EdgeDiff: Edge-aware Diffusion Network for Building Reconstruction from Point Clouds

Supplementary Material

A. Overview

The supplementary material is organized into the following sections:

- **Sec. B.** Detailed training and inference algorithms.
- **Sec. C.** Qualitative evaluation of the proposed edge attention mechanism.
- **Sec. D.** Comprehensive qualitative comparisons against state-of-the-art method.
- **Sec. E.** Evaluation of model generalization on unseen datasets.
- **Sec. F.** Reconstruction results demonstrated across the entire Tallinn city.

B. Training and Inference Details

B.1. Training Details

Training Algorithm. Algorithm 1 provides the pseudo-code of EdgeDiff training procedure. EdgeDiff first diffuse the ground truth wireframe into a distribution of random gaussian noise. The model is then trained to reverse this process, allowing it to learn the wireframe structure of the point clouds.

Training Loss. Following PBWR [1], we employ a comprehensive loss function that combines L1 distance loss for midpoint and component regression ($\mathcal{L}_{mid}$, $\mathcal{L}_{comp}$), cross-entropy loss for confidence scores and quadrant classification ($\mathcal{L}_{con}$, $\mathcal{L}_{quad}$), edge similarity loss ($\mathcal{L}_{sim}$) for structural refinement, and mean squared error loss ($\mathcal{L}_{attn}$) for edge attention supervision:

$$\begin{aligned}\mathcal{L} =&\lambda_{mid}\mathcal{L}_{mid} + \lambda_{comp}\mathcal{L}_{comp} + \lambda_{con}\mathcal{L}_{con} + \\ &\lambda_{quad}\mathcal{L}_{quad} + \lambda_{sim}\mathcal{L}_{sim} + \lambda_{attn}\mathcal{L}_{attn}\end{aligned} \tag{1}$$

where we set $\lambda_{mid} = 5$, $\lambda_{comp} = 1$, $\lambda_{con} = 1$, $\lambda_{quad} = 2$, $\lambda_{sim} = 2$, and $\lambda_{attn} = 5$ to balance the trade-off between geometric accuracy and structural consistency in the wireframe reconstruction process.

B.2. Inference Details

Inference Algorithm. The EdgeDiff inference process follows a denoising paradigm that transforms Gaussian noise into precise wireframes through iterative refinement, as detailed in Algorithm 2.

Edge Renewal. At each inference step, the network generates N_{eval} edges with corresponding confidence scores. We implement an adaptive filtering mechanism that retains edges whose confidence scores exceed a threshold, while replacing lower-confidence edges with new parameters sampled from a gaussian distribution.

Sampling Strategy. We analyzed the impact of sampling strategy on reconstruction quality, as shown in Fig. 1. Our experiments demonstrate that while using either DDIM [2] or edge renewal mechanism alone leads to marginal performance degradation in EF1 metrics, their combination yields superior results. Notably, our analysis reveals that optimal performance is achieved with single-step denoising, with additional iterations yielding negligible improvements. This efficiency can be attributed to three key design choices: the edge attention mechanism that provides strong geometric priors, the midpoint-offset parameterization that simplifies optimization, and the edge renewal mechanism that ensures high-quality predictions. These architectural decisions collectively demonstrate the effectiveness of our noise-to-wireframe paradigm.

Algorithm 1 EdgeDiff Training

```
def train_loss(point_clouds, gt_wireframes):
    """
    point_clouds: [B, N, 7] # points with RGB and
        intensity
    gt_wireframes: [B, M, 6] # M edges, each with
        midpoint-offset
    # B: batch size
    # N: number of points
    # M: number of edges
    """

    # Encode point cloud features and edge_attention
    enc_xyz, f_embed = point_embedding(point_clouds)
    f_enc = point_encoder(f_embed)
    pred_attn = edge_attention_generation(f_embed)
    gt_attention = gt_edge_attention_generation(
        gt_wireframes)

    # Pad gt wireframes to N_train
    wf = pad_wireframe(gt_wireframes) # [B, N_train, 6]

    # Signal scaling
    wf = wf * scale # scale=2.0 as default

    # Corrupt wireframes with noise
    t = randint(0, T) # timestep
    eps = normal(mean=0, std=1) # noise: [B,
        num_queries, 6]
    wf_noised = sqrt(alpha_bar[t]) * wf + \
    sqrt(1 - alpha_bar[t]) * eps

    # Predict
    wf_pred = edge_aware_denoising(wf_noised, f_enc, t,
         query_xyz, pred_attn)

    # Calculate losses
    loss = wireframe_loss(wf_pred, gt_wireframes) + \
         attention_loss(pred_attn, gt_attention)

    return loss
```

Algorithm 2 EdgeDiff Sampling

```
def infer(point_clouds, sampling_steps, T):
   """
   point_clouds: [B, N, 7] # points with RGB and
       intensity
   sampling_steps: number of sample steps
   T: total diffusion steps
   """

   # Encode point cloud features and edge attention
   enc_xyz, f_embed = point_embedding(point_clouds)
   f_enc = point_encoder(f_embed)
   pred_attn = edge_attention_generation(f_embed)

   # Initialize with random noise
   wf_t = normal(mean=0, std=1) # [B, N_eval, 6]

   # Sample timesteps uniformly
   times = reversed(linspace(-1, T, sampling_steps))
   time_pairs = list(zip(times[:-1], times[1:]))

   for t_curr, t_next in zip(time_pairs):
      # Edge-aware denoising prediction
      wf_pred = edge_aware_denoising(wf_t, f_enc,
          t_curr, query_xyz, pred_attn)

      # DDIM update step
      wf_t = ddim_step(wf_t, wf_pred, t_curr, t_next)

      # Edge renewal
      wf_t = renew_edges(wf_t)

   return wf_pred
```

C. Visualization of Edge Attention

We present comprehensive visualizations of edge attention weights in Fig. 2. Our supervised edge attention mechanism effectively captures geometric features along building edges, showing strong correlation with ground truth patterns. This enables accurate wireframe reconstruction while preserving structural integrity. In contrast, unsupervised attention weights exhibit scattered patterns, inadequately highlighting edge-relevant features. The clear distinction between attention maps empirically validates our design. Furthermore, improved reconstruction quality through supervised attention underscores the importance of edge-aware feature learning in wireframe denoising.

D. Qualitative Comparison with SOTA

Fig. 3 presents extensive visual comparisons against PBWR [1] on diverse building examples. Our EdgeDiff demonstrates superior performance by accurately capturing fine-grained roof details while maintaining global structural coherence. The edge-aware denoising process enables robust reconstruction of complex roof structures and preserves building topology, particularly in densely connected regions. These qualitative results validate the effectiveness of our noise-to-wireframe paradigm in generating high-fidelity building reconstructions from point clouds.

E. Model Generalization

To further evaluate the generalization capability of our model, we perform extensive experiments on additional

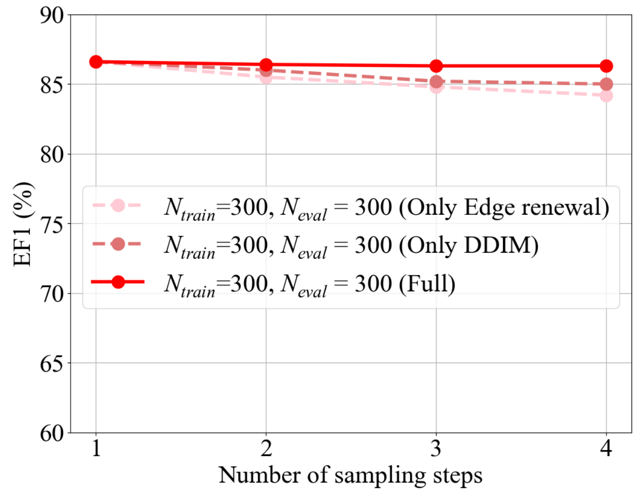

Figure 1. Evaluation of the contributions of the DDIM and Edge renewal mechanism on performance.

datasets, including Hiiumaa, Keila, Loksa, and Sillamäe. These datasets contain only point clouds and lack ground truth wireframe annotations. Hence, we apply the model trained on the Building3D dataset [3] to these datasets without fine-tuning. The qualitative results, shown in Fig. 4, Fig. 5, Fig. 6, and Fig. 7, demonstrate that our method generalizes effectively to unseen datasets.

F. Large-scale Reconstruction Results of Tallinn City

In Fig. 8, we showcase large-scale building reconstruction results on the Tallinn city dataset [3], demonstrating EdgeDiff's capability to reconstruct complex architectural structures across an extensive urban region comprising 36,084 buildings. We present close-up views of a specific region, followed by detailed visualization of an individual building, including its point cloud, predicted wireframe, and predicted mesh representation.

References

[1] Shangfeng Huang, Ruisheng Wang, Bo Guo, and Hongxin Yang. Pbwr: Parametric-building-wireframe reconstruction from aerial lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27778–27787, 2024. 1, 2, 4

[2] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *ICLR*, 2021. 1

[3] Ruisheng Wang, Shangfeng Huang, and Hongxin Yang. Building3d: A urban-scale dataset and benchmarks for learning roof structures from point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20076–20086, 2023. 2

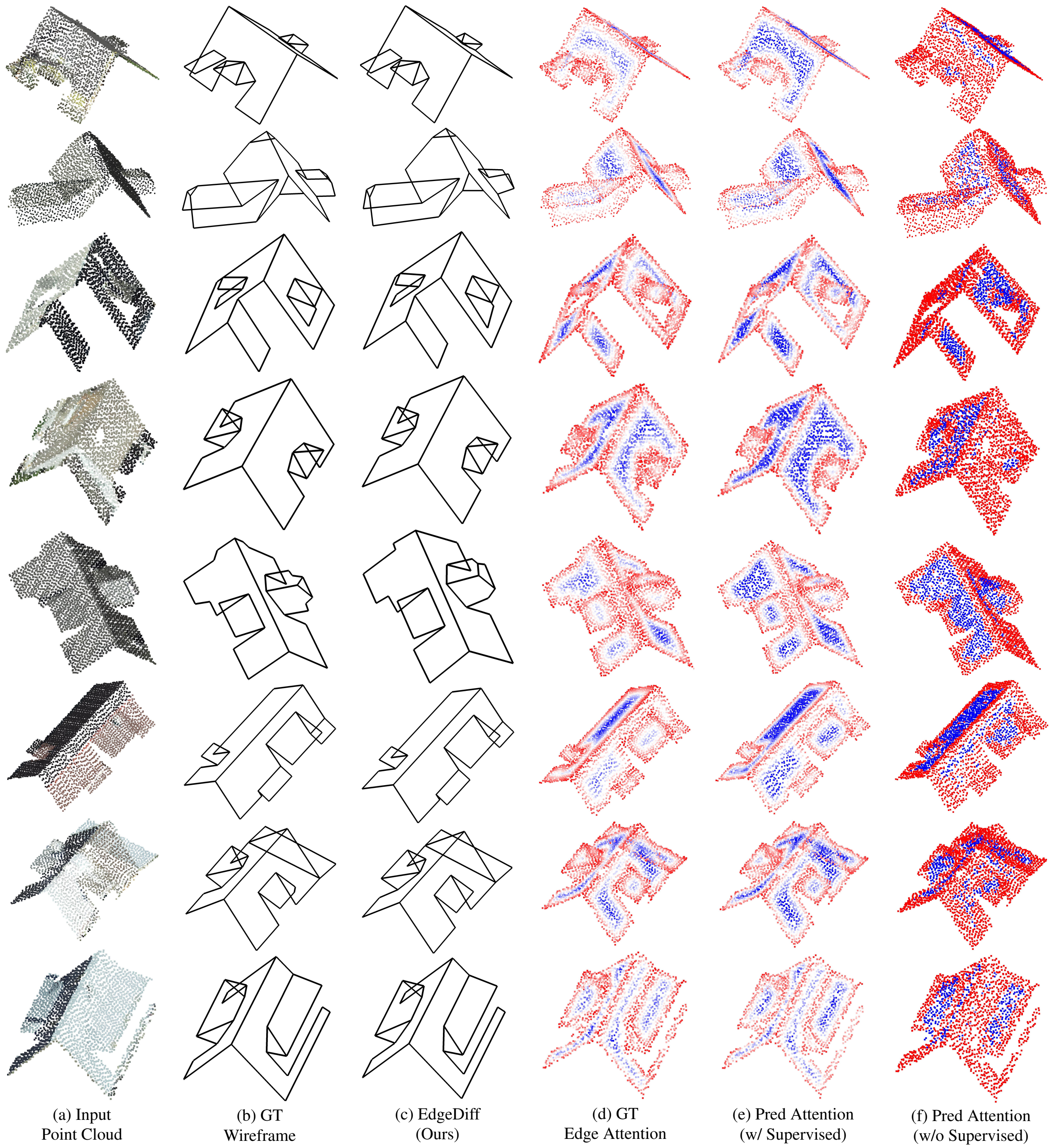

Figure 2. Comparison of supervised and unsupervised edge attention weights in wireframe reconstruction. Blue indicates lower weights, while red represents higher weights. Default settings of EdgeDiff (c) use supervised attention (e).

Figure 3. Qualitative comparison with the existing state-of-the-art method.

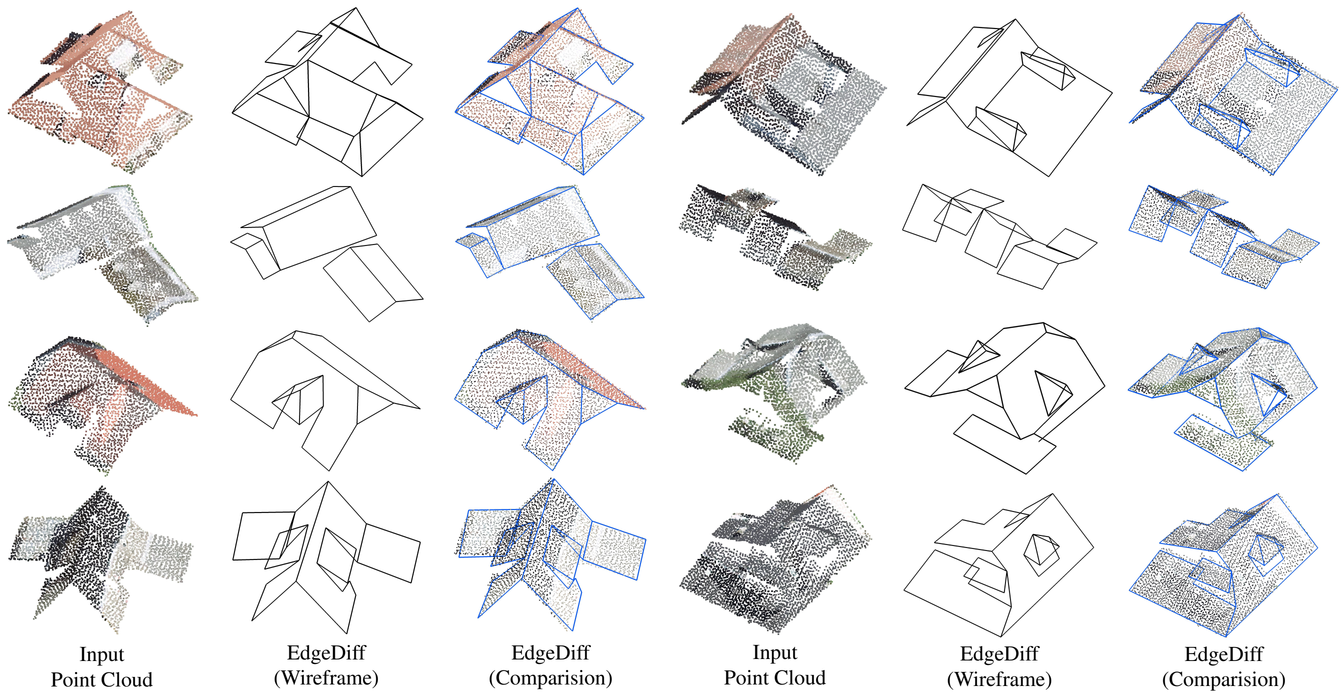

Figure 4. Visualization of model generalization. Qualitative results on the Hiiumaa dataset.

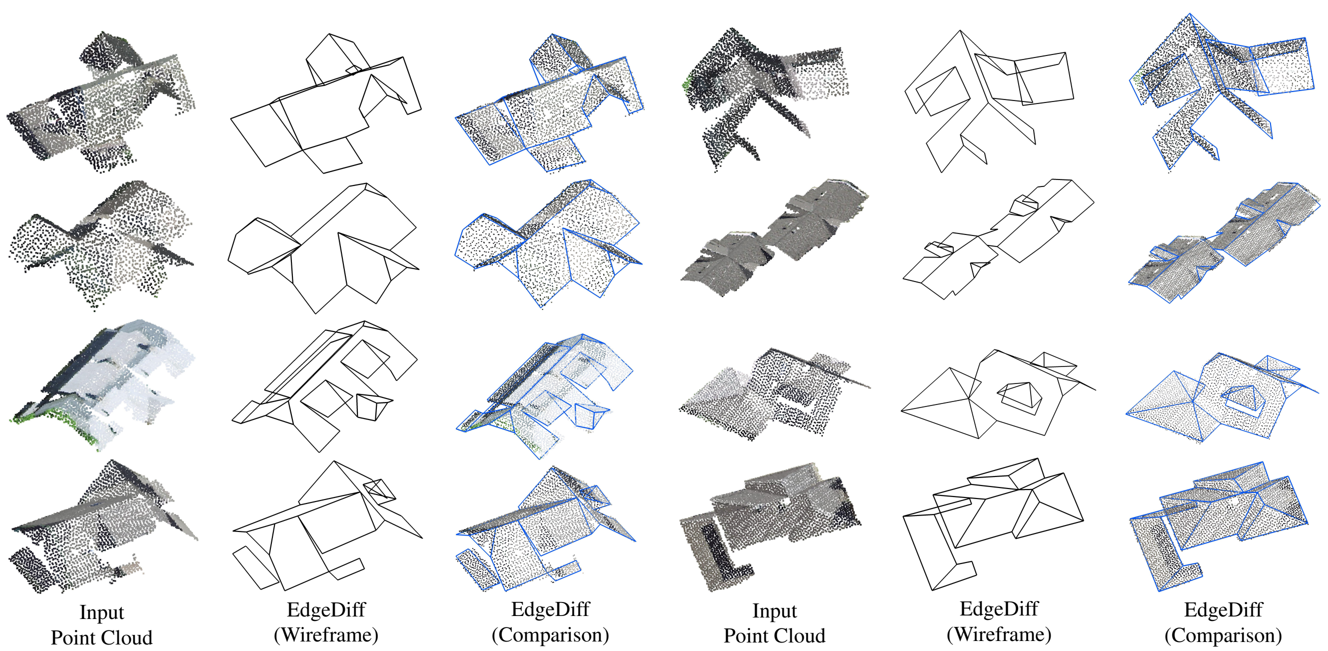

Figure 5. Visualization of model generalization. Qualitative results on the Keila dataset.

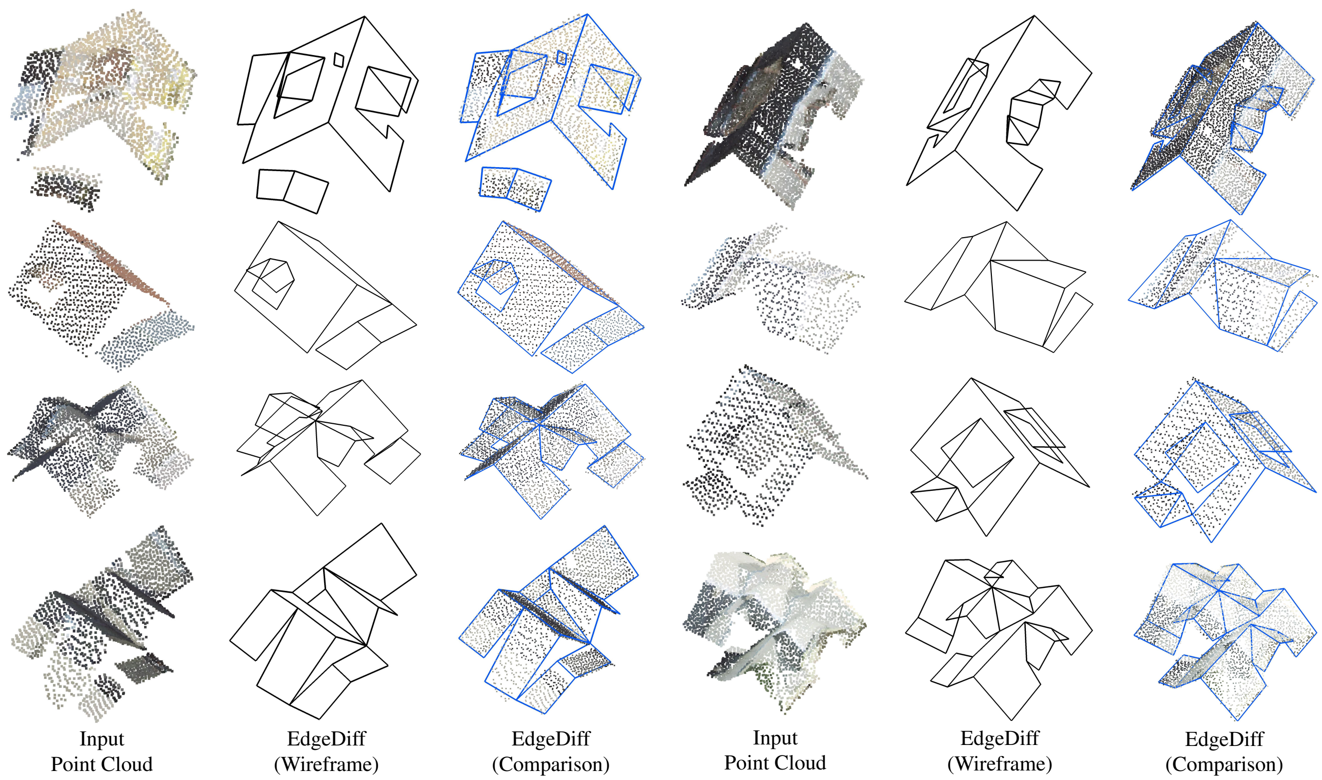

Figure 6. Visualization of model generalization. Qualitative results on the Loksa dataset.

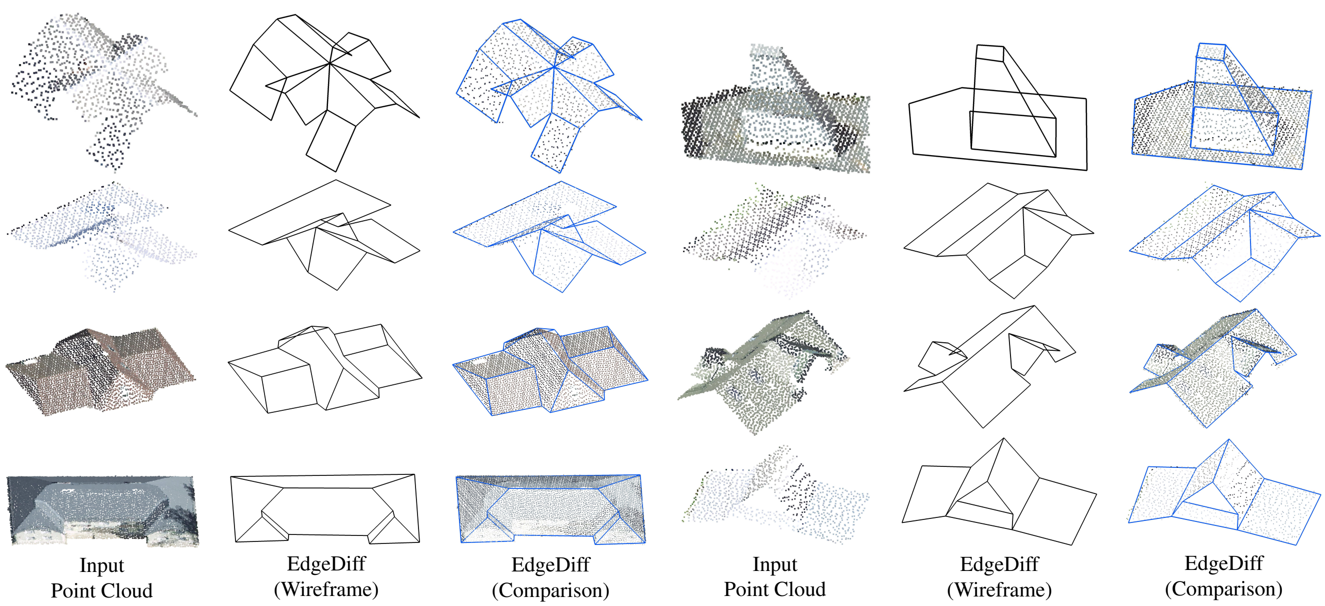

Figure 7. Visualization of model generalization. Qualitative results on the Sillamae dataset.

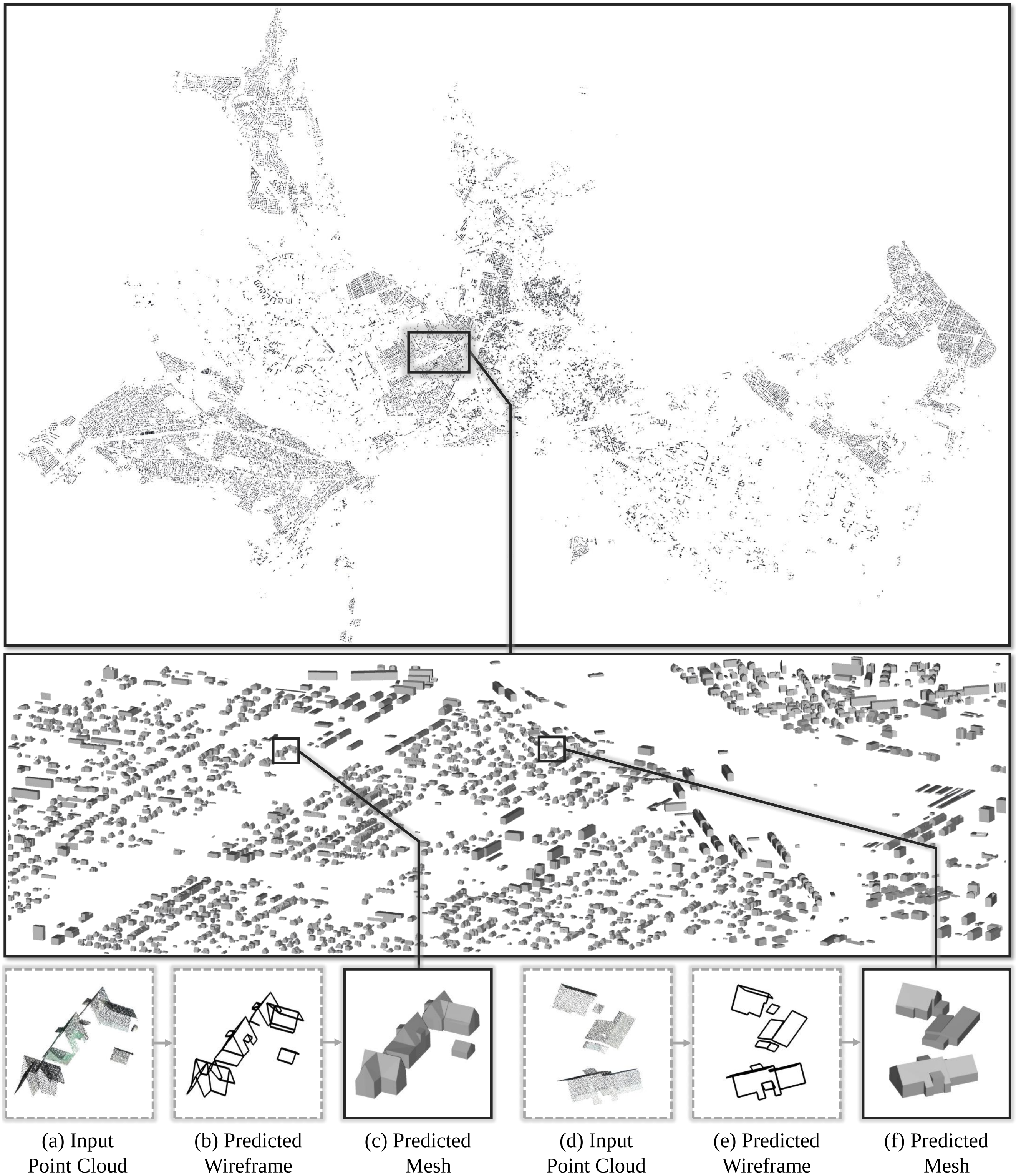

Figure 8. Reconstruction of Tallinn city buildings with EdgeDiff.