNet*, the Encoder learns the coding features of the noisy loop from  $\tilde{\mathcal{F}}$ . We then concatenate  $\tilde{E}_{pair}$  with the features of *Encoder*( $\tilde{\mathcal{F}}$ ) in the last dimension and feed it into the Decoder to output the prediction result from  $\tilde{C}$ . We use the **Mean Squared Error (MSE)** loss to supervise the training:

$$L_{stitch} = \text{MSE}(\tilde{C}, C). \quad (10)$$

During inference, the output of the diffusion model is used to predict geometric relationships between edge pairs, which enables the correction of topological inconsistencies between loops. Specifically, we first consider each pair of loops: if an edge  $e_k$  in  $loop_i$  and an edge  $e_l$  in  $loop_j$  are the shared edges according to the network prediction result, then we put the edge pair  $(e_k, e_l)$  into a set  $E_{merge}$  to be merged. We then traverse every edge pair in  $E_{merge}$  and verify whether the two edges are in the same direction. If not, one of the edges is flipped if it is reversed concerning its vertices. Finally, we average the vertex coordinates of the current two edges across their associated duplicates and generate a geometrically closed-loop roof structure. The robustness and effectiveness of *LoopStitchingNet* are validated through our ablation study (Section 5.4).

## 5 Experimental Results

This section evaluates the performance of *Loops2Roofs* on synthetic and real roof datasets with diverse architectural styles and geometric variations. Extensive quantitative and qualitative analysis, supported by comparisons to state-of-the-art methods and comprehensive ablation studies, validate the effectiveness of our approach.

### 5.1 Setup

**Implementation details.** We implemented our neural networks in PyTorch and conducted experiments using an NVIDIA A6000 GPU. To ensure robust and dataset-specific training, models were trained independently for each dataset. The AdamW optimizer was utilized with a learning rate of  $5 \times 10^{-4}$ , a weight decay of 0.05, and a batch size of 64. The training process for each dataset converged within approximately 24 hours after  $\sim 1,000$  epochs. For the *LoopStitchingNet* component, we employed the Adam optimizer with default parameters, with a learning rate of 0.001 and a batch size of 256. The model was trained for 300 epochs. The VAE for conditional image input tasks shared identical training parameters with the *LoopStitchingNet*, except for an increased batch size of 512. Our implementation supports up to 12 loops per roof and 12 vertices per loop, sufficient for all datasets used in our experiments. The architecture is scalable and can accommodate larger structures by increasing the positional encoding dimensionality.

**Dataset preparation.** We evaluated the performance of our approach on three diverse and widely used building roof datasets: (1) RoofGAN [Qian et al. 2021], a dataset featuring 502 CAD-style roof geometries annotated with incident relationships; (2) City3D [Huang et al. 2022], a large-scale building reconstruction dataset consisting of 20k real-world buildings models; and (3) LC2WF [Luo

et al. 2022], a collection of 2,539 complex synthetic roofs with ground-truth 3D wireframe models [Ren et al. 2021].

We designated the z-axis as the vertical direction. For both RoofGAN and City3D datasets, which initially contain wall geometries, we first filtered out all wall planes (i.e., faces orthogonal to the xy-plane). City3D includes non-Manhattan-world roof geometries. To address this, we performed principal direction analysis by computing the cumulative edge length along different orientations in the xy-plane and then aligning the dominant direction with the x-axis. For each loop, we select the vertex closest to the origin as the starting (first) vertex, and the remaining vertices are ordered in a clockwise manner. When multiple loops are present, they are ordered according to the distances of their respective starting vertices to the origin.

We implemented eight canonical transformations using the dihedral group  $D_4$  symmetries for all datasets, including identity transformation, rotations of  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$ , along with four reflections (horizontal, vertical, and two diagonal transformations). During training, we normalized all roof coordinates to the range  $[-1, 1]$ . An exception is made for City3D, where we preserved the original aspect ratio in the xy-plane to accommodate its characteristic elongated roof structures. All datasets were randomly split into 90%, 5%, and 5% for training, validation, and testing, respectively, and the same split was used for unconditional and conditional generation.

**Evaluation metrics.** To quantitatively evaluate the quality of the generated roofs, we utilized three widely used metrics, each designed to capture distinct aspects of geometric and topological fidelity.

- *Recursive Minimum Matching Distance (RMMD)*, introduced in RoofGAN [Qian et al. 2021], evaluates the realism and fidelity of generated roof models. RMMD quantifies geometric and topological differences by recursively calculating the minimum matching distances among vertices and faces between pairs of roof models. To ensure consistency (RMMD is sensitive to scale and translation), each model is normalized by aligning its axis-aligned bounding box to a canonical square frame through uniform scaling and translation. The range of RMDD is from zero to infinity. Lower RMMD values indicate greater geometric fidelity.
- *Coverage (COV)* measures the percentage of reference roof geometries covered by the generated ones. Through RMMD’s recursive approach, we calculate the distance between each model in the generated and the reference sets. We then calculate the proportion of ground-truth models successfully matched by the generated roof models. The range of COV is between zero and one, and a higher COV value indicates more diverse and well-represented roof geometries.
- *Fréchet Inception Distance (FID)* [Heusel et al. 2017] assesses the diversity of generated roof models with rasterized images. Each roof model is converted into a surface normal image, where the RGB channels encode the 3D surface normal vectors, effectively capturing the orientation and structure of the model. The background is set to white to isolate the model surface, enabling FID to objectively measure the difference in distribution between generated

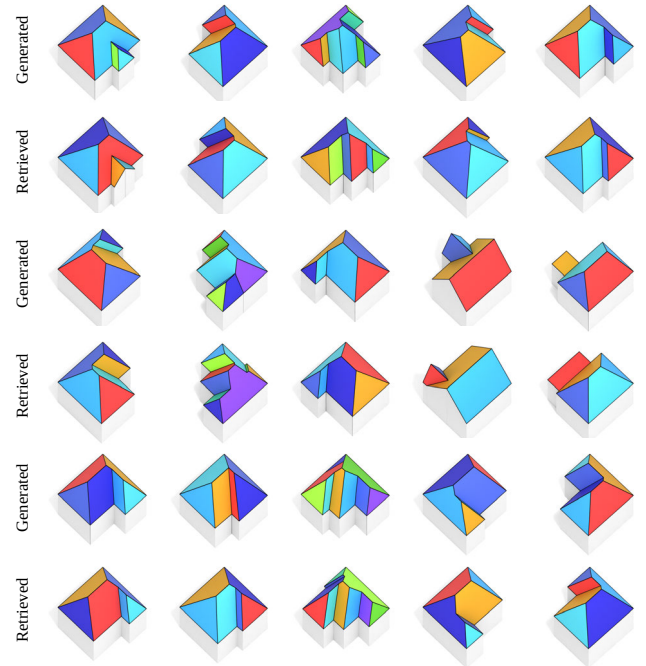


Fig. 6. Unconditionally generated shapes (rows 1, 3, 5) by our proposed method and their corresponding retrieved matches from the training dataset (rows 2, 4, 6).

and real roof models regarding structural integrity and surface orientation. To compute FID scores, we used Blender [Community 2018] to generate top-down normal maps of the roofs.  $FID = 0$  indicates the two sets of images are identical and higher values indicate a poorer generative model.

**User study.** In addition to the quantitative metrics, we conducted a user study to assess the perceptual realism of *Loops2Roofs* compared to other methods. We generated 100 roof samples using each method and used a **two-alternative forced-choice (2AFC)** study to sort the samples. Specifically, we showed each participant pairs of randomly selected images from different methods and asked them to choose the one perceived as more realistic. Each selected image had its perceptual score incremented (with a score of +1), and the deselected image decreased (with a score of -1). If no images were selected, no score was assigned. The average scores were calculated to summarize user preferences. The study involved 31 participants in total, with an average age of 28. Among them, three have a background in architecture, four have experience in urban reconstruction or modeling, and the others are novices with no prior experience.

## 5.2 Evaluation

**Unconditional roof generation.** We first evaluated *Loops2Roofs* on the unconditional roof generation task, which aims at synthesizing roof models from pure noise. The novelty of *Loops2Roofs* is highlighted through a comparison of generated shapes and their closest counterparts from the training set, identified using the **Recursive Minimum Matching Distance (RMMD)** metric. Figure 6

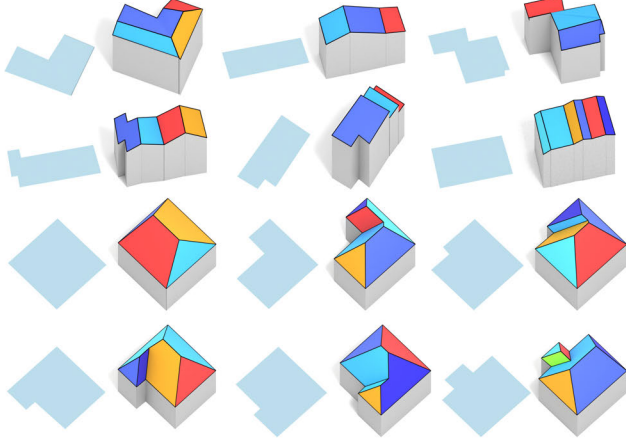


Fig. 7. Examples of 3D roof structures generated by our model conditioned on 2D building footprints. The light blue planar polygons represent roof silhouettes. Corresponding 3D roof reconstructions are shown adjacent to each footprint on the right.

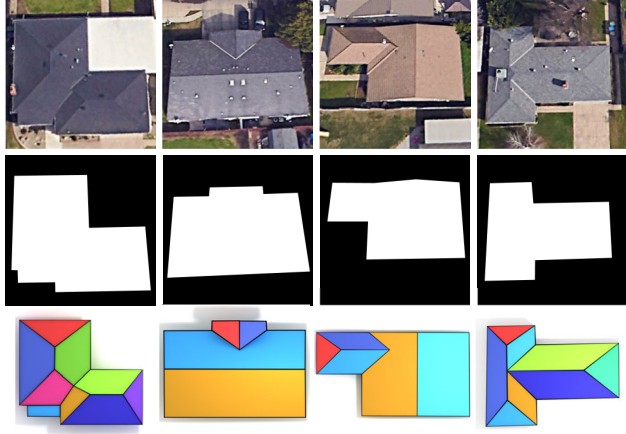


Fig. 8. Illustration of 3D roof generation (bottom row), conditioned on 2D building footprints (middle row) extracted from real-world aerial images (top row).

presents our generated results (rows 1, 3, 5) alongside their nearest neighbors from the training dataset (rows 2, 4, 6). These results show that *Loops2Roofs* generates realistic and visually accurate roof models while ensuring distinct topology and geometrical variations compared to the training samples.

*Image-conditioned roof generation.* To demonstrate the versatility of our approach, we further presented a practical application: controllable roof generation conditioned on input images. By leveraging a rooftop image  $I$ , we incorporated loop constraints for image-guided roof reconstruction. We trained a VAE to extract image embeddings, which are subsequently encoded as conditions in the Transformer. The image feature embedding is obtained by

$$\hat{V}_{i,j}^t \leftarrow \text{Linear} \left( \left[ V_{i,j}^t, \mathbf{1}(i), \mathbf{1}(j), t, I \right] \right). \quad (11)$$

To determine the number of polygons and vertices, we extract image features  $I$  using ResNet14 [He et al. 2016] and inject it into the

Table 1. Quantitative and Qualitative (Last Column) Comparison on The RoofGAN Dataset

| Method    | FID↓          | RMMD↓        | COV↑         | MMD↓          | COV* ↑       | User Score↑ |
|-----------|---------------|--------------|--------------|---------------|--------------|-------------|
| PQ-Net    | <u>27.105</u> | 0.203        | 0.928        | 0.0091        | 0.593        | 0.45        |
| House-GAN | 31.773        | <u>0.195</u> | 0.912        | 0.0096        | 0.575        | 0.47        |
| PolyGen   | 79.532        | 0.685        | <u>0.953</u> | 0.0111        | 0.484        | 0.41        |
| Roof-GAN  | <b>22.235</b> | 0.196        | 0.926        | 0.0084        | 0.639        | <u>0.62</u> |
| MeshGPT   | 98.948        | \            | \            | 0.0085        | <u>0.641</u> | 0.42        |
| MeshXL    | 64.054        | \            | \            | <b>0.0078</b> | 0.552        | 0.51        |
| Ours      | 28.215        | <b>0.173</b> | <b>0.958</b> | <u>0.0083</u> | <b>0.645</b> | <b>0.78</b> |

The best result of each metric is shown in **bold**, and the second-best is underlined.

auto-regressive loop index generation model (Section 4.1) through cross-attention in the Transformer decoder, enabling the generation of a loop index sequence conditioned on the input image.

As illustrated in Figure 7, the generated results effectively preserve the geometric features of input silhouettes. This capability enables architects and designers to create custom roof boundaries and derive corresponding 3D roof structures that ensure architectural validity while aligning with the design intent. Figure 8 further demonstrates our real-world applicability, where we manually extract 2D building footprints and specify the loop condition (the number of loops and the number of vertices per loop) from aerial images to generate corresponding 3D roofs.

### 5.3 Comparison

*Comparison on RoofGAN dataset.* We compared against the representative graph-constrained 3D roof generation method, RoofGAN [Qian et al. 2021]. RoofGAN is restricted to four predefined rectangular roof primitives (i.e., horizontal-gable, vertical-gable, horizontal-hip, and vertical-hip), which limits its applicability to more diverse datasets. Therefore, we compared with RoofGAN on their proposed roof geometry dataset for a fair comparison. We also compared with a Seq2Seq-based CAD structure generator (PQ-Net [Wu et al. 2020]) and a floorplan generative model (HouseGAN [Nauata et al. 2020]). We further evaluated sequence-based mesh generation methods by training PolyGen [Nash et al. 2020] and MeshGPT [Siddiqui et al. 2024] on this dataset. Since MeshGPT operates exclusively on triangulated meshes, we first applied a triangulation step to the roof geometry before feeding it into the network. Since RMMD and COV are designed to evaluate both geometric and topological fidelity, they are defined on loop representations and cannot be directly applied to the triangle mesh outputs of MeshGPT. To enable fair cross-representation comparisons, we adopt the **minimum matching distance (MMD)**, computed from point clouds sampled from generated meshes, as a standard measure of geometric fidelity. Based on this, we also compute COV\* using the MMD.

Table 1 reports the quantitative results. Although RoofGAN achieves the best FID score, *Loops2Roofs* outperforms all RMMD- and COV-based methods. This indicates its ability to generate authentic, diverse roof models with geometric variations. The higher FID score of RoofGAN can be attributed to its use of templates composed of multiple loops, which ensures greater local completeness and more consistent normal maps during evaluation. In contrast, *Loops2Roofs* operates at a finer granularity, treating individual loops

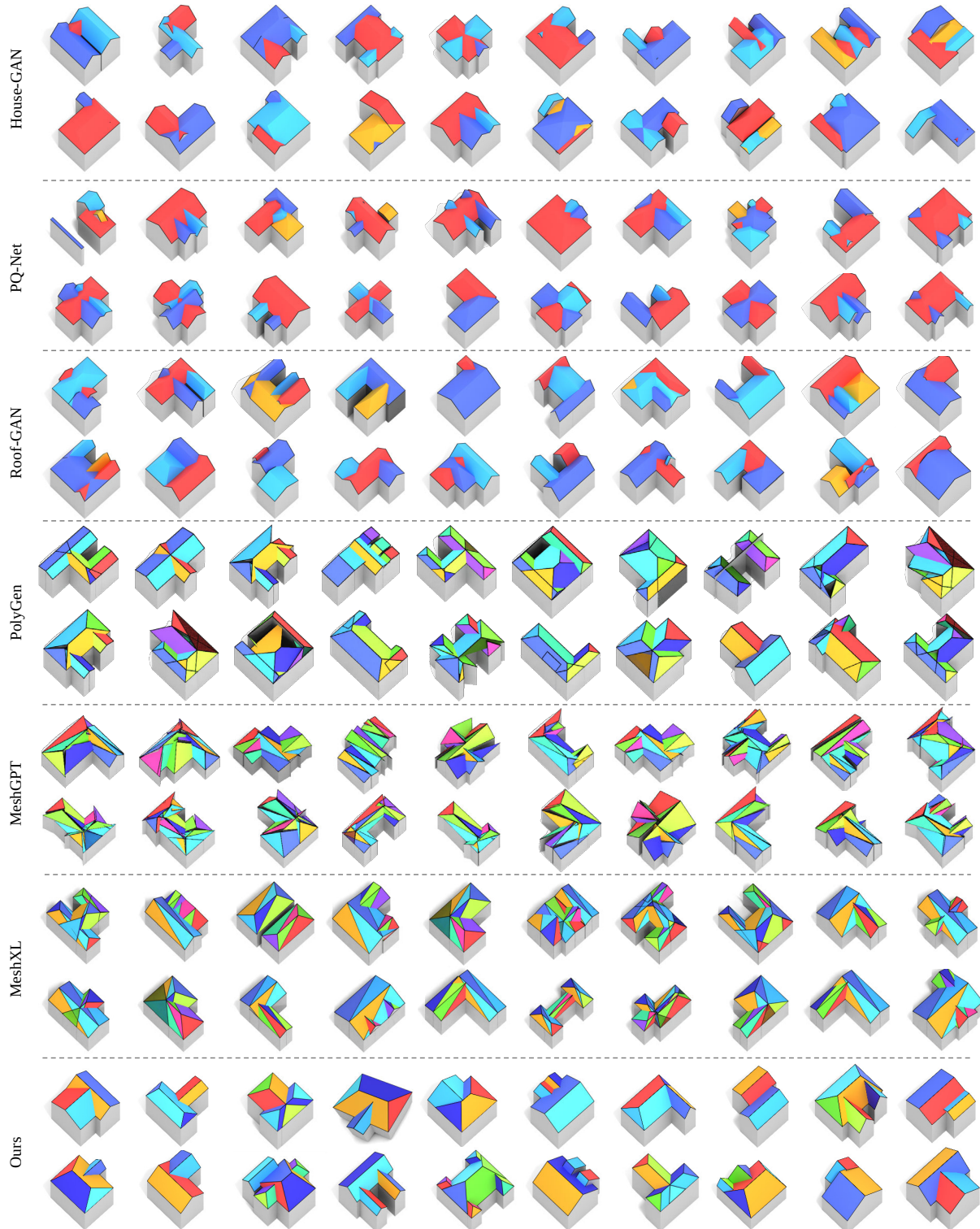


Fig. 9. Visual comparison of generated roof models on the Roof-GAN dataset. Our results demonstrate more complete and geometrically consistent structures. In contrast, the other methods suffer from geometric misalignments and inconsistencies.

Table 2. Quantitative Comparison on City3D and LC2WF Datasets

| Dataset | Method        | FID↓          | RMMD↓        | COV↑         | MMD↓         | COV* ↑       | User Score↑ |
|---------|---------------|---------------|--------------|--------------|--------------|--------------|-------------|
| City3D  | PQ-Net        | 51.178        | 0.802        | 0.613        | 0.025        | 0.631        | 0.32        |
|         | DeepCAD       | 40.597        | 1.088        | 0.561        | 0.030        | 0.652        | 0.35        |
|         | BrepGen       | 65.492        | 0.972        | 0.477        | 0.033        | 0.516        | 0.18        |
|         | PolyGen       | 46.541        | 0.853        | 0.431        | 0.027        | 0.608        | 0.38        |
|         | MeshGPT       | 87.623        | \            | \            | 0.024        | <b>0.663</b> | 0.29        |
|         | MeshXL        | <b>21.804</b> | \            | \            | <u>0.023</u> | 0.591        | <u>0.48</u> |
|         | Ours          | <u>39.440</u> | <b>0.699</b> | <b>0.679</b> | <b>0.022</b> | <u>0.655</u> | <b>0.69</b> |
| LC2WF   | PQ-Net        | 114.41        | 0.705        | 0.528        | 0.025        | 0.623        | 0.13        |
|         | DeepCAD       | 127.66        | 0.836        | 0.539        | 0.019        | 0.659        | 0.16        |
|         | BrepGen       | 76.448        | 0.437        | <u>0.591</u> | <u>0.018</u> | <u>0.662</u> | 0.33        |
|         | PolyGen       | 86.905        | 0.477        | 0.424        | 0.179        | 0.159        | 0.50        |
|         | RoofSynthesis | 80.554        | <u>0.403</u> | 0.556        | 0.125        | 0.182        | <u>0.63</u> |
|         | MeshGPT       | <u>59.706</u> | \            | \            | 0.131        | 0.198        | 0.47        |
|         | MeshXL        | 104.74        | \            | \            | 0.130        | 0.158        | 0.52        |
|         | Ours          | <b>23.941</b> | <b>0.262</b> | <b>0.797</b> | <b>0.013</b> | <b>0.698</b> | <b>0.67</b> |

as the minimal unit, which may yield less locally complete structures but allows greater flexibility in generated roof geometries. The user study results in the last column of Table 1 show that the models generated by our method were selected as the most realistic.

Qualitative comparisons in Figure 9 reveal that both House-GAN and PQ-Net suffer from pronounced misalignments and geometric inconsistencies, compromising the visual plausibility of their results. PolyGen, which represents 3D roofs as  $n$ -gon meshes, exhibits limited robustness during inference due to its separate vertex and face sequence predicting approach. Small mistakes in early vertex predictions can propagate through the sequence, leading to geometric inconsistencies between adjacent polygons. MeshGPT generates triangular meshes but performs poorly on the RoofGAN dataset due to its limited number of roof samples, revealing weak generalization capabilities. In contrast, our generated results, leveraging the flexible loop-based representation and topology-repairing capabilities of *LoopStitchingNet*, achieve topology and geometric details that are competitive against those of Roof-GAN. While Roof-GAN is specifically designed to handle the four rectangular roof primitives presented in the dataset, our approach can be broadly applied to datasets containing a wider variety of roof primitive types.

*Comparison on City3D and LC2WF dataset.* In addition to the mesh generation methods PolyGen and MeshGPT, we also compared our approach with state-of-the-art deep generation methods for producing CAD geometries, including PQ-Net [Wu et al. 2020], DeepCAD [Wu et al. 2021], BrepGen [Xu et al. 2024]. We trained them from scratch on the roof datasets used in our approach. We also conducted a comparison with RoofSynthesis [Ren et al. 2021], which relies on geometric and topological optimization. We compare their generation results on the LC2WF dataset provided by the authors.

We sought to preserve each method’s original neural architecture as closely as possible and to modify the input/output to fit our problem. We adapted the part geometry autoencoder in PQ-Net by converting it into a loop VAE to suit the roof generation task. This modification shifts the focus from encoding voxelized parts to encoding loops, enabling the extraction of latent loop features. A Transformer network served as the backbone of the loop VAE. We

then trained PQ-Net’s Seq2Seq autoencoder to learn each roof’s features, which consist of a sequence of loop features, yielding a 256-dimensional latent vector. Similar to PQ-Net, we employed latent GANs for 3D shape generation.

We also made minimal adjustments to DeepCAD’s autoencoder. The original autoencoder in DeepCAD processed CAD sequences, including the sequences of CAD command types and their corresponding parameters. We replaced the CAD sequences with roof sequences, modifying the original prediction of a command type to estimate the number of vertices in a loop within the roof while adjusting the prediction of parameters to forecast the 3D coordinates of these vertices. The revised autoencoder encoded each roof into a 64-dimensional latent vector. As in DeepCAD, we applied the latent-GAN technique to perform roof generation on the learned latent space.

For BrepGen, we retained its VAE + latent diffusion module pipeline but replaced its shape geometry VAEs with a loop VAE identical to the one used in the modified PQ-Net. Each loop is encoded into a 16-dimensional latent vector. Subsequently, we trained BrepGen’s Transformer-based denoising network to refine the loop sequences. The modified BrepGen can be regarded as the latent diffusion variant of our approach, where diffusion is directly applied to the coordinates of the loops (rather than latent loop features).

As reported in Table 2, *Loops2Roofs* achieves significantly better performance across the majority of metrics on both datasets. Benefiting from the flexibility of the loop-based roof representation, *Loops2Roofs* seamlessly accommodates various datasets.

Figure 10 shows the qualitative comparison of the City3D and LC2WF datasets. Unlike Roof-GAN, which targets a limited scope, these datasets present a broader range of roof primitive types and significantly increased geometric complexity. Although the previous CAD generation works also use loop sequences as input and manage to achieve geometric diversity, they often fail to maintain coherent topological relationships between roof faces, resulting in less plausible outputs. In contrast, *Loops2Roofs* demonstrates robust performance and adaptability. This performance difference is attributed to two main factors stemming from the nature of loop sequences as discrete data representing highly condensed geometric information: (1) The discrete characteristic results in a less smooth latent space. (2) The already compact nature of loop sequences leaves little room for meaningful dimensionality reduction through VAE encoding, which is employed in all other competing methods.

#### 5.4 Ablation Study

The ablation study validates the distinct features of our proposed pipeline.

*Continuous vs. discrete loop representation.* Discrete representations are widely used in CAD processing for their regularity and compatibility with existing systems [Wu et al. 2021]. In our diffusion-based framework, we investigated continuous and discrete representations to determine which better preserves geometric precision and enhances generation quality. The discrete approach quantizes vertex coordinates to fixed intervals, potentially reducing noise and computational complexity, whereas continuous representation preserves the full geometric precision. We compare the denoising performance on continuous coordinates (ranging in

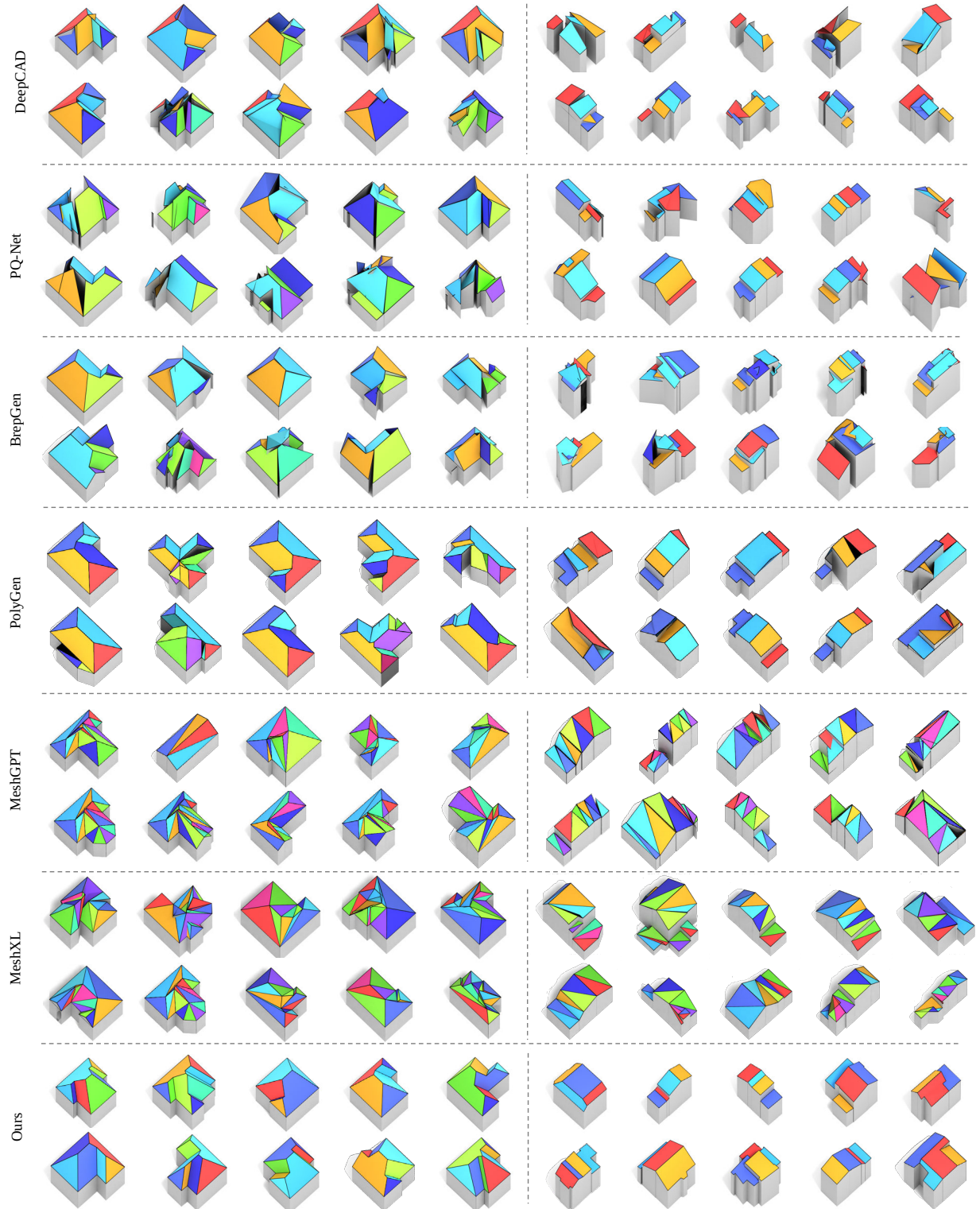


Fig. 10. Visual comparison on the LC2WF dataset (left) and the City3D dataset (right). Our method generates roof models with improved geometric fidelity and topological consistency, resulting in more plausible 3D structures across diverse architectural styles.

Table 3. Quantitative Results of Ablation Studies on The LC2WF Dataset

| Representation | Threshold ( $\tau$ ) | FID↓          | RMMD↓        | COV↑         |
|----------------|----------------------|---------------|--------------|--------------|
| Discrete       | -                    | 42.162        | 0.366        | 0.743        |
| CtD            | -                    | 31.246        | 0.281        | 0.781        |
| Continuous     | $\tau = 0.0$         | 27.147        | 0.276        | 0.776        |
|                | $\tau = 0.1$         | 26.508        | <u>0.265</u> | 0.786        |
|                | $\tau = 0.2$         | <u>25.163</u> | 0.267        | <u>0.789</u> |
|                | $\tau = 0.3$         | 25.933        | 0.273        | 0.782        |
|                | StitchingNet         | <b>23.941</b> | <b>0.262</b> | <b>0.797</b> |

We compare the generation performance of discrete, continuous, and continuous-then-discrete (CtD) [Shabani et al. 2023] loop representations. The effectiveness of our neural stitching network is verified by comparing it against a snapping-based heuristic with varying thresholds.

[-1,1]) with their discretized counterparts obtained through *analog bits* [Chen et al. 2022b] quantization. Table 3 shows that the continuous representation consistently outperforms its discrete counterpart in our model, yielding superior generation quality across multiple metrics. This suggests that for roof geometry generation, the benefits of maintaining continuous coordinate values outweigh the potential advantages of discretization.

HouseDiffusion [Shabani et al. 2023] introduced a **continuous-then-discrete (CtD)** denoising strategy that first applies continuous denoising and transitions to discrete denoising in later iterations to generate high-quality floorplans. However, determining the optimal transition point is non-trivial and typically requires empirical tuning. In contrast, our approach explicitly infers both geometric and topological relationships without relying on hand-tuned parameters, resulting in more robust performance. This advantage is supported by the superior performance shown in Table 3.

**Loop stitching network.** We introduced a neural approach of *LoopStitchingNet* to recover the exact adjacency relationship between different loops. Firstly, we conducted an ablation study to demonstrate its effectiveness by replacing *LoopStitchingNet* with a post-processing based on a snapping heuristic. A Euclidean distance threshold was used to determine whether two vertices should be considered as sharing the same position. We varied the threshold and recorded performance using thresholds of 0.0 (indicating no post-processing), 0.1, 0.2, and 0.3, respectively. As shown in Table 3, the results validate the effectiveness of *LoopStitchingNet*, as it significantly enhances geometric accuracy and resolves adjacency relationships more robustly than heuristic snapping. We also compared the runtime of *LoopStitchingNet* and the heuristic method on LC2WF. While *LoopStitchingNet* takes 0.036 seconds per sample compared to 0.021 seconds for the heuristic-based method, it eliminates the need for manual threshold tuning, an error-prone and time-consuming process that varies by dataset and often produces inconsistent results. Notably, the optimal threshold varies across datasets, making it difficult to determine a universal value. Larger thresholds risk merging distinct elements, while smaller ones may fail to identify duplicate geometry.

Next, we evaluated the stability of *LoopStitchingNet*. Since determining the exact success rate is challenging due to the lack of ground truth data for loop stitching, we used the simulated dataset from Section 4.3, where ground truth edge relationships are available. As shown in Table 4, *LoopStitchingNet* performs reliably on

Table 4. Performance Evaluation of *LoopStitchingNet* on the LC2WF Test Dataset

| Method                     | Accuracy     | Precision    | Recall       | F1 Score     |
|----------------------------|--------------|--------------|--------------|--------------|
| <b>StitchingNet (MSE)</b>  | <b>0.987</b> | <b>0.948</b> | <b>0.994</b> | <b>0.971</b> |
| StitchingNet (BCE)         | 0.977        | 0.000        | 0.000        | 0.000        |
| StitchingNet (W-BCE)       | 0.985        | 0.941        | 0.988        | 0.964        |
| Heuristic ( $\tau = 0.2$ ) | 0.946        | 0.921        | 0.966        | 0.952        |

Table 5. Ablation Study Comparing Point-Level and Loop-Level Generation on The LC2WF Dataset

| Method       | KL Div. ↓     | Avg. Loops (Gen/GT) | MaxMD ↓       | Valid. | Uniqueness ↑  |
|--------------|---------------|---------------------|---------------|--------|---------------|
| Point (Ours) | <b>0.0165</b> | 7.06 / 6.79         | <b>0.1279</b> | 1.000  | <b>0.7483</b> |
| Loop         | 0.3963        | 6.16 / 6.79         | 0.1373        | 1.000  | 0.5425        |

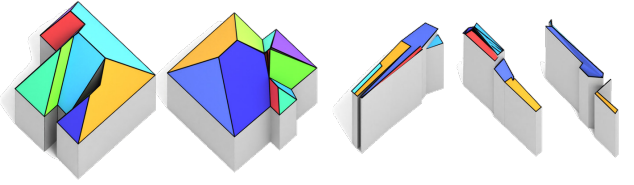


Fig. 11. Illustration of typical failure cases for our method. We face limitations in determining sufficient vertices for proper loop connections (shown in the left two models) or generating satisfactory results for elongated roof structures (shown in the right three models).

the LC2WF test dataset, successfully handling most cases and outperforming the heuristic-based method.

We further evaluated the choice of the MSE loss in Equation (10) through an ablation study comparing MSE, standard **binary cross-entropy (BCE)**, and **weighted BCE (W-BCE)**, as reported in Table 4. Due to the extreme class imbalance in the stitching task, standard BCE leads to degenerate solutions with near-zero recall. While W-BCE achieves performance comparable to MSE with carefully tuned class weights, MSE consistently provides more stable optimization and smoother gradients, resulting in the best overall performance without requiring heuristic class balancing.

**Loop and Vertex Index Generation Network.** Finally, we conduct an ablation study to compare two variants of our autoregressive model: point-level generation (ours) and loop-level generation. In the loop-level variant, the model directly predicts a sequence  $\{N_1, N_2, \dots, N_i\}$ , where each  $N_i$  denotes the number of vertices within the  $i$ -th loop. Both models are trained on the LC2WF training set and evaluated on the corresponding test set.

For evaluation, we perform inference on 1,000 samples per model and assess performance using three key criteria: distribution fidelity, structural validity, and diversity.

- **Distribution Fidelity** is measured using the **Kullback-Leibler (KL)** divergence, the average number of loops, and the **Maximum Mean Discrepancy (MaxMD)**.
- **Structural Validity** quantifies the proportion of generated sequences that form valid closed polygons.
- **Diversity** is defined as the ratio of unique sequences to the total number of generated samples.

Table 5 shows that the point-level model consistently matches or outperforms the loop-level variant across all metrics. In particular,

the point-level formulation achieves notably higher diversity and better distribution fidelity (KL), indicating that point-level modeling more effectively captures the fine-grained topological variability inherent in diverse roof structures.

## 6 Conclusions and Future Work

We have introduced a novel generative diffusion framework for roof model generation. At the core of our approach is a loop-based roof representation, in which roofs are represented by a sequence of 3D polygonal loops. This representation offers the flexibility to model complex roof structures in diverse styles. By leveraging this adaptable representation, complex roof modeling becomes a task of generating discrete polygonal primitives. This transformation significantly simplifies the modeling process while preserving design fidelity. Our approach employs a two-stage learning framework that generates polygonal loops sequentially and then merges them to form a topologically consistent compact roof model. Extensive quantitative and qualitative evaluations show that *Loops2Roofs* surpasses existing approaches, consistently generating 3D roof models with greater diversity and structural complexity.

**Limitations and future work.** Currently, our approach does not explicitly enforce co-planarity, though non-planar faces are rare in practice. During data preprocessing, a non-planar face is detected by sampling a set of points, defining a reference plane from the first three non-collinear points, and verifying that not all remaining vertices lie within a tolerance of this plane. Upon detection, we apply least-squares fitting to project vertices onto an estimated plane. This pre-processing step has a negligible impact on the final results, as non-planar faces are rare and the induced vertex adjustments are minimal. Second, our loop-based polygonal representation is unsuitable for modeling non-planar surfaces, such as curved domes or barrel roofs. Accurately representing such geometry requires a large number of loops to approximate the curved surfaces with planar sub-faces, each represented as a loop. Additionally, the method also occasionally faces loop connection issues, as seen in the two leftmost examples of Figure 11. This challenge arises from the pre-defined vertex count for each loop, which may result in insufficient vertices for seamless alignment. Furthermore, the method is unsuitable for generating elongated roof structures, as illustrated in the three rightmost examples in Figure 11.

In future work, we plan to investigate **Boundary Representation (B-rep)**, a more advanced geometric modeling approach capable of precisely capturing planar and curved surfaces. This exploration aims at enhancing both the geometric fidelity and applicability of our approach. We also intend to investigate dynamic vertex count adjustment during generation to address connectivity challenges associated with fixed loop configurations. Finally, we would like to incorporate 3D representations (e.g., 3D voxels or point clouds) as input conditions by using powerful neural feature extractors.

## References

- Oswin Aichholzer and Franz Aurenhammer. 1996. Straight skeletons for general polygonal figures in the plane. In *Proceedings of the International Computing and Combinatorics Conference*. Springer, 117–126.
- Oswin Aichholzer, Franz Aurenhammer, David Alberts, and Bernd Gärtner. 1996. *A Novel Type of Skeleton for Polygons*. Springer.
- Fatemeh Alidoost, Hossein Arefi, and Michael Hahn. 2020. Y-shaped convolutional neural network for 3D roof elements extraction to reconstruct building models from a single aerial image. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* V-2-2020, (2020), 321–328.
- Antonio Alliegro, Yawar Siddiqui, Tatiana Tommasi, and Matthias Nießner. 2023. Polydiff: Generating 3d polygonal meshes with diffusion models. arXiv:2312.11417. Retrieved from <https://arxiv.org/abs/2312.11417>
- Murat Arikian, Michael Schwärzler, Simon Flöry, Michael Wimmer, and Stefan Maierhofer. 2013. O-snap: Optimization-based snapping for modeling architecture. *ACM Transactions on Graphics* 32, 1 (2013), 15 pages.
- Jean-Philippe Bauchet and Florent Lafarge. 2020. Kinetic shape reconstruction. *ACM Transactions on Graphics* 39, 5 (2020), 156:1–156:14.
- Therese Biedl, Martin Held, Stefan Huber, Dominik Kaaser, and Peter Palfrader. 2015. Weighted straight skeletons in the plane. *Computational Geometry* 48, 2 (2015), 120–133.
- Filip Biljecki, Jantien Stoter, Hugo Ledoux, Sisi Zlatanova, and Arzu Çöltekin. 2015. Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information* 4, 4 (2015), 2842–2889.
- Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, Sameh Khamis, et al. 2022. Efficient geometry-aware 3D generative adversarial networks. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. 16123–16133.
- Kai-Hung Chang, Chin-Yi Cheng, Jieliang Luo, Shingo Murata, Mehdi Nourbakhsh, and Yoshito Tsuji. 2021. Building-GAN: Graph-conditioned architectural volumetric design generation. In *Proceedings of the IEEE International Conference on Computer Vision*. 11956–11965.
- Anne-Laure Chauve, Patrick Labatut, and Jean-Philippe Pons. 2010. Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. 1261–1268.
- Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. 2022b. Analog bits: Generating discrete data using diffusion models with self-conditioning. arXiv:2208.04202. Retrieved from <https://arxiv.org/abs/2208.04202>
- Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Zhongang Cai, Lei Yang, Gang Yu, Guosheng Lin, and Chi Zhang. 2025. MeshAnything: Artist-created mesh generation with autoregressive transformers. In *International Conference on Learning Representations*, Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (Eds.). 2025 (2025), 51369–51389.
- Zhaiyu Chen, Hugo Ledoux, Seyran Khademi, and Liangliang Nan. 2022a. Reconstructing compact building models from point clouds using deep implicit fields. *ISPRS Journal of Photogrammetry and Remote Sensing* 194 (2022), 58–73.
- Zhaiyu Chen, Yilei Shi, Liangliang Nan, Zhitong Xiong, and Xiao Xiang Zhu. 2024b. PolyGNN: Polyhedron-based graph neural network for 3D building reconstruction from point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 218 (2024), 693–706.
- Blender Online Community. 2018. *Blender - a 3D Modelling and Rendering Package*. Blender Foundation, Stichting Blender Foundation, Amsterdam. Retrieved from <http://www.blender.org>
- Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems* 34 (2021), 8780–8794.
- Johannes Edelsbrunner, Ulrich Krispel, Sven Havemann, Alexei Sourin, and Dieter W. Fellner. 2016. Constructive roofs from solid building primitives. *Transactions on Computational Science XXVI* (2016), 17–40.
- David Eppstein and Jeff Erickson. 1998. Raising roofs, crashing cycles, and playing pool: Applications of a data structure for finding pairwise interactions. In *Proceedings of the 14th Annual Symposium on Computational Geometry*. 58–67.
- Dieter Finkenzeller. 2008. Detailed building facades. *IEEE Computer Graphics and Applications* 28, 3 (2008), 58–66.
- Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. 2022a. Get3d: A generative model of high quality 3d textured shapes learned from images. In *Proceedings of the Advances in Neural Information Processing Systems*. 31841–31854.
- Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao Zhang. 2019. SDM-NET: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics* 38, 6 (2019), 1–15.
- Xifeng Gao, Kui Wu, and Zherong Pan. 2022b. Low-poly mesh generation for building models. In *ACM SIGGRAPH 2022 Conference Proceedings*. 1–9.
- Michael Garland and Paul S. Heckbert. 1997. Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'97)*. 209–216.
- Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. 2022. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. 10696–10706.
- Jianwei Guo, Yanchao Liu, Xin Song, Haoyu Liu, Xiaopeng Zhang, and Zhanglin Cheng. 2024. Line-based 3D building abstraction and polygonal surface reconstruction from images. *IEEE Transactions on Visualization and Computer Graphics* 30, 7 (2024), 3283–3297. DOI: <https://doi.org/10.1109/TVCG.2022.3230369>

- Jianwei Guo, HaoBo Qin, Yinchang Zhou, Xin Chen, Liangliang Nan, and Hui Huang. 2024. Fast building instance proxy reconstruction for large urban scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 11 (2024), 7267–7282.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. 770–778.
- Liu He and Daniel Aliaga. 2023. GlobalMapper: Arbitrary-shaped urban layout generation. In *Proceedings of the IEEE International Conference on Computer Vision*. 454–464.
- Martin Held and Peter Pfaff. 2017. Straight skeletons with additive and multiplicative weights and their application to the algorithmic generation of roofs and terrains. *Computer-Aided Design* 92 (2017), 33–41.
- Sijin He, Xin Chen, Anqi Pang, Xianfang Zeng, Wei Cheng, Yijun Fu, Fukun Yin, Zhibin Wang, Jingyi Yu, and Gang Yu. 2024. Meshxl: Neural coordinate field for generative 3d foundation models. In *Proceedings of the Advances in Neural Information Processing Systems*. 97141–97166.
- Andre Henn, Gerhard Gröger, Viktor Stroh, and Lutz Plümer. 2013. Model driven reconstruction of roofs from sparse LiDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 76 (2013), 17–29.
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. 6629–6640.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20)*.
- Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. 2022. Video diffusion models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS'22)*.
- Ruizhen Hu, Zeyu Huang, Yuhan Tang, Oliver Van Kaick, Hao Zhang, and Hui Huang. 2020. Graph2plan: Learning floorplan generation from layout graphs. *ACM Transactions on Graphics* 39, 4 (2020), 118–1.
- Hai Huang, Claus Brenner, and Monika Sester. 2013. A generative statistical approach to automatic 3D building roof reconstruction from laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing* 79 (2013), 29–43.
- Jin Huang, Jantien Stoter, Ravi Peters, and Liangliang Nan. 2022. City3D: Large-scale building reconstruction from airborne LiDAR point clouds. *Remote Sensing* 14, 9 (2022), 2254.
- Tom Kelly, John Femiani, Peter Wonka, and Niloy J. Mitra. 2017. BigSUR: Large-scale structured urban reconstruction. *ACM Trans. Graph.* 36, 6 (2017).
- Tom Kelly, Paul Guerrero, Anthony Steed, Peter Wonka, and Niloy J. Mitra. 2018. FrankGAN: Guided detail synthesis for building mass models using style-synchronized GANs. *ACM Transactions on Graphics* 37, 6 (2018), 14 pages.
- Tom Kelly and Peter Wonka. 2011. Interactive architectural modeling with procedural extrusions. *ACM Transactions on Graphics* 30, 2 (2011), 1–15.
- Robert G. Laycock and A. M. Day. 2003. Automatically generating roof models from building footprints. In *11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*. 81–84.
- Li Li, Nan Song, Fei Sun, Xinyi Liu, Ruisheng Wang, Jian Yao, and Shaosheng Cao. 2022. Point2Roof: End-to-end 3D building roof modeling from airborne LiDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 193 (2022), 17–28.
- Hui Lin, Jizhou Gao, Yu Zhou, Guiliang Lu, Mao Ye, Chenxi Zhang, Ligang Liu, and Ruigang Yang. 2013. Semantic decomposition and reconstruction of residential scenes from LiDAR data. *ACM Transactions on Graphics* 32, 4 (2013), 1–10.
- Jiayi Liu, Hou In Ivan Tam, Ali Mahdavi-Amiri, and Manolis Savva. 2024b. CAGE: Controllable articulation generation. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. 17880–17889.
- Yang Liu, Chuanchen Luo, Lue Fan, Naiyan Wang, Junran Peng, and Zhaoxiang Zhang. 2024a. Citygaussian: Real-time high-quality large-scale scene rendering with gaussians. In *Proceedings of the European Conference on Computer Vision*. Springer, 265–282.
- Yicheng Luo, Jing Ren, Xuefei Zhe, Di Kang, Yajing Xu, Peter Wonka, and Linchao Bao. 2022. Lc2wf: Learning to construct 3d building wireframes from 3d line clouds. In *Proceedings of the British Machine Vision Conference*.
- Aron Monszpart, Nicolas Mellado, Gabriel J. Brostow, and Niloy J. Mitra. 2015. RAPter: Rebuilding man-made scenes with regular arrangements of planes. *ACM Transactions on Graphics* 34, 4 (2015), 103:1–103:12.
- Pascal Müller, Peter Wonka, Simon Haegler, Andreas Ulmer, and Luc Van Gool. 2006. Procedural modeling of buildings. *ACM Trans. Graph.* 25, 3 (July 2006), 614–623.
- Przemyslaw Musialski, Peter Wonka, Daniel G. Aliaga, Michael Wimmer, Luc Van Gool, and Werner Purgathofer. 2013. A survey of urban reconstruction. *Computer Graphics Forum* 32, 6 (2013), 146–177.
- Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*. 807–814.
- Liangliang Nan, Andrei Sharf, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. 2010. Smartboxes for interactive urban reconstruction. In *ACM SIGGRAPH 2010 Papers (SIGGRAPH'10)*.
- Liangliang Nan and Peter Wonka. 2017. Polyfit: Polygonal surface reconstruction from point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*. 2353–2361.
- Charlie Nash, Yaroslav Ganin, S. M. Ali Eslami, and Peter Battaglia. 2020. Polygen: An autoregressive generative model of 3d meshes. In *Proceedings of the International Conference on Machine Learning*. PMLR, 7220–7229.
- Nelson Nauata, Kai-Hung Chang, Chin-Yi Cheng, Greg Mori, and Yasutaka Furukawa. 2020. House-gan: Relational generative adversarial networks for graph-constrained house layout generation. In *Proceedings of the European Conference on Computer Vision*. Springer, 162–177.
- Nelson Nauata, Sepidehsadat Hosseini, Kai-Hung Chang, Hang Chu, Chin-Yi Cheng, and Yasutaka Furukawa. 2021. House-gan++: Generative adversarial layout refinement network towards intelligent computational agent for professional architects. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. 13632–13641.
- Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. 2022. Point-e: A system for generating 3d point clouds from complex prompts. arXiv:1409.0473. Retrieved from <https://arxiv.org/abs/1701.00133>
- Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *Proceedings of the International Conference on Machine Learning*. PMLR, 8162–8171.
- Zhexi Peng, Kun Zhou, and Tianjia Shao. 2025. Gaussian-plus-SDF SLAM: High-fidelity 3D reconstruction at 150+ fps. *Computational Visual Media* 11, 6 (2025), 1195–1208.
- Ryan Po, Wang Yifan, Vladislav Golyanik, Kfir Aberman, Jonathan T. Barron, Amit Bermano, Eric Chan, Tali Dekel, Aleksander Holynski, Angjoo Kanazawa, et al. 2024. State of the art on diffusion models for visual computing. In *Proceedings of the Computer Graphics Forum*. e15063.
- Yiming Qian, Hao Zhang, and Yasutaka Furukawa. 2021. Roof-gan: Learning to generate roof geometry and relations for residential houses. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. 2796–2805.
- Jing Ren, Biao Zhang, Bojian Wu, Jianqiang Huang, Lubin Fan, Maks Ovsjanikov, and Peter Wonka. 2021. Intuitive and efficient roof modeling for reconstruction and synthesis. *ACM Transactions on Graphics* 40, 6 (2021), 17 pages.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. 10684–10695.
- David Salinas, Florent Lafarge, and Pierre Alliez. 2015. Structure-aware mesh decimation. *Computer Graphics Forum* 34, 6 (2015), 211–227.
- Michael Schwarz and Pascal Müller. 2015. Advanced procedural modeling of architecture. *ACM Transactions on Graphics* 34, 4 (2015), 1–12.
- Pratheba Selvaraju, Mohamed Nabail, Marios Loizou, Maria Maslioukova, Melinos Averkiou, Andreas Andreou, Siddhartha Chaudhuri, and Evangelos Kalogerakis. 2021. Buildingnet: Learning to label 3d buildings. In *Proceedings of the IEEE International Conference on Computer Vision*. 10397–10407.
- Mohammad Amin Shabani, Sepidehsadat Hosseini, and Yasutaka Furukawa. 2023. Housediffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. 5466–5475.
- Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. 2024. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. 19615–19625.
- Kenichi Sugihara and Yoshitugu Hayashi. 2008. Automatic generation of 3D building models with multiple roofs. *Tsinghua Science and Technology* 13, S1 (2008), 368–374.
- Jiaxiang Tang, Zhaoshuo Li, Zekun Hao, Xian Liu, Gang Zeng, Ming-Yu Liu, and Qingsheng Zhang. 2025. Edgerunner: Auto-regressive auto-encoder for artistic mesh generation. In *Proceedings of the International Conference on Learning Representations*.
- Aaron Van Den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. 6309–6318.
- Carlos A. Vanegas, Daniel G. Aliaga, and Bedrich Benes. 2010. Building reconstruction using manhattan-world grammars. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. IEEE, 358–365.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. 6000–6010.
- Yannick Verdie, Florent Lafarge, and Pierre Alliez. 2015. LOD generation for urban scenes. *ACM Transactions on Graphics* 34, 3 (2015), 30:1–30:14.
- Vivek Verma, Rakesh Kumar, and Stephen Hsu. 2006. 3D building detection and modeling from aerial LiDAR data. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. IEEE, 2213–2220.
- Hanxiao Wang, Mingyang Zhao, Yiqun Wang, Weize Quan, and Dong-Ming Yan. 2024. VQ-CAD: Computer-aided design model generation with vector quantized diffusion. *Computer Aided Geometric Design* 111 (2024), 102327.

- Ruisheng Wang, Shangfeng Huang, and Hongxin Yang. 2023. Building3d: A urban-scale dataset and benchmarks for learning roof structures from point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*. 20076–20086.
- Ruisheng Wang, Jiju Peethambaran, and Dong Chen. 2018. Lidar point clouds to 3-D urban models : A review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11, 2 (2018), 606–627.
- Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation* 1, 2 (1989), 270–280.
- Rundi Wu, Chang Xiao, and Changxi Zheng. 2021. Deepcad: A deep generative network for computer-aided design models. In *Proceedings of the IEEE International Conference on Computer Vision*. 6772–6782.
- Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. 2020. Pq-net: A generative part seq2seq network for 3d shapes. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. 829–838.
- Zhen Xing, Qijun Feng, Haoran Chen, Qi Dai, Han Hu, Hang Xu, Zuxuan Wu, and Yu-Gang Jiang. 2024. A survey on video diffusion models. *Computing Surveys* 57, 2 (2024), 1–42.
- Xiang Xu, Joseph Lambourne, Pradeep Jayaraman, Zhengqing Wang, Karl Willis, and Yasutaka Furukawa. 2024. BrepGen: A B-rep generative diffusion model with structured latent geometry. *ACM Transactions on Graphics* 43, 4 (2024), 1–14.
- Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. 2023. Diffusion models: A comprehensive survey of methods and applications. *Computing Surveys* 56, 4 (2023), 1–39.
- Zhensheng Yuan, Haozhi Huang, Zhen Xiong, Di Wang, and Guanghua Yang. 2025. Robust and efficient 3D gaussian splatting for urban scene reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision*. 26209–26219.
- Huayi Zeng, Jiaye Wu, and Yasutaka Furukawa. 2018. Neural procedural reconstruction for residential buildings. In *Proceedings of the European Conference on Computer Vision*. 737–753.
- Biao Zhang, Matthias Nießner, and Peter Wonka. 2022. 3dilg: Irregular latent grids for 3d generative modeling. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'22)*. 21871–21885.
- Wenyuan Zhang, Zhixin Li, and Jie Shan. 2021. Optimal model fitting for building reconstruction from point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021), 9636–9650.
- Xiaowei Zhang and Daniel Aliaga. 2022. Procedural roof generation from a single satellite image. *Computer Graphics Forum* 41, 2 (2022), 249–260.
- Linqi Zhou, Yilun Du, and Jiajun Wu. 2021. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*. 5826–5835.
- Qian-Yi Zhou and Ulrich Neumann. 2010. 2.5 d dual contouring: A robust approach to creating building models from aerial lidar point clouds. In *Proceedings of the European Conference on Computer Vision*. Springer, 115–128.

Received 5 September 2025; revised 5 February 2026; accepted 26 March 2026