PatchScene: Patch-based Voxel Diffusion for Large-Scale Scene Completion

Supplementary Material

6. Method Details

6.1. Voxel Diffusion Model

The denoising network of PatchScene is based on a 3D U-Net architecture consisting of five hierarchical levels, with channel dimensions of (32, 64, 128, 256, 256), respectively. During the denoising process, the time embedding and positional embedding are summed and injected into each level of the network to condition the feature transformation.

For the sparse input point cloud, PatchScene first encodes it and performs a forward pass through the U-Net, during which the intermediate features at all levels are cached. When processing noisy inputs, these cached sparse-conditioning features are added to the corresponding inputs at each U-Net level, enabling the network to produce x0-predictions conditioned on the sparse observations.

During inference, a pretrained denoising network is used to convert the predicted noise into the final completed point cloud, effectively reconstructing high-resolution geometry from the noisy and incomplete measurements.

6.2. Position Embedding

To preserve the absolute spatial context of local crops within the global Bird's-Eye-View (BEV) coordinate system, we introduce a Global-Aware Patch Position Embedding strategy. Unlike standard Vision Transformers that assign fixed positional indices to patches within a normalized image grid, our approach handles dynamic crops sampled from arbitrary locations in the large-scale scene.

More formally, consider the global BEV domain discretized into a grid of resolution $H \times W$. We initialize a learnable global position embedding table $\mathbf{E}_{\text{global}} \in \mathbb{R}^{H \times W \times C}$, where C denotes the embedding dimension. Each entry in $\mathbf{E}_{\text{global}}$ corresponds to a unique voxel coordinate on the BEV plane.

Given a local patch p defined by its top-left global coordinates (x_p, y_p) and spatial dimensions $h \times w$, we explicitly retrieve the spatial priors from the global table. Instead of assigning a single discrete ID to the patch, we first extract the sub-region of embeddings corresponding to the patch's physical coverage:

$$\mathbf{E}^{(p)}_{\text{local}} = \text{Crop}(\mathbf{E}_{\text{global}}, x_p, y_p, h, w) \in \mathbb{R}^{h \times w \times C}, \quad (10)$$

where $\text{Crop}(\cdot)$ denotes the slicing operation: $\mathbf{E}_{\text{global}}[y_p : y_p + h,\ x_p : x_p + w]$.

To obtain a unified position descriptor invariant to the internal spatial structure of the patch, we aggregate the dense spatial embeddings via Global Average Pooling (GAP):

$$\mathbf{e}_p = \frac{1}{h \cdot w} \sum_{i=1}^{h} \sum_{j=1}^{w} \mathbf{E}^{(p)}_{\text{local}}(i, j), \quad (11)$$

where $\mathbf{e}_p \in \mathbb{R}^C$ represents the final position embedding vector for patch p. This vector is subsequently fused with the patch's feature representation. This formulation ensures that the model remains aware of the absolute global location of each patch, enabling effective spatial reasoning across disjointed local views.

6.3. Completion Upsampling

Although the denoising network recovers the underlying geometric structures, the output point cloud may exhibit non-uniform density distribution due to the irregularity of the sampling process. To mitigate this issue and generate a high-fidelity representation suitable for visualization and downstream tasks, we employ a voxel-guided uniform resampling strategy as a post-processing step.

To formalize the procedure, the denoised point cloud is denoted as $\mathcal{P}_{raw} \in \mathbb{R}^{M \times 3}$. The 3D space is then discretized into a voxel grid $\mathcal{G}$ with resolution r (e.g., $r = 0.15625$m). We identify the set of occupied voxels $\mathcal{O}$ by mapping each point $p \in \mathcal{P}_{raw}$ to its corresponding voxel index $\mathbf{v}_i \in \mathbb{Z}^3$:

$$\mathcal{O} = \{\mathbf{v}_i \mid \exists p \in \mathcal{P}_{raw}, \lfloor (p - p_{min})/r \rfloor = \mathbf{v}_i\}, \quad (12)$$

where p_{min} represents the origin of the bounding volume. This step effectively consolidates the local geometry and eliminates density variations within local neighborhoods.

To achieve a target density of N_{target} points (e.g., 10^6), we perform uniform stochastic sampling within each activated voxel. The number of points to be generated per voxel is calculated as $k = \lceil N_{target}/|\mathcal{O}| \rceil$. For each occupied voxel $\mathbf{v} \in \mathcal{O}$ with geometric center $\mathbf{c}_\mathbf{v}$, we generate a set of dense points $\{q_j\}_{j=1}^{k}$ via:

$$q_j = \mathbf{c}_\mathbf{v} + \delta_j, \quad \text{where } \delta_j \sim \mathcal{U}\left[-\frac{r}{2}, \frac{r}{2}\right]^3. \quad (13)$$

Here, $\mathcal{U}$ denotes the uniform distribution. This process ensures that the final point cloud $\mathcal{P}_{dense}$ is uniformly distributed across the underlying surface manifold, filling minor sparsity gaps while preserving the global structure reconstructed by the network.

To validate the efficacy of the proposed Voxel-guided Density Refinement strategy, we perform a quantitative comparison between completion results generated with the upsampling module enabled and those obtained without it, as reported in Table 6. Empirical results demonstrate

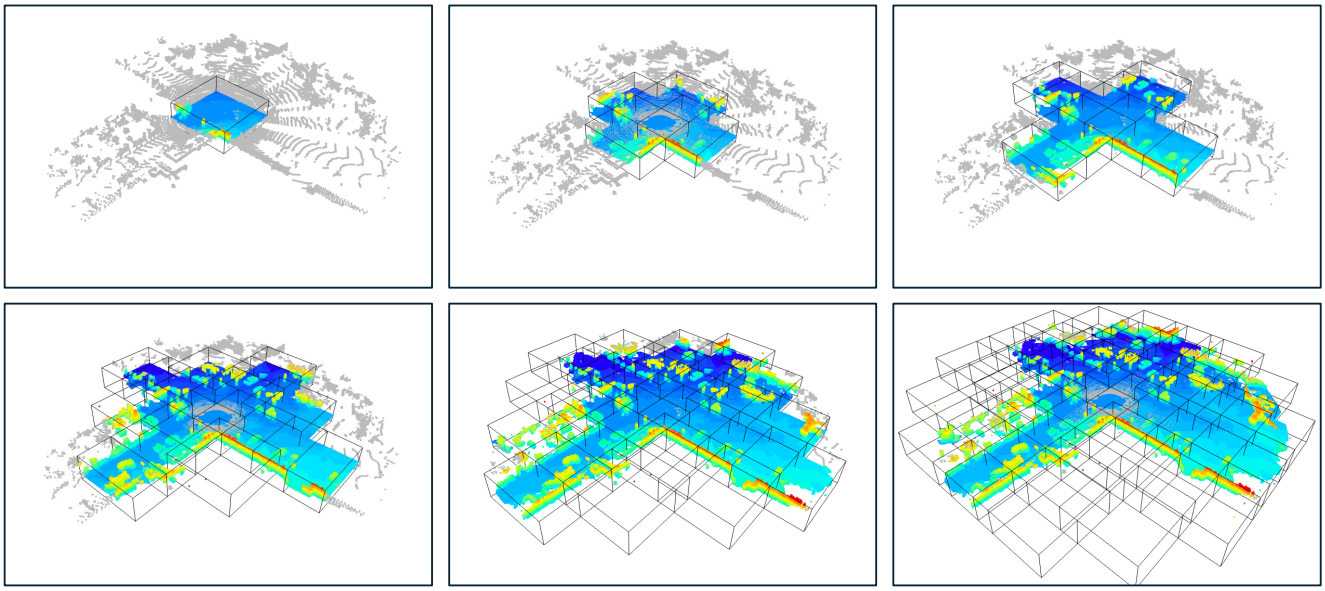

Figure 6. Step-by-step visualization of the completion process.

that the refined output yields consistent improvements in geometric fidelity and distributional alignment. In particular, the proposed strategy delivers superior performance in terms of Chamfer Distance and Jensen–Shannon Divergence, evaluated both in 3D space and in Bird's-Eye-View projections. This confirms the importance of density homogenization for achieving high-quality point cloud completion.

Model	**CD↓**	**JSD 3D↓**	**JSD BEV↓**
w/o Upsampling	0.331	0.456	0.380
Upsampling	0.319	0.444	0.371

Table 6. Ablation of Completion Upsampling

6.4. Diffusion Generation Direction

In the main paper, we introduced the Annular-Flow Diffusion Completion strategy, motivated by the radial density decay of LiDAR sensors. This section provides the formal definition of the annular regions $\{\mathcal{R}_\ell\}$ and outlines the corresponding inference procedure.

The concentric annular regions $\{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_L\}$ are constructed based on the *Chebyshev distance* (or topological distance) from the sensor center on the grid graph.

The grid coordinates of the ego-vehicle (or the scene center) are denoted by (c_x, c_y). The topological distance of a grid location $D(p_{i,j})$ from this center is defined as

$$D(p_{i,j}) = \max(|i - c_x|, |j - c_y|). \tag{14}$$

Accordingly, the ℓ-th annular region $\mathcal{R}_\ell$ is defined as the set of patches with a specific distance from the center. In our implementation, as illustrated in Fig. 6, we group these patches into discrete inference steps:

$$\mathcal{R}_\ell = \{p_{i,j} \in \mathcal{P} \mid D(p_{i,j}) = \ell - 1\}. \tag{15}$$

6.5. Parallel Acceleration

While the *Annular-Flow* strategy imposes a strict sequential dependency between concentric rings (i.e., the generation of $\mathcal{R}_\ell$ relies on the completion of $\mathcal{R}_{\ell-1}$), the inference process for patches *within* the same ring is spatially decoupled. Specifically, conditioned on the fixed context provided by the inner completed regions $\mathcal{C}_{\ell-1} = \bigcup_{k=0}^{\ell-1} \mathcal{R}_k$, the posterior distributions of patches in the current ring $\mathcal{R}_\ell$ are mutually independent.

Leveraging this independence, we implement a parallelized inference scheme to expedite the inference. Instead of synthesizing patches serially, which would otherwise introduce substantial latency in large-scale scenes, we aggregate all $N_\ell = |\mathcal{R}_\ell|$ patches belonging to the current ring into a single batch tensor $\mathbf{B}_\ell \in \mathbb{R}^{N_\ell \times C \times H \times W}$. The diffusion model ϵ_θ processes this batch simultaneously in a single forward pass on the GPU:

$$\mathbf{z}_{t-1}^{(\mathcal{R}_\ell)} \leftarrow \text{Denoise}(\mathbf{z}_t^{(\mathcal{R}_\ell)}, \mathbf{x}_{obs}^{(\mathcal{R}_\ell)}, \mathcal{C}_{\ell-1}), \tag{16}$$

where $\mathbf{z}^{(\mathcal{R}_\ell)}$ represents the batched latent states of all patches in ring ℓ.

This strategy substantially reduces wall-clock inference time by consolidating the patches within each ring into a single batched forward pass. Concretely, the number of model evaluations decreases from N_{total} (one pass per patch under a fully serial scheme) to only L (one pass per annular

layer). While the overall amount of computation remains proportional to N_{total}, executing fewer but larger forward passes significantly lowers latency and improves throughput on modern GPUs. Consequently, the proposed approach provides an efficient and scalable inference pipeline for large-scale environments while preserving the center-outward generation structure.

7. Implementation Details

7.1. Train Details

We train our model on an L20 × 8 machine setup using the SemanticKITTI dataset. Following the protocols of LiDiff [20] and LiDPM [14], we use sequence 0008 as the validation set and the remaining sequences for training. When reproducing ScoreLiDAR [38], we strictly follow the original implementation and use the same official evaluation code.

7.2. Metrics

We assess the quality of point cloud completion using a suite of standard geometric and distributional metrics, including Chamfer Distance (CD), 3D Jensen–Shannon Divergence (JSD), BEV JSD, and Voxel IoU evaluated at multiple resolutions (0.5 m, 0.2 m, and 0.1 m). To further quantify temporal stability, we compute frame-to-frame Root Mean Squared Error (RMSE) across consecutive frames. Detailed definitions and computational procedures for each metric are provided below.

Chamfer Distance (CD) The ground-truth point cloud is denoted as $\mathcal{G} = \{g_i\}_{i=1}^{N_g} \subset \mathbb{R}^3$, and the predicted point cloud as $\mathcal{P} = \{p_j\}_{j=1}^{N_p} \subset \mathbb{R}^3$. The symmetric squared ℓ_2 Chamfer Distance (CD) is employed to evaluate the geometric similarity between these two point clouds:

$$\mathrm{CD}(\mathcal{P},\mathcal{G}) = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}\min_{g\in\mathcal{G}}\|p-g\|_2^2 + \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}}\min_{p\in\mathcal{P}}\|g-p\|_2^2 \tag{17}$$

CD measures the average closest-point discrepancy in both directions, providing a dense geometric comparison between the two point clouds.

Jensen–Shannon Divergence (JSD) JSD measures the similarity between probability distributions over voxelized occupancy grids. Given a voxel set V, we convert each point cloud into an occupancy probability distribution $P_{\mathcal{X}}(v) \in [0,1]$ over $v \in V$. Define the mixture distribution:

$$M(v) = \frac{1}{2}\big(P_{\mathcal{P}}(v) + P_{\mathcal{G}}(v)\big)$$

The JSD is computed as:

$$\begin{aligned}\mathrm{JSD}(P_{\mathcal{P}}\|P_{\mathcal{G}}) = &\frac{1}{2}\sum_{v\in V}P_{\mathcal{P}}(v)\log\frac{P_{\mathcal{P}}(v)}{M(v)} \\ &+\frac{1}{2}\sum_{v\in V}P_{\mathcal{G}}(v)\log\frac{P_{\mathcal{G}}(v)}{M(v)}\end{aligned} \tag{18}$$

To ensure numerical stability, all probabilities are clipped with a small constant ϵ.

When V represents a full 3D voxel grid, Eq. (18) evaluates the similarity of volumetric occupancy distributions, emphasizing global geometric structure. For Bird's-Eye View (BEV), we project the 3D space onto a horizontal 2D voxel grid by aggregating along the vertical dimension. Eq. (18) is then computed on this 2D distribution, focusing on planar structural consistency while ignoring height variations.

Voxel IoU at Multiple Resolutions Voxel intersection-over-union (IoU) evaluates the spatial overlap between the predicted and ground-truth voxel occupancies. The binary occupancy of a voxel v with respect to a point set $\mathcal{X}$ is defined as:

$$O_{\mathcal{X}}(v) = \begin{cases}1, & \text{if voxel } v \text{ contains at least one point from } \mathcal{X},\\ 0, & \text{otherwise.}\end{cases}$$

IoU is defined as:

$$\mathrm{IoU}(\mathcal{P},\mathcal{G}) = \frac{\sum_{v\in V}(O_{\mathcal{P}}(v)\wedge O_{\mathcal{G}}(v))}{\sum_{v\in V}(O_{\mathcal{P}}(v)\vee O_{\mathcal{G}}(v))} \tag{19}$$

We compute IoU at three commonly used voxel sizes to capture completion quality at multiple spatial scales:

$$\text{voxel size} \in \{0.5\,\text{m},\ 0.2\,\text{m},\ 0.1\,\text{m}\}$$

Frame-to-Frame RMSE To evaluate temporal smoothness and frame-to-frame consistency, we compute RMSE between consecutive voxelized occupancy distributions. For two consecutive frames $\mathcal{X}_t$ and $\mathcal{X}_{t+1}$ with voxelized probabilities $P_{\mathcal{X}_t}(v)$, the RMSE is defined as:

$$\mathrm{RMSE}_{t\to t+1} = \sqrt{\frac{1}{|V|}\sum_{v\in V}\big(P_{\mathcal{X}_t}(v) - P_{\mathcal{X}_{t+1}}(v)\big)^2} \tag{20}$$

The overall temporal inconsistency for a sequence of length T is given by the average:

$$\overline{\mathrm{RMSE}} = \frac{1}{T-1}\sum_{t=1}^{T-1}\mathrm{RMSE}_{t\to t+1}$$

This metric penalizes abrupt spatial occupancy changes across time, reflecting the temporal coherence of the predicted point clouds.

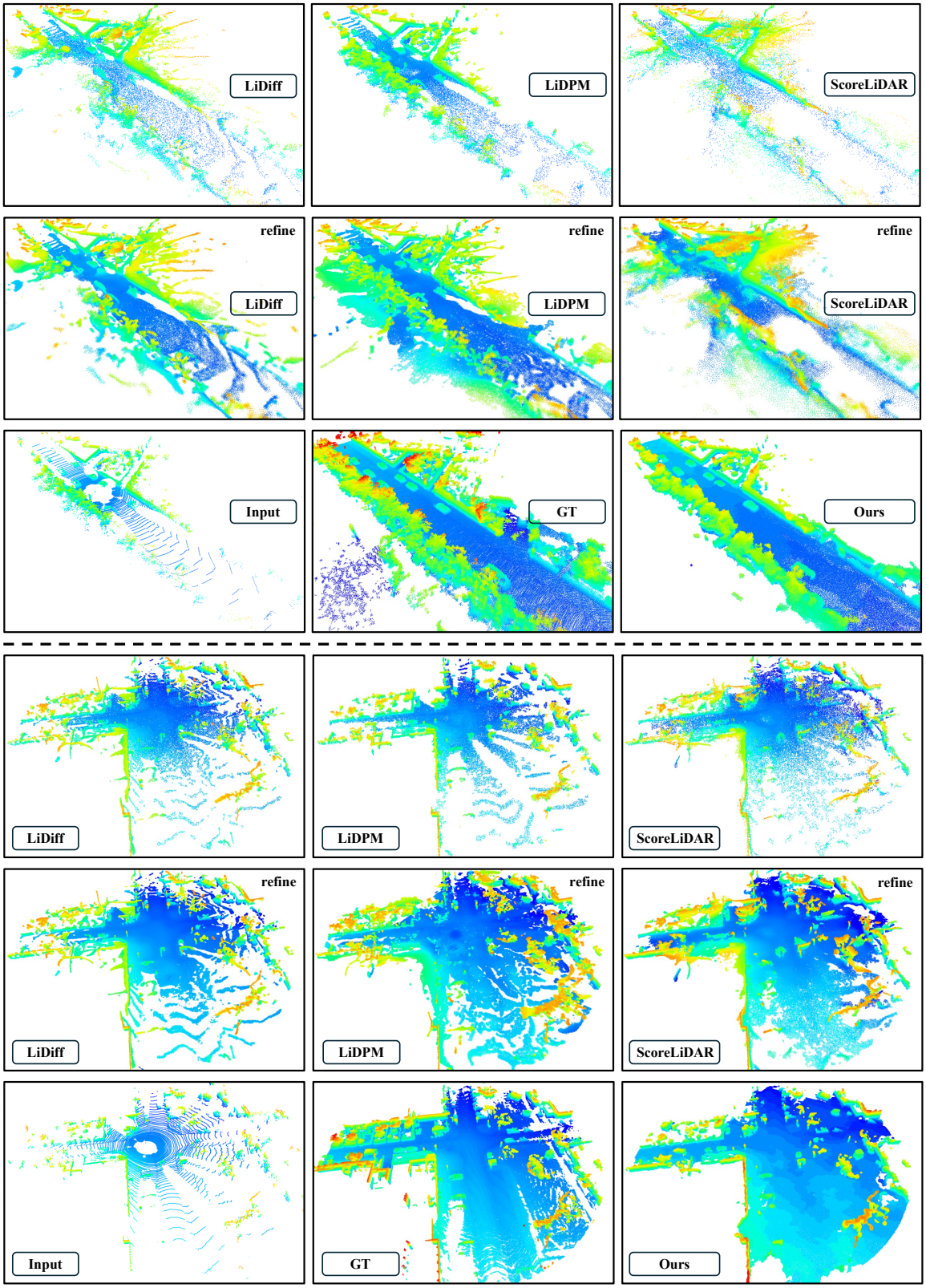

Figure 7. Comparisons across a variety of complex scenes.

8. Experiment Details

8.1. Compared Methods

To further demonstrate the effectiveness of our approach, we present qualitative comparisons across a variety of complex scenes, as shown in Fig. 7. In challenging environments with large occlusions or severe sparsity, existing methods often produce substantial missing regions; even their refined versions still exhibit numerous unfilled holes. Moreover, these baselines frequently generate hallucinated structures near object boundaries, leading to noticeable inconsistencies with the underlying geometry and causing the completed results to deviate significantly from the ground truth. In contrast, our method delivers completion results that are both smooth and spatially coherent, achieving full scene coverage with remarkably high fidelity. The reconstructed shapes preserve fine-grained geometric details, and the resulting dense point clouds exhibit superior realism and structural accuracy compared to all competing approaches.

8.2. RMSE for Other Methods

As shown in Table 7, we additionally report the RMSE metric to evaluate the temporal consistency of different completion methods across consecutive frames. RMSE reflects the stability of voxel occupancy predictions between two adjacent scans, where lower scores indicate more coherent and physically plausible temporal behavior.

Across both directions ($t_0 \rightarrow t_1$ and $t_1 \rightarrow t_0$), PatchScene achieves substantially lower RMSE than all baselines, including both LiDiff and ScoreLiDAR and their refined variants. The baselines exhibit RMSE values in the range of 0.160–0.172, demonstrating noticeable temporal fluctuations and inconsistency in their reconstructed geometry. Even after refinement, their RMSE worsens slightly, suggesting that the refinement process may introduce frame-specific artifacts or overfitting to individual scans rather than improving global temporal smoothness.

In comparison, our method achieves RMSE values of 0.086 and 0.081, representing nearly a 50% improvement over the best competing methods. This strong performance indicates that PatchScene maintains highly consistent structural predictions over time, producing stable and coherent dense completions even when the sparse LiDAR inputs vary significantly between frames. The remarkable temporal smoothness also highlights the robustness of our model to sensor noise, scanning sparsity, and viewpoint changes, underscoring its suitability for downstream applications such as tracking, mapping, and long-range scene understanding.

8.3. Temporal Consistency

To evaluate the efficacy of our Temporal Fusion module in preserving temporal consistency, we visualize the frame-to-frame evolution of the completed point clouds on the PatchScene dataset, as shown in Fig. 8. The sequence is arranged chronologically from left to right and top to bottom.

Method	**RMSE↓**	
	t_0 to t_1	t_1 to t_0
LiDiff	0.160	0.163
LiDiff(refine)	0.168	0.172
ScoreLiDAR	0.163	0.166
ScoreLiDAR(refine)	0.170	0.172
Ours	**0.086**	**0.081**

Table 7. RMSE for Other Methods

In this visualization, interframe geometric discrepancies are explicitly highlighted by rendering points that differ between the current and preceding frames in red. As observed, the completion results demonstrate high stability when the Temporal Fusion module is applied. The limited number of red points indicates that geometric deviations between adjacent frames are minimal, providing evidence that the method produces coherent structures while effectively suppressing temporal flickering artifacts.

8.4. Infinite Expansion

To further validate the scalability of our method to substantially larger spatial extents, we present additional large-range completion results in Fig. 9. Specifically, we apply a model trained only within a 20 m sensing range to scenes expanded to 50 m, and compare the input scans, our predicted completions, and the corresponding ground truth across three representative scenarios. In all cases, PatchScene successfully reconstructs highly realistic, dense, and detailed point clouds despite the significant increase in spatial coverage.

Notably, PatchScene is able to infer and complete geometric structures even in regions occluded from LiDAR sensing. In regions with incomplete ground truth caused by limited visibility, our method generates smooth, continuous, and physically plausible geometry, demonstrating strong generalization capability and exceptional robustness in large-scale, unseen environments.

9. Additional Results

9.1. Results on the KITTI360 Dataset

To further verify the generalization capability of our proposed method across different sensor configurations and environments, we conducted comparative experiments on the KITTI-360 dataset. We compared our approach against several impressive methods, including LMSCNet [25], LODE [6], MID [31], LiDiff [20], and ScoreLidar [38].

The quantitative results are summarized in Table 8. As

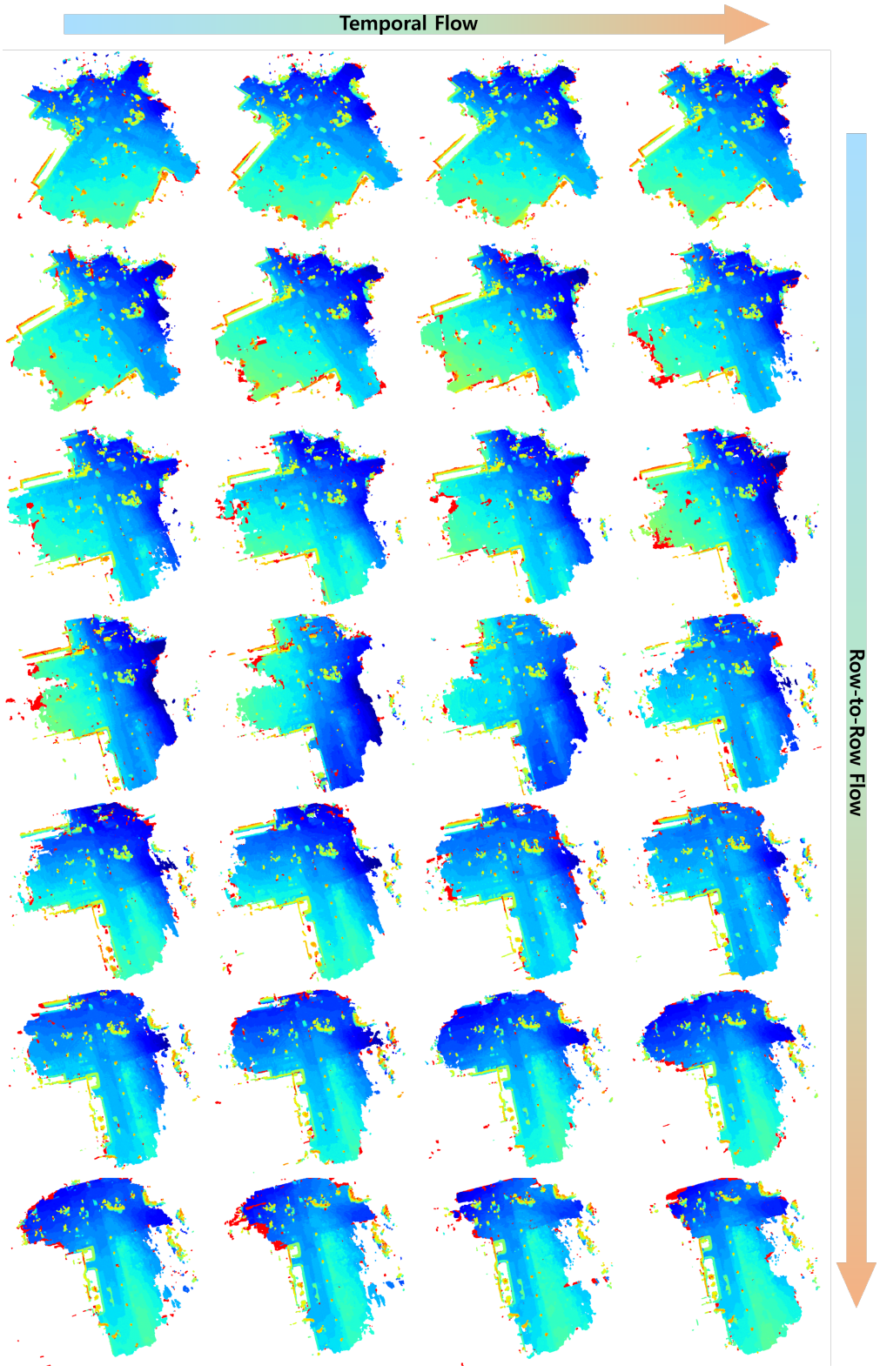

Figure 8. Completion status of consecutive frames.

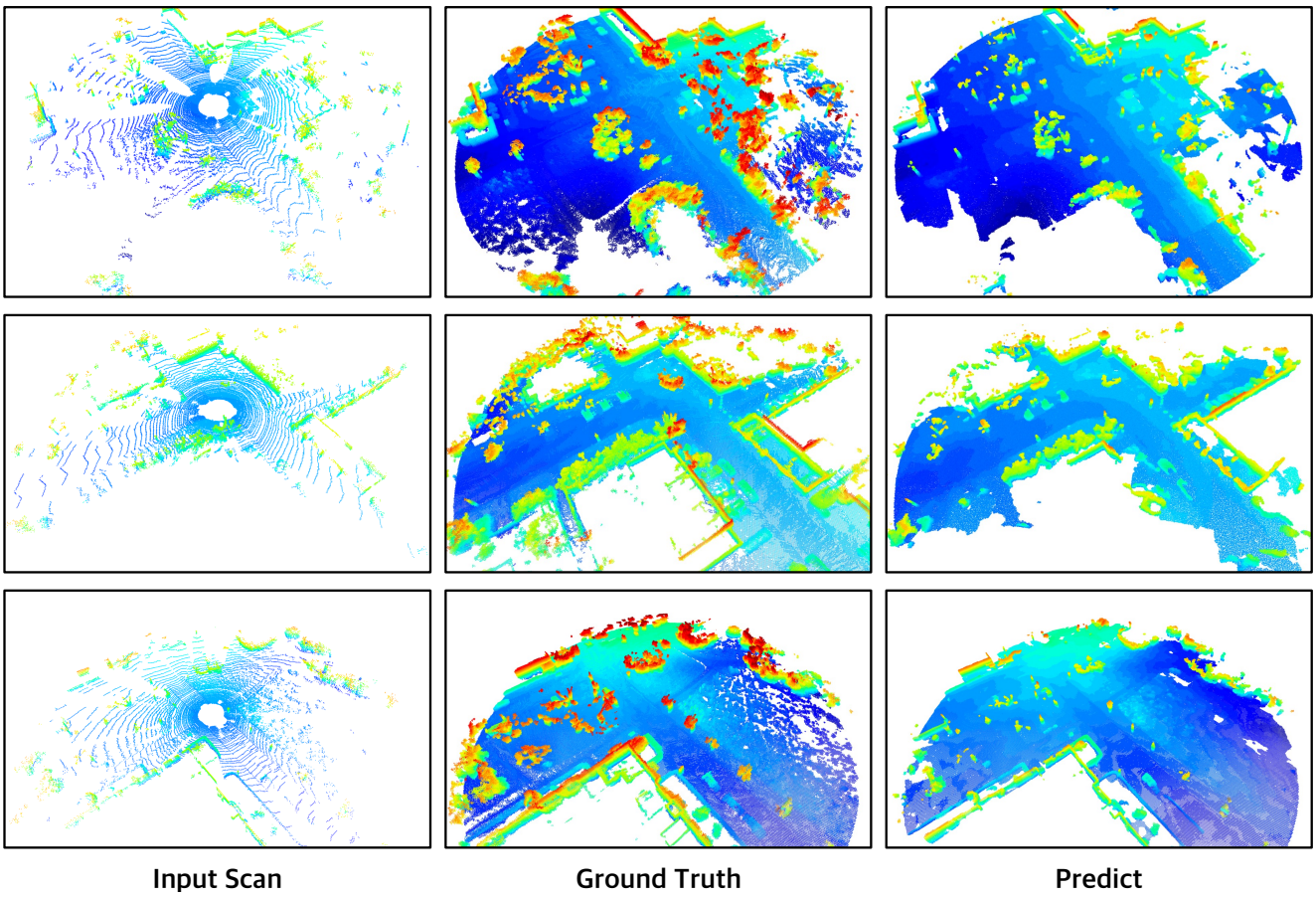

Figure 9. Additional large-range completion results.

shown, our method consistently outperforms all competing approaches across all evaluation metrics, demonstrating superior robustness. Specifically:

Geometric Accuracy. In terms of Chamfer Distance (CD), our method achieves a score of 0.356, reducing the error by approximately 21% compared to the second-best method, ScoreLidar (refine) (0.452). This indicates that our generated points are significantly closer to the ground truth geometry.

Volumetric Completeness. Most notably, our method exhibits a substantial advantage in Voxel IoU metrics. At IoU (0.5 m), we achieve 51.8%, surpassing the previous state-of-the-art (LiDiff-refine, 33.43%) by a remarkable margin of over 18 percentage points.

Distribution Fidelity. Our method also achieves the lowest Jensen-Shannon Divergence (JSD) in both 3D and BEV evaluations, further confirming that the statistical distribution of our completed point clouds aligns best with the real-world data.

Method	CD↓	JSD 3D↓	JSD BEV↓	Voxel IoU↑		
				0.5	0.2	0.1
LMSCNet [25]	0.979	-	0.496	26.17	9.21	2.88
LODE [6]	1.565	-	0.483	33.06	15.24	4.68
MID [31]	0.637	-	0.476	33.05	21.32	11.30
LiDiff [20]	0.564	-	0.459	33.23	17.55	4.88
LiDiff(refine)	0.517	-	0.446	33.43	22.04	11.84
ScoreLidar [38]	0.472	-	0.444	-	-	-
ScoreLidar(refine)	0.452	-	0.437	-	-	-
Ours	**0.356**	**0.424**	**0.341**	**51.8**	**42.5**	**21.3**

Table 8. Comparison of our method with existing approaches on KITTI360.

9.2. Results on the propriety dataset

To benchmark point cloud completion against real-world complexity, we introduce our proprietary dataset, a high-fidelity LiDAR dataset comprising 550 clips characterized by a significant long-tailed distribution of over 40 distinct scene categories. As shown in Fig. 10, this dataset goes beyond standard urban cruising by explicitly incorporating diverse road topologies (e.g., complex intersections, U-

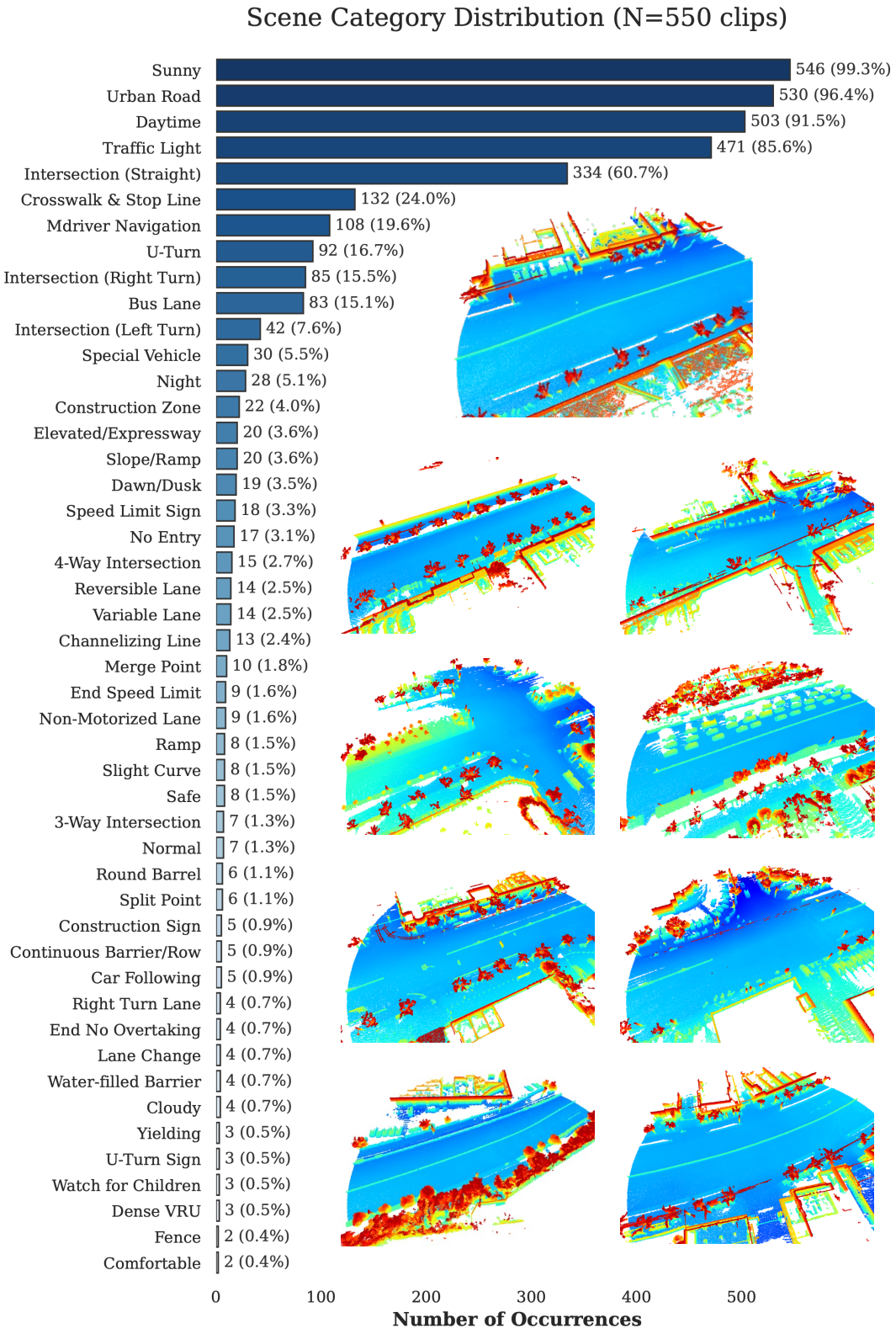

Figure 10. Overview of the propriety dataset: diverse scene category distribution and visualization of ground-truth dense point clouds.

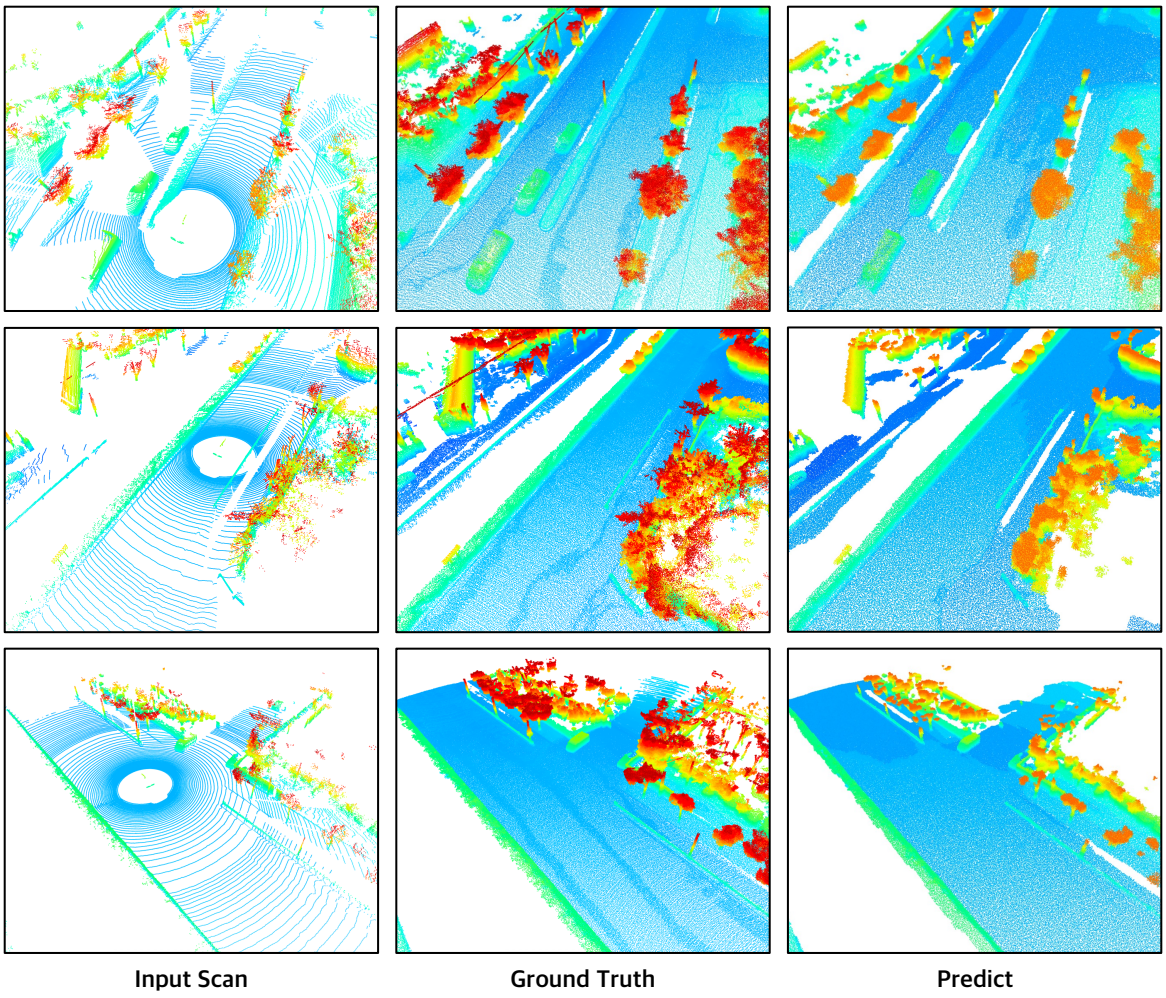

Figure 11. Visualization of the completion results of three distinct scenes in propriety dataset.

turns, and ramps), environmental variations (e.g., night and dawn/dusk), and safety-critical corner cases (e.g., construction zones and special vehicles). This rich semantic diversity ensures a rigorous evaluation of the model's ability to recover geometric details across both canonical driving scenarios and challenging, low-probability events.

We further evaluated the point cloud completion performance of our proposed method on the propriety dataset dataset to validate its effectiveness in complex, real-world scenarios. The quantitative results are reported in Table 9. Our method achieves a Chamfer Distance (CD) of 0.199 and a Voxel IoU of 0.595 (0.5 m). When compared with performance benchmarks typically observed on standard datasets, these metrics indicate that our approach maintains high competitiveness and robustness, even when facing the diverse and challenging road topologies.

To provide a more intuitive assessment, we visualize the completion results of three distinct scenes in Fig. 11.

Method	**CD↓**	**JSD 3D↓**	**JSD BEV↓**	**Voxel IoU↑**		
				0.5	0.2	0.1
Ours	0.199	0.334	0.315	0.595	0.496	0.246

Table 9. Our point cloud completion metrics on the propriety dataset.

As demonstrated in the visualizations, our method generates high-fidelity completions with remarkable geometric details. Specifically, the recovered road surfaces are exceptionally clean and smooth, exhibiting minimal noise. Crucially, the structural boundaries, such as road edges and curbs, are preserved with high sharpness. Furthermore, the completion of dynamic objects is highly precise; the geometry of vehicles is reconstructed with fine-grained detail, verifying the model's ability to handle both static background and foreground objects effectively.