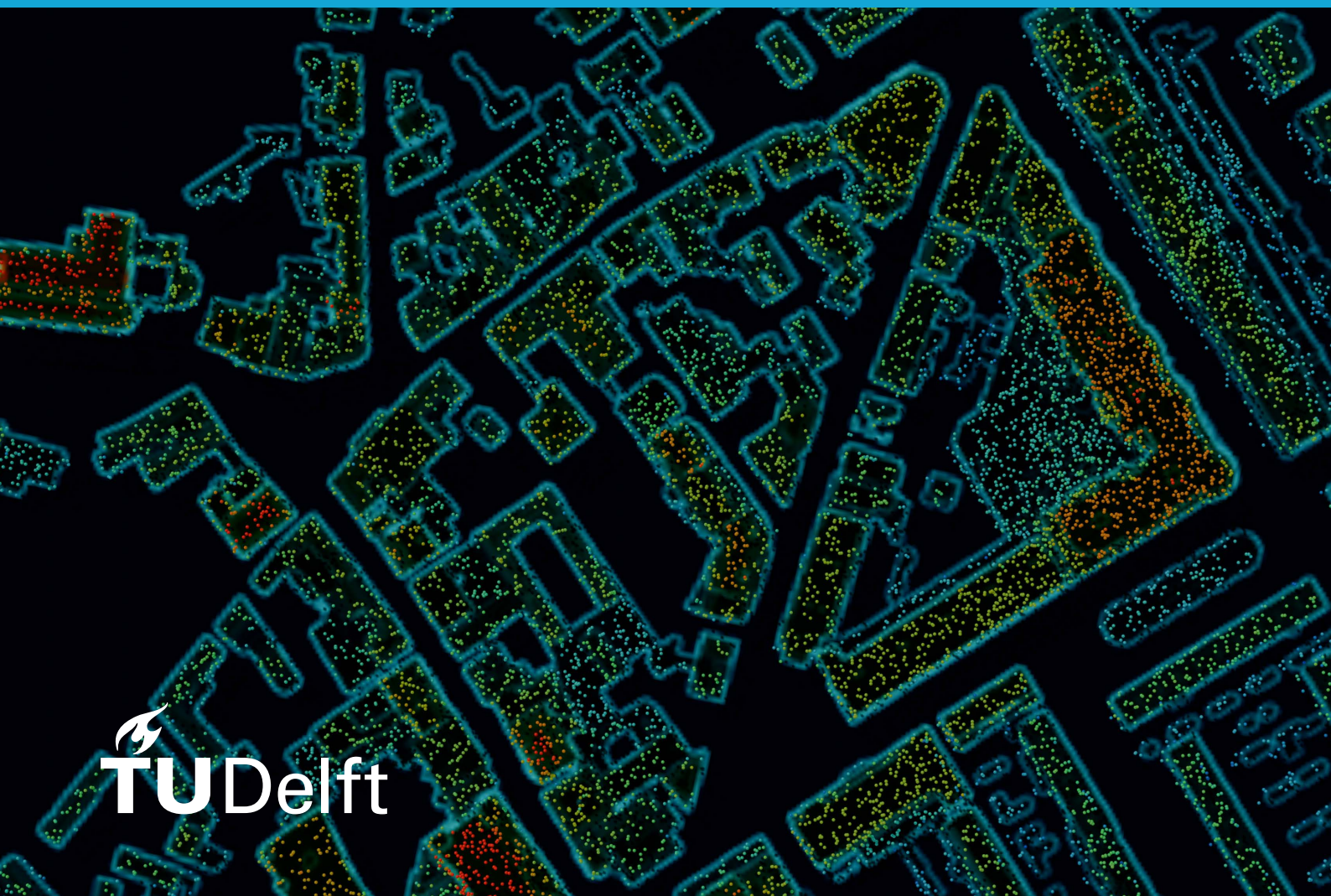
MSc thesis in Geomatics

Structure guided roof heightmap completion

Xiaduo Zhao

2025



TU Delft

MSc thesis in Geomatics

Structure Guided Roof Heightmap Completion

Xiaduo Zhao

June 2025

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Geomatics

The work in this thesis was carried out in the:

3D geoinformation group
Delft University of Technology

Supervisors:	Dr. Weixiao Gao
	Prof. Hugo Ledoux
Co-reader:	Ravi Peters

Abstract

Urban digital twins rely on accurate rooftop geometry, yet airborne lidar point clouds are frequently sparse and incomplete, leading to substantial information loss in building reconstruction. This thesis investigates diffusion–based learning as a remedy for high-fidelity roof recovery under severe data corruption.

This thesis proposes a two-stage framework that operates on 2.5D height-map representations. Stage I introduces a dual-task diffusion model that jointly performs roof height-map completion and roof-line prediction. A novel Bidirectional Control Module enables reciprocal conditioning between the two tasks, enforcing geometric consistency during the denoising process. Stage II employs a patch-based diffusion upsampler equipped with positional embeddings and a domain-specific global context encoder to synthesise high-resolution height maps while remaining computationally tractable for large and variably-sized buildings. A rigorous preprocessing pipeline further yields two challenging benchmarks, S80_I30 and S80_I80, derived from 160k real-world building samples.

Extensive experiments conducted on these datasets demonstrate the effectiveness of the proposed approach. Under moderate corruption (S80_I30), the completion model attains an *RMSE* of **0.89** m and a Chamfer distance of **0.06**, improving upon the state-of-the-art RoofDiffusion baseline by 13.2% and 17.3%, respectively. In the severe setting (S80_I80), the method sustains a 13.5% *RMSE* reduction. The upsampling stage delivers an additional 10% *RMSE* gain over the best classical interpolator, and the end-to-end pipeline achieves *RMSE* values of 0.91 m (moderate) and 1.42 m (severe).

The thesis contributes: (i) a structurally-aware diffusion framework for roof completion, (ii) a scalable patch-based upsampler, and (iii) public benchmarks that reflect real lidar degradation. Collectively, these advances close a critical gap between theoretical research and practical generation of LOD2.2 building models, facilitating more reliable urban analytics and planning applications.

The code and datasets are available at `https://github.com/xdzhaooo/msc_roof_completion`.

Acknowledgements

As I reflect on the journey of completing this master's thesis, my heart is filled with gratitude for the countless individuals who have supported me along the way.

First and foremost, I extend my deepest appreciation to my supervisors, Dr. Weixiao Gao and Prof. Hugo Ledoux. Their guidance, patience, and insightful feedback have been invaluable in shaping this work. Their encouragement pushed me to explore beyond my limits, and their expertise illuminated paths I could not have navigated alone.

To my friends, who became my family in the Netherlands, thank you for the laughter, late-night conversations, and unwavering support. You made every challenge feel surmountable and every moment there unforgettable. To my roommates, your warmth and camaraderie transformed our shared space into a home. Whether it was sharing meals, debating ideas, or simply being there, your presence made all the difference.

Finally, to my family back home, your love and belief in me carried me through moments of doubt.

Delft has been more than a place of study—it has been a chapter of growth, connection, and discovery. Thank you all for being part of it.

. . .

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

lidar Light Detection and Ranging

DSM Digital Surface Model

LOD Level of Detail

AHN Actueel Hoogtebestand Nederland

RMSE Root Mean Square Error

MAE Mean Absolute Error

IDW Inverse Distance Weighting

STA Spatial Transformer Attention

CLIP Contrastive Language-Image Pre-training

BCM Bidirectional Control Module

PE Positional Embedding

GC Global Condition

RC-Full Roof Completion - Full Model

RC-Ablated Roof Completion - Ablated Model

UP-Full Upsampling - Full Model

UP-NoGC Upsampling - No Global Context

UP-NoPE Upsampling - No Positional Embedding

UP-Patch Upsampling - Patch-Based Processing

UP-FullImg Upsampling - Full-Image Processing

S80-i30 Sparsity 80%, Incompleteness 30%

S80-i80 Sparsity 80%, Incompleteness 80%

CNN Convolutional Neural Network

GAN Generative Adversarial Network

1. Introduction

1.1. Background

3D city models are increasingly used in a wide range of applications, including urban planning (Biljecki et al., 2015), aircraft and car navigation (Gruen, 2008), energy management (Agugiaro, 2016), and disaster response (Agrawal and Gupta, 2017). Common data sources for constructing these models include aerial imagery, satellite imagery, and airborne lidar point clouds, with airborne lidar scanning providing high-precision raw measurements (Wang et al., 2018). Techniques for constructing concise 3D building representations from these data sources have been extensively investigated (Holzmann et al., 2018; Huang et al., 2022; Nan and Wonka, 2017). However, airborne lidar point clouds often suffer from low point density and incompleteness due to occlusions, noise, and sparsity, particularly when scanning large urban areas (Figure 1.1,Figure 1.2) (Rottensteiner, 2009; Lafarge and Mallet, 2012). These data imperfections often cause conventional reconstruction algorithms to produce incomplete or topologically incorrect models. While one approach is to design more complex reconstruction algorithms, an alternative strategy, which this thesis adopts, is to first complete the point cloud data before modeling.

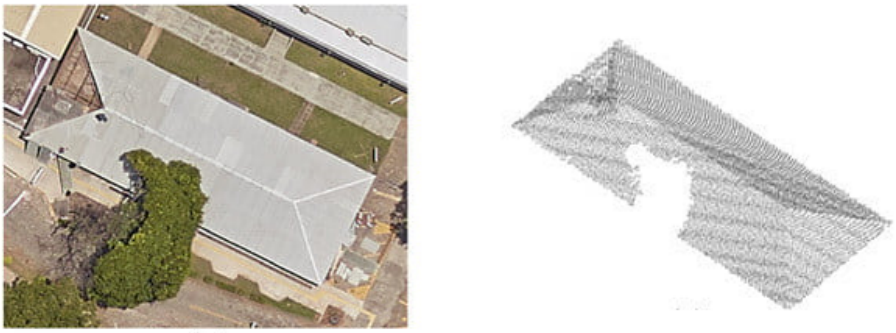


Figure 1.1.: An example of an airborne lidar point cloud exhibiting occlusions and sparsity.

While existing methods have made progress in general point cloud completion (Tesema et al., 2024), they face significant limitations when applied to roof shape reconstruction, particularly in preserving fine architectural details, maintaining sharp edges, and ensuring accurate scale alignment (Gao et al., 2024). These challenges are especially pronounced in complex buildings with multiple roof planes and irregular structures, where current approaches often result in the loss of small-scale features (Figure 1.3), over-smoothing of sharp edges (Figure 1.4), and normalization inconsistencies (Figure 1.5) (Gao et al., 2024).

While 3D point cloud-based completion faces these challenges, an alternative approach leverages 2D representations through Digital Surface Models (DSMs), also called heightmaps.

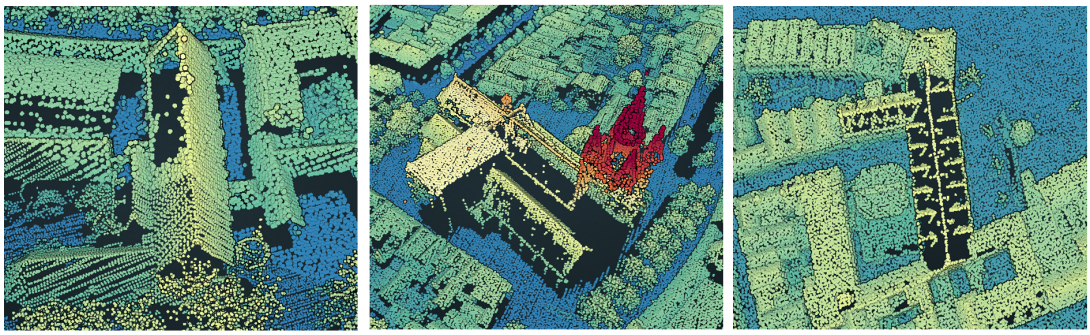


Figure 1.2.: Examples of point-cloud incompleteness in airborne-lidar roof scans. **Left**: a large, continuous roof area is missing due to scanning–angle *obstruction*, where external factors limit the laser's reach. **Centre**: severe gaps around the ridge and beneath the tower arise from geometric *self-occlusion*, i.e. the building itself blocks the laser. **Right**: widespread missing data result from an *overly complex structure*; dense neighbouring objects (e.g. other buildings or vegetation) obstruct the laser and prevent full coverage.
(Figure from `ahn2.pointclouds.nl`)

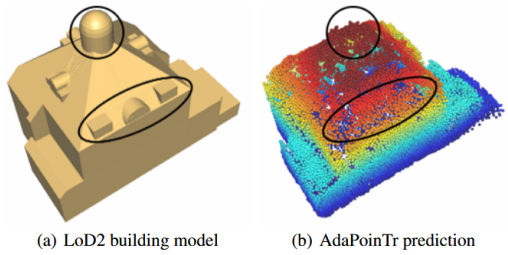


Figure 1.3.: Loss of local details (e.g., rooftop and dormers) in a point cloud predicted by AdaPoinTr. (Figure from Gao et al. (2024))

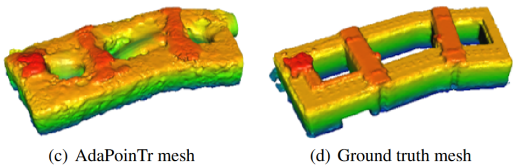


Figure 1.4.: Over-smoothing of sharp features at the intersection of piecewise planar surfaces (e.g., where facades meet roofs). (Figure from Gao et al. (2024))

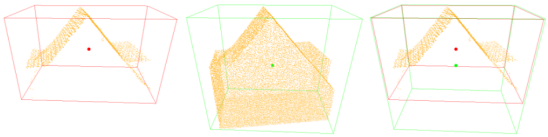


Figure 1.5.: Normalization issues in point cloud completion. Inaccurate alignment of partial point clouds to ground truth can lead to training or prediction failures. (Figure from Gao et al. (2024))

heightmaps are single-channel images where pixel values represent the elevation of natural and artificial features. Rasterizing airborne lidar point clouds allows for working with more manageable 2D data while preserving crucial height information (Liu, 2008). This 2.5D representation offers several advantages over direct 3D point cloud processing, including reduced computational complexity, the use of well-established 2D image processing techniques, and a regular data structure (Musialski et al., 2013; Haala and Kada, 2010). Moreover, as most buildings in practice are represented as 2.5D models with "jump edges," the heightmap assumption is valid for the majority of real-world cases (Biljecki et al., 2015). Furthermore, building footprints—which define the 2D extent of structures—can often be readily obtained from geoinformation databases (e.g., cadastral maps), providing a reliable and readily available source of context for the reconstruction process.

Given the remarkable ability of diffusion models to capture complex data distributions and produce fine-detailed, high-quality results, they present a promising solution for image inpainting,denoising and completion tasks (Saharia et al., 2022a). Building on these benefits, RoofDiffusion recently demonstrated how diffusion models can be effectively applied to heightmaps for inferring missing roof segments (Lo et al., 2024), showing promising results in handling complex roof structures. However, as a purely data-driven method, its performance is fundamentally constrained by the diversity and scope of its training data, limiting its ability to generalize to novel architectural styles or geographical contexts. Furthermore, its reliance on implicitly learned priors compromises its ability to preserve fine geometric details and enforce global structural consistency, especially for large-scale buildings or in cases of severe data corruption.

This research aims to further leverage and explore diffusion-based approaches using heightmaps to address the limitations of existing roof reconstruction methods for airborne lidar point clouds. Specifically, this work develops an enhanced diffusion framework and compiles a dataset enriched with diverse roof shapes, architectural styles, and authentic occlusion scenarios. By focusing on preserving fine-grained details, maintaining accurate geometric alignments, and ensuring robust performance across varied building scales, the goal is to achieve more realistic and reliable roof heightmaps from severely incomplete point cloud data. These high-fidelity reconstructions, with their preserved architectural details and precise geometric features, can serve as an ideal foundation for more detailed modeling, significantly simplifying the generation of detailed 3D city models that include elements such as dormers, chimneys, and other roof installations.

1.2. Research Questions

To address the challenges outlined above and to guide the research toward its objectives, this thesis is structured around a central research question and three focused sub-questions.

How can diffusion models be leveraged to accurately complete roof point clouds from 2D heightmaps, while accommodating complex roof geometries and varying scales?

This research investigates the capability of diffusion models to address challenges in completing roof shapes from heightmaps derived from sparse, noisy, and corrupted lidar data. The primary objective is to develop and validate an enhanced diffusion framework for this task. The following sub-questions guide this research:

RQ1: How can a diverse, large-scale building heightmap dataset be constructed, integrating various architectural forms and conditional data, while ensuring category balance and meeting the rigorous demands of deep learning tasks?

The construction of a diverse and large-scale dataset is a foundational step in training robust deep learning models, particularly for diffusion-based approaches that rely on data quality and variety to generalize effectively. For the task of roof heightmap completion, the dataset must encompass a wide range of architectural forms (e.g., flat, gabled, hipped, or complex roof geometries) and incorporate conditional data such as building footprints and lidar-derived sparse heightmaps. This sub-question addresses the challenge of curating a dataset that captures the variability in roof structures while ensuring category balance to prevent model bias toward overrepresented roof types. Additionally, the dataset must be sufficiently large to support the data-intensive nature of diffusion models.

RQ2: What strategies can be employed to integrate heightmaps and additional contextual information into the diffusion process to enhance completion fidelity?

Diffusion models excel at generating high-quality outputs by iteratively refining noisy data, but their performance in roof heightmap completion depends on effectively incorporating input heightmaps and contextual information. This sub-question investigates strategies for conditioning the diffusion model on sparse or corrupted 2D heightmaps derived from lidar data, alongside additional contextual cues, to enhance the fidelity of the reconstructed roof point clouds and ensure alignment with real-world roof characteristics.

RQ3: How can significant variations in building size be handled to ensure consistent reconstruction accuracy?

Buildings often exhibit significant variability in size, posing a challenge for diffusion models in maintaining consistent reconstruction accuracy across scales. This sub-question explores methods to ensure that the proposed diffusion framework can effectively handle scale disparities in building heightmaps, producing detailed roof point cloud completions regardless of building footprint size. Addressing this issue is essential for real-world applicability, as urban environments contain a mix of buildings with diverse sizes and roof geometries. Large buildings may require higher-resolution heightmaps, increasing computational demands, while small buildings may suffer from insufficient detail in sparse lidar data.

1.3. Contributions

This thesis makes several key contributions to the field of 3D building model reconstruction, specifically focusing on diffusion-based roof heightmap completion from sparse lidar data. The main contributions are:

- **A Novel Dual-Task Diffusion Framework for High-Fidelity Roof Completion.** This thesis proposes a novel framework that jointly performs heightmap completion and roofline prediction. Although rooflines provide powerful structural guidance, they are challenging to acquire directly from raw data, often requiring advanced processing or being unavailable altogether. By framing roofline prediction as a learnable, intermediate task that is jointly optimized with heightmap completion, the proposed approach circumvents the need for pre-existing roofline data. The core of this framework is a custom-designed Bidirectional Control Module (BCM) that facilitates mutual information exchange between the two tasks during the diffusion process. This joint optimization strategy leverages explicit structural guidance from the predicted rooflines to significantly enhance the geometric accuracy and structural integrity of the completed heightmaps, particularly for complex roof geometries and under severe data corruption.
- **A Scalable Patch-Based Diffusion Model for High-Resolution Heightmap Upsampling.** To address the challenge of handling large and variably-sized buildings, a patch-based upsampling diffusion model is developed. This model efficiently generates high-resolution heightmaps from low-resolution inputs by processing them in smaller, manageable patches. A key innovation is the integration of a domain-specific global context encoder, based on a ResNet architecture, which ensures global structural consistency across patches. This approach overcomes the computational limitations of traditional methods and enables consistent, high-quality reconstruction across different building scales without introducing boundary artifacts.
- **A Rigorous Data Curation Pipeline and New Benchmarks for Roof Reconstruction.** This work introduces a systematic data preparation pipeline to construct a large-scale, high-quality dataset for training and evaluation. Starting from the Building-PCC dataset, a rigorous filtering process is applied based on quantitative sparsity and incompleteness metrics to select challenging cases. It also details a process for generating precise roofline ground truth labels. Furthermore, two specialized benchmark datasets, S80_i30 and S80_i80, are constructed and released to facilitate standardized evaluation of model robustness against moderate and severe data corruption.

1.4. Thesis Organization

This thesis is structured to systematically present the research on diffusion-based roof heightmap completion, with each chapter building upon the previous one to provide a comprehensive exploration of the topic. The organization of the thesis is as follows:

Chapter 1 (Introduction) outlines the motivation and objectives of the research. It introduces the research questions, highlights the contributions of the work, and provides an overview of the thesis structure to guide the reader.

Chapter 2 (Preliminaries) establishes the foundational concepts necessary for understanding the proposed approach. It covers the theoretical underpinnings of diffusion models, including

the forward and backward processes, noise prediction networks, conditional diffusion frameworks, latent diffusion models, variational autoencoders (VAEs), and training objectives. Additionally, it introduces the 2.5D representation for roof modeling, which is critical to the proposed methodology.

Chapter 3 (Related Work) surveys the existing literature on 2D and 3D completion methods relevant to roof heightmap completion. It discusses 2D completion techniques such as DSM/-DEM inpainting and depth map completion, as well as 3D completion approaches including point cloud, mesh, wireframe, and multimodal methods. The chapter also explores strategies for enhancing model control, optimizing diffusion models, and enabling multi-scale generation. It concludes by identifying a research gap that motivates the proposed approach.

Chapter 4 (Methodology) presents the proposed diffusion-based framework for roof heightmap completion. It details the design of a novel mesh-heightmap-edgemap dataset, the development of a heightmap completion model, a roofline prediction model, and their joint optimization. The chapter also describes a patch-based diffusion up-sampling strategy and defines the evaluation metrics and standards used to assess the framework's performance.

Chapter 5 (Experiment Setup) describes the experimental setup employed to evaluate the proposed framework. It includes details on dataset preparation, training strategies, implementation specifics, and the results obtained. The chapter also presents an ablation study to analyze the contributions of different components of the proposed method and evaluates its performance across various datasets.

Chapter 6 (Limitations and Future Work) discusses the limitations of the proposed approach and outlines potential directions for future research to address these challenges. It summarizes the key findings and contributions of the thesis, and provides a critical reflection on the work and suggests avenues for further improvement.

2. Preliminaries

This chapter provides the theoretical foundations for the diffusion-based generative models used in this thesis. It begins with an introduction to diffusion models and their underlying principles, followed by a discussion of conditional diffusion frameworks and latent diffusion models.

2.1. Diffusion Model

Diffusion models represent a class of generative models that learn to reverse a gradual noising process. They have emerged as powerful tools for generating high-quality data across various domains, including images, audio, and text. The fundamental concept behind diffusion models involves two key processes: a forward process that systematically adds noise to data and a backward process that learns to reverse this corruption (Ho et al., 2021).

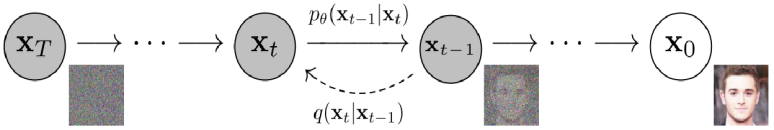


Figure 2.1.: The diffusion process, consisting of a forward noising process and a backward denoising process. Figure from (Ho et al., 2020).

2.1.1. Forward Process: Noise Addition

The forward process, also known as the diffusion process, involves gradually adding Gaussian noise to data samples according to a predefined schedule. Given a data point $\mathbf{x}_0$ sampled from the real data distribution $q(\mathbf{x}_0)$, we define a Markov chain that progressively adds noise over T timesteps:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \tag{2.1}$$

where $\{\beta_t \in (0,1)\}_{t=1}^T$ represents the noise schedule. After sufficient steps, the data point $\mathbf{x}_T$ approaches an isotropic Gaussian distribution $\mathcal{N}(0, \mathbf{I})$, effectively destroying the original information.

A key property of this process is that we can directly sample $\mathbf{x}_t$ at any arbitrary timestep t using:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) \tag{2.2}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$.

2.1.2. Backward Process: Denoising

The backward process, also referred to as the reverse process or denoising process, aims to learn the reverse transitions $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. Starting from pure noise $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$, this process gradually denoises the sample to generate data points resembling the original distribution.

The reverse process is parameterized as:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)) \tag{2.3}$$

Often, for simplicity, $\Sigma_\theta(\mathbf{x}_t, t)$ is fixed to match the forward process variance, and only $\mu_\theta(\mathbf{x}_t, t)$ is learned.

2.1.3. Noise Prediction Network

Instead of directly predicting the mean μ_θ of the reverse process, modern diffusion models often employ a noise prediction network $\epsilon_\theta(\mathbf{x}_t, t)$ that predicts the noise component added during the forward process. This approach has been shown to improve training stability and sample quality.

Given the noise prediction network ϵ_θ, the mean of the reverse distribution can be expressed as:

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right) \tag{2.4}$$

The sampling process typically employs either the deterministic DDIM sampler or stochastic DDPM sampler to generate samples from noise.

2.1.4. Conditional Diffusion Framework

Conditional diffusion models extend the standard denoising diffusion probabilistic models by incorporating conditioning information $\mathbf{c}$ (e.g., class labels, text embeddings, structural priors), enabling controlled and guided generation. The conditional noise prediction network is modified as:

$$\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}),$$

where $\mathbf{x}_t$ is the noisy input at timestep t, and $\mathbf{c}$ is the conditioning signal.

Several strategies exist for introducing conditional information:

- **Concatenation (Early Fusion)**: The conditioning $\mathbf{c}$ is concatenated with $\mathbf{x}_t$ or intermediate features along the channel dimension. This approach is simple and computationally efficient, but may struggle when $\mathbf{c}$ and $\mathbf{x}_t$ are semantically mismatched or structurally different.
- **Cross-Attention (Late Fusion)**: Conditioning is injected via cross-attention modules placed within the U-Net. This allows the network to attend selectively to relevant features in $\mathbf{c}$, enabling flexible and fine-grained control. It is particularly effective for complex or structured conditions (e.g., text, edge maps), but incurs higher computational cost.
- **Feature-wise Linear Modulation (FiLM)**: The network modulates intermediate features by applying affine transformations conditioned on $\mathbf{c}$:

  $$\text{FiLM}(\mathbf{h}) = \gamma(\mathbf{c}) \cdot \mathbf{h} + \beta(\mathbf{c}),$$

  where $\mathbf{h}$ is a hidden feature map. FiLM provides a lightweight and general mechanism for conditioning, suitable for scalar or low-dimensional inputs, but may lack the expressiveness needed for rich structured data.
- **Classifier Guidance**: During sampling, a pretrained classifier provides gradients to steer the denoising process toward samples consistent with a target class. While effective, this method is limited to inference and requires a separate classifier, making it less flexible and harder to scale.
- **Classifier-Free Guidance**: The model is trained jointly with and without $\mathbf{c}$, allowing guided sampling by interpolating between the two predictions (Ho and Salimans, 2022):

  $$\epsilon_{\text{guided}} = (1 + w)\, \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}) - w\, \epsilon_\theta(\mathbf{x}_t, t),$$

  where w is a guidance strength parameter. This method is simple, effective, and does not require additional models, making it widely adopted in modern diffusion systems.

The conditional sampling step proceeds as:

$$\mathbf{x}_{t-1} = \mu_\theta(\mathbf{x}_t, t, \mathbf{c}) + \sigma_t \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I}),$$

where μ_θ is computed based on the conditional noise estimate. The choice of conditioning mechanism should consider the nature of $\mathbf{c}$, model complexity, and the trade-off between expressivity and efficiency.

2.1.5. Training Objective

The training objective for diffusion models is typically derived from a variational bound on the negative log-likelihood of the data. Unlike the conventional ϵ-prediction approach, which directly predicts the noise ϵ added during the forward process, the *v-prediction* parameterization redefines the prediction target as a variable v. V-prediction is the combination of the noise ϵ and the clean data $\mathbf{x}_0$, offering improved numerical stability and training efficiency (Ho et al., 2021). The v-parameter is defined as:

$$v = \alpha_t \epsilon - \sigma_t \mathbf{x}_0,$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is the Gaussian noise added to the clean data $\mathbf{x}_0$, $\alpha_t = \sqrt{\bar{\alpha}_t}$ and $\sigma_t = \sqrt{1 - \bar{\alpha}_t}$ are coefficients determined by the noise schedule, and $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \epsilon$ is the noisy sample at timestep t.

For standard diffusion models, the simplified v-prediction objective is:

$$\mathcal{L}^v_{\text{simple}} = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[\|v - v_\theta(\mathbf{x}_t, t)\|_2^2\right], \tag{2.5}$$

where $v_\theta(\mathbf{x}_t, t)$ is the model's prediction of v, and t is uniformly sampled from $\{1, 2, \ldots, T\}$.

For conditional diffusion models, which incorporate a conditioning variable $\mathbf{c}$ (e.g., class labels or text embeddings), the v-prediction objective becomes:

$$\mathcal{L}^v_{\text{cond}} = \mathbb{E}_{t,\mathbf{x}_0,\epsilon,\mathbf{c}}\left[\|v - v_\theta(\mathbf{x}_t, t, \mathbf{c})\|_2^2\right]. \tag{2.6}$$

For Latent Diffusion Models (LDMs) (Rombach et al., 2022), the objective is adapted to operate in a latent space, where the data $\mathbf{x}_0$ is encoded into a latent representation $\mathbf{z}_0 = \mathcal{E}(\mathbf{x}_0)$ via an encoder $\mathcal{E}$. The v-prediction objective is:

$$\mathcal{L}^v_{\text{LDM}} = \mathbb{E}_{t,\mathbf{z}_0,\epsilon,\mathbf{c}}\left[\|v - v_\theta(\mathbf{z}_t, t, \mathbf{c})\|_2^2\right], \tag{2.7}$$

where $\mathbf{z}_t = \alpha_t \mathbf{z}_0 + \sigma_t \epsilon$ is the noisy latent at timestep t, and $v_\theta(\mathbf{z}_t, t, \mathbf{c})$ predicts v in the latent space.

In practice, timestep-dependent weighting schemes or specialized loss functions may be applied to the v-prediction objectives to improve training stability and enhance the quality of generated samples. These adjustments often prioritize specific timesteps or modify the loss to better align with the generative process (Ho et al., 2021).

2.2. 2.5D Representation for Roof Model

A 2.5D heightmap, often used interchangeably with a digital elevation model (DEM) in geospatial applications, represents a surface by assigning a single height value to each point within a two-dimensional grid (Weibel and Heller, 1992). In this thesis, a 2.5D heightmap specifically captures the elevation of roof surfaces above a reference plane (the lowest height of the corresponding building model) across a regular grid of spatial coordinates. Formally, this can be defined as a function $h : \mathbb{R}^2 \to \mathbb{R}$, where $h(x, y)$ represents the roof height at coordinates (x, y). This grid is typically discretized with a fixed resolution, striking a balance between the level of detail and computational efficiency.

The adoption of 2.5D heightmaps in this thesis is motivated by their practicality and computational efficiency in the context of diffusion-based roof heightmap completion. Real-world buildings are often effectively represented as 2.5D models, which can capture roof geometry while accommodating "jump edges"—abrupt height changes at boundaries like walls or roof edges (Biljecki et al., 2015). These jump edges are naturally aligned with the heightmap format, making it a suitable choice for processing building data derived from sources such as lidar or stereo imagery (Rottensteiner, 2005) (see Figure 2.2). Compared to full 3D diffu-

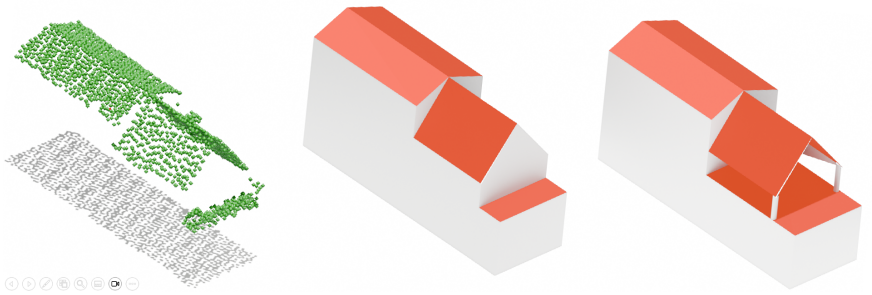


Figure 2.2.: Illustration of 2.5D representation and its limitations. From left to right: a 2.5D points, the corresponding 2.5D building model, and a 3D model with overhangs that cannot be represented in 2.5D.

sion methods, which are computationally demanding and complex to implement due to their handling of volumetric data, heightmap diffusion operates on a simpler scalar field. However, it is important to acknowledge the limitations of the 2.5D format. It cannot represent complex features like overhangs or multi-layered structures, which necessitate full 3D modeling (see Figure 2.2). For the specific objectives of this thesis, the 2.5D heightmap is selected as the primary representation due to its advantageous balance of simplicity, expressiveness, and compatibility with diffusion-based completion tasks.

In this work, the heightmap is represented using unsigned 16-bit integer (uint16) precision, where values range from 0 to 65,535. This range corresponds to physical heights of 0 to 255 meters, with each meter subdivided into 255 discrete bins, yielding a resolution of approximately 3.92 mm per bin. This representation is sufficient to model most building roofs, as typical urban structures rarely exceed 255 meters in height, and the fine granularity of 255 bins per meter accommodates precise height variations. The uint16 format optimizes memory usage and computational efficiency, making it practical for diffusion-based processing on large-scale datasets. This 2.5D representation is particularly well-suited for diffusion-based methods because it supports Image-based operations, such as those employed in convolutional neural networks or denoising diffusion processes.

3. Related Work

This chapter provides a comprehensive review of the existing literature relevant to the research presented in this thesis. It will delve into various techniques for 2D and 3D reconstruction, with a particular focus on their application to roof completion and heightmap generation. The discussion will also cover strategies for enhancing model control and enabling large-scale, multi-scale generation. Furthermore, an examination of existing datasets pertinent to building and roof reconstruction will be undertaken, and key research gaps in the current state-of-the-art will be identified. The following table provides an initial classification of reconstruction methods, which will be expanded upon in the subsequent sections.

Table 3.1.: Classification of Reconstruction Methods

Dimension	Category	Method
2D	DSM/DEM Inpainting	Linear interpolation
		IDW interpolation (Shepard, 1968)
		Spline fitting (Keys, 1981)
	Learning-based Methods	CNN approaches (Lee et al., 2022)
		Adversarial training frameworks (Tsuji et al., 2018; Khan et al., 2021)
	Diffusion-based Methods	Palette framework (Saharia et al., 2022b)
		RoofDiffusion (Lo et al., 2024)
	Depth Map Completion	SparseDC (Long et al., 2024)
		Self-supervised methods (Cai et al., 2023)
		Scale Propagation Network (Wang et al., 2025)
		Few-shot diffusion (Zheng et al., 2023)
3D	Point Cloud Completion	3D Convolution-based (Wu et al., 2020; Yuan et al., 2018)
		Transformer frameworks (Yu et al., 2021, 2023)
		SeedFormer (Zhou et al., 2022)
		Parameterized planes (Chen et al., 2025)
		GAN-based approaches (Xie et al., 2021)
		PointDif (Luo and Hu, 2021)
		3DQD (Li et al., 2023b)
	Mesh Completion	DiffComplete (Chu et al., 2023)
	Wireframe Generation	Traditional wireframe (Zou et al., 2020)
		PBWR (Cheng et al., 2022)
		Point2Building (Liu et al., 2024b)
Multimodal	Fusion Methods	Image-point cloud fusion (Zhang et al., 2021)
		SDFusion (Cheng et al., 2023)
		Conditional IMLE (Arora et al., 2021)

3.1. 2D Completion Methods

2D depth map completion and generation has been extensively studied in computer vision, with approaches ranging from traditional interpolation methods to advanced deep learning techniques.

3.1.1. DSM/DEM Inpainting

Inpainting of Digital Surface Models (DSMs) and Digital Elevation Models (DEMs) is a critical area in geomatics and geoinformation fields, focusing on filling missing regions in heightmaps. Traditional interpolation methods, such as linear interpolation, IDW (Shepard, 1968), and spline fitting (Keys, 1981), rely on geometric principles to estimate missing values. These methods are effective for small voids but struggle with large-scale occlusions or complex structural patterns due to their limited capacity to capture boundary geometry and structural dependencies. For example, when reconstructing building rooftops with complex architectural features like dormers, chimneys, or multi-level structures, traditional interpolation methods often produce overly smooth surfaces that fail to preserve sharp edges and geometric details. In cases where large portions of a building are occluded by vegetation or shadows, these methods may create unrealistic height transitions that do not respect the underlying building geometry.

Learning-based methods utilize deep neural networks to learn complex spatial patterns from data. Recent studies (Tsuji et al., 2018) have employed adversarial training frameworks to learn high-dimensional feature representations from large-scale heightmap datasets, outperforming traditional interpolation in handling severe sparsity and noise (Khan et al., 2021). However, these methods primarily target natural terrain completion, with less focus on the challenges of urban DSM restoration, such as extreme sparsity, tree occlusion, and fine structural details.

Diffusion-based methods have introduced a new paradigm for image completion. The Palette framework (Saharia et al., 2022b) established foundational techniques for image-to-image translation through sophisticated diffusion processes, demonstrating exceptional performance in inpainting, super-resolution, and translation. It introduced critical guidance mechanisms to handle complex, irregular missing regions while maintaining structural coherence. Building on this, RoofDiffusion (Lo et al., 2024) proposes a specialized diffusion-based framework for digital surface model completion. This method employs a conditional diffusion model that leverages building footprints to achieve strong performance in severely corrupted scenarios, handling up to 99% point sparsity and 80% regional incompleteness. Its integration of multi-Gaussian masking and tree modeling enhances robustness by synthesizing realistic noise patterns, making it particularly effective for corrupted roof heightmaps.

3.1.2. Depth Map Completion

Depth completion aims to predict dense per-pixel depth maps from sparse depth measurements, with applications in autonomous driving, 3D reconstruction, and robotics. Early methods, such as those using CNNs (Lee et al., 2022), adapted architectures to handle irregular point distributions, while continuous confidence masks (Truong Giang et al., 2021) provided more nuanced input pixel weighting compared to binary masks. Multi-stage approaches incorporating semantic features, such as surface normals and RGB information, further improved accuracy

(Nazir et al., 2022). However, most methods target depth maps from a driver's perspective, which are typically uniformly distributed, sparse, and lack large missing areas, differing from the characteristics of airborne lidar data.

Recent research has addressed more challenging scenarios. SparseDC (Long et al., 2024) is designed for completing sparse, non-uniform depth maps by employing a Stable Feature Filling Mechanism (SFFM) to fill unstable depth features with stable image features, making it suitable for real-world low-quality depth maps. Self-supervised methods, such as those by Cai et al. (2023), leverage 3D-aware features and multi-view geometric consistency to achieve high-accuracy completion without ground truth data, which is ideal for scenarios with limited annotations. The Scale Propagation Network (Wang et al., 2025) enhances model generalization through a novel architecture based on SP-Norm and a ConvNeXt V2 backbone, achieving state-of-the-art performance across various sparse depth map types.

Diffusion models have also been applied to depth completion. A few-shot learning paradigm proposed by Zheng et al. (2023) utilizes pre-trained denoising diffusion probabilistic models, demonstrating effective completion with limited training samples (e.g., 12.5% of the KITTI dataset), making it suitable for data-scarce scenarios.

3.1.3. Applicability in Roof Completion Tasks

Recent literature has increasingly recognized the potential of diffusion models for architectural reconstruction tasks. The seminal work by Saharia et al. (2022b) established diffusion models as state-of-the-art for image inpainting, denoising, and super-resolution, demonstrating their capability to handle incomplete and noisy datasets through iterative refinement processes. This foundational work has inspired subsequent research to explore diffusion models for specialized reconstruction tasks, including roof heightmap completion.

The choice of 2D heightmap representation over full 3D approaches in roof reconstruction literature reflects a pragmatic balance between computational feasibility and representational adequacy. Biljecki et al. (2015) demonstrated that most real-world building applications can be effectively handled using 2.5D models with "jump edges," establishing a theoretical foundation that has influenced subsequent research directions. This representation choice has become prevalent in the roof reconstruction literature, as evidenced by works such as RoofDiffusion (Lo et al., 2024), which adopts heightmap-based approaches for practical deployment considerations.

The application of diffusion models to roof reconstruction remains a relatively unexplored area, with limited prior work available for comparison. Lo et al. (2024) in RoofDiffusion represents the primary example of diffusion-based roof completion, explicitly adopting heightmap representation for their framework. This choice contrasts with earlier non-diffusion approaches like Ren et al. (2021); Qian et al. (2021), which focused on small-scale 3D mesh reconstruction. The scarcity of diffusion-based roof reconstruction methods makes it difficult to identify clear trends in representation choices, though RoofDiffusion's heightmap approach provides a valuable reference point for understanding the potential and limitations of 2D representations in this domain.

3.2. 3D Completion and Generation Methods

Heightmaps, often represented as 2.5D grids with height values, share conceptual similarities with 3D representations like point clouds and meshes, particularly in handling incomplete or noisy data. This section reviews existing literature on 3D completion and generation, categorizing methods by data format and exploring how these techniques can inspire heightmap completion strategies.

3.2.1. Point Cloud Completion Methods

Point cloud completion involves reconstructing complete 3D shapes from partial or sparse point cloud data, a task with direct relevance to heightmap completion due to the shared goal of filling missing regions while preserving geometry. Key approaches include:

3D Convolution-based Methods: These methods convert point clouds into structured representations, such as voxel grids or distance fields, to apply 3D convolutions (Wu et al., 2020). While offering high accuracy, they are computationally intensive and may struggle with fine details (Yuan et al., 2018). For heightmaps, this suggests potential adaptations using 2D convolutions on grid representations, though computational costs remain a concern.

Transformer Frameworks: Recent developments, such as those by Yu et al. (2021, 2023), use anchor points to guide the upsampling process, integrating local and global features with attention mechanisms. This approach could be adapted to heightmaps by identifying key structural points, like building corners, to anchor the completion process. SeedFormer (Zhou et al., 2022) introduces a novel shape representation called Patch Seeds, designed to capture both global structures and local patterns from partial point clouds. The Upsample Transformer extends the transformer architecture by incorporating fundamental point generation operations, enhancing spatial and semantic relationships to preserve fine details during coarse-to-fine completion. Chen et al. (2025) propose to represent planar point clouds using parameterized planes, and achieves shape completion indirectly by predicting the combination and segmentation of proxy planes via a Transformer-based network.

GAN-based Approaches: Generative Adversarial Networks (GANs) have been applied to point cloud completion, leveraging the discriminator to evaluate completion quality (Xie et al., 2021). However, these methods may produce multiple interpretations with varying styles, making them less suitable for applications requiring high geometric accuracy, such as building reconstruction. For heightmaps, this suggests caution in using GANs for tasks needing precise structural fidelity (Chen et al., 2020).

Diffusion-based Approaches: Diffusion models have gained attention for their ability to iteratively denoise and complete point clouds by learning a probabilistic distribution of 3D shapes (Luo and Hu, 2021; Li et al., 2023b). Luo and Hu (2021) were the first to propose sampling point clouds to a fixed number of points and applying a diffusion model for denoising-based generation (Diffusion Probabilistic Models for 3D Point Cloud Generation). PointDif introduces a conditional point generator, which aggregates features extracted by the backbone and uses them as conditions to guide point-to-point recovery from noisy point clouds. This process helps the backbone capture both local and global geometric priors, as well as the global point density distribution of the object. 3DQD leverages a Vector Quantized Variational Autoencoder (VQ-VAE) to encode local geometric details, and employs a discrete diffusion

model for shape generation, which improves 3D generation accuracy while reducing training complexity.

3.2.2. Mesh Completion and Generation Methods

Mesh-based methods focus on completing or generating 3D meshes, which provide continuous surface representations, offering insights for heightmap completion where surface continuity is crucial. A notable example is DiffComplete (Chu et al., 2023), which uses diffusion models to generate a complete 3D model represented by a truncated signed distance field (TSDF). This approach highlights the versatility of diffusion models across data forms, suggesting that similar techniques could be applied to heightmaps to ensure smooth transitions in completed regions. However, the conversion between heightmaps and meshes may introduce additional complexity, limiting direct applicability but offering inspiration for surface-based completion strategies.

3.2.3. Wireframe Generation Methods

Wireframe models represent 3D structures using edges and vertices and are particularly useful for architectural applications like building reconstruction, which aligns with the goals of heightmap completion. For instance, Zou et al. (2020) generate wireframe models from partial images or other inputs, capturing structural semantics. This approach can inspire heightmap completion by first predicting structural elements, such as building edges or rooflines, and then using these as priors to guide the filling of missing regions. PBWR (Cheng et al., 2022) introduces an end-to-end framework for 3D building wireframe reconstruction that directly regresses edge parameters from aerial lidar point clouds by leveraging the self-attention mechanism of Transformers. A more recent generative approach, Point2Building (Liu et al., 2024b), reconstructs polygonal building meshes from airborne LiDAR by autoregressively generating sequences of vertices and faces, allowing it to adapt to diverse geometries by learning directly from raw point cloud data.

3.2.4. Multimodal Completion Methods

Multimodal approaches integrate data from different sources, such as images and point clouds, to leverage complementary information. Methods proposed by Zhang et al. (2021) combine single or multiple view images with partial point clouds, enhancing completion accuracy by incorporating texture and semantic cues from the images. Further advancing this, SDFusion (Cheng et al., 2023) presents a diffusion-based framework that supports various input modalities, including images, text, and partial shapes. It uses an encoder-decoder architecture to handle shape completion and reconstruction tasks flexibly. Another notable technique is the use of conditional Implicit Maximum Likelihood Estimation (IMLE) to generate diverse and complete shapes from partial inputs, which avoids the mode collapse issue often seen in GANs (Arora et al., 2021). This suggests that for heightmap completion, integrating satellite imagery, aerial photos, or building footprints with height data could improve results, drawing from the success of multimodal 3D completion. The primary challenge lies in aligning different data modalities and ensuring consistency, but the potential for enhanced accuracy is significant, especially in scenarios with limited heightmap data.

3.2.5. Inspirations for Heightmap Completion

The reviewed 3D completion and generation methods provide several inspirations for heightmap completion, addressing challenges like sparsity, noise, and structural coherence:

Diffusion Models: Their success in point cloud and mesh completion suggests adapting them to heightmaps, treating the maps as 2D grids for probabilistic completion. This could enhance robustness, as seen in RoofDiffusion for DSM completion, by modeling complex patterns and maintaining structural integrity.

Multimodal Integration: Inspired by multimodal 3D methods, combining heightmaps with other data sources, such as satellite imagery or building footprints, can leverage complementary information. This approach could address data scarcity and improve completion in urban environments with occlusions.

Structural Priors: Wireframe generation techniques suggest that predicting structural elements like building edges from partial heightmaps can provide strong priors for accurate completion. This could ensure that completed regions respect architectural constraints, enhancing practical applicability.

These inspirations highlight the potential for cross-pollination between 3D and heightmap completion, although challenges like computational efficiency and domain-specific adaptations require further exploration.

3.3. Strategies for Enhancing Model Completion and Control

3.3.1. Controlling the Generative Result

Control methods are critical for guiding the generation process of diffusion models through additional conditions, thereby improving the controllability of the output. ControlNet (Zhang et al., 2023) enhances pre-trained text-to-image diffusion models with conditions like edge and depth maps, boosting spatial consistency. Imagic (Kawar et al., 2022) offers a text-prompt-based method for real image editing that supports complex semantic adjustments, such as local height changes, making it ideal for non-rigid edits.

3.3.2. Optimization of Diffusion Models

Image inpainting aims to restore missing regions of an image seamlessly. For example, RePaint (Lugmayr et al., 2022) employs a pre-trained unconditional denoising diffusion probabilistic model (DDPM) with a resampling mechanism to optimize the generation process, effectively addressing extreme mask scenarios and enabling free-form inpainting. Similarly, StrDiffusion (Liu et al., 2024a) introduces a structure-guided texture denoising approach, enhancing large-hole inpainting by ensuring semantic consistency between masked and unmasked regions. In 3D shape completion, diffusion models also show remarkable potential. DiffComplete (Chu et al., 2023) presents a diffusion-based method optimized for sparse point clouds that generates diverse, high-fidelity 3D shapes. Meanwhile, ComPC (Huang et al., 2025) proposes a zero-shot framework that leverages pre-trained 2D diffusion priors and Gaussian splatting to achieve cross-category 3D point cloud completion without additional training.

Multi-task optimization seeks to extend diffusion models across related tasks or improve performance by integrating complementary techniques. Difftsr (Zhang et al., 2024b) applies diffusion models to blind text image super-resolution using a dual diffusion mechanism with cross-attention. SGDM (Horita et al., 2023) tackles large-hole image completion with a cascaded approach to structure and texture generation. Furthermore, Cho et al. (2025) introduce a feature disentanglement training framework that enhances controllability by separating spatial content masks and style embeddings.

3.4. Enabling Multi-Scale and Large-Scale Generation

In generative models, particularly those based on diffusion and U-Net architectures, handling large-scale input images presents a significant challenge. As image size increases, the computational complexity rises substantially: the complexity of convolution operations scales linearly with image size ($H \times W$), while the complexity of global self-attention mechanisms can reach $O((H \times W)^2)$, leading to a surge in memory and computational costs.

3.4.1. Patch-Based Processing

Patch-based methods decompose large images into smaller blocks to reduce computational complexity. This approach allows models to operate on local regions, significantly decreasing memory requirements and computational load. Wang et al. (2023) introduced Patch Diffusion, a random block-based training method for diffusion models that enables independent or collaborative denoising operations on individual patches. By processing images in smaller chunks, the method accelerates training and improves data efficiency, making it particularly suitable for high-resolution image generation. A key advantage of patch-based processing is its modularity, allowing seamless integration with existing U-Net architectures without structural modifications. Ding et al. (2023) propose Patch-DM, an effective denoising diffusion model for generating high-resolution images (e.g., 1024×512) while training on small image patches (e.g., 64×64). The algorithm features a novel feature window patching strategy to avoid boundary artifacts when synthesizing large images. This strategy systematically crops and combines partial features from adjacent patches to predict features for offset image patches, enabling seamless generation of the entire image.

3.4.2. Architectural Optimizations

Efficient neural network architectures can optimize attention mechanisms or replace traditional convolution operations to reduce the computational complexity of processing large images. The Swin Transformer (Liu et al., 2021) implements a shifted window mechanism that achieves local self-attention, reducing the quadratic complexity of traditional global self-attention to linear. When integrated as a backbone in U-Net architectures, the Swin Transformer excels at processing high-resolution images. Its hierarchical structure enables feature capture at various scales, reducing computational costs while preserving generation quality. Further architectural optimization is demonstrated in the Sana framework (Xie et al., 2024), which employs linear attention mechanisms in Diffusion Transformers (DiT). This approach approximates global attention while reducing computational overhead, maintaining the ability to generate

high-quality images and making high-resolution generation more accessible on consumer-grade GPUs.

3.4.3. Latent Space Diffusion

Latent space diffusion methods significantly reduce computational demands by performing the diffusion process on compressed representations rather than in pixel space. Rombach et al. (2022) proposed Latent Diffusion Models (LDMs), which run the diffusion process in the latent space of a pretrained autoencoder. Images are first compressed into a lower-dimensional latent space where the diffusion process is executed. This method substantially reduces resource requirements while preserving image details through the autoencoder's reconstruction capabilities. LDMs have demonstrated strong performance in tasks including image inpainting, unconditional image generation, and super-resolution. Additionally, LDMs support conditional inputs such as text or bounding boxes through cross-attention layers, establishing them as versatile generative frameworks.

3.4.4. Tuning-Free High-Resolution Generation

Tuning-free methods allow pretrained models to generate images at resolutions higher than those used during training, avoiding expensive retraining. He et al. (2023) introduced ScaleCrafter, a method that dynamically adjusts the receptive field of convolution kernels in a U-Net to enable diffusion models trained on low-resolution images to generate high-resolution outputs. The method addresses object repetition and structural inconsistency issues in high-resolution generation through techniques including re-dilation, dispersed convolution, and noise-suppressed classifier-free guidance. Similarly, Zhang et al. (2024a) proposed HiDiffusion, which enables models trained on low-resolution data to generate high-resolution images through specific inference-time techniques, optimizing the diffusion process for computational efficiency.

3.5. Building and Roof-Related Datasets

The reconstruction of architectural structures requires point cloud data with corruption and corresponding ground truth representations, such as meshes, for training. Wichmann et al. (2018) introduced the RoofN3D dataset, which contains point clouds and automatically generated building models. However, the automatically generated models are not sufficiently detailed, which is a major limitation. While Ren et al. (2021); Qian et al. (2021) built custom roof completion datasets, these focus on small buildings and a limited number of examples.

Currently, the Poznan3D dataset, used by the authors of RoofDiffusion, was created by filtering 13k noise-free complex building models and introducing simulated noise to induce point cloud degradation. However, this approach may not sufficiently represent the diversity of real-world noise, potentially reducing the generalization capabilities of the model.

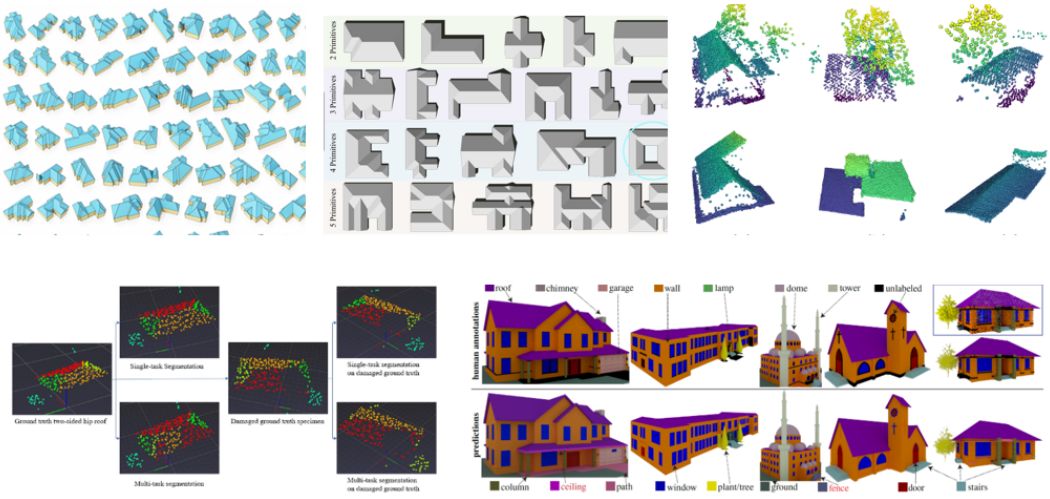


Figure 3.1.: Examples of building and roof datasets. Top row showcases datasets from (Ren et al., 2021; Qian et al., 2021) and the Poznan3D dataset from RoofDiffusion (Lo et al., 2024). Bottom row displays examples from the RoofN3D (Wichmann et al., 2018) and BuildingNet (Selvaraju et al., 2021) datasets.

3.6. Research Gaps

Through the comprehensive review of existing methods, several critical research gaps emerge that limit the current state-of-the-art in roof heightmap completion.

Scalability Limitations in High-Resolution Processing The review of multi-scale generation methods reveals that while techniques like patch-based processing and architectural optimizations show promise, their application to roof completion remains limited. Current diffusion-based methods, including RoofDiffusion, operate on fixed image sizes and face computational bottlenecks when processing high-resolution heightmaps. The analysis of attention mechanisms demonstrates that as image dimensions increase, computational complexity escalates from linear for convolution operations to quadratic ($O((H \times W)^2)$) for self-attention mechanisms. Despite advances in Swin Transformer architectures and latent space diffusion that address these computational challenges in general image generation, roof completion methods have not yet incorporated these scalability solutions. This gap becomes particularly pronounced when considering large-scale urban datasets or complex architectural structures that require higher resolution processing, directly addressing RQ3's focus on handling variable image sizes.

Insufficient Control Mechanisms for Guided Generation The examination of control strategies reveals a paradox in current roof reconstruction approaches. While general-purpose frameworks like ControlNet demonstrate sophisticated conditional guidance using edge maps and depth information, their adaptation to roof completion faces unique challenges. The review shows that acquiring reliable control inputs for roof structures is inherently difficult due to the noisy and incomplete nature of point cloud data from which heightmaps are derived. Moreover, existing roof completion methods, particularly RoofDiffusion, primarily rely on building footprints as the sole control mechanism, underutilizing the potential of multimodal

guidance. The analysis of 3D completion methods suggests that structural priors from wireframe generation and multimodal integration could significantly enhance control, yet these approaches remain underexplored in roof heightmap completion, highlighting a critical gap related to RQ2's emphasis on improving reconstruction fidelity through enhanced guidance.

Dataset Limitations and Real-World Representation The review of building and roof-related datasets exposes fundamental limitations in current training resources. While datasets like RoofN3D provide point cloud data with corresponding models, the automatically generated ground truth lacks sufficient architectural detail. The Poznan3D dataset, despite containing 13k building models, relies on simulated noise patterns that may not adequately represent real-world data degradation scenarios. This synthetic approach contrasts with the complexity of actual airborne lidar data, where occlusions from vegetation, shadows, and occlusion conditions create diverse corruption patterns. The comparison with depth completion datasets reveals that roof completion datasets lack the scale and diversity needed for robust deep learning model training, directly impacting RQ1's goal of developing effective completion methods for varied building types and noise conditions.

Underutilized Multimodal Integration Potential The review of 3D completion methods demonstrates the effectiveness of multimodal approaches that combine point clouds with images, achieving superior results through complementary information fusion. However, the translation of these successful strategies to roof heightmap completion remains limited. While multimodal integration shows promise in general 3D completion tasks, the specific challenges of aligning heightmaps with auxiliary data sources like satellite imagery, building footprints, or edge maps in roof reconstruction contexts have not been adequately addressed. The analysis suggests that roof completion could benefit significantly from the integration strategies developed for 3D shape completion, yet current methods largely operate in isolation without leveraging complementary data modalities, representing an underexplored avenue for enhancing reconstruction accuracy and robustness as emphasized in RQ2.

4. Methodology

This chapter details the proposed two-stage methodology for heightmap completion, where each stage is trained independently. The overall framework, depicted in Figure 4.1, consists of two main stages: the first stage is a dual-task diffusion model for heightmap completion and roofline prediction, and the second stage is a patch-based diffusion model for high-resolution upsampling.

4.1. Diffusion-Based Roof Completion

This section outlines the proposed framework for diffusion-based roof heightmap completion, focusing on the architectural design and joint optimization strategy. The diffusion principles and training methodologies, as detailed in Chapter 2, are summarized to provide context for the model architecture and integration mechanisms.

Drawing inspiration from the Structure–Guided Diffusion Model (SGDM) (Horita et al., 2023), which jointly models structure and texture maps to enhance structural fidelity for image inpainting, and the DiffTSR framework (Zhang et al., 2024b), which couples a text diffusion model with an image super–resolution diffusion model through a lightweight fusion block to improve textual accuracy, this work adopts a similar dual-task paradigm. In our setting, the Heightmap Completion Diffusion model is responsible for synthesising fine-grained geometric details, whereas the Roofline Prediction Diffusion model provides an explicit structural prior. The proposed Bidirectional Control Module (BCM) acts analogously to the fusion block in DiffTSR, enabling the two diffusion processes to exchange information at every reverse step while keeping the individual model footprints low. Following DiffTSR, both diffusers are first trained independently and are subsequently fine-tuned jointly via the BCM, which mitigates GPU memory overhead without compromising interaction between tasks.

A key element in this framework is the building footprint, which, while simple, serves a critical function. Primarily, it is used to define the area of interest for processing. During training, the footprint acts as a mask to crop the input heightmaps, ensuring the model focuses exclusively on the building structure. Similarly, the loss is calculated only within the masked region defined by the footprint, concentrating the learning process on relevant areas. It is important to note that due to its low intrinsic information content—being a binary representation—the footprint is not concatenated with the model's input channels. Instead, its role is strictly functional, serving as a spatial mask to guide both training and loss computation.

4.1.1. Heightmap Completion Diffusion

The Heightmap Completion Diffusion model reconstructs complete heightmaps from corrupted inputs using a standard diffusion process. This model builds upon the RoofDiffusion baseline (Lo et al., 2024), which employs the Palette (Saharia et al., 2022a) U-Net architecture.

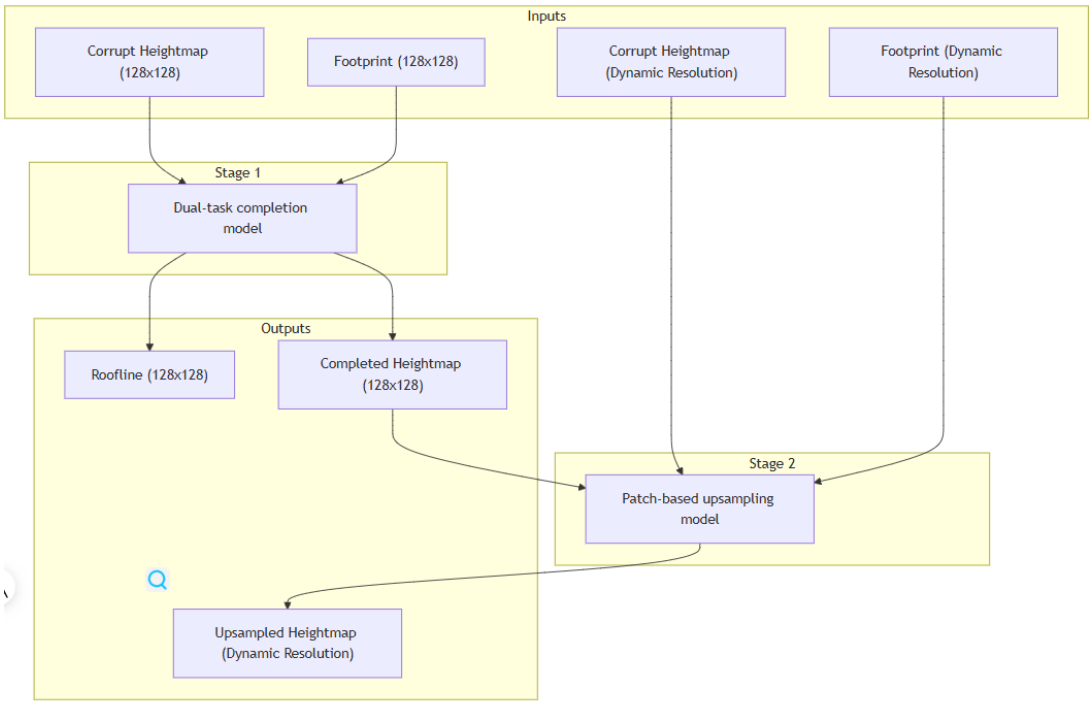


Figure 4.1.: Overview of the proposed framework. The first stage involves a dual-task diffusion model for heightmap completion and roofline prediction, while the second stage employs a patch-based diffusion model for high-resolution upsampling.

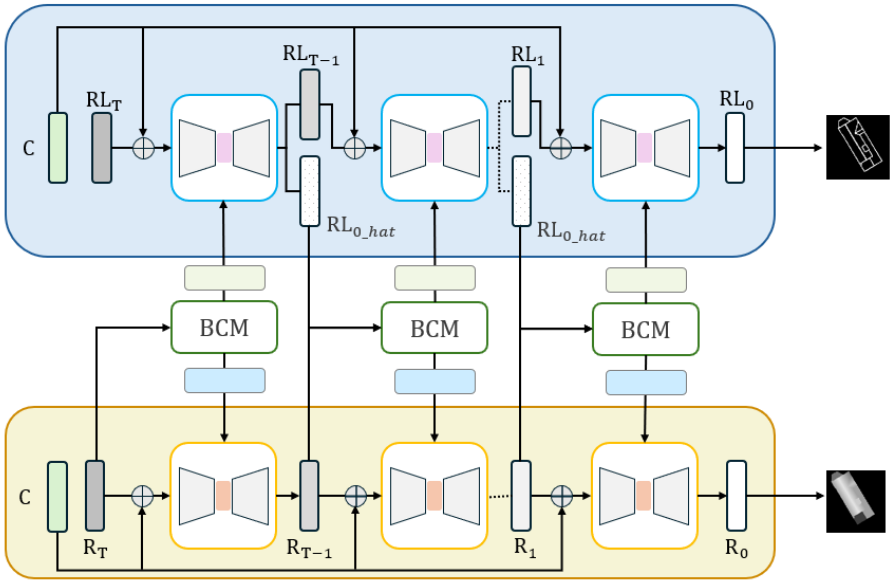


Figure 4.2.: The first stage is the diffusion-based roof completion, which is a dual-task diffusion model for heightmap completion and roofline prediction.

However, the baseline RoofDiffusion operates in isolation and lacks the capability to incorporate complementary information from related tasks such as roofline prediction. To address this limitation, this work adopts a similar Palette-style U-Net structure to capture spatial dependencies, augmented by spatial transformer attention blocks to effectively integrate structural information from roofline predictions during denoising.

Let $\mathbf{R}_0 \in \mathbb{R}^{H\times W}$ represent the ground-truth heightmap, and $\mathbf{C} \in \mathbb{R}^{H\times W}$ denote the corrupted roof input, which serves as a conditioning input. Since the corrupted roof $\mathbf{C}$ contains partial information, the denoising process starts from $\mathbf{C}$ noised to timestep T rather than from pure noise. The forward diffusion process, as defined in Chapter 2, introduces noise to the heightmap, producing a noisy heightmap $\mathbf{R}_t$ at time t.

To enhance the denoising process, a spatial transformer attention mechanism is incorporated to fuse roofline information into the model. This mechanism, detailed in Appendix 3, aligns and integrates spatial features from the corrupted input $\mathbf{C}$ and the intermediate roofline estimate $\hat{\mathbf{RL}}_t$ (obtained from the Roofline Prediction Diffusion model at timestep t), enabling the model to better capture geometric and contextual details. The reverse process is parameterized by a neural network $\epsilon_\theta(\mathbf{R}_t, t, \mathbf{C}, \hat{\mathbf{RL}}_t)$, which conditions on the corrupted roof $\mathbf{C}$ and $\hat{\mathbf{RL}}_t$ to predict the noise component ϵ, with the training objective:

$$\mathcal{L}_{\text{heightmap}} = \mathbb{E}_{\mathbf{R}_0,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(\mathbf{R}_t, t, \mathbf{C}, \hat{\mathbf{RL}}_t)\|^2\right]. \tag{4.1}$$

This produces an intermediate denoised heightmap $\mathbf{R}_t$ at each timestep t with enhanced context to the underlying roof geometry.

4.1.2. Roofline Prediction Diffusion

The Roofline Prediction Diffusion model generates roofline structures from corrupted inputs using a diffusion framework. Let $\mathbf{RL}_0 \in \mathbb{R}^{C\times H\times W}$ denote the ground-truth roofline, and $\mathbf{C} \in \mathbb{R}^{H\times W}$ represent the corrupted roof input, which serves as a conditioning input. Similar to the heightmap completion model, this model is augmented with spatial transformer attention (detailed in Appendix 3) to integrate geometric information from the heightmap completion process. The forward diffusion process, as outlined in Chapter 2, introduces noise to the roofline, producing a noisy roofline $\mathbf{RL}_t$ at time t. The reverse process is parameterized by a neural network $\mathbf{v}_\phi(\mathbf{RL}_t, t, \mathbf{C}, \hat{\mathbf{R}}_t)$, which conditions on the corrupted roof $\mathbf{C}$ and the intermediate heightmap estimate $\hat{\mathbf{R}}_t$, adopting the v-prediction parameterization (Ho et al., 2022), with the training objective:

$$\mathcal{L}_{\text{roofline}} = \mathbb{E}_{\mathbf{RL}_0,\epsilon,t}\left[\|\mathbf{v} - \mathbf{v}_\phi(\mathbf{RL}_t, t, \mathbf{C}, \hat{\mathbf{R}}_t)\|^2\right]. \tag{4.2}$$

To enhance the model's ability to handle imbalanced roofline features, a weighted binary cross-entropy (BCE) loss is incorporated, inspired by methods in edge detection (Liu et al., 2017). This loss addresses the class imbalance between edge and non-edge pixels by assigning different weights to positive and negative samples. A detailed explanation of the loss function and its implementation is provided in Appendix A.2.2. The final training objective combines both the diffusion loss and the weighted BCE loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{roofline}} + \lambda \cdot \mathcal{L}_{\text{cross-entropy}}, \tag{4.3}$$

where $\mathcal{L}_{\text{cross-entropy}}$ represents the weighted BCE loss and λ is a hyperparameter controlling its contribution (set to 0.5 by default). The Roofline Prediction Diffusion model also adopts

a Palette-style U-Net architecture similar to the heightmap completion model, processing the corrupted roof input $\mathbf{C}$ conditioned on the noisy heightmap $\mathbf{R}_t$, yielding a roofline prediction $\mathbf{RL}_t$ at each step, which directly represents the final roofline.

4.1.3. Bidirectional Control Module

The Bidirectional Control Module (BCM) is a separately trained neural network designed to enforce structural consistency between the heightmap completion and roofline prediction models during the joint reverse diffusion process. It functions as a mediator that refines the predictions of the two specialist models at each timestep.

The process at each reverse step t is iterative. First, the Heightmap Completion model and the Roofline Prediction model generate initial, unguided predictions of the clean outputs, $\hat{\mathbf{R}}_0^{\text{initial}}$ and $\hat{\mathbf{RL}}_0^{\text{initial}}$, respectively. These preliminary predictions are then passed as input to the BCM. The BCM, which uses a shared-encoder, dual-decoder architecture, processes these initial estimates and produces refined, more coherent predictions, $\hat{\mathbf{R}}_0^{\text{refined}}$ and $\hat{\mathbf{RL}}_0^{\text{refined}}$.

These refined predictions from the BCM are not the final output for the step. Instead, they are fed back to the specialist models as external conditioning signals. The Heightmap Completion model performs its final denoising calculation for step t conditioned on the BCM's refined roofline ($\hat{\mathbf{RL}}_0^{\text{refined}}$), while the Roofline Prediction model conditions on the refined heightmap ($\hat{\mathbf{R}}_0^{\text{refined}}$). This ensures that the final update for step t is guided by mutually consistent structural information. This iterative refinement process is detailed in Algorithm 1.

The joint optimization objective for finetuning the system with the BCM combines the individual losses:

$$\mathcal{L}_{\text{joint}} = \lambda_1 \mathcal{L}_{\text{heightmap}} + \lambda_2 \mathcal{L}_{\text{roofline}}, \tag{4.4}$$

where λ_1, λ_2 are empirically tuned hyperparameters. The sampling process is detailed in Algorithm 1.

Algorithm 1 Bidirectional Sampling with BCM Refinement

Input: Corrupted heightmap $\mathbf{C}$, timesteps T
Models: Heightmap model ϵ_θ, Roofline model $\mathbf{v}_\phi$, BCM
Initialize $\mathbf{R}_T \sim \mathcal{N}(0, \mathbf{I})$, $\mathbf{RL}_T \sim \mathcal{N}(0, \mathbf{I})$
for $t = T$ to 1 **do**
▷ 1. Get initial unguided predictions
  $\hat{\mathbf{R}}_0^{\text{initial}} \leftarrow \text{Predict}_H(\mathbf{R}_t, t, \mathbf{C})$
  $\hat{\mathbf{RL}}_0^{\text{initial}} \leftarrow \text{Predict}_{RL}(\mathbf{RL}_t, t, \mathbf{C})$
▷ 2. BCM refines the predictions
  $[\hat{\mathbf{R}}_0^{\text{refined}}, \hat{\mathbf{RL}}_0^{\text{refined}}] \leftarrow \text{BCM}(\hat{\mathbf{R}}_0^{\text{initial}}, \hat{\mathbf{RL}}_0^{\text{initial}}, \mathbf{C})$
▷ 3. Perform reverse step using BCM-refined guidance
  $\mathbf{R}_{t-1} \leftarrow \text{ReverseStep}_H(\mathbf{R}_t, t, \mathbf{C}, \text{condition} = \hat{\mathbf{RL}}_0^{\text{refined}})$
  $\mathbf{RL}_{t-1} \leftarrow \text{ReverseStep}_{RL}(\mathbf{RL}_t, t, \mathbf{C}, \text{condition} = \hat{\mathbf{R}}_0^{\text{refined}})$
end for
Output: Completed heightmap $\mathbf{R}_0$, predicted roofline $\mathbf{RL}_0$

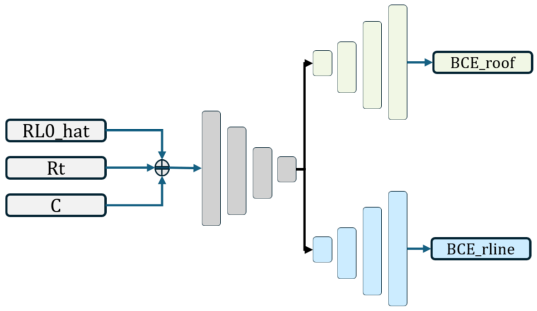


Figure 4.3.: The Bidirectional Control Module (BCM) facilitates information exchange between the heightmap completion and roofline prediction models.

The BCM ensures that the Heightmap Completion Diffusion and Roofline Prediction Diffusion models collaboratively refine their outputs, aligning the completed heightmap with the predicted roofline. This bidirectional approach, inspired by multi-modal diffusion frameworks (Li et al., 2023a), enhances structural fidelity in the roof completion task.

4.2. Patch-Based Up-Sampling

In the context of building heightmap completion and upsampling, patch-based methods offer notable advantages over conventional whole-image techniques. Heightmaps representing buildings vary significantly in scale, requiring varying resolutions to accurately depict intricate details in larger structures. Processing entire high-resolution heightmaps directly, however, imposes substantial computational and memory demands. By adopting a patch-based approach, the workload is divided into smaller, manageable segments, facilitating efficient computation even at high resolutions. Moreover, the use of fixed-size patches allows the model to maintain a consistent level of detail reconstruction across the heightmap, regardless of its overall dimensions. A primary concern with this method is ensuring seamless integration at patch boundaries to prevent visible artifacts. This is addressed through a feature collage strategy that incorporates information from adjacent patches, enhancing both local coherence and global semantic consistency.

4.2.1. Patch-based Denoising Diffusion Model

The Patch-based Denoising Diffusion Model is adapted from the Patch-DM framework (Ding et al., 2023) to generate high-resolution heightmaps by processing fixed-size patches, thereby balancing computational efficiency with reconstruction quality. Unlike the original Patch-DM, which uses low-resolution color images as input, the approach in this thesis modifies the input to be a concatenation of the low-resolution heightmap and the corrupted heightmap. To enhance the upsampling capability and improve reconstruction quality, the low-resolution heightmaps undergo a carefully designed preprocessing strategy. Instead of simple interpolation, Gaussian blur is applied to the low-resolution ground truth (LR GT). This preprocessing serves two purposes: (1) it simulates realistic degradation where low-resolution inputs naturally exhibit smoothness due to limited sampling density, and (2) it creates a more challenging training scenario that encourages the model to learn sophisticated detail generation rather than merely

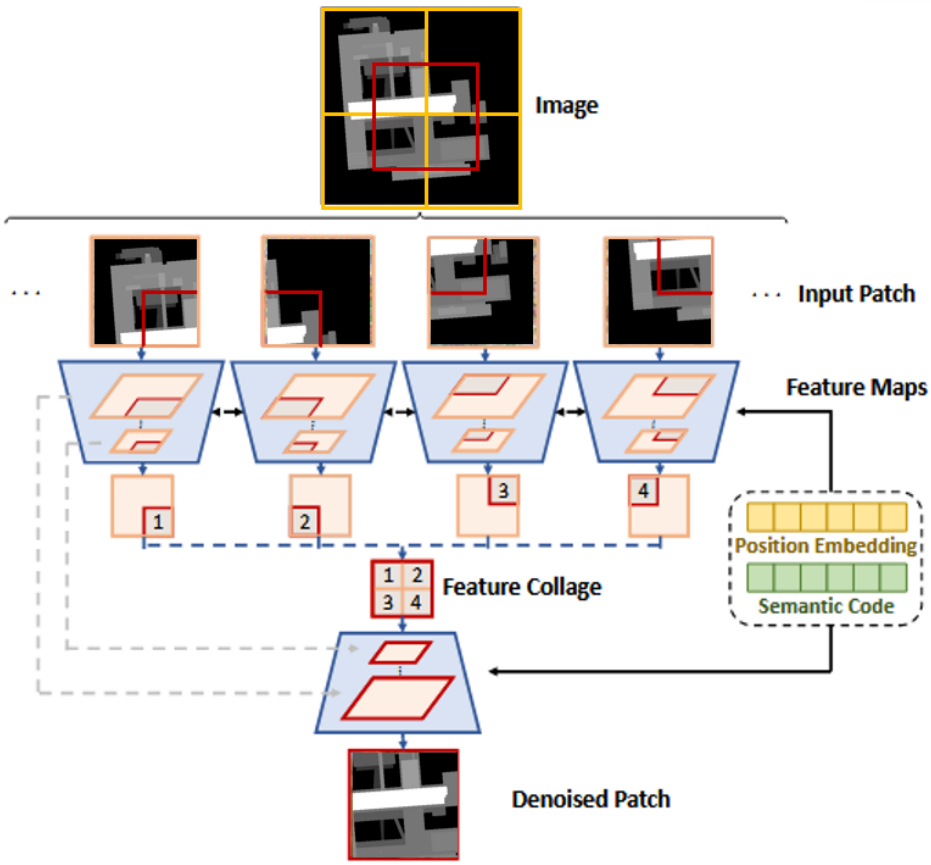


Figure 4.4.: The proposed patch-based denoising diffusion model for heightmap upsampling.

copying high-frequency information from sharp LR inputs. The Gaussian blur operation with a small kernel ($\sigma = 0.8$) removes fine details from the LR GT while preserving major structural information, forcing the diffusion model to reconstruct missing high-frequency details.

This modification enables the model to leverage both complete low-resolution structural information and sparse high-resolution details for enhanced reconstruction. The remaining diffusion process follows the original Patch-DM framework, as illustrated in Figure 4.4. Given a high-resolution heightmap $x_0 \in \mathbb{R}^{C\times H\times W}$, it is segmented into non-overlapping patches $x_0^{(i,j)} \in \mathbb{R}^{C\times h\times w}$, where i and j denote the patch coordinates. Similarly, the low-resolution heightmap is upsampled to the target resolution using a simple interpolation method and divided into corresponding patches $y^{(i,j)}$. $x_0^{(i,j)}$ is then noised to $x_t^{(i,j)}$ at timestep t using the forward diffusion process, and $y^{(i,j)}$ is used as one of the conditioning inputs.

The Patch-DM architecture is built upon a UNet-like framework(see Appendix A.5 for details), enhanced with a feature collage mechanism to manage patch boundaries and incorporate low-resolution conditioning. It comprises an encoder $f_{E\theta}$ and a decoder $f_{D\theta}$, with the following steps (4.5):

1. **Encoding**: Each noisy high-resolution patch $x_t^{(i,j)}$ is paired with its corresponding low-

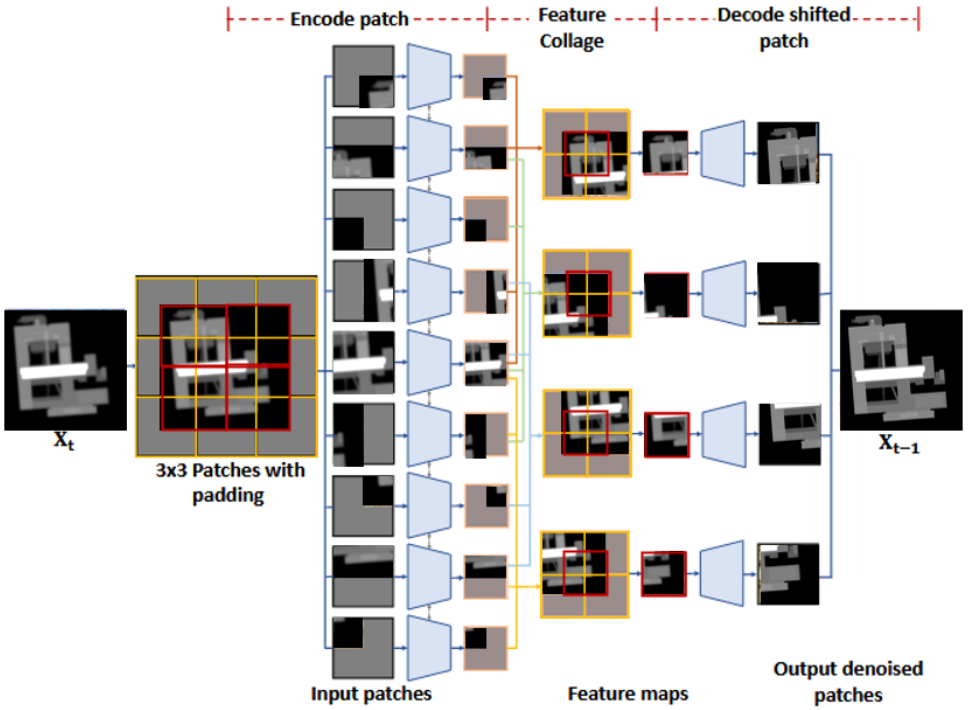


Figure 4.5.: Illustration of the patch division and feature collage mechanism.

resolution patch $y^{(i,j)}$ (preprocessed with Gaussian blur and upsampled via nearest-neighbor interpolation) through channel-wise concatenation and processed by the encoder to yield feature maps:

$$[z_1^{(i,j)}, z_2^{(i,j)}, \ldots, z_n^{(i,j)}] = f_{E\theta}(x_t^{(i,j)}, t), \tag{4.5}$$

where t represents the timestep, $f_{E\theta}$ is the encoder, and $z_k^{(i,j)}$ represents the feature map for the k-th channel.

2. **Feature Collage**: To achieve coherence across patch boundaries, each feature map $z_k^{(i,j)}$ is split into four quadrants. These quadrants are then combined with quadrants from neighboring patches to form a shifted feature map $\hat{z}_k'^{(i,j)}$:

$$\hat{z}_k'^{(i,j)} = \left[P_1(z_k^{(i,j)}), P_2(z_k^{(i,j+1)}), P_3(z_k^{(i+1,j)}), P_4(z_k^{(i+1,j+1)})\right], \tag{4.6}$$

where P_1, P_2, P_3, P_4 extract the top-left, top-right, bottom-left, and bottom-right quadrants, respectively.

3. **Decoding**: The collaged feature maps $\hat{z}_k'^{(i,j)}$ are fed into the decoder to predict the noise for the shifted patch:

$$\epsilon_t'^{(i,j)} = f_{D\theta}([\hat{z}_1'^{(i,j)}, \hat{z}_2'^{(i,j)}, \ldots, \hat{z}_n'^{(i,j)}], t). \tag{4.7}$$

To ensure spatial and contextual consistency, the proposed approach incorporates two critical components: position embeddings and global conditioning. Position embeddings $P^{(i,j)}$ encode the top-left coordinates of each patch, providing spatial location information to help the model correctly position patches relative to each other. For global conditioning, a domain-specific autoencoder is developed and trained on low-resolution heightmaps. Unlike original Patch-DM (Ding et al., 2023), which employs a pre-trained CLIP encoder for natural images, this task requires an encoder tailored to geometric data. Therefore, an autoencoder with a ResNet50-based encoder was trained from scratch. The encoder component of this trained autoencoder is then used as a global condition encoder. It extracts semantic features from the complete low-resolution heightmap, providing overall structural context to each patch during processing. This ensures that local patch reconstructions remain consistent with the global building geometry. Each patch is processed with both its position embedding $P^{(i,j)}$ and the global condition G, enhancing the model's ability to produce globally coherent and spatially accurate outputs.

In summary, the conditioning information for each patch in the upsampling process consists of four components: the corresponding patch from the corrupted heightmap $\mathbf{C}$, the low-resolution heightmap patch $y^{(i,j)}$, the positional embedding $P^{(i,j)}$, and the global conditioning vector G. It is worth noting that while the roofline predicted in the first stage provides structural guidance for completion, it is not used as a condition in this upsampling stage. This is because the roofline is generated at the same low resolution as the completed heightmap and thus lacks the fine-grained detail necessary to provide meaningful additional guidance for high-resolution synthesis.

At inference, the process begins with noise patches $x_T^{(i,j)} \sim \mathcal{N}(0, I)$, and the reverse diffusion is iteratively applied, conditioned on $y^{(i,j)}$, to generate high-resolution patches. These patches are subsequently assembled into the complete heightmap, with padding employed at the borders to facilitate feature collage for edge patches. This methodology ensures that the Patch-DM efficiently produces detailed heightmaps suitable for building reconstruction, adeptly handling the variability in scale inherent to such applications.

4.3. Evaluation Metrics

To assess the performance of the heightmap reconstruction, three primary metrics are employed: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and a 2D Chamfer Distance. These metrics quantify the discrepancy between the predicted heightmap and the ground truth, expressed in meters, and provide complementary insights into the model's accuracy.

Root Mean Square Error (RMSE): This metric calculates the square root of the average squared differences between the predicted height values ($\hat{z}_i$) and the ground truth height values (z_{gt_i}). The formula is given by:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{z}_i - z_{gt_i})^2}, \tag{4.8}$$

where N represents the total number of pixels. RMSE emphasizes larger errors due to the squaring operation, making it particularly sensitive to significant deviations and thus effective for evaluating overall reconstruction fidelity.

Mean Absolute Error (MAE): MAE measures the average of the absolute differences between predicted and true height values, defined as:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{z}_i - z_{gt_i}|, \tag{4.9}$$

where N is the total pixel count. By focusing on absolute differences, MAE provides a robust estimate of the typical error magnitude, less influenced by outliers compared to RMSE. This makes it suitable for assessing the general consistency of the predictions.

Chamfer Distance (2D): Unlike the commonly used 3D point cloud Chamfer distance, this thesis employs a 2D variant specifically designed for heightmap evaluation. This metric measures the geometric similarity between the spatial distribution of non-zero pixels in the predicted and ground truth heightmaps. The 2D Chamfer distance is computed as:

$$\text{Chamfer} = \frac{1}{2}\left(\frac{1}{|P_{gt}|}\sum_{p\in P_{gt}}\min_{q\in P_{pred}}\|p-q\|_2 + \frac{1}{|P_{pred}|}\sum_{q\in P_{pred}}\min_{p\in P_{gt}}\|q-p\|_2\right), \tag{4.10}$$

where P_{gt} and P_{pred} represent the sets of 2D pixel coordinates with non-zero values in the ground truth and predicted heightmaps, respectively, and $\|\cdot\|_2$ denotes the Euclidean distance in 2D space. This bidirectional metric captures both coverage and spatial accuracy by measuring the average minimum distance from ground truth pixels to predicted pixels and vice versa. The 2D formulation specifically targets the geometric fidelity of roof structure boundaries.

5. Experiments and Results

5.1. Dataset Preparation

Figure 5.1 showcases examples from the dataset generated for this thesis. The following sections describe the data source and the preprocessing steps involved in its creation.

5.1.1. Data Source

This thesis leverages the Building-Pcc dataset (Gao et al., 2024), a collection of 3D models generated from synthetic reference point clouds, which includes approximately 300k structures in the Dutch cities of Rotterdam and The Hague. The models achieve a Level of Detail (LOD) of approximately 2.2, offering precise depictions of roof configurations, including multi-pitched and unconventional designs. Moreover, the dataset explicitly represents roof features such as dormer windows, chimneys, and elevated facade components like roof towers and spires. This granularity meets the precision demands of the analysis.

To ensure the training aligns with practical applications, real-world point cloud data from AHN3 and AHN4 were integrated. The Actueel Hoogtebestand Nederland (AHN) is the Netherlands' national elevation repository, with AHN3 and AHN4 being its third and fourth editions, respectively, each delivering high-resolution height data. Using building footprints to segment these point clouds, a total of 600k building-specific point clouds were compiled, with each structure represented by dual point clouds—one from AHN3 and one from AHN4.

The decision to exclude AHN5, the latest edition of the AHN dataset with higher resolution, was made to maintain consistency and comparability within the dataset. AHN3 and AHN4 already effectively capture real-world challenges such as partial missingness due to occlusions

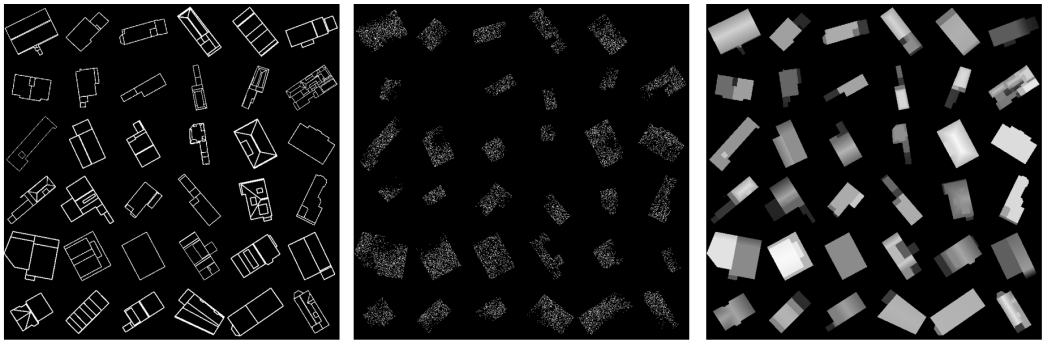
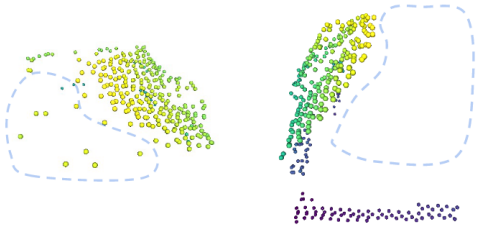


Figure 5.1.: Examples from the generated dataset: (a) roofline map, (b) corrupted heightmap, and (c) ground-truth heightmap.

or signal absorption. The increased resolution of AHN5 is unlikely to fundamentally alter the nature of these challenges, as the primary issues (e.g., incomplete building structures) remain across versions.

5.1.2. Preprocessing and Filtering of Point Cloud Data Based on Sparsity and Incompleteness Metrics

To ensure the quality and applicability of the dataset for this thesis, a systematic preprocessing and filtering approach was applied to the point cloud data and corresponding models. This process is essential for isolating data instances that exhibit significant defects, with sparsity and incompleteness serving as the core indicators for evaluating point cloud imperfections. These two concepts, though interrelated, differ in their origins and manifestations, necessitating distinct treatment during data preparation (see Figure 5.2).



(a) Sparse **(b)** Incomplete

Figure 5.2.: Illustration of point cloud defects: (a) sparsity—random uniform removal of points; (b) incompleteness—systematic absence of points in specific regions.(Figure from Lo et al. (2024))

Sparsity refers to the random absence of points within a point cloud, resulting in pixel absences that are typically uniformly distributed. This phenomenon arises from real-world factors such as sensor noise, insufficient sampling, or environmental interference. Sparsity is quantified as a percentage, calculated as the ratio of randomly removed pixels to the total number of pixels within the building's footprint. A sparsity level of 50% indicates that half of the pixels in the footprint are absent.Sparsity reflects an overall reduction in data density but does not imply systematic absences in specific regions.

Incompleteness, in contrast, reflects *contiguous* regions where roof points are completely missing. This thesis measures incompleteness with the *Hausdorff distance* (HD) between two binary masks:

1. M_{foot} – all raster cells that fall inside the building footprint, and
2. M_{ahn} – the subset of cells that contain at least one point in the AHN heightmap.

The HD is defined as $\mathrm{HD}(M_{\text{foot}}, M_{\text{ahn}}) = \max\left\{ \sup_{x \in M_{\text{foot}}} \inf_{y \in M_{\text{ahn}}} d(x,y),\ \sup_{y \in M_{\text{ahn}}} \inf_{x \in M_{\text{foot}}} d(x,y) \right\}$, where $d(\cdot,\cdot)$ denotes the Euclidean distance on the raster grid. Intuitively, the HD returns the largest gap between any occupied footprint cell and the nearest observed cell in the corrupted heightmap; a high value therefore signals a wide area that lacks data due to occlusion or other acquisition constraints. For instance, an HD of 18 pixels ($\approx 5.4\,\text{m}$ at the chosen $0.3\,\text{m}$ resolution) indicates a substantial void.

To make HD values comparable across buildings of different sizes, we report a *normalised Hausdorff distance* obtained by dividing the raw pixel HD by $L_{\max}$, the length of the longer side of the buildingś footprint bounding box. This yields a dimensionless score in the range $[0, 1]$ that removes scale bias while retaining sensitivity to large missing regions. Unlike sparsity—which is characterised by randomly scattered missing pixels—incompleteness thus manifests as spatially coherent blanks highlighted by a large (normalised) Hausdorff distance.

Building on these definitions, the preprocessing phase involved the following steps to curate a dataset tailored to studying complex point cloud defects:

1. **Removal of Simple Buildings**: Buildings with overly simplistic geometries, such as those with entirely single flat roofs, and those with footprints smaller than 50 square meters were all excluded. These structures lack the geometric complexity required to investigate intricate point cloud defects and are thus unsuitable for the research objectives.

2. **Removal of Mismatched Data**: Instances where the point cloud data and the corresponding model showed evident inconsistencies were manually identified and removed. This step ensures the consistency and reliability of the data pairs, which is critical for subsequent analysis.

3. **Filtering Based on Defect Metrics**: To identify and exclude point clouds with pronounced sparsity or incompleteness, three metrics were applied, measured on planar heightmaps by comparing the heightmap generated from the AHN point cloud with that of the reference model:

    - **Grid Coverage Completeness**: This metric measures the fraction of footprint cells that contain at least one AHN point. A low coverage value may arise from *(i)* randomly scattered gaps (high sparsity) or *(ii)* contiguous voids (high incompleteness); it therefore provides a joint indicator for both kinds of defects by reflecting how well the heightmap covers the expected building extent.

    - **Chamfer Distance**: This measures the bidirectional average distance between the point sets of the AHN and reference heightmaps, reflecting the combined impact of sparsity and incompleteness. Random point absences due to sparsity increase the average distance, while regional voids from incompleteness further amplify this effect, providing a comprehensive defect assessment.

    - **Hausdorff Distance**: This metric evaluates the maximum deviation between the two heightmaps, pinpointing severe local defects caused by incompleteness. For instance, significant voids in the heightmap due to occlusions result in large deviations, emphasizing localized extreme absences.

Using standard thresholds for these metrics, 160k point cloud-model pairs were selected from the original dataset. These pairs exhibit notable sparsity or incompleteness defects while retaining diverse and unique geometric shapes, making them well-suited for analyzing complex point cloud processing scenarios. From the 160k point cloud-model pairs in the dataset, 3k pairs were randomly selected to form a test set for the final evaluation which is not included in the training and validation sets. This test set retains the characteristics of the larger dataset, including notable sparsity or incompleteness defects and diverse, unique geometric shapes, ensuring it is representative for evaluating model performance in complex point cloud processing scenarios.

To further refine the dataset, stricter thresholds were applied, targeting higher defect levels and more intricate geometries. This yielded a subset of 14k pairs, which prioritizes challenging cases and severe defects, offering a robust foundation for training.

In conclusion, by distinguishing between sparsity and incompleteness and implementing a systematic filtering process based on planar heightmaps, a large-scale point cloud dataset was constructed for this thesis research. Additionally, a meticulously curated subset was derived to facilitate in-depth analysis, ensuring a solid basis for the subsequent experiments.

5.1.3. heightmap Generation

For heightmap creation, the urban point cloud density from AHN3 and AHN4 was considered, averaging around 10 points per cubic meter. A grid resolution of 0.3 meters was selected as the projection unit, with the highest point in each grid cell designated as the roof height. To support processing via the U-Net architecture, heightmap dimensions start at 128 pixels and increase in steps of 64 pixels, up to a maximum of 1024 pixels. Following this procedure, approximately 85% of the models fit within the 128-pixel size, indicating that most buildings are within the range of 19.2m $\times$ 19.2m to 38.4m $\times$ 38.4m, which is a reasonable size. Other buildings are larger.

5.1.4. Acquisition of Roof Line Ground Truth Labels

For multi-task training purposes in this thesis, roof line ground truth labels were derived by rendering top-down perspectives of the models using Blender. Specifically, edges between adjacent faces were marked as rooflines if the angle between their normals exceeded a 5-degree threshold. In these renderings, roof fold lines were assigned a white color, while all other regions were set to black. This technique provides multiple benefits:

- It extracts semantically clear rooflines directly from structured 3D models, avoiding typical edge detection errors like false positives and negatives.
- The process is highly customizable, ensuring consistent viewpoint, resolution, and color schemes across renderings, thus yielding standardized, high-quality labels suitable as ground truth for multi-task learning.

This process ensures the resulting rooflines are inherently aligned with the corresponding heightmaps, sharing identical dimensions and scale.

5.1.5. Roof heightmap Normalization

Given the wide range of building heights and the constrained value domain required by diffusion models, a tailored normalization method was essential. To isolate the roof structure, we first identify the lowest height value among roof pixels and subtract it from the entire heightmap, establishing a zero baseline for the roof. For each building, indexed as the i-th structure, we define δ_i as the roof height range—the difference between its maximum and minimum heights. Across all buildings, z is then calculated as the 99th percentile of these δ_i values, excluding the top 1% to reduce outlier effects. Analysis of 13,000 buildings determined z to be 10 meters.

The normalized heightmap value, x, for a given height h in the map is computed as:

$$x = \frac{2(h - h_{\min})}{z} - 1 \tag{5.1}$$

Here, $h_{\min}$ represents the minimum roof height specific to that building. This normalization ensures that height variations are mapped to the interval $[-1, 1]$, accommodating the majority of roof structures based on the selected z threshold.
+*Practical note:* during preprocessing we additionally record the true mid-height value, $h_{\text{mid}} = (h_{\min}+h_{\max})/2$, for every building. Together with the per-building height range, this metadata allows the network outputs to be denormalised after inference via $\hat{h} = \frac{z}{2}(\hat{x}+1)+h_{\min}$, restoring metric roof heights without loss of information.

5.1.6. Construction of S80_i80 and S80_i30 Benchmarks

To evaluate the robustness of the proposed methods under varying levels of data corruption, two benchmark datasets, S80_i80 and S80_i30, were constructed from the test set of 3,000 point cloud-model pairs. *Because the native sparsity and incompleteness present in the AHN heightmaps (cf. Figure 5.1) are relatively modest, additional degradation was applied to create a more challenging testbed.* For the S80_i80 dataset, 80% of the points were randomly removed to simulate high sparsity (80%) and severe incompleteness (80%) caused by systematic occlusions. Similarly, the S80_i30 dataset was generated by removing 80% of the points, representing moderate sparsity (80%) and lower incompleteness (30%). In practice the benchmarks are generated through *(i)* random point thinning to realise the target sparsity, followed by *(ii)* the removal of randomly sampled contiguous patches to emulate realistic occlusion-driven incompleteness. Footprint masks were retained as conditional inputs to guide structure-aware processing. These datasets were designed to test the models' performance in handling extreme and moderate corruption scenarios, respectively, ensuring a comprehensive evaluation of reconstruction capabilities.

5.2. Implementation Details

5.2.1. Training strategy

The training of the proposed two-stage model is performed sequentially.The models were trained on a single NVIDIA RTX 3090 (24GB) GPU, each requiring over 150 GPU hours to complete 500,000 training timesteps, a commonly recommended training steps in the community.

Stage 1: Diffusion-Based Roof Completion. The first stage involves a multi-step training process for the dual-task diffusion model:

1. **Independent Training of Specialist Models:** Initially, the Heightmap Completion Diffusion model and the Roofline Prediction Diffusion model are trained separately. This allows each model to learn its specific task effectively.
2. **BCM Training with Frozen Specialists:** Subsequently, the Bidirectional Control Module (BCM) is trained. During this phase, the parameters of the pre-trained Heightmap Completion and Roofline Prediction models are frozen. The BCM learns to mediate and refine the outputs of these specialist models, enforcing structural consistency.

This staged training approach for the first stage, inspired by methodologies like DiffTSR (Zhang et al., 2024b), helps to manage GPU memory overhead while ensuring effective interaction between the heightmap completion and roofline prediction tasks.

Stage 2: Patch-Based Up-Sampling. The second stage, which performs high-resolution upsampling, involves training the Patch-based Denoising Diffusion Model. This model is trained independently after the first stage is complete. It takes the completed heightmap from Stage 1 as one of its inputs and is trained to generate high-resolution heightmaps by processing fixed-size patches. Further details on the specific training configuration, including hyperparameters and dataset splits for both stages, are provided in Appendix A.6.

5.2.2. Evaluation Protocol on Local Dataset

To evaluate the proposed framework and for a fair comparison, all baseline methods—including *RoofDiffusion* as well as the classical interpolation techniques were retrained and/or tested using the same training/validation splits as this thesis proposed. Owing to the project's focus on localized performance and limited time, results on the original RoofDiffusion benchmark are not reported.

5.2.3. Baseline Implementations

To evaluate the proposed methods, a range of classical and learning-based baseline approaches are selected for the roof completion and heightmap upsampling tasks.

Roof Completion Baselines

To comprehensively evaluate the effectiveness of the proposed approach, a diverse set of baseline methods spanning both classical and state-of-the-art techniques were selected for comparison in the roof heightmap completion task. These methods represent different paradigms in sparse data interpolation and reconstruction:

- **IDW**: Inverse Distance Weighting interpolation, which assigns weights based on the inverse of the distance to known points.
- **Nearest**: Nearest-neighbor interpolation, which selects the value of the closest known point.
- **Spline**: Radial basis function interpolation using bivariate splines for smooth surface reconstruction.
- **Perona-Malik** (Perona and Malik, 1990): Anisotropic diffusion inpainting with iterative filtering to preserve edges.
- **RoofDiffusion** (Lo et al., 2024): A diffusion-based deep learning model for roof height completion, retrained on our dataset to ensure a fair comparison.

Heightmap Upsampling Baselines

For the heightmap upsampling task, a comprehensive set of classical interpolation techniques were employed as baselines to establish performance benchmarks against the proposed diffusion-based approach. These methods represent well-established mathematical interpolation paradigms that have been widely used in image processing and spatial data reconstruction applications. Each technique offers distinct characteristics in terms of computational complexity, smoothness, and edge handling capabilities. All baseline methods were implemented using standard libraries such as SciPy or OpenCV to ensure reproducibility and fair comparison:

- **IDW**: Distance-weighted interpolation applied to upsampled predictions.
- **Nearest**: Standard nearest-neighbor interpolation.
- **Bilinear**: Linear interpolation in two dimensions.
- **Bicubic**: Cubic interpolation for smoother results in 2D.
- **Lanczos** (Turkowski, 1990): Windowed sinc interpolation for high-quality upsampling.

5.2.4. Architecture Variants

The proposed framework comprises two models: one for sparse roof completion and another for heightmap upsampling. Several architectural variants are developed to facilitate ablation studies and assess the effectiveness of individual components.

Roof Completion Variants

The roof completion model builds upon the RoofDiffusion backbone by incorporating an auxiliary branch for roofline prediction. The following variants are evaluated:

- **RC-Full**: The complete model featuring dual-task training with roofline-guided prediction.
- **RC-Ablated**: A simplified variant excluding the auxiliary roofline prediction branch.

Heightmap Upsampling Variants

The upsampling model integrates two enhancements: a global context condition and position embeddings. The following ablation variants, summarized in Table A.8 (Appendix A), are designed:

- **UP-Full**: The complete model incorporating both global context conditions and positional encodings.
- **UP-NoGC**: A variant excluding the global context condition branch.
- **UP-NoPE**: A variant excluding position embeddings.
- **UP-Patch**: The proposed patch-based processing approach using 64×64 patches.
- **UP-FullImg**: A full-image processing variant using the same architecture without patch decomposition.

5.3. Roof Heightmap Completion Results

5.3.1. Quantitative Comparison

The quantitative results, presented in Table 5.1, demonstrate significant performance variations among different completion methods, with clear patterns emerging across the moderate (S80_i30) and severe (S80_i80) corruption scenarios. The proposed method consistently achieves superior performance in terms of RMSE and Chamfer Distance metrics, while maintaining competitive MAE scores.

In the moderate corruption scenario (S80_i30), the proposed method achieves the best RMSE of 0.89, representing a 13.2% improvement over the closest competitor, RoofDiffusion (1.02). For Chamfer Distance, the proposed approach yields 0.06, demonstrating an 17.3% improvement over RoofDiffusion's 0.07. Interestingly, the Nearest-neighbor method achieves the best MAE (0.22), suggesting its effectiveness in handling local pixel-level errors in moderate corruption scenarios.

The performance gap becomes more pronounced under severe corruption (S80_i80). The proposed method maintains its superiority with an RMSE of 1.11, achieving a 13.5% improvement over RoofDiffusion (1.28) and a 12.9% improvement over Nearest (1.27). The Chamfer Distance improvement is even more significant, with the proposed method achieving 0.06 compared to RoofDiffusion's 0.07, indicating better preservation of geometric structure under extreme degradation.

Classical interpolation methods show mixed performance. While IDW and Spline demonstrate reasonable RMSE values (1.04 and 1.03 for S80_i30), they deteriorate significantly under severe corruption, with Spline's RMSE increasing to 1.79 in S80_i80. Perona-Malik consistently underperforms across all scenarios, with RMSE values exceeding 4.70, highlighting its unsuitability for this specific reconstruction task.

Table 5.1.: Quantitative comparison of roof heightmap completion methods under moderate (S80_i30) and severe (S80_i80) corruption scenarios. Best results are highlighted in bold.

	S80_i30 (Moderate)			S80_i80 (Severe)		
Methods	RMSE	MAE	Chamfer	RMSE	MAE	Chamfer
IDW	1.0392	0.2391	0.0757	1.3642	0.4039	0.2192
NEAREST	1.06	**0.2186**	0.0731	1.2687	**0.3365**	0.0731
SPLINE	1.0274	0.2246	0.0879	1.7888	0.6103	0.6242
PERONA MALIK	4.7112	2.4031	1.5064	4.7946	2.4852	6.8649
RoofDiffusion	1.0229	0.2312	0.074	1.2776	0.3941	0.074
Ours	**0.8878**	0.2637	**0.0612**	**1.1057**	0.3859	**0.0641**

5.3.2. Qualitative Results

The qualitative evaluation reveals significant visual differences between methods, particularly highlighting the limitations of traditional interpolation approaches under challenging corruption scenarios. Figures 5.3 and 5.4 present comparative results for moderate (S80_i30) and severe (S80_i80) corruption scenarios respectively.

Moderate Corruption Scenario (S80_i30)

Under moderate corruption conditions, traditional interpolation methods demonstrate varying degrees of effectiveness but consistently exhibit fundamental limitations. As shown in Figure 5.3, IDW produces rapid gap filling while maintaining basic structural shapes, yet suffers from obvious interpolation artifacts, blurred boundaries, and bias toward low-frequency reconstruction that fails to preserve local details or gradient transitions. Nearest-neighbor interpolation, while computationally efficient, generates extensive block-wise artifacts with poor smoothness and severe detail loss. Spline interpolation provides smoother transitions compared to IDW but lacks accurate structural boundary reconstruction, with notable distortions observed in complex samples (e.g., Sample 5).

The Perona-Malik method shows inconsistent performance, with reasonable local structure preservation in some cases (Sample 2) but significant reconstruction failures in others (Samples 1 and 6), indicating high sensitivity to input data distribution. RoofDiffusion demonstrates relatively balanced performance with good smoothness and structural integrity, though it still exhibits over-smoothing tendencies and occasional artifacts in complex regions.

The proposed method achieves superior reconstruction quality across all samples, with results closely matching the ground truth. The method demonstrates exceptional structure preservation capabilities (particularly evident in Samples 4 and 6), maintains clear details with natural transitions, and consistently produces the most faithful reconstructions among all evaluated approaches.

Severe Corruption Scenario (S80_i80)

Under extreme data sparsity, the performance gap between learning-based and traditional methods becomes dramatically pronounced, as demonstrated in Figure 5.4. Traditional interpolation techniques largely fail to provide meaningful reconstruction. IDW produces crude approximations with obvious blur and void regions, particularly in severely damaged samples (Samples 3 and 5), where over-reliance on distance-based weighting creates unnatural transitions and uneven internal distributions. Nearest-neighbor interpolation generates large color blocks with abrupt transitions, proving highly sensitive to corruption and unable to reconstruct coherent structural boundaries.

Spline interpolation, while showing reasonable transitions in simpler cases (Sample 4), exhibits severe structural collapse or unrealistic elevation changes in complex scenarios (Samples 1 and 3), with boundary oscillations compromising reconstruction stability. The Perona-Malik approach produces obviously distorted results with anomalous structures that deviate significantly from expected roof geometries across multiple samples.

RoofDiffusion demonstrates the advantages of learning-based approaches, generating smooth and structurally reasonable reconstructions that significantly outperform traditional methods. The method shows evidence of shape priors through good boundary fitting and smooth region handling, though it still exhibits detail loss in complex structural cases (Sample 6).

The proposed method achieves the most faithful reconstruction across all corruption levels, demonstrating exceptional robustness to sparse data conditions. The approach successfully captures complete boundary morphology and slope information, being virtually the only method capable of accurately reconstructing complex elongated structures (Samples 3 and 5). This superior performance under extreme conditions validates the effectiveness of the

roofline-guided completion strategy and highlights the critical importance of incorporating structural priors in sparse heightmap reconstruction tasks.

5.3.3. Discussion

The quantitative and qualitative results underscore the superior performance of the proposed method in roof heightmap completion, particularly under both moderate $S80\_i30$ and severe $S80\_i80$ corruption scenarios. The method's consistent excellence in RMSE and Chamfer Distance metrics, coupled with visually faithful reconstructions, highlights its robustness and ability to preserve geometric structures. However, the method does not achieve the best MAE scores, with Nearest-neighbor interpolation outperforming it in both scenarios. This observation warrants a deeper analysis of the metrics and the characteristics of heightmap data.

Understanding Metric Discrepancies: MAE vs. RMSE and Chamfer Distance

Heightmap images, particularly those representing roof structures, are inherently smooth, with gradual elevation changes and limited high-frequency details. This smoothness reduces the presence of complex patterns that might otherwise amplify errors in local regions. MAE, as a metric, computes the average absolute error across all pixels, giving equal weight to all deviations. In smooth heightmaps, where most regions exhibit small, uniform errors, the Nearest-neighbor method benefits from its simplicity, producing low local errors in flat or gradually varying areas. This explains its superior MAE performance, as it effectively minimizes pixel-level deviations in less complex regions, which dominate the heightmap.

In contrast, the proposed deep learning-based method excels at reconstructing regions with abrupt changes, such as structural edges or sharp elevation transitions, which are critical for preserving the overall geometry of roof heightmaps. These regions, while sparse, contribute significantly to large errors when mishandled by interpolation methods like Nearest-neighbor or IDW, which struggle with blocky artifacts or blurred boundaries. RMSE, by squaring errors, amplifies the impact of these large deviations, thereby highlighting the method's ability to accurately capture edges and structural details. Similarly, the Chamfer Distance, which evaluates geometric fidelity, further underscores the method's strength in preserving boundary morphology and slope information, as evidenced by the 17.3% and 13.5% improvements over RoofDiffusion in $S80\_i30$ and $S80\_i80$, respectively.

This discrepancy between MAE and other metrics reveals a key insight: while MAE is useful for assessing average pixel-level accuracy, it may under-represent the significance of errors in critical regions like edges, which constitute a small fraction of the heightmap but are essential for structural integrity. In applications where geometric accuracy and visual fidelity are paramount, RMSE and Chamfer Distance provide a more comprehensive evaluation of performance, aligning with the qualitative superiority observed in the reconstructions (Figures 5.3 and 5.4).

Limitations and Applicability Considerations

A limitation of the proposed method lies in its dependence on the structure prediction module, which infers structural information primarily from the building footprint and sparse heightmap

data. In cases of extreme data sparsity, such as severe corruption scenarios (S80_i80), insufficient input information may lead to erroneous outputs or overly smoothed reconstructions, despite the method's superior performance compared to other approaches. Furthermore, the computational complexity of the deep learning-based model poses challenges in resource-constrained environments, where simpler interpolation techniques may be preferred. These trade-offs suggest that the suitability of the method depends on application-specific requirements, balancing the need for high-fidelity reconstruction against constraints like real-time processing.

In conclusion, the proposed method's superior performance in RMSE and Chamfer Distance, driven by its ability to handle edges and complex structures, makes it particularly well-suited for roof heightmap completion. The lower MAE scores of simpler methods like Nearest-neighbor reflect the smooth nature of heightmaps, where local errors are less pronounced, but do not detract from the overall effectiveness of the proposed approach in achieving geometrically accurate and visually coherent reconstructions.

5.4. Heightmap Upsampling Results

This section presents a comprehensive evaluation of the proposed patch-based upsampling model for roof heightmaps. The evaluation comprises quantitative comparisons, qualitative assessments, and a discussion of the model's performance, limitations, and potential improvements. Experiments are conducted in two scenarios: (1) upsampling ground-truth low-resolution (LR) heightmaps with varying corruption levels (S80_i30 and S80_i80), and (2) upsampling outputs from a roof completion model to simulate real-world pipelines.

5.4.1. Quantitative Comparison

The quantitative evaluation compares the proposed method against classical interpolation techniques, including Inverse Distance Weighting (IDW), Nearest Neighbor, Bilinear, Cubic, and Lanczos, using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as metrics. The evaluation protocol computes per-image averages before cross-image aggregation to account for variable heightmap dimensions. Due to computational constraints for large-scale heightmaps, Chamfer Distance is excluded.

Table 5.2 summarizes the performance on the ground-truth (GT) dataset and the S80_i30 and S80_i80 scenarios. The proposed method achieves RMSE values of 0.437 and 0.435 for S80_i30 and S80_i80, respectively, representing an approximately 10% improvement over the best classical baseline (Bilinear: 0.490). Similarly, its MAE values of 0.086266 and 0.0853 outperform Bilinear's 0.087. Among classical methods, Bilinear interpolation performs best, followed by Cubic and Lanczos, while Nearest Neighbor exhibits the highest RMSE (0.618).

Notably, the proposed method demonstrates robust performance across corruption scenarios, with a minimal RMSE difference of 0.03 between S80_i30 and S80_i80. This stability highlights the model's ability to handle varying levels of data degradation, making it suitable for practical applications.

To evaluate the complete pipeline, the upsampling methods are tested using outputs from a roof completion model as input, simulating real-world scenarios where heightmaps are first completed from sparse data before upsampling. Table 5.3 presents the results. The proposed

Table 5.2.: Quantitative comparison of upsampling methods. Interpolation methods are evaluated on the ground-truth (GT) dataset. The proposed deep learning model (Ours) is evaluated on GT low-resolution data with corrupted heightmaps as input (GT→S80_i30 and GT→S80_i80). "–" indicates not applicable.

Method	**GT**		**GT→S80_i30**		**GT→S80_i80**	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
IDW	0.535396	0.101994	–	–	–	–
Nearest	0.618101	0.088653	–	–	–	–
Bilinear	0.490263	0.087479	–	–	–	–
Cubic	0.510711	0.101233	–	–	–	–
Lanczos	0.519653	0.113311	–	–	–	–
Ours	–	–	0.437917	0.086266	0.435049	0.084973

method outperforms classical methods, achieving RMSE values of 0.909 and 1.419 for S80_i30 and S80_i80, respectively, compared to Bilinear's 1.178 and 1.689. MAE values of 0.439 and 0.848 further demonstrate the model's superiority. Classical methods exhibit significantly higher errors, particularly in the S80_i80 scenario, indicating their sensitivity to input quality degradation.

Table 5.3.: Quantitative evaluation results: Performance of upsampling methods when using roof completion model output as input. All methods are evaluated on S80_i30 and S80_i80 datasets, with the input being the completed heightmaps from the roof completion stage rather than ground-truth data.

	S80_i30		S80_i80	
Methods	RMSE	MAE	RMSE	MAE
IDW	0.950098	0.469048	1.595715	0.879998
Nearest	1.242432	0.610033	1.741563	1.031157
Bilinear	1.178433	0.605272	1.689419	1.024969
Cubic	1.191716	0.611404	1.702138	1.031706
Lanczos	1.195722	0.616110	1.705208	1.035350
Ours	**0.908743**	**0.438605**	**1.419351**	**0.847748**

5.4.2. Qualitative Results

Qualitative evaluations provide insights into the visual quality of the upsampled heightmaps, focusing on edge sharpness, structural fidelity, and detail generation. Figures 5.5, 5.6, 5.7, and 5.8 present visual comparisons across methods and scenarios.

Specifically, Figure 5.5 compares classical interpolation methods (Nearest Neighbor, Bilinear, Cubic, Lanczos, and IDW) on low-resolution ground-truth heightmaps. This figure highlights the inherent limitations of these traditional techniques, such as blocky artifacts from Nearest Neighbor, blurring from Bilinear and Cubic, ringing artifacts from Lanczos, and difficulties with smooth transitions in complex regions by IDW.

Figure 5.6 illustrates the performance of the proposed method throughout the upsampling pipeline for both S80_i30 and S80_i80 corruption scenarios. It showcases the low-resolution input, the corrupted inputs at different severities, the corresponding outputs from proposed model, and the ground-truth heightmaps. This comparison demonstrates the model's ability to generate sharp edges, although minor irregularities in edge linearity can be observed in complex roof structures.

Figures 5.7 and 5.8 provide a closer look at the qualitative results of the proposed method for the S80_i30 and S80_i80 scenarios, respectively. These figures emphasize the model's strength in producing well-defined edges even under significant data corruption. However, they also reveal that while the upsampled heightmaps are visually improved in terms of resolution and edge definition, the model tends to be conservative in generating new structural details that are not present in the low-resolution input. This is a key characteristic of the proposed approach: it excels at high-quality upsampling and edge preservation rather than true super-resolution involving the synthesis of entirely new fine-grained geometric features. The visual results confirm the quantitative findings, showing that the model effectively enhances the resolution and clarity of roof heightmaps while maintaining structural integrity.

5.4.3. Discussion

The proposed patch-based upsampling model demonstrates significant advantages in the super-resolution of roof heightmaps, particularly in generating sharp edges and achieving competitive quantitative performance. The patch-based approach not only reduces computational complexity but also produces upsampled images without obvious artifacts or stitching traces, as evidenced in the qualitative results. This artifact-free enlargement capability establishes a solid foundation for future enhancements. However, the model's inability to generate new structural details and minor issues with edge linearity highlight key limitations that warrant further investigation.

Strengths

The model consistently outperforms classical interpolation methods across all evaluation scenarios. Quantitatively, it achieves an approximately 10% improvement over the best classical baseline (Bilinear) with RMSE values of 0.437917 and 0.435049 for S80_i30 and S80_i80, respectively, compared to Bilinear's 0.490263. This improvement reflects the model's superior ability to handle edge regions where classical methods typically introduce blurring or interpolation artifacts.

The patch-based architecture provides significant computational advantages, reducing both training and inference complexity while maintaining reconstruction quality. Unlike full-image processing approaches, this strategy enables efficient processing of large-scale heightmaps without memory constraints. The architecture is also inherently well-suited for handling multi-resolution data. By processing local patches independently, the model can efficiently adapt to input heightmaps of varying resolutions, enabling flexible application to datasets with diverse spatial scales without requiring extensive architectural modifications.

Qualitatively, the model produces seamless, artifact-free reconstructions with sharp edge preservation, as demonstrated in the visual comparisons. The absence of stitching artifacts between

patches, despite the segmented processing, validates the effectiveness of the global condition and positional embedding design.

In the complete pipeline evaluation (Table 5.3), the model maintains its superior performance when processing outputs from the roof completion stage, achieving RMSEs of 0.908743 and 1.419351 for S80_i30 and S80_i80, respectively. This demonstrates its practical applicability in end-to-end systems.

Limitations

The primary limitation of the proposed method lies in its conservative approach to detail generation. Rather than performing true super-resolution by synthesizing high-frequency structural details absent in the low-resolution input, the model primarily functions as a high-quality upsampler. Across both S80_i30 and S80_i80 scenarios, the outputs exhibit enhanced resolution and improved edge sharpness but do not introduce new geometric features such as additional roof ridges or fine-grained surface textures.

This conservative behavior is particularly noticeable when comparing the model outputs with ground-truth heightmaps, where the reconstructed surfaces closely follow the input heightmap topology without generating plausible missing details. For applications requiring detailed 3D building models, this limitation may restrict the method's utility where fine-grained geometric accuracy is paramount.

The model also exhibits occasional irregularities in maintaining perfect edge linearity, particularly visible in complex roof geometries with multiple intersecting planes. While subtle, these may affect applications requiring precise geometric consistency, such as structural analysis or architectural modeling.

Furthermore, the patch-based processing approach, while computationally efficient, inherently limits the model's ability to capture very long-range spatial dependencies that might be crucial for generating coherent large-scale structural patterns. This architectural constraint may contribute to the conservative reconstruction behavior.

Causal Analysis

The observed limitations can be attributed to several interconnected architectural and methodological factors:

Global Condition Encoding Strategy: The model's reliance on global conditions derived from low-resolution inputs creates an inherent bias toward structural fidelity rather than detail innovation. The global condition encoding captures coarse geometric patterns from the input heightmap, effectively constraining the generative process to remain faithful to the existing topology. While this design ensures consistency and prevents the hallucination of implausible structures, it simultaneously limits the model's capacity to infer and synthesize missing high-frequency details.

Patch-Based Processing Constraints: The 64×64 patch-based architecture, while computationally efficient, inherently limits the model's receptive field for capturing long-range spatial relationships. This constraint may manifest as an inability to maintain perfect geometric consistency across patch boundaries in complex roof structures and a limited capacity to generate coherent large-scale patterns that would require an understanding of building-wide

architectural principles. Introducing data sources with higher information density, such as aerial imagery, can enable the capture of more fine-grained details and enhance structural understanding.

Dataset Characteristics and Diversity: The Building-PCC dataset, while extensive, may exhibit inherent biases toward certain architectural styles prevalent in the two Dutch cities it covers. The limited geographical and architectural diversity could restrict the model's ability to learn and generate diverse roof detail patterns, particularly those involving subtle geometric variations or complex multi-level structures that are less common in the training data.

5.5. Ablation Study and Analysis

To further investigate the contributions of the key components in the proposed framework, a series of ablation studies was performed. Specifically, the influence of the roofline prediction module in the heightmap completion network, and the effects of the global condition and positional embedding in the patch-based upsampling network, were evaluated. All ablation experiments followed the training settings described in Section 4.2.1, ensuring a fair comparison. The evaluation was conducted on the S80_i30 dataset, which has moderate corruption.

5.5.1. Effect of Roofline Prediction Module

To quantify the contribution of the roofline prediction module, a variant of the proposed heightmap completion network was constructed by removing this component. This ablated variant is equivalent to the baseline RoofDiffusion model, retaining all other modules. The comparison between the full model and its variant without roofline prediction is presented in Table 5.4. The results show that removing the roofline prediction significantly degrades reconstruction accuracy, which confirms the crucial role of this additional supervision in improving geometric fidelity.

Table 5.4.: Ablation study on the roofline prediction module in the heightmap completion network.

Model Variant	**RMSE ↓**	**MAE ↓**	**Chamfer ↓**
Without roofline prediction (baseline)	1.02	0.23	0.07
With roofline prediction (full)	0.89	0.26	0.06

5.5.2. Effect of Global Condition and Positional Embedding

In the patch-based upsampling network, both the global condition and positional embedding play crucial roles. The global condition guides the upsampling process by providing overall contextual information, while positional embedding aids the network in distinguishing the spatial location of each patch. To validate their contributions, variants without these components were trained under the same settings.

The comparison, presented in Table 5.5, demonstrates the critical roles of both components. Global conditions provide overall semantic guidance to the model, ensuring that generated

images maintain semantic relevance and consistency at a global scale. Without global conditions, the model relies solely on positional embedding and local context, resulting in weaker overall semantics where the model may generate unnecessary details while losing important structural information.

Positional embedding provides spatial position information to the model, helping it correctly place image patches and ensuring spatial structural accuracy. Without positional embedding, the generated images exhibit distortions with severely incorrect relative positional relationships between patches and a lack of proper adjacency, as shown in Figure 5.10.

These results demonstrate the critical role of both components in maintaining coherent heightmap reconstruction.

Table 5.5.: Ablation study on the global condition and positional embedding in the patch-based upsampling network.

Model Variant	**RMSE ↓**	**MAE ↓**
without global condition	1.76	0.60
without positional embedding	1.53	0.65
with both components (full)	0.43	0.08

5.5.3. Computational Efficiency: Patch-Based vs. Full-Image Processing

A critical advantage of the patch-based upsampling approach is its superior computational efficiency compared to full-image processing architectures. To quantify this benefit, the maximum input resolution capabilities of the patch-based approach were compared against a variant that processes complete heightmaps using the same underlying diffusion architecture. This comparison evaluates the practical scalability limitations imposed by the attention mechanism's complexity in transformer-based models.

The experimental setup involved testing both approaches on an NVIDIA RTX 3090 GPU with 24GB of VRAM during inference. For the full-image processing variant, the input and output dimensions were modified while maintaining an identical network architecture.

Maximum Resolution Capabilities

The results demonstrate a dramatic difference in scalability between the two approaches. The patch-based upsampling method successfully processes heightmaps up to **1024×1024** resolution without memory constraints or significant performance degradation. In contrast, the full-image processing variant reaches its computational limit at **256×256** resolution, representing a 16-fold reduction in processable area.

This substantial performance gap stems from the quadratic scaling behavior of attention mechanisms with respect to input sequence length. In transformer-based diffusion models, the computational complexity of self-attention operations scales as $O(n^2)$, where n is the number of spatial tokens (pixels). For a full-image approach processing a 1024×1024 heightmap, the attention computation must handle $1024^2 = 1,048,576$ tokens, resulting in prohibitively large computational requirements that quickly exceed available GPU memory.

5.5.4. Patch-Based Efficiency Analysis

The patch-based approach circumvents this limitation by decomposing large heightmaps into smaller 64×64 patches, each containing only 4,096 tokens. This reduces the attention complexity per patch by several orders of magnitude compared to full-image processing. While multiple patches require sequential or batched processing, the cumulative computational cost remains significantly lower than the quadratic scaling penalty of full-image attention.

Additionally, the patch-based approach enables efficient parallel processing strategies, where multiple patches can be processed simultaneously within available memory constraints. This parallelization capability further improves practical throughput and enables scalable deployment on systems with varying computational resources.

Practical Implications

The 4× improvement in maximum processable resolution (1024×1024 vs. 256×256) has substantial implications for real-world applications:

- **Coverage Area**: The patch-based approach can process building footprints covering up to 16 times larger spatial areas at the same resolution, or alternatively, provide 16 times higher spatial detail for the same coverage area.
- **Deployment Flexibility**: The reduced memory requirements enable deployment on a broader range of hardware configurations, from high-end research GPUs to more accessible consumer graphics cards.
- **Scalability**: The approach naturally scales to even larger resolutions through increased patch decomposition, whereas full-image methods face hard computational limits.

This computational efficiency analysis validates the architectural choice of patch-based processing as both a practical necessity for handling large-scale heightmaps and a strategic advantage for scalable deployment. The trade-off between local patch processing and global context integration proves highly favorable, achieving substantial computational savings while maintaining reconstruction quality through the global condition and positional embedding mechanisms.

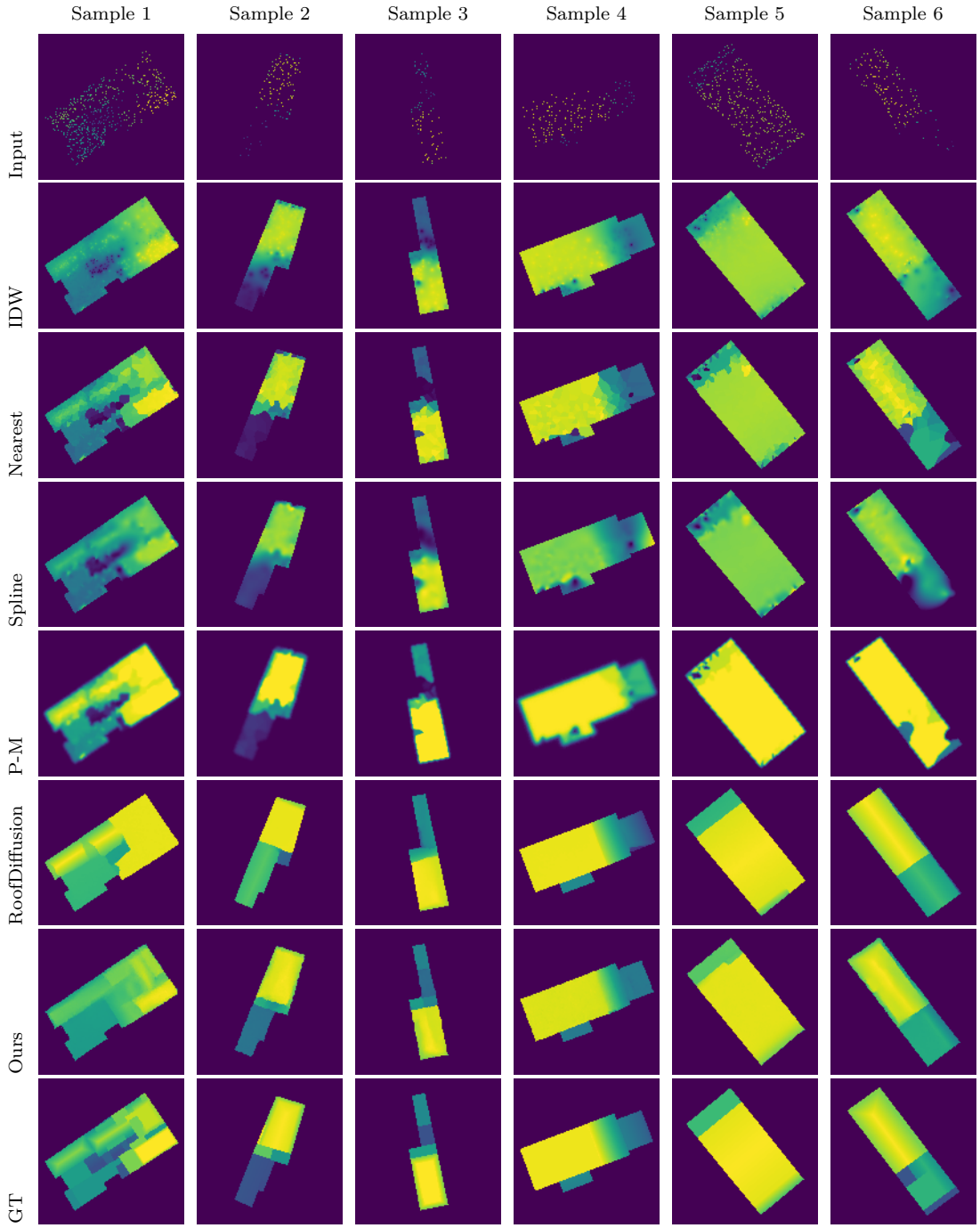


Figure 5.3.: Qualitative comparison of roof heightmap completion methods on the S80_i30 scenario. Each row shows results from a different method across all building samples, while each column represents a different building sample.

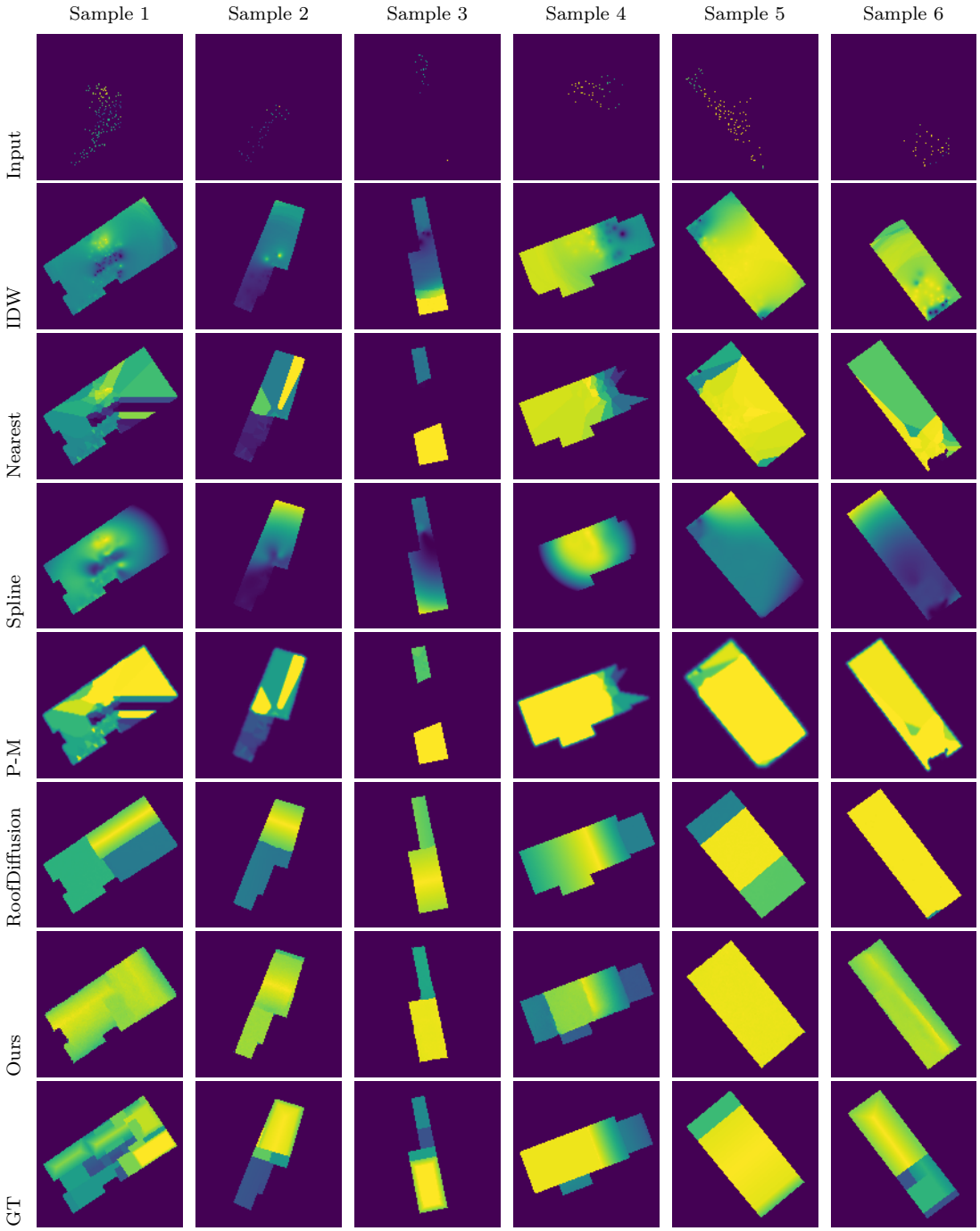


Figure 5.4.: Qualitative comparison of roof heightmap completion methods on the S80_i80 scenario. Each row shows results from a different method across all building samples, while each column represents a different building sample.

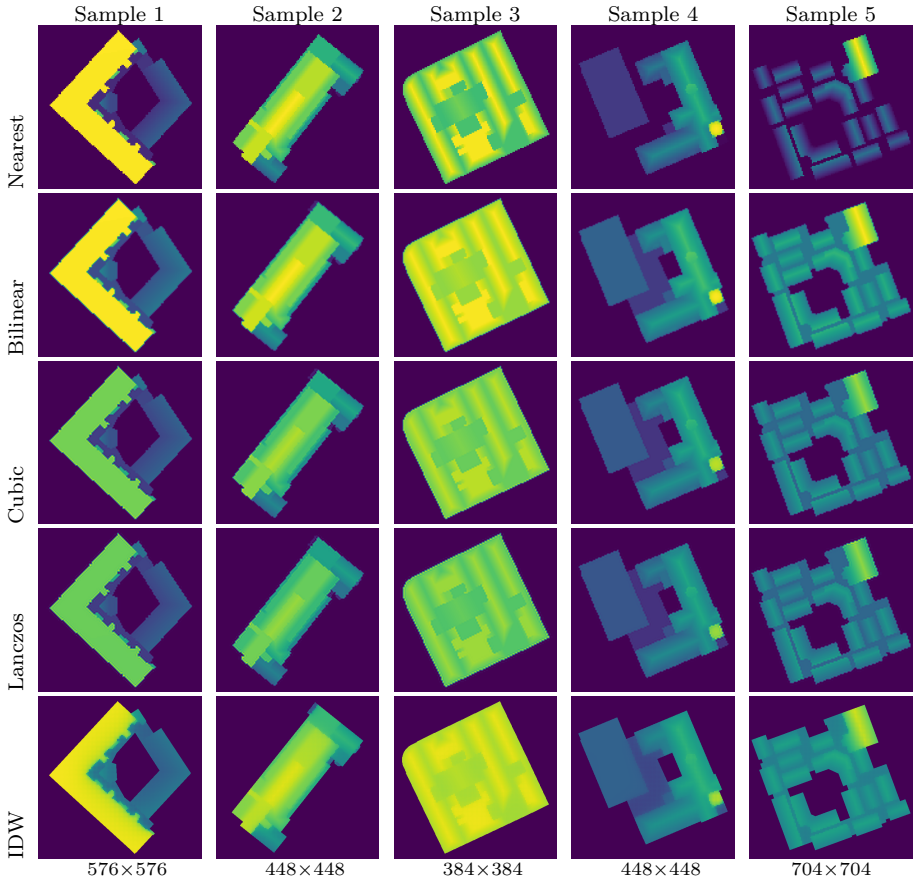


Figure 5.5.: Visual comparison of classical interpolation methods applied to low-resolution ground-truth heightmaps. Rows represent different methods: Nearest Neighbor, Bilinear, Cubic, Lanczos, and Inverse Distance Weighting (IDW). Columns correspond to distinct building samples, showcasing variations in roof geometry. Nearest Neighbor produces blocky artifacts, while Bilinear and Cubic introduce blurring at edges. Lanczos preserves some edge details but introduces ringing artifacts, and IDW struggles with smooth transitions in complex regions.

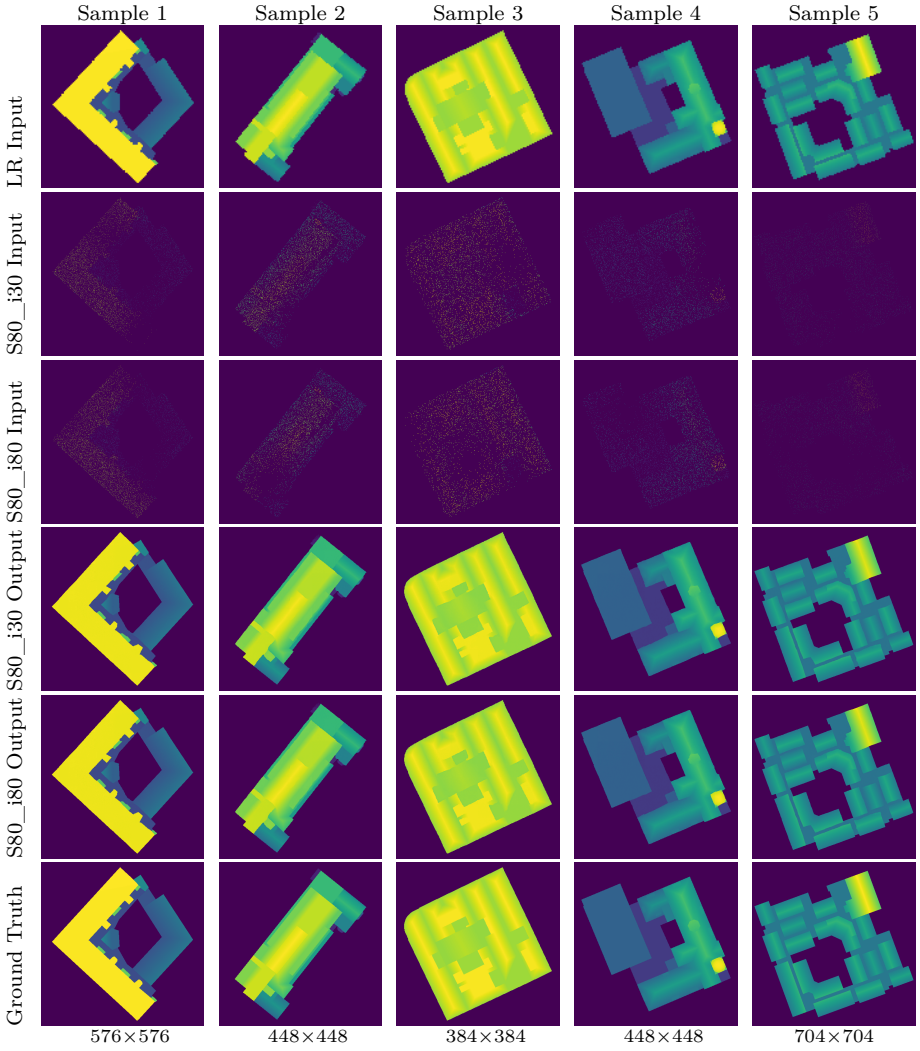


Figure 5.6.: Visual comparison of the proposed method across the upsampling pipeline for S80_i30 and S80_i80 scenarios. Rows display: (1) low-resolution input heightmap, (2) S80_i30 corrupted input, (3) S80_i80 corrupted input, (4) S80_i30 output, (5) S80_i80 output, and (6) ground-truth heightmap. Columns represent different building samples. The proposed method generates sharp edges but exhibits minor irregularities in edge linearity, particularly in complex roof structures.

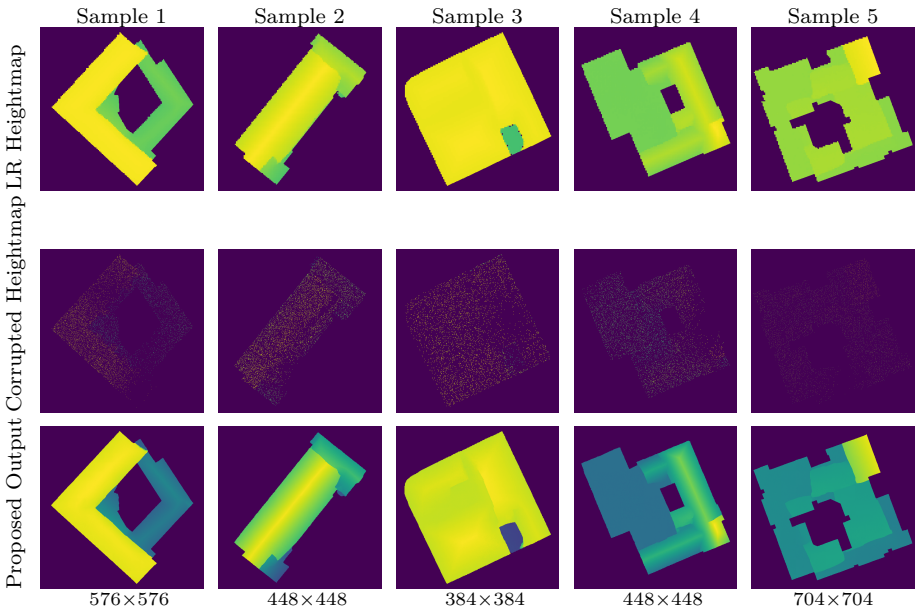


Figure 5.7.: Qualitative results for the proposed method on the S80_i30 scenario. Rows show: (1) low-resolution heightmap input, (2) corrupted heightmap input, and (3) output of the proposed method. Columns represent different building samples. The method produces sharp, well-defined edges but struggles with generating new structural details and maintaining edge linearity in complex geometries.

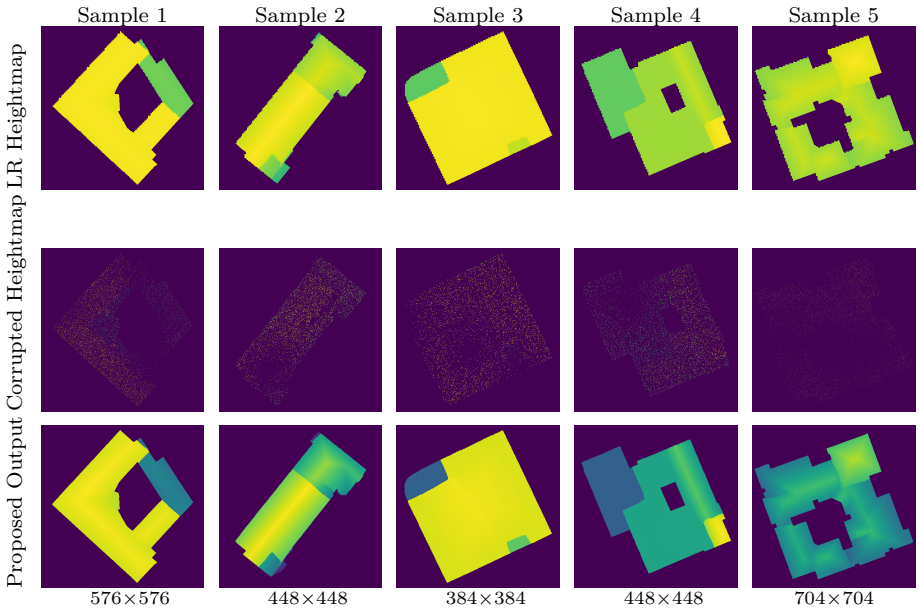


Figure 5.8.: Qualitative results for the proposed method on the S80_i80 scenario. Rows show: (1) low-resolution heightmap input, (2) corrupted heightmap input, and (3) output of the proposed method. Columns represent different building samples. The method maintains edge sharpness under severe corruption but fails to introduce new structural details, with visible edge irregularities in intricate roof structures.

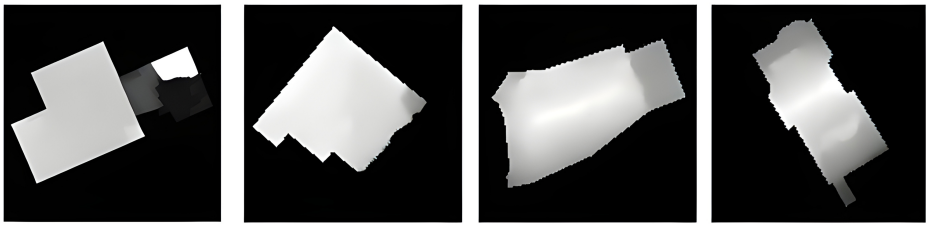


Figure 5.9.: Qualitative ablation result for the patch-based upsampling model excluding the global condition.

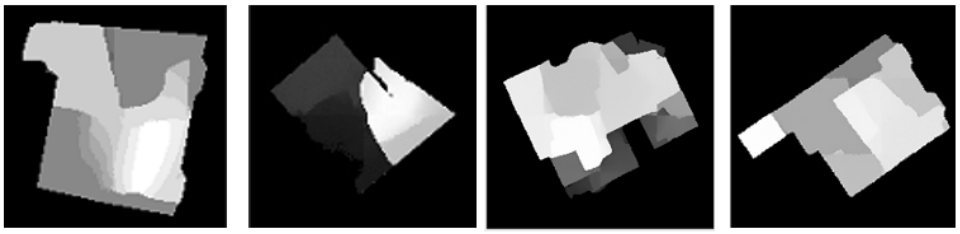


Figure 5.10.: Qualitative ablation result for the patch-based upsampling model excluding positional embedding.

6. Conclusion, Limitations, and Future Work

6.1. Conclusion

This thesis addressed the challenge of reconstructing high-fidelity 3D roof models from sparse and incomplete airborne lidar data. The primary research objective was to develop a framework based on diffusion models, operating on 2.5D heightmap representations, to overcome the limitations of existing methods in preserving geometric detail, handling sharp features, and accommodating scale variability.

To this end, a novel two-stage framework was proposed. The first stage introduced a dual-task diffusion model that jointly performs heightmap completion and roofline prediction. A key innovation, the Bidirectional Control Module (BCM), facilitates mutual information exchange between these tasks, leveraging explicit structural guidance from predicted rooflines to enhance the geometric accuracy of the completed heightmaps. The second stage presented a scalable, patch-based diffusion model for high-resolution heightmap upsampling. This model effectively addresses scale variance in buildings by processing large heightmaps in manageable patches, while a domain-specific global context encoder ensures structural consistency across the entire building. Furthermore, this research contributed a rigorous data curation pipeline and two new benchmark datasets, S80_i30 and S80_i80, to facilitate standardized evaluation of model robustness against data corruption.

The experimental results validate the effectiveness of the proposed framework. The dual-task completion model significantly outperformed state-of-the-art baselines in geometric accuracy, as measured by RMSE and Chamfer Distance, particularly under severe corruption scenarios. Similarly, the patch-based upsampling model demonstrated superior performance over classical interpolation methods, producing sharp, artifact-free reconstructions while maintaining robustness to input quality degradation. In conclusion, this work presents a comprehensive and effective solution for roof reconstruction, advancing the potential for automated generation of detailed and accurate LOD3.0 3D city models from imperfect real-world data.

6.2. Limitations

Despite the promising results, this research has several limitations that warrant consideration.

- **Dependency on Input Data Quality:** While the completion model shows robustness, its performance is fundamentally dependent on the quality of the input data. In scenarios of extreme sparsity, the structural information inferred from the building footprint and sparse heightmap may be insufficient, potentially leading to topologically incorrect or overly smoothed reconstructions.

- **Conservative Detail Generation in Upsampling:** The proposed upsampling model excels as a high-quality upscaler, preserving edges and structure effectively. However, it does not perform true super-resolution by generating new, plausible high-frequency geometric details (e.g., small dormers, vents) that are absent in the low-resolution input. Its behavior is primarily conservative, limiting its utility for applications that require the synthesis of fine-grained architectural features.
- **Architectural and Dataset Constraints:** The patch-based upsampling architecture, while computationally efficient, has an inherently limited receptive field that may constrain its ability to model long-range spatial dependencies in exceptionally large or complex buildings. Furthermore, the models were trained and evaluated on the Building-PCC dataset, which covers two Dutch cities. Generalizability to different architectural styles and urban typologies in other geographic regions remains unvalidated.
- **Computational Expense:** As with most deep learning approaches, the proposed framework is computationally more intensive than traditional interpolation techniques. This may pose a barrier to adoption in real-time applications or in environments with limited computational resources.

6.3. Future Work

Building upon the findings and limitations of this thesis, several avenues for future research are identified.

- **Advancing Towards True Super-Resolution:** Future work could focus on enhancing the upsampling model's capability to generate plausible, high-frequency details. This could be achieved by integrating more powerful generative priors, exploring adversarial training objectives to encourage detail synthesis, or designing a model that explicitly learns a residual component representing fine-grained features.
- **Exploring Globally-Aware Architectures:** To overcome the receptive field limitations of the patch-based approach, future research could investigate hybrid or hierarchical architectures. For instance, a model could combine the efficiency of local patch processing with more sophisticated mechanisms for capturing global context, such as multi-scale transformers or attention across patch-level feature summaries.
- **Enhancing Generalizability through Diverse Data and Multimodal Fusion:** A crucial next step is to improve the model's generalizability by training and evaluating it on a more diverse dataset encompassing a wider range of architectural styles and geographic locations. Furthermore, integrating complementary data modalities, such as aerial or satellite imagery, could provide rich semantic and textural information to guide the reconstruction process, potentially overcoming ambiguities present in sparse height data alone.
- **Leveraging Aerial Imagery for Richer Context:** A promising direction is to leverage high-resolution aerial imagery to extract strong structural priors. Future work could focus on predicting accurate building footprints and detailed rooflines directly from images, which are often challenging to obtain from lidar data alone. This explicit guidance could significantly enhance heightmap completion. Furthermore, the rich color and texture information can serve as a strong prior to resolve ambiguities in sparse lidar data,

guide the generation of fine-grained details (e.g., dormers, chimneys), and improve the semantic realism of the reconstructed roof structures.

- **Transition to Direct 3D Vector Representation:** An ambitious direction would be to move beyond raster-based heightmaps towards the direct generation of 3D vector representations, such as parametric planes or wireframes, which are more aligned with standard 3D modeling formats like CityGML. This would constitute a more fundamental solution to 3D reconstruction and would simplify downstream applications.
- **Model Optimization for Practical Deployment:** To enhance the practical applicability of the framework, future efforts could be directed towards model optimization. Techniques such as knowledge distillation, quantization, and pruning could be explored to develop lightweight versions of the models that retain high performance while significantly reducing computational and memory requirements.

A. Appendix

This appendix provides detailed information to ensure the reproducibility of the experiments presented in this thesis. It covers the data preprocessing pipeline, model architectures, and training configurations.

A.1. Data and Preprocessing Details

A.1.1. Filtering Metrics

To curate a high-quality dataset from the raw point clouds, several metrics were employed to filter out simple or inconsistent building models. The key metrics—sparsity and incompleteness—are defined based on a comparison between the ground-truth building footprint (derived from the 3D model) and the roof point cloud projected onto a 2D grid.

- **Sparsity**: Measures the overall density reduction in the point cloud due to random factors like sensor noise. It is the percentage of pixels within the ground-truth footprint that are missing from the lidar-derived roof map.

$$\text{Sparsity} = \left(1 - \frac{\text{Count}(\text{Pixels}_{\text{lidar}} \cap \text{Pixels}_{\text{Footprint}})}{\text{Count}(\text{Pixels}_{\text{Footprint}})}\right) \times 100\% \tag{A.1}$$

- **Incompleteness**: Quantifies systematic data loss, often from occlusion. It is measured using geometric distance metrics between the ground-truth footprint points and the observed lidar points. In this work, Hausdorff distances on the 2D plane are used as primary indicators. The Hausdorff distance between two point sets, P_{gt} and P_{obs}, is defined as:

$$d_H(P_{\text{gt}}, P_{\text{obs}}) = \max\left(\sup_{p \in P_{\text{gt}}} \inf_{q \in P_{\text{obs}}} \|p - q\|, \sup_{q \in P_{\text{obs}}} \inf_{p \in P_{\text{gt}}} \|q - p\|\right), \tag{A.2}$$

  where $\|\cdot\|$ denotes the Euclidean distance.

Algorithm 2 outlines the computational steps for deriving these metrics for each building.

A.1.2. heightmap Normalization

To constrain the height values to the range $[-1, 1]$ required by the diffusion model, a tailored normalization was applied. The normalization constant, z, was determined empirically from a statistical analysis of 13,000 buildings in the dataset. For each building, the height range, δ_i, was calculated as the difference between the maximum and minimum roof heights. The 99th percentile of all δ_i values was selected as z to mitigate the effect of outliers.

Algorithm 2 Data Preprocessing and Metric Calculation

Input: lidar point cloud file (`laspath`), 3D model file (`plypath`)
Output: Dictionary of calculated metrics for one building
procedure ProcessBuilding(laspath, plypath)
    Load mesh from `plypath`
    Calculate mesh bounds to determine the grid size and resolution (pixel size $\approx 1.5 \times$ average point distance)
    Project mesh to a 2D grid to create the ground-truth `footprint_map`
    Project lidar points from `laspath` to a 2D grid to create the `roof_map`
    Binarize both maps (0 for background, 255 for building)
    `footprint_pixels` ← Number of non-zero pixels in `footprint_map`
    `roof_pixels` ← Number of non-zero pixels in `roof_map`
    `intersection` ← Number of non-zero pixels in (`footprint_map` AND `roof_map`)
    `sparsity_rate` ← 1.0 - (`intersection` / `footprint_pixels`)
    Convert maps to 2D point sets `P_footprint` and `P_roof`
    `chamfer_dist`, `hausdorff_dist` ← Calculate distances between `P_footprint` and `P_roof`
    return `sparsity_rate`, `chamfer_dist`, `hausdorff_dist`
end procedure

The analysis determined $z = 10$ meters. A conceptual representation of the quantile analysis is shown in Table A.1.

Table A.1.: Conceptual Quantile Analysis for Roof Height Range (δ).

Percentile	**Height Range (δ_i) in meters**
50th (Median)	3.5
75th	6.2
90th	8.1
95th	9.0
99th	**10.0**
99.9th	14.2

A.2. Model Architecture and Hyperparameters

A.2.1. Model Architectures

The framework consists of three core models: a U-Net for completion/roofline prediction, the BCM for control, and a patch-based upsampler with a semantic encoder.

Roof Completion and Roofline Prediction Model (Palette-style U-Net)

Both the heightmap completion and roofline prediction models use a Palette-style U-Net architecture. The detailed configuration is presented in Table A.2. To balance computational

efficiency with model expressiveness, spatial transformer attention blocks are strategically applied only at the deeper layers (4× and 8× downsampling levels) where the spatial resolution is reduced, minimizing the computational overhead while maintaining the model's ability to capture long-range spatial dependencies. The total parameter count is approximately **125.6M**.

Table A.2.: Architecture of the Palette-style U-Net with Spatial Transformer.

Layer Type	Output Resolution	Channels	Heads	Details
— *Encoder* —				
Input	128x128	2	-	Corrupted Map + Footprint
Conv	128x128	64	-	Initial Convolution
Down Block 1	64x64	128	-	2 x ResBlock, Downsample
Down Block 2	32x32	256	-	2 x ResBlock, Downsample
Down Block 3	16x16	512	4	2 x ResBlock + Spatial Transformer, Downsample
Down Block 4	8x8	1024	4	2 x ResBlock + Spatial Transformer, Downsample
— *Bottleneck* —				
ResBlock	8x8	1024	4	2 x ResBlock + Spatial Transformer
— *Decoder* —				
Up Block 1	16x16	512	4	2 x ResBlock + Spatial Transformer, Upsample
Up Block 2	32x32	256	4	2 x ResBlock + Spatial Transformer, Upsample
Up Block 3	64x64	128	-	2 x ResBlock, Upsample
Up Block 4	128x128	64	-	2 x ResBlock, Upsample
Conv	128x128	1	-	Final Convolution

Bidirectional Control Module (BCM)

The BCM is the core component that facilitates interaction between the heightmap and roofline models. It is implemented as a smaller, dedicated U-Net that takes the concatenated outputs from both specialist models as input and uses cross-attention to generate a fused representation. Its architecture is detailed in Table A.3. The total parameter count for this module is approximately **16.8M**.

Patch-Based Upsampling Model

The upsampling model consists of a U-Net that processes patches and a semantic encoder that provides global context.

Semantic Encoder The global context encoder is an autoencoder based on a modified ResNet-50 architecture, designed for efficiency and effectiveness on heightmap data. Its structure is detailed in Table A.4. The total parameter count is **48.7M**.

Upsampler U-Net The main upsampling network is a Palette-style U-Net, configured to process 64x64 patches. Its architecture is detailed in Table A.5. Unlike the completion model, it does not use spatial transformer attention, prioritizing computational efficiency for patch-based processing.

Spatial Transformer Attention Block

The spatial transformer attention blocks used throughout the architecture implement a cross-attention mechanism that allows the model to attend to relevant spatial features from different resolution levels. For computational efficiency, these attention blocks are only deployed at the deeper layers of the U-Net (corresponding to $4\times$ and $8\times$ downsampling), where the reduced spatial dimensions make the attention computation more tractable while still capturing essential long-range dependencies. The detailed implementation is presented in Algorithm 3.

The key innovation of this spatial transformer is the context selection mechanism (lines 11-18), which dynamically chooses the appropriate context features based on the input feature map's channel dimensions. This allows the attention mechanism to operate effectively across different resolution levels in the U-Net architecture.

A.2.2. Training Hyperparameters

Key hyperparameters used for training the models are summarized in Table A.6.

Bidirectional Control Module (BCM) Training Hyperparameters The Bidirectional Control Module (BCM) is trained separately after the initial training of the heightmap completion and roofline prediction models. During BCM training, the weights of the specialist models are frozen. Key hyperparameters for training the BCM are detailed in Table A.7. These are derived from the 'mom.yaml' configuration file, specifically tailored for the BCM's role in mediating the two tasks.

Weighted BCE Loss for Roofline Prediction

To address the significant class imbalance between edge pixels (positive class) and non-edge pixels (negative class) in the roofline prediction task, a weighted binary cross-entropy (BCE) loss was implemented. This approach, inspired by techniques used in edge detection literature, ensures that the model does not become biased towards the majority (non-edge) class. The implementation involves several key steps based on the `cross_entropy_loss_RCF` function in the codebase.

First, the ground-truth roofline label, which may contain continuous values from rendering, is preprocessed. It is binarized using a threshold of 0.1 to create a clean binary map of edges (1) and non-edges (0). Following this, a Gaussian blur with a kernel size of 3 and a sigma of 1.0 is applied to the binarized label. This step introduces soft labels, which helps to regularize the model and improve its generalization by penalizing over-confident predictions.

The core of the loss function is its weighting scheme. The weights for the BCE loss are dynamically calculated for each batch to counteract the class imbalance. Let N_p be the number

of positive samples (edge pixels) and N_n be the number of negative samples (non-edge pixels) in a given batch. The weights for each class are assigned as follows:

- The weight for positive samples is set to $\frac{N_n}{N_p+N_n}$.
- The weight for negative samples is set to $\beta \cdot \frac{N_p}{N_p+N_n}$, where β is a hyperparameter set to 1.1 to slightly increase the importance of finding all positive samples.

This scheme gives more weight to the minority class (edges), forcing the model to pay more attention to them.

Additionally, to handle ambiguity in the ground-truth labels, pixels with a value in the range $(0, 0.3]$ after the blurring step are considered uncertain. These pixels are assigned a weight of 0, effectively excluding them from the loss calculation.

The final loss for a given prediction $\mathbf{P}$ and preprocessed label $\mathbf{L}$ is computed as a weighted BCE:

$$\mathcal{L}_{\text{BCE}} = \text{mean}(\text{BCE}(\mathbf{P}, \mathbf{L}, \mathbf{M})), \tag{A.3}$$

where $\mathbf{M}$ is the weight mask derived from the rules described above. This specialized loss is then combined with the main diffusion loss $\mathcal{L}_{\text{roofline}}$ to form the total objective for the roofline prediction model, as shown in Equation 4.3.

A.2.3. Ablation Study Configurations

Ablation studies were conducted to validate the contribution of key architectural components. The configurations are summarized in Table A.8.

A.2.4. Computational Efficiency Variants

To evaluate the computational advantages of patch-based processing, two architectural variants were designed for direct comparison:

- **UP-Patch (Proposed)**: The standard patch-based upsampling model that decomposes input heightmaps into 64×64 patches for processing. This approach includes global context conditions and positional embeddings to maintain coherence across patches.
- **UP-FullImg**: A variant that processes complete heightmaps using the identical underlying diffusion architecture, including the same attention mechanisms and layer configurations as the patch-based model, but without patch decomposition.

The key difference lies in the input processing strategy: UP-Patch operates on local 64×64 patches with global conditioning, while UP-FullImg processes the entire heightmap simultaneously. Both variants share the same:

- Palette-style U-Net architecture with spatial transformer attention blocks
- Attention head configurations (4 heads at deeper layers)
- Layer channel dimensions and residual connections
- Training hyperparameters and optimization settings

This controlled comparison isolates the impact of patch-based processing on computational efficiency while maintaining architectural consistency.

A.3. Qualitative Completion Results

This section presents additional qualitative results for the roof completion task on two challenging benchmark configurations: s80_i80 (80% sparsity, 80% incompleteness) and s80_i30 (80% sparsity, 30% incompleteness).

It is important to note that the quality of the roofline prediction is contingent on the information available in the corrupted heightmap and the building footprint. When these inputs are severely degraded, the model's predictions may exhibit uncertainty, resulting in artifacts such as repeated edges or extraneous lines. This phenomenon is more apparent in the heavily corrupted s80_i80 benchmark.

Figure A.1.: Qualitative completion results for the s80_i80 benchmark (Samples 1-5). From left to right: Ground Truth, Corrupted Input, RoofDiffusion (Baseline), Ours (Heightmap), and Ours (Roofline).

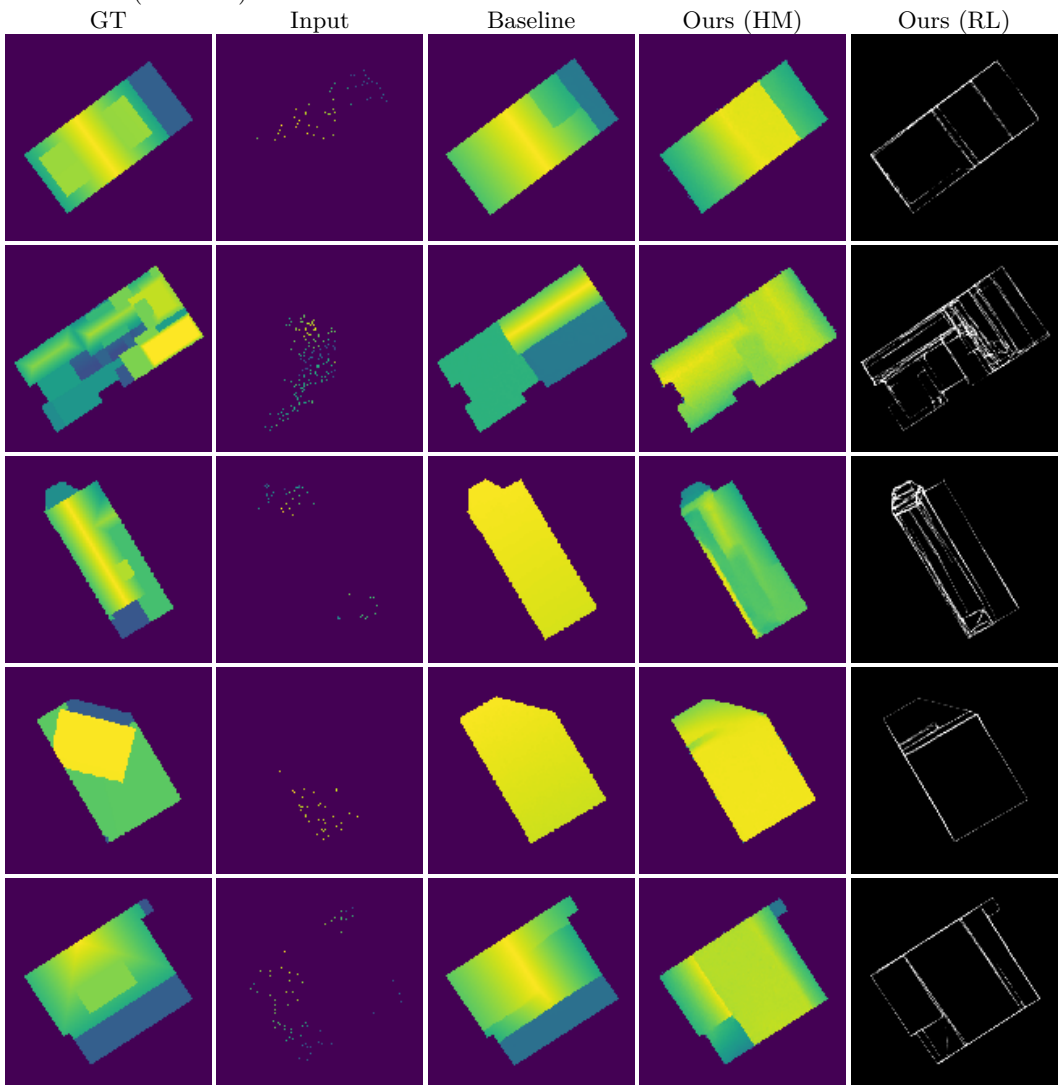


Figure A.2.: Qualitative completion results for the s80_i80 benchmark (Samples 6-10).

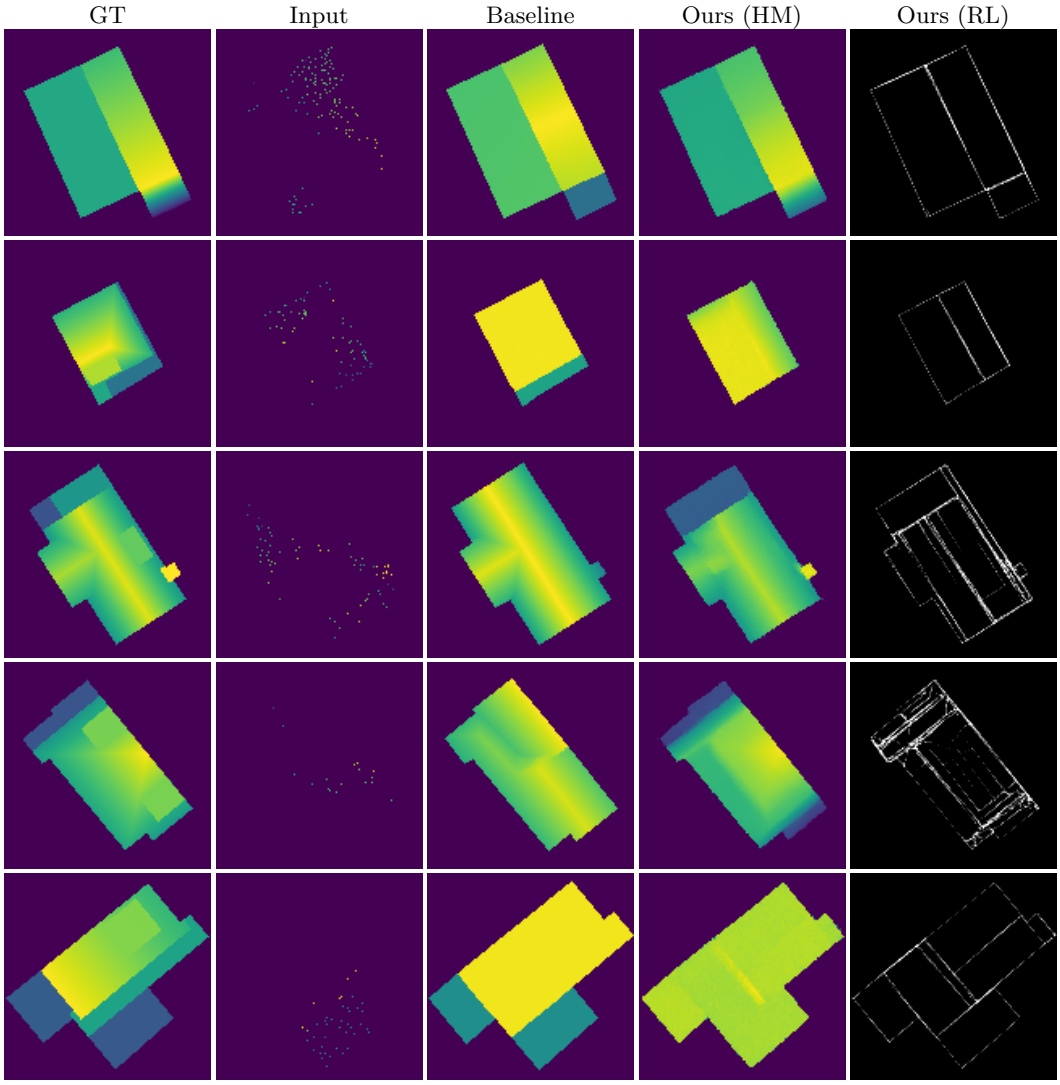


Figure A.3.: Qualitative completion results for the s80_i30 benchmark (Samples 1-5). From left to right: Ground Truth, Corrupted Input, RoofDiffusion (Baseline), Ours (Heightmap), and Ours (Roofline).

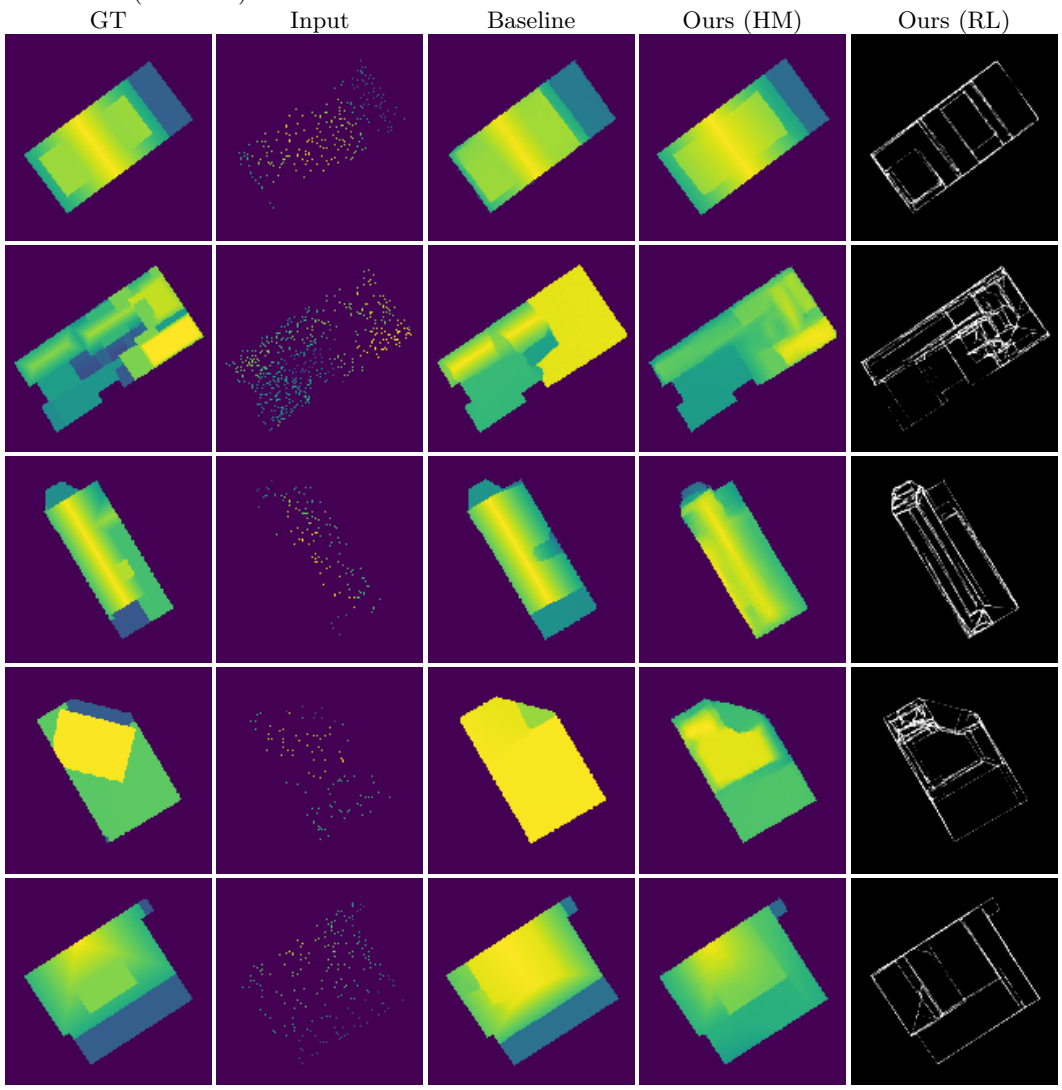


Figure A.4.: Qualitative completion results for the s80_i30 benchmark (Samples 6-10).

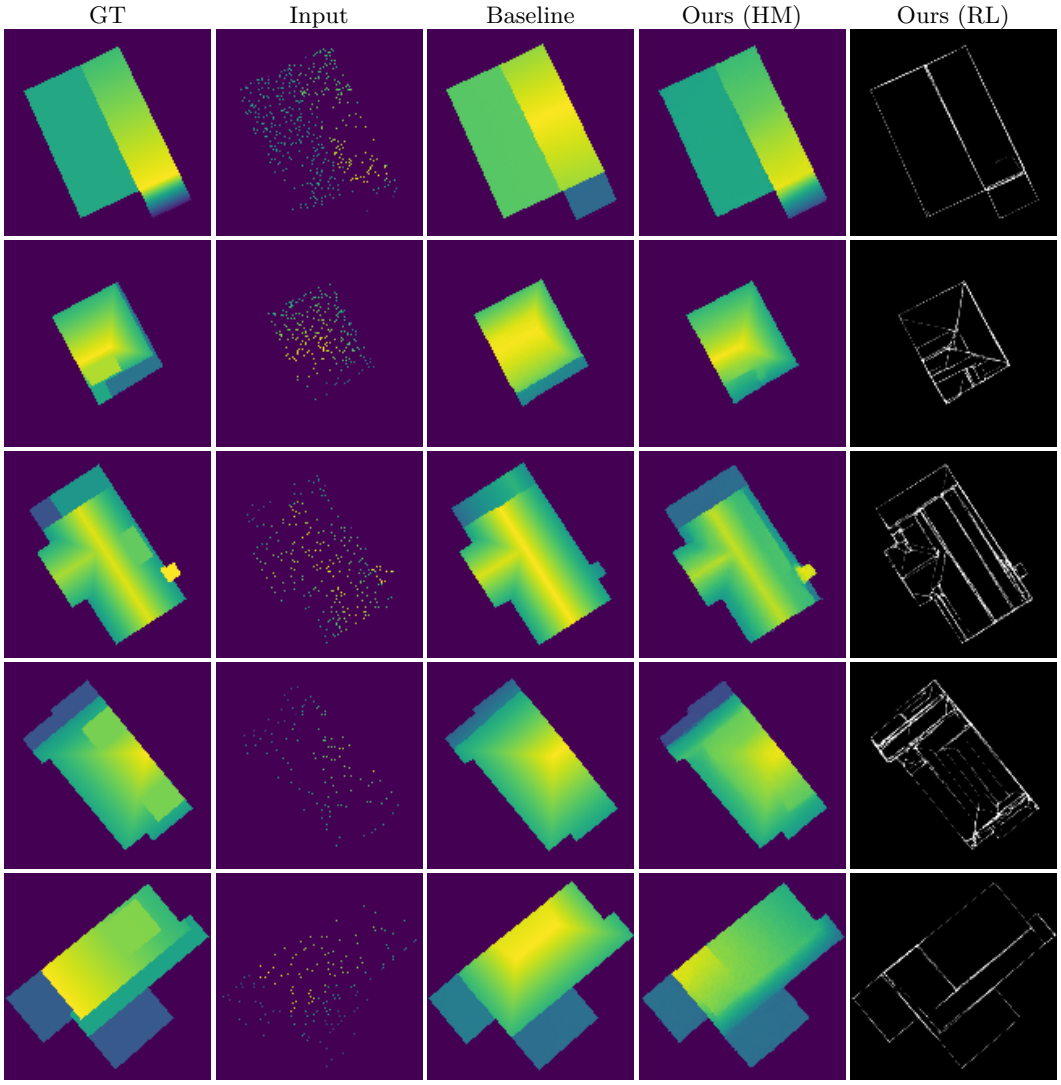


Table A.3.: Architecture of the Bidirectional Control Module (BCM) U-Net.

Layer Type	**Output Resolution**	**Channels**	**Heads**	**Details**
— *Encoder* —				
Input	128x128	3	-	Heightmap + Roofline + Footprint
Conv	128x128	64	-	Initial Convolution
Down Block 1	64x64	64	-	1 x ResBlock, Downsample
Down Block 2	32x32	128	1	1 x ResBlock + Cross-Attention, Downsample
Down Block 3	16x16	256	1	1 x ResBlock + Cross-Attention, Downsample
— *Bottleneck* —				
ResBlock	16x16	512	1	1 x ResBlock + Cross-Attention
— *Decoder* —				
Up Block 1	32x32	256	1	1 x ResBlock + Cross-Attention, Upsample
Up Block 2	64x64	128	1	1 x ResBlock + Cross-Attention, Upsample
Up Block 3	128x128	64	-	1 x ResBlock, Upsample
— *Output Heads* —				
Head 1 (Heightmap)	128x128	1	-	GroupNorm, SiLU, 3x3 Conv
Head 2 (Roofline)	128x128	1	-	GroupNorm, SiLU, 3x3 Conv

Table A.4.: Architecture of the Semantic Encoder.

Module	Layer Name	Output Shape	Details
— Encoder (ResNet-50 based) —			
Input	-	(B, 1, 128, 128)	Low-Res Heightmap
Stem	Sequential	(B, 64, 32, 32)	2 x Conv, MaxPool
Layer 1	Bottleneck x 3	(B, 256, 32, 32)	+ Local Attention
Layer 2	Bottleneck x 4	(B, 512, 16, 16)	+ Local Attention
Layer 3	Bottleneck x 6	(B, 1024, 8, 8)	+ Local Attention
Layer 4	Bottleneck x 3	(B, 2048, 4, 4)	+ SE Block
Pooling	AdaptiveAvgPool	(B, 2048)	Global + Spatial Pooling
FC Layers	Sequential	(B, 768)	3 x Linear, ReLU, Dropout
— Decoder —			
FC Layers	Sequential	(B, 1024, 4, 4)	From Latent to Feature Map
Decoder 1	DecoderBlock	(B, 512, 8, 8)	Upsample, Conv, Attention
Decoder 2	DecoderBlock	(B, 256, 16, 16)	Upsample, Conv, Attention
Decoder 3	DecoderBlock	(B, 128, 32, 32)	Upsample, Conv
Decoder 4	DecoderBlock	(B, 64, 64, 64)	Upsample, Conv
Decoder 5	DecoderBlock	(B, 32, 128, 128)	Upsample, Conv
Final Conv	Sequential	(B, 1, 128, 128)	5 x Conv, Tanh

Table A.5.: Architecture of the Upsampler U-Net for Patch-Based Processing.

Layer Type	Output Resolution	Channels	Heads	Details
— Encoder —				
Input	64x64	2	-	Corrupted Patch + Low-Res Patch
Conv	64x64	64	-	Initial Convolution
Down Block 1	32x32	128	-	2 x ResBlock, Downsample
Down Block 2	16x16	256	-	2 x ResBlock, Downsample
Down Block 3	8x8	512	-	2 x ResBlock, Downsample
Down Block 4	4x4	1024	-	2 x ResBlock, Downsample
— Bottleneck —				
ResBlock	4x4	1024	-	2 x ResBlock
— Decoder —				
Up Block 1	8x8	512	-	2 x ResBlock, Upsample
Up Block 2	16x16	256	-	2 x ResBlock, Upsample
Up Block 3	32x32	128	-	2 x ResBlock, Upsample
Up Block 4	64x64	64	-	2 x ResBlock, Upsample
Conv	64x64	1	-	Final Convolution

Algorithm 3 Spatial Transformer Attention Block

Input: Feature map $x \in \mathbb{R}^{B\times C\times H\times W}$, Context list context $\in \{\mathbb{R}^{B\times C_i\times H_i\times W_i}\}$
Parameters: Number of heads n_h, head channels d_h, dropout rate p, cross-attention only flag
Output: Enhanced feature map $x' \in \mathbb{R}^{B\times C\times H\times W}$
procedure SpatialTransformerAttention(x, context)
    $x_{\text{in}} \leftarrow x$ ▷ Store input for residual connection
    $d_{\text{inner}} \leftarrow n_h \times d_h$ ▷ Inner dimension
    // Normalize and project input
    $x \leftarrow \text{GroupNorm}(x)$
    $x \leftarrow \text{Conv2D}_{1\times1}(x, d_{\text{inner}})$ ▷ Project to inner dimension
    $x \leftarrow \text{Rearrange}(x, \text{'b c h w -> b (h w) c'})$ ▷ Flatten spatial dimensions
    // Process context based on input channels
    context_dims $\leftarrow [64, 128]$ ▷ Supported context dimensions
    ct_index $\leftarrow \text{IndexOf}(\text{context\_dims}, C//4)$ ▷ Find matching context
    ctx $\leftarrow$ context[ct_index] ▷ Select appropriate context
    if ctx is 4D tensor **then**
        ctx $\leftarrow$ ContextCompressor(ctx) ▷ Optional compression
        ctx $\leftarrow \text{Conv2D}_{1\times1}(\text{ctx}, d_{\text{inner}})$ ▷ Project context
        ctx $\leftarrow \text{Rearrange}(\text{ctx}, \text{'b c h w -> b (h w) c'})$ ▷ Flatten context
    end if
    // Apply transformer blocks
    for each block in transformer_blocks **do**
        if cross-attention only mode **then**
            $x \leftarrow \text{BasicCrossTransformerBlock}(x, \text{context} = \text{ctx})$
        else
            $x \leftarrow \text{BasicTransformerBlock}(x, \text{context} = \text{ctx})$ ▷ Self + Cross attention
        end if
    end for
    // Project back and add residual
    $x \leftarrow \text{Rearrange}(x, \text{'b (h w) c -> b c h w'}, h = H, w = W)$
    $x \leftarrow \text{ZeroConv2D}_{1\times1}(x, C)$ ▷ Zero-initialized output projection
    return $x + x_{\text{in}}$ ▷ Residual connection
end procedure
procedure BasicTransformerBlock(x, context)
    $x \leftarrow x + \text{SelfAttention}(\text{LayerNorm}(x))$ ▷ Self-attention with residual
    $x \leftarrow x + \text{CrossAttention}(\text{LayerNorm}(x), \text{context})$ ▷ Cross-attention
    $x \leftarrow x + \text{FeedForward}(\text{LayerNorm}(x))$ ▷ MLP with residual
    return x
end procedure
procedure BasicCrossTransformerBlock(x, context)
    $x \leftarrow x + \text{CrossAttention}(\text{LayerNorm}(x), \text{context})$ ▷ Cross-attention only
    $x \leftarrow x + \text{FeedForward}(\text{LayerNorm}(x))$ ▷ MLP with residual
    return x
end procedure

Table A.6.: Key Training Hyperparameters.

Parameter	**Completion/Roofline**	**Upsampling**
Optimizer	Adam	Adam
Learning Rate	7×10^{-5}	7×10^{-5}
Adam Betas	(0.9, 0.999)	(0.9, 0.999)
Weight Decay	0.0	0.0
LR Schedule	Linear Decay	Linear Decay
Batch Size (effective)	128	8
Gradient Accumulation	4	1
— *Diffusion* —		
Noise Schedule (Train)	Cosine	Linear
Timesteps (Train)	2000	2000
Noise Schedule (Test)	Cosine	Linear
Timesteps (Test)	1000	1000
— *Loss Weights* —		
Joint Loss λ_1 (Heightmap)	1.0 (Implicit)	N/A
Joint Loss λ_2 (Roofline)	1.0 (Implicit)	N/A
Roofline BCE Weight λ	1.0	N/A

Table A.7.: BCM Training Hyperparameters (derived from mom.yaml).

Parameter	**Value**
Optimizer	Adam
Learning Rate	2×10^{-4}
Adam Betas	(0.9, 0.999)
Weight Decay	0.00
LR Scheduler Start Factor	0.1
LR Scheduler Total Iters	20
Batch Size (per GPU)	24
EMA Decay	0.9999
EMA Start Iteration	1
EMA Iteration Frequency	1

Table A.8.: Configuration Differences in Ablation Studies.

Model	**Variant ID**	**Description of Change**
Roof Completion	RC-Full	Complete model with dual-task training and BCM.
	RC-Ablated	Roofline prediction branch and BCM removed.
Upsampling	UP-Full	Complete model with global context and position embeddings.
	UP-NoGC	Global context (semantic encoder) branch removed.
	UP-NoPE	Positional embeddings removed from patch processing.
	UP-Patch	Patch-based processing (64×64 patches, proposed method).
	UP-FullImg	Full-image processing variant (same architecture, no patches).

Bibliography

Agrawal, S. and Gupta, R. D. (2017). Web gis and its architecture: a review. *Arabian Journal of Geosciences*, 10:518.

Agugiaro, G. (2016). Energy planning tools and citygml-based 3d virtual city models: experiences from trento (italy). *Applied Geomatics*, 8:41–56.

Arora, H., Mishra, S., Peng, S., Li, K., and Mahdavi-Amiri, A. (2021). Multimodal shape completion via imle. *arXiv preprint arXiv:2106.16237*.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., and Çöltekin, A. (2015). Applications of 3d city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4):2842–2889.

Cai, Y., Shen, T., Huang, S.-S., and Huang, H. (2023). Self-supervised depth completion guided by 3d perception and geometry consistency.

Chen, J., Yi, J. S. K., Kahoush, M., Cho, E. S., and Cho, Y. K. (2020). Point cloud scene completion of obstructed building facades with generative adversarial inpainting. *Sensors*, 20(18):5029.

Chen, Z., Wang, Y., Nan, L., and Zhu, X. X. (2025). Parametric point cloud completion for polygonal surface reconstruction. https://arxiv.org/abs/2503.08363. CVPR 2025.

Cheng, S., Li, Y., Chen, H., Wang, L., Li, J., Wu, Z., and Chen, C. L. P. (2022). Pbwr: A transformer-based framework for 3d building wireframe reconstruction from point clouds. In *European Conference on Computer Vision*, pages 593–609. Springer.

Cheng, Y.-C., Lee, H.-Y., Tulyakov, S., Schwing, A. G., and Gui, L.-Y. (2023). SDFusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4456–4465.

Cho, W., Ravi, H., Harikumar, M., Khuc, V., Singh, K. K., Lu, J., Inouye, D. I., and Kale, A. (2025). Enhanced controllability of diffusion models via feature disentanglement and realism-enhanced sampling methods.

Chu, R., Xie, E., Mo, S., Li, Z., Nießner, M., Fu, C.-W., and Jia, J. (2023). Diffcomplete: Diffusion-based generative 3d shape completion.

Ding, Z., Zhang, M., Wu, J., and Tu, Z. (2023). Patched Denoising Diffusion Models For High-Resolution Image Synthesis. *arXiv preprint arXiv:2308.01316*.

Gao, W., Peters, R., and Stoter, J. (2024). Building-pcc: Building point cloud completion benchmarks. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-4/W5-2024:179–186.

Gruen, A. (2008). Reality-based generation of virtual environments for digital earth. *International Journal of Digital Earth*, 1(1):88–106.

Haala, N. and Kada, M. (2010). An update on automatic 3d building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6):570–580.

He, Y., Yang, S., Chen, H., Cun, X., Xia, M., Zhang, Y., Wang, X., He, R., Chen, Q., and Shan, Y. (2023). Scalecrafter: Tuning-free higher-resolution visual generation with diffusion models.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*.

Ho, J. and Salimans, T. (2022). Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. (2021). Improved denoising diffusion probabilistic models. *International Conference on Machine Learning (ICML)*, pages 4181–4191.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. (2022). Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Holzmann, T., Maurer, M., Fraundorfer, F., and Bischof, H. (2018). Semantically aware urban 3d reconstruction with plane-based regularization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 468–483, Munich, Germany. Springer International Publishing.

Horita, D., Yang, J., Chen, D., Koyama, Y., Aizawa, K., and Sebe, N. (2023). A structure-guided diffusion model for large-hole image completion.

Huang, J., Stoter, J., Peters, R., and Nan, L. (2022). City3d: Large-scale building reconstruction from airborne lidar point clouds. *Remote Sensing*, 14(9):2254.

Huang, T., Yan, Z., Zhao, Y., and Lee, G. H. (2025). Compc: Completing a 3d point cloud with 2d diffusion priors.

Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., and Irani, M. (2022). Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276*.

Keys, R. G. (1981). Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160.

Khan, M. F. F., Troncoso Aldas, N. D., Kumar, A., Advani, S., and Narayanan, V. (2021). Sparse to dense depth completion using a generative adversarial network with intelligent sampling strategies. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 5528–5536, New York, NY, USA. Association for Computing Machinery.

Lafarge, F. and Mallet, C. (2012). Creating large-scale city models from 3d point clouds: A robust approach with hybrid representation. *International Journal of Computer Vision*, 99:69–85.

Lee, Y., Park, S., Kang, B., and Park, H. (2022). Confidence guided depth completion network.

Li, F. et al. (2023a). Unidiffuser: One transformer fits all distributions in multi-modal diffusion. *arXiv preprint arXiv:2303.06555*.

Li, Y., Dou, Y., Chen, X., Ni, B., Sun, Y., Liu, Y., and Wang, F. (2023b). 3dqd: Generalized deep 3d shape prior via part-discretized diffusion process.

Liu, H., Wang, Y., Qian, B., Wang, M., and Rui, Y. (2024a). Structure matters: Tackling the semantic discrepancy in diffusion models for image inpainting.

Liu, X. (2008). Liu x. airborne lidar for dem generation: Some critical issues. progress in physical geography. *Progress in Physical Geography - PROG PHYS GEOG*, 32:31–49.

Liu, Y., Cheng, M.-M., Hu, X., Wang, K., and Bai, X. (2017). Richer convolutional features for edge detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3000–3009.

Liu, Y., Obukhov, A., Wegner, J. D., and Schindler, K. (2024b). Point2building: Reconstructing buildings from airborne lidar point clouds.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows.

Lo, K. S.-H., Peters, J., and Spellman, E. (2024). Roofdiffusion: Constructing roofs from severely corrupted point data via diffusion. *arXiv e-prints*.

Long, C., Zhang, W., Chen, Z., Wang, H., Liu, Y., Tong, P., Cao, Z., Dong, Z., and Yang, B. (2024). Sparsedc: Depth completion from sparse and non-uniform inputs. *Information Fusion*, 110:102470.

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., and Gool, L. V. (2022). Repaint: Inpainting using denoising diffusion probabilistic models.

Luo, S. and Hu, W. (2021). Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Musialski, P., Wonka, P., Aliaga, D. G., Wimmer, M., Gool, L., and Purgathofer, W. (2013). A survey of urban reconstruction. *Comput. Graph. Forum*, 32(6):146–177.

Nan, L. and Wonka, P. (2017). Polyfit: Polygonal surface reconstruction from point clouds. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2372–2380.

Nazir, D., Liwicki, M., Stricker, D., and Afzal, M. Z. (2022). Semattnet: Towards attention-based semantic aware guided depth completion.

Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639.

Qian, Y., Zhang, H., and Furukawa, Y. (2021). Roof-gan: Learning to generate roof geometry and relations for residential houses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2796–2805.

Ren, J., Zhang, B., Wu, B., Huang, J., Fan, L., Ovsjanikov, M., and Wonka, P. (2021). Intuitive and efficient roof modeling for reconstruction and synthesis. *arXiv preprint arXiv:2109.07683*.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models.

Rottensteiner, F. (2005). Automatic generation of high-quality building models from lidar data. *IEEE Computer Graphics and Applications*, 23(6):42–50.

Rottensteiner, F. (2009). Isprs test project on urban classification and 3d building reconstruction: Evaluation of building reconstruction results. Test project report, International Society for Photogrammetry and Remote Sensing (ISPRS).

Saharia, C., Chan, W., Chang, H., Lee, C. A., Ho, J., Salimans, T., Fleet, D. J., and Norouzi, M. (2022a). Palette: Image-to-image diffusion models.

Saharia, C. et al. (2022b). Palette: Image-to-image diffusion models. In *Advances in Neural Information Processing Systems*.

Selvaraju, P. et al. (2021). Buildingnet: Learning to label 3d buildings. In *Proceedings of the IEEE International Conference on Computer Vision*.

Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, pages 517–524. ACM.

Tesema, K. W., Hill, L., Jones, M. W., Ahmad, M. I., and Tam, G. K. L. (2024). Point cloud completion: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 30(10):6880–6899.

Truong Giang, K., Song, S., Kim, D., and Choi, S. (2021). Sequential depth completion with confidence estimation for 3d model reconstruction. *IEEE Robotics and Automation Letters*, 6(2):327–334.

Tsuji, Y., Chishiro, H., and Kato, S. (2018). Non-guided depth completion with adversarial networks. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1109–1114.

Turkowski, K. (1990). Filters for common resampling tasks. In *Graphics Gems*, pages 147–165, San Diego, CA, USA. Academic Press Professional, Inc.

Wang, H., Yang, M., Zheng, X., and Hua, G. (2025). Scale propagation network for generalizable depth completion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):1908–1922.

Wang, R., Peethambaran, J., and Chen, D. (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):606–627.

Wang, Z., Jiang, Y., Zheng, H., Wang, P., He, P., Wang, Z., Chen, W., and Zhou, M. (2023). Patch diffusion: Faster and more data-efficient training of diffusion models.

Weibel, R. and Heller, M. (1992). Digital terrain modelling. *Geographical Information Systems*, 1:269–297.

Wichmann, A. et al. (2018). Roofn3d: A dataset for roof reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 145:85–100.

Wu, W., Qi, Z., and Fuxin, L. (2020). Pointconv: Deep convolutional networks on 3d point clouds.

Xie, C., Wang, C., Zhang, B., Yang, H., Chen, D., and Wen, F. (2021). Style-based point generator with adversarial rendering for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4619–4628.

Xie, E., Chen, J., Chen, J., Cai, H., Tang, H., Lin, Y., Zhang, Z., Li, M., Zhu, L., Lu, Y., and Han, S. (2024). Sana: Efficient high-resolution image synthesis with linear diffusion transformers.

Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., and Zhou, J. (2021). Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12498–12507.

Yu, X., Rao, Y., Wang, Z., Lu, J., and Zhou, J. (2023). Adapointr: Diverse point cloud completion with adaptive geometry-aware transformers.

Yuan, W. et al. (2018). Pcn: Point completion network. In *International Conference on 3D Vision*.

Zhang, L., Rao, A., and Agrawala, M. (2023). Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*.

Zhang, S., Chen, Z., Zhao, Z., Chen, Y., Tang, Y., and Liang, J. (2024a). Hidiffusion: Unlocking higher-resolution creativity and efficiency in pretrained diffusion models.

Zhang, X., Feng, Y., Li, S., Zou, C., Wan, H., Zhao, X., Guo, Y., and Gao, Y. (2021). View-guided point cloud completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15890–15899.

Zhang, Y., Zhang, J., Li, H., Wang, Z., Hou, L., Zou, D., and Bian, L. (2024b). Diffusion-based blind text image super-resolution.

Zheng, Y. et al. (2023). Few-shot depth completion using denoising diffusion probabilistic models. *arXiv preprint arXiv:2303.10326*.

Zhou, H., Cao, Y., Chu, W., Zhu, J., Lu, T., Tai, Y., and Wang, C. (2022). Seedformer: Patch seeds based point cloud completion with upsample transformer.

Zou, K.-K., Zhang, Z., Ma, W.-C., Wu, Z., and Pan, X. (2020). Learning to parse wireframes in images of man-made environments. In *European Conference on Computer Vision*, pages 684–701. Springer.

Colophon

This document was typeset using LaTeX, using the KOMA-Script class `scrbook`. The main font is Palatino.