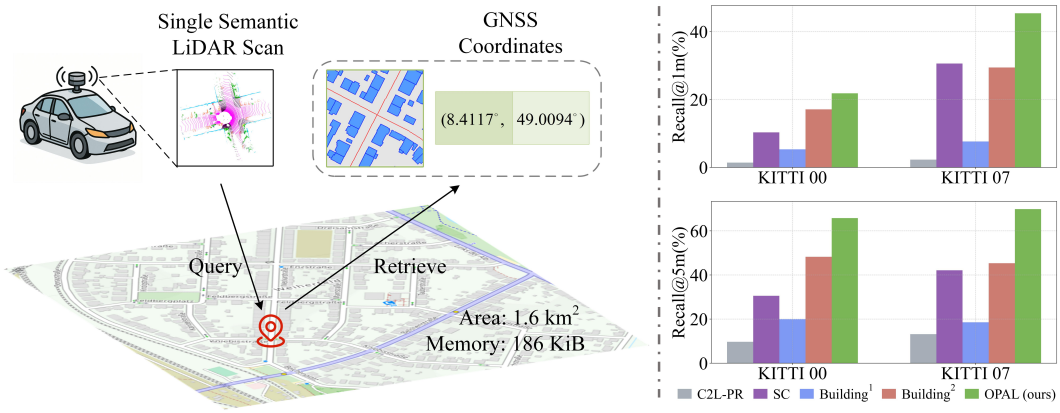
OPAL: Visibility-aware LiDAR-to-OpenStreetMap Place Recognition via Adaptive Radial Fusion

Shuhao Kang[1*] Martin Y. Liao[2*] Yan Xia[3†] Olaf Wysocki[4] Boris Jutzi[1,5] Daniel Cremers[1]

[1] Technical University of Munich [2] Wuhan University
[3] University of Science and Technology of China [4] University of Cambridge [5] Karlsruhe Institute of Technology



(a) Problem Setup (b) Performance

Figure 1: (a) Point cloud-to-OpenStreetMap (P2O) place recognition estimates the geographic location of a LiDAR scan by matching the semantic point cloud to geo-referenced OpenStreetMap tiles. (b) shows the evaluation results on KITTI [1] dataset.

Abstract: LiDAR place recognition is a critical capability for autonomous navigation and cross-modal localization in large-scale outdoor environments. Existing approaches predominantly depend on pre-built 3D dense maps or aerial imagery, which impose significant storage overhead and lack real-time adaptability. In this paper, we propose OPAL - **O**penStreetMap-based LiDAR **P**lace recognition via **A**daptive radia**L** fusion, which leverages OpenStreetMap (OSM) as a lightweight and up-to-date prior. Our key innovation lies in bridging the domain disparity between sparse LiDAR scans and structured OSM data through two carefully designed components. First, a cross-modal visibility mask that identifies observable regions from both modalities to guide feature alignment. Second, an adaptive radial fusion module that dynamically consolidates radial features into discriminative global descriptors. Extensive experiments on KITTI and KITTI-360 datasets demonstrate OPAL's superiority, achieving 15.98% higher recall at 1m threshold for top-1 retrieved matches, along with 12× faster inference speed compared to the state-of-the-art approach. Code and data are publicly available at https://github.com/kang-1-2-3/OPAL.

Keywords: Place Recognition, OpenStreetMap, Point Cloud

1 Introduction

Accurate and reliable localization is crucial for autonomous vehicles and robots operating in large-scale urban environments, where GNSS signals are often degraded or blocked due to structural

* Equal contribution. † Corresponding author.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

obstructions. Place recognition addresses this need by retrieving the most likely location from a reference database, based on a query that reflects the robot's current perception. Compared to image-based place recognition methods [2, 3, 4], which are sensitive to photometric variations caused by changing weather and seasons [5], LiDAR point clouds maintain robustness under varying illumination and meteorological conditions. Moreover, point cloud offers precise depth measurements and rich geometric detail, making them effective for accurate localization in outdoor environments [6, 7].

Most existing point cloud-based place recognition methods rely on pre-built 3D maps [8, 9, 10, 11] or satellite images [12] as reference database. However, constructing a city-scale point cloud map is both time-consuming and costly to maintain, while storage demands remain prohibitively high for large-scale deployments. Although aerial images are more compact than 3D point cloud maps, they are still expensive to capture, generally not free, and heavy to store at high resolution. Moreover, they are highly sensitive to weather, seasonal changes, and lighting conditions. In contrast, OSM provides a globally accessible, compact geospatial database comprising infrastructure, architectural elements, points of interest, land-use classifications and other stationary urban features [13]. Remarkably, it is extremely storage-efficient: only 186 KiB (as shown in Fig.1(a)), compared with 19.61 MiB for a bird's-eye view (BEV) point cloud image (as in BEVPlace++[11]) or 8.22 GiB for the raw KITTI 00 sequence. Moreover, OSM data is continuously updated by volunteers and organizations, with weekly snapshots released. Its timely and rich geometric primitives and semantic elements enable reliable place recognition, mirroring human navigation's use of spatial and semantic cues [14, 15]. Cho et al. [16] first developed a place recognition descriptor for point cloud-to-OpenStreetMap (P2O) place recognition by calculating the shortest distances to building structures at fixed angular intervals around the sensor. Lee and Ryu [17] proposed a learning-based place recognition method and integrated it into simultaneous localization and mapping (SLAM), while it requires an accurate orientation prior for initialization. Overall, current single-frame P2O place recognition methods are still limited in accuracy, robustness and efficiency.

In this paper, we present OPAL, a novel P2O place recognition framework that achieves meter-level localization accuracy using a single LiDAR scan, while maintaining real-time computational performance. The OPAL pipeline begins by projecting the query point cloud and OSM data into BEV representation, generating the visibility mask as an additional input to alleviate viewpoint disparity. A Siamese convolutional neural network (CNN) processes these polar representations to extract local feature maps. The adaptive radial fusion (ARF) module then dynamically weights radial-wise features based on their contextual importance, enabling optimized feature aggregation across varying distances and robustness to viewpoint change. Experiments on KITTI and KITTI-360 [18] datasets demonstrate that our method significantly outperforms both hand-crafted and learning-based baselines across various environments. The main contributions include:

1. We propose a novel pipeline for P2O place recognition. Compared to existing methods, our approach substantially improves accuracy, robustness, and computational efficiency.
2. We introduce visibility mask to resolve the viewpoint disparity between cross-modal inputs. The visibility mask significantly improves cross-modality feature alignment by focusing on mutually visible regions and ignoring modality-specific occlusions.
3. We propose the ARF module to dynamically fuse radial features into the global descriptor. This adaptive strategy preserves geometric structure while maintaining real-time efficiency.

2 Related Work

We review point cloud place recognition research through two perspectives: uni-modal point cloud place recognition approaches and cross-modal approaches that bridge different sensor domains.

Uni-modal point cloud place recognition. Early breakthroughs in point cloud-to-point cloud place recognition were led by PointNetVLAD [8], which combined PointNet [19] with the NetVLAD [8] aggregation layer to produce global descriptors from raw point clouds. Transformer-based architectures have also been explored for capturing long-range dependencies and contextual se-

mantics [20, 21, 22], leveraging attention mechanisms to improve feature expressiveness. MinkLoc3D [23] employed a voxel-FPN architecture with generalized mean pooling (GeM) for compact global descriptors. Recently, CASSPR [24] proposed a hybrid voxel-point dual-branch framework using hierarchical cross-attention to effectively fuse multi-level features, significantly boosting performance on sparse single-frame scans. Although these methods leverage the rich spatial information from LiDAR data to achieve strong performance, their scalability is limited by the high cost and maintenance overhead associated with constructing and updating dense, city-scale point cloud maps. These practical limitations pose a major obstacle in consumer-grade applications.

Cross-Modal point cloud place recognition. For image-to-point (I2P) cloud place recognition, Cattaneo et al. [25] and Li et al. [26] established a shared global feature space for feature matching and retrieval. C2L-PR [27] improved I2P place recognition via modality alignment and orientation voting. For point cloud-to-aerial image place recognition, Tang et al. [12] proposed a self-supervised localization approach based on 2D occupancy map matching. Beyond I2P place recognition, recent efforts have extended cross-modal localization to natural language queries [28, 29, 30].

OpenStreetMap-based approaches are most related to our method. OpenStreetSLAM [31] integrated visual odometry with map priors to improve trajectory accuracy, while subsequent methods [32, 33] focused on road or building structure alignment with OSM data. Suger and Burgard [34] introduced a Monte Carlo localization framework that aligns semantic features from LiDAR with the OSM data for outer-urban navigation. Yan et al. [35] proposed a compact 4-bit descriptor that encoded the street intersections and building gaps for efficient global localization. Bieringer et al. [36] utilized Level of Detail 3 (LOD3) models for outdoor map-based positioning. Sequential frames generally improve accuracy through spatial consistency, yet single-scan place recognition remains critical in unknown or dynamic environments that lack accurate maps. Besides, methods developed for sequential point cloud localization often struggle in the single-frame setting, where limited observations and the absence of motion constraints significantly hinder performance. For the single-frame P2O place recognition, Cho et al. [16] proposed a hand-crafted descriptor by extracting the shortest distance to buildings at fixed angular intervals for cross-modality feature matching, later improved by Li et al. [37] with directional boundary features. However, these methods exhibit strong dependence on building structures, limiting practical applicability. Although Lee and Ryu [17] introduced learning-based descriptors, their method requires IMU-based orientation priors for initialization. To summarize, existing solutions suffer from three key limitations: reliance on sequential inputs, limited accuracy and robustness, and inefficient descriptor generation. In contrast, OPAL leverages both geometric and topological cues to achieve accurate, robust, and generalizable localization across environments.

3 Methodology

The P2O place recognition task aims to localize a query LiDAR point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$ by matching it against a geo-referenced OSM database $\mathbb{O}$, where each of the N points in $\mathcal{P}$ is represented by a 3D Cartesian coordinate (x, y, z). Since our approach incorporates cross-modality data, $\mathcal{P}$ and $\mathbb{O}$ require pre-processing before being fed into the framework. $\mathcal{P}$ is first enhanced by concatenating per-point semantic labels as $\mathcal{P}' \in \mathbb{R}^{N\times 4}$. The original OSM data $\mathbb{O}$ is stored in structured format and represents various entities, including areas, ways, and nodes, in geographic coordinate system. Following the OrienterNet [14], we rasterize the areas, ways, and nodes into a 3-channel grid map with a fixed sampling distance Δ_o in local 2D East-North coordinate system. From this projected map, we densely sample m map tiles $\mathbb{O} = \{\mathcal{O}_i\}_{i=1}^{m}$ along the ego-vehicle trajectory to construct the OSM database, where each tile $\mathcal{O}_i$ corresponds to an $H \times W$ meters region centered at geographic coordinates (lat_i, lon_i). Details of the OSM tile database are given in Appendix B.1.

Fig. 2 illustrates OPAL's pipeline. The pipeline begins by computing visibility masks to resolve occlusion patterns caused by viewpoint disparities (Sec. 3.1). Next, a Siamese polar CNN architecture is employed to extract deep feature maps from both modalities (Sec. 3.2), which are subsequently aggregated into compact global descriptors through the proposed ARF module (Sec. 3.3).

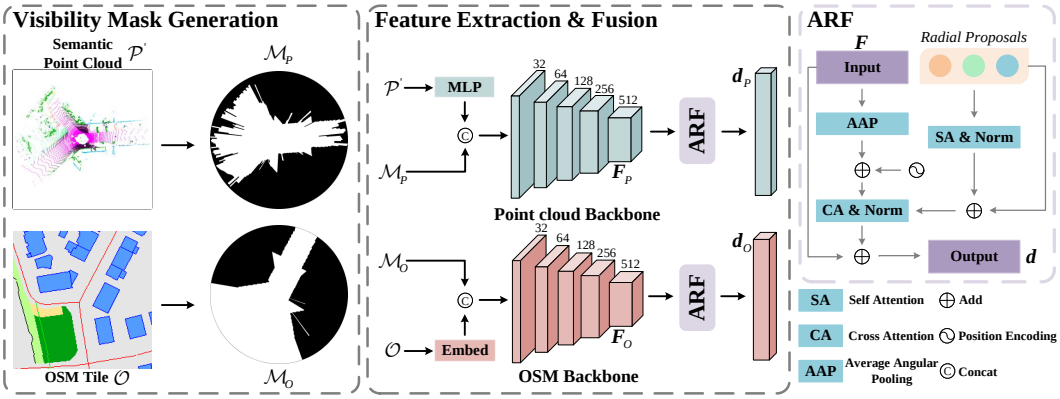


Figure 2: Overview of proposed OPAL. Given a semantic point cloud frame $\mathcal{P}'$ and OSM tile $\mathcal{O}$, OPAL computes visibility masks to bridge the occlusion difference, then extracts polar BEV features via a Siamese encoder, and lastly generates discriminative descriptors using ARF for place retrieval.

3.1 Visibility Mask Generation

The concept of visibility alignment originates from prior work in occlusion handling for image matching [38], visual localization [12, 14], and 3D building reconstruction [39]. In the P2O place recognition, this challenge remains significant due to the modality gap between LiDAR scans and OSM data. Effective visibility handling becomes crucial for robust cross-modal matching.

To address this issue, we compute visibility masks $\mathcal{M}$ for both point cloud and OSM data to resolve occlusion discrepancies. Given a point cloud frame $\mathcal{P}' \in \mathbb{R}^{N\times 4}$, we first project it onto a polar BEV grid with U radial rings and V angular sectors, assigning points within each cell (u, v). The radial and angular resolutions are given by $\Delta_r = \frac{L}{U}$ and $\Delta_s = \frac{2\pi}{V}$, where L is defined as the maximum valid range of LiDAR. For each polar cell (u, v), the corresponding radial distance $r_{u,v}$ and azimuth angle $\phi_{u,v}$ are defined as:

$$\begin{aligned} r_{u,v} &= (u + 0.5)\Delta_r, \quad u \in \{0, \dots, U-1\}, \\ \phi_{u,v} &= (v + 0.5)\Delta_s, \quad v \in \{0, \dots, V-1\}. \end{aligned} \tag{1}$$

Through ray casting, cells are classified as visible $\mathcal{M}_P(u, v) = 1$ if they lie within the line-of-sight before the last measured return in a sector. Conversely, cells are marked as occluded $\mathcal{M}_P(u, v) = 0$ if they are behind the last valid range return :

$$\mathcal{M}_P(u, v) = \begin{cases} 1, & r_{u,v} \le \max(r_{[:,v]}) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\max(r_{[:,v]})$ is range of the last valid return in v-th sector.

For the OSM tile $\mathcal{O} \in \mathbb{R}^{H\times W\times 3}$ defined in the 2D Cartesian coordinate system, we convert it into a polar BEV grid with U rings and V sectors, as in the point cloud branch. For each polar cell (u, v), the corresponding Cartesian coordinates $(x_{u,v}, y_{u,v})$ are computed via:

$$x_{u,v} = r_{u,v} \cos \phi_{u,v}, \quad y_{u,v} = r_{u,v} \sin \phi_{u,v}, \tag{3}$$

and then the polar representation is obtained by bilinear interpolation of $\mathcal{O}$ at $(x_{u,v}, y_{u,v})$.

As OSM data lacks explicit range measurements, visibility estimation relies on semantic cues. Here, we select the "building" elements from the area channel as occluders, owing to their vertical extent and structural continuity, which consistently obstruct sensor visibility in both urban and suburban environments. Through ray-casting in each sector, cells are classified as occluded $\mathcal{M}_O(u, v) = 0$ if they lie further than the nearest "building" element; otherwise, they are classified as visible $\mathcal{M}_O(u, v) = 1$. The process is formatted as:

$$\mathcal{M}_O(u, v) = \begin{cases} 0, & u > \min(u') \text{ if } \exists \mathcal{O}(u', v) = \text{"building"} \\ 1, & \text{otherwise} \end{cases} \tag{4}$$

where $\mathcal{O}(u', v)$ is the building elements in v-th sector.

Remark. Unlike prior approaches [12, 14] that estimate the visible or confidential mask with a neural network, our visibility mask generation is fully deterministic and leverages the complementary strengths of both modalities. For LiDAR, valid range measurements directly yield visibility masks. For OSM, which lacks range information, we approximate the visibility mask using building masks, as buildings are the primary occluders in urban scenes. By eliminating the approximation errors and training overhead of learned visibility estimation, our method preserves geometric consistency across modalities while ensuring computational efficiency.

3.2 Feature Extraction

As shown in Fig. 2, our feature extraction pipeline processes both modalities through parallel yet symmetric branches. The augmented point cloud $\mathcal{P}' \in \mathbb{R}^{N\times 4}$ is first passed through a lightweight multilayer perceptron (MLP) to generate C_{pem}-dimensional per-point features. These features are then splatted onto the polar BEV grid, where grid-wise features are aggregated using max pooling, resulting in a dense feature map $\boldsymbol{F}_P \in \mathbb{R}^{U\times V\times C_{pem}}$. This representation is concatenated with the visibility mask $\mathcal{M}_P \in \mathbb{R}^{U\times V\times 1}$, and processed by the encoder of PolarNet [40], yielding the local feature map $\boldsymbol{F'}_P \in \mathbb{R}^{Z\times T\times C}$.

For the OSM branch, we embed each channel of the rasterized map tile $\mathcal{O} \in \mathbb{R}^{H\times W\times 3}$ into a C_{oem}-channel feature, generating dense semantic feature $\boldsymbol{F}_O \in \mathbb{R}^{H\times W\times(3\times C_{oem})}$ in the Cartesian coordinate system. Then $\boldsymbol{F}_O$ is transformed into polar BEV feature map via bilinear sampling with Eq. (3), and concatenated with the visibility mask $\mathcal{M}_O \in \mathbb{R}^{U\times V\times 1}$ to form a visibility-aware input. Finally, the OSM features are processed through a separate PolarNet [40] encoder (with weights independent of the point cloud branch) to produce the final feature map $\boldsymbol{F'}_O \in \mathbb{R}^{Z\times T\times C}$.

3.3 Adaptive Radial Fusion

Given the extracted local features $\boldsymbol{F} \in \mathbb{R}^{Z\times T\times C}$ from the point cloud and OSM tile, the next step is to aggregate them into a global descriptor for efficient retrieval. This global descriptor is expected to be both representative and robust to orientation variations. Existing solutions entail critical trade-offs: frequency-domain methods [41, 42] and range projection approach [22] sacrifice spatial relationships for rotation invariance, while sampling-based approaches like BEVPlace++ [11] suffer from high computational overhead. These limitations hinder cross-modal place recognition by either compromising geometric fidelity or reducing system efficiency.

To preserve geometric completeness and ensure rotation robustness, we introduce the ARF module (shown in the last column of Fig. 2). The module first extracts radial features through angular average pooling (AAP):

$$\boldsymbol{F}_r = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{F}[:, t, :] + \boldsymbol{E}_{re}, \tag{5}$$

where $\boldsymbol{F}_r$ represents the radially compressed features, and $\boldsymbol{E}_{re} \in \mathbb{R}^{Z\times C}$ encodes the ring-order information using cosine position encoding [43]. Building upon the radial-wise features, we introduce trainable *radial proposals* $\boldsymbol{Q} \in \mathbb{R}^{Z\times C}$ as in [44, 45], to adaptively track and fuse the radial-wise features based on significance. These *radial proposals* are implemented as trainable model parameters that undergo a two-stage attention refinement process. First, inter-proposal communication is enhanced with a self-attention module:

$$\boldsymbol{Q}' = \mathrm{softmax}(\frac{\boldsymbol{Q}\boldsymbol{Q}^\top}{\sqrt{C}})\boldsymbol{Q}. \tag{6}$$

This enables the proposals to capture global contextual awareness while suppressing redundant correlations. The refined proposals then selectively aggregate information from the radial features via a cross-attention mechanism:

$$\boldsymbol{F}'_r = \mathrm{softmax}(\frac{\boldsymbol{Q}'\boldsymbol{F}_r^\top}{\sqrt{C}})\boldsymbol{F}_r. \tag{7}$$

This attention mechanism dynamically weights each radial feature based on its geometric salience for place recognition, allowing the model to focus on discriminative spatial patterns while maintaining robustness to orientation variations. The refined radial features $\boldsymbol{F}'_r$ are then combined with the original $\boldsymbol{F}_r$ via a residual connection and transformed into the global descriptor $\boldsymbol{d}$ finally:

$$\boldsymbol{d} = \boldsymbol{W}(Flatten(\boldsymbol{F}_r + \boldsymbol{F}'_r)), \tag{8}$$

where $Flatten(\cdot)$ denotes the operation that flattens the radial-wise features into a one-dimensional vector, and $\boldsymbol{W}$ is a fully connected layer that projects this vector to the global descriptor $\boldsymbol{d}$.

Remark. The key innovation lies in combining LiDAR's native radial geometry with learned attention. AAP preserves sensor's scanning pattern while ensuring rotation robustness, and the trainable radial proposals adaptively weight features based on their importance. The proposed ARF surpasses fixed aggregation methods without compromising computational efficiency.

4 Experiments

4.1 Experimental Setup

We validate the proposed method on two public datasets: KITTI [1] and KITTI-360 [18]. The corresponding OSM data is collected from the OpenStreetMap official website[1]. Our model is trained exclusively on the KITTI dataset and evaluated in a zero-shot setting on KITTI-360 to assess its generalization capability. For evaluation, we adopt the standard Recall@Km metric, which measures the proportion of queries whose top-1 retrieved match falls within K-meters of the ground-truth (GT) location. To prevent overlap between the training and validation sets, sequences 00 and 07 from KITTI, and sequences 00, 05, 06, and 09 from KITTI-360 are used for evaluation. Additional details about the train/test splits are provided in Appendix. A.

Baseline methods. Due to the limited attention of P2O place recognition research, we adapt three place recognition baselines for comparison: a) Building [16]: the pioneering P2O place recognition method that first utilizes a global key for fast matching, followed by a fine-grained descriptor for precise localization. We re-implement both the original two-stage version (Building^2) and a simplified one-stage variant (Building^1) for comparison. b) SC [46]: a widely-used point cloud-to-point cloud place recognition method based on hand-crafted descriptors. Following Li et al. [37], we extract building points from both LiDAR scans and OSM data to facilitate descriptor extraction and matching. c) C2L-PR [27]: A hybrid framework for image-to-point cloud place recognition. C2L-PR first extracts hand-crafted features from point cloud semantics (road, parking, sidewalk, other-ground, building, fence, other-structure, vegetation, terrain) and OSM data (building, parking, grass, forest, fence, wall, road), then learns the descriptors via an embedding network. We re-trained the modified C2L-PR with the same setting as OPAL for fair comparison.

4.2 Place Recognition Results

KITTI. As shown in Tab. 1, OPAL achieves superior place recognition performance on KITTI sequences 00 and 07, outperforming all baseline methods by a large margin. Specifically, it achieves significant improvements of 4.73%, 17.55%, and 17.47% in R@1m/5m/10m metrics on sequence 00, and 15.98%, 24.53%, and 24.52% on sequence 07 compared to the state-of-the-art method, Building^2 [16]. Fig. 3 illustrates OPAL's accurate localization results along the trajectory across diverse environments, highlighting the robust performance of our method under various conditions. Fig. 4 shows

Table 1: Recall@Km of top-1 retrieved results on the KITTI dataset.

Method	Seq 00			Seq 07		
	R@1	R@5	R@10	R@1	R@5	R@10
SC [46]	10.31	30.54	31.16	30.61	42.42	43.42
Building^1 [16]	5.31	19.89	20.52	7.63	18.53	19.35
Building^2 [16]	17.09	48.23	48.93	29.43	45.32	45.78
C2L-PR [27]	1.39	9.69	12.20	2.27	13.17	17.26
OPAL	**21.82**	**65.78**	**66.40**	**45.41**	**69.85**	**70.30**
OPAL-Rot	21.49	66.46	67.14	46.14	70.12	70.30

[1] https://www.openstreetmap.org/

Figure 3: Top-1 retrieved results @5m threshold on the 00 sequence of the KITTI dataset. Black points • denote OSM tile locations, while red • and green • indicate the wrong and correct retrieved results, respectively.

Figure 4: Examples of LiDAR queries and their top-1 retrieved matches on KITTI. Red rectangles □ represent wrong retrieved results and green □ represent correct retrieved results. Legends for point cloud and OSM tile are shown in the Appendix B.1.

the qualitative results of the baseline methods and our OPAL. These quantitative and qualitative results demonstrate OPAL's effectiveness in enhancing localization accuracy and robustness.

Robust to rotation. To assess rotational robustness, we apply random z-axis rotations uniformly sampled from $[0, 2\pi]$ to each query point cloud to simulate view change. As shown in the last row of Tab. 1, OPAL remains robust under these transformations.

Table 2: Recall@Km of top-1 retrieved results on the KITTI-360 dataset.

Method	Seq 00			Seq 05			Seq 06			Seq 09		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
SC [46]	15.14	39.61	40.66	3.69	16.69	17.18	4.14	14.22	14.59	13.92	27.45	28.07
Building¹ [16]	5.22	15.76	17.07	0.87	3.91	4.61	0.60	3.18	3.79	4.21	12.16	13.27
Building² [16]	**17.12**	**49.61**	**51.29**	4.23	15.94	16.29	3.00	12.82	13.39	18.28	41.92	42.64
C2L-PR [27]	1.70	8.23	10.93	0.81	4.72	6.36	0.45	3.35	4.50	1.69	9.59	12.45
OPAL	14.92	42.82	44.18	**7.74**	**30.49**	**31.55**	**7.71**	**36.38**	**37.54**	**27.89**	**60.96**	**61.92**

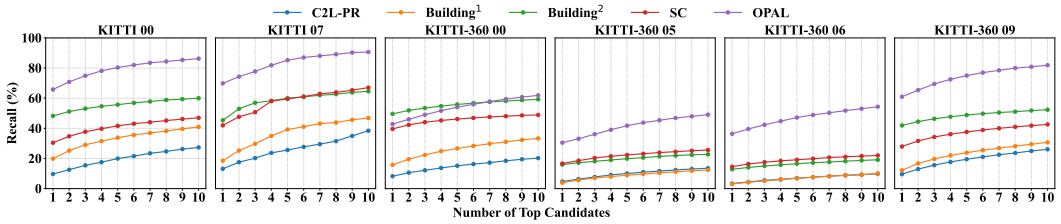


Figure 5: Recall curves @5m of top-N candidates on the KITTI and KITTI-360 datasets.

Zero-shot generalization on KITTI-360. As shown in Tab. 2, while OPAL shows slightly reduced performance in building-dominated urban environments (sequence 00) compared to the Building² [16], it achieves significant improvements over baselines in other sequences 05, 06, 09, with performance gains of 14.55%, 23.56%, and 19.04% at R@5m, respectively. Fig. 5 presents the recall curves of top candidates within a 5-meter threshold on the KITTI and KITTI-360 datasets, where OPAL consistently outperforms baseline methods across diverse scenes. These results highlight OPAL's strong generalization capability across diverse environments.

Runtime performance. We evaluate our method on a desktop with an Intel i9-13900K CPU and NVIDIA RTX 4090 GPU and report the results in Tab. 3. Our OPAL achieves high efficiency, processing point clouds in 1.91 ms and OSM tiles in 5.14 ms, resulting in a total runtime of only 7.05 ms. This corresponds to a throughput exceeding 140 FPS for descriptor generation, enabling deployment in time-sensitive applications.

Table 3: Descriptor generation runtime (ms).

Method	Point Cloud	OSM Tile	Total
SC [46]	31.54	16.68	48.22
Building [16]	29.86	54.87	84.73
C2L-PR [27]	219.08	316.71	535.79
OPAL	**1.91**	**5.14**	**7.05**

4.3 Ablation Study

We conduct ablation studies on KITTI sequence 00 to evaluate the impact of three components: the visibility mask, ARF module, and the effect of semantic labels in the point cloud.

Visibility mask and ARF. Tab. 4 presents a systematic comparison of seven architectural variants, all trained from scratch. Variant $[A]$ represents a simple baseline with PolarNet [40] for feature extraction and global average pooling (GAP) for feature aggregation, without the visibility mask and ARF module. $[B]$ introduces ARF alone, yielding notable performance gains. For variants $[A]$ and $[B]$, the visibility mask is removed during both training and evaluation. With the visibility mask, variants $[C]$ and $[D]$ adopt GAP and GeM, while $[E]$ and $[F]$ use NetVLAD [2] and MixVPR [47], respectively. All variants, however, show limited performance in the P2O place recognition task, mainly because global pooling-based aggregators discard critical local features, and NetVLAD/MixVPR are tailored for front-view images rather than BEV point clouds, making them rotation-sensitive and less effective in our setting. $[G]$ employs the average angular pooling (AAP) for feature aggregation and achieves moderate performance. Finally, variant $[H]$ combines both the visibility mask and ARF module, achieving the best performance, confirming the effectiveness of their joint contribution.

Table 4: Ablation study on visibility mask and ARF module.

ID	VM	FA	R@1	R@5	R@10
[A]		GAP	3.79	22.46	24.82
[B]		ARF	21.03	62.21	63.27
[C]	✓	GAP	5.84	17.95	19.18
[D]	✓	GeM	1.56	9.36	10.52
[E]	✓	NetVLAD	6.39	30.70	32.26
[F]	✓	MixVPR	1.78	7.69	8.17
[G]	✓	AAP	17.68	51.62	52.17
[H]	✓	ARF	**21.82**	**65.78**	**66.40**

VM: visibility mask; FA: feature aggregation.

Effect of point cloud semantic label accuracy. Tab. 5 illustrates the impact of semantic labels on P2O place recognition performance. With the ground truth labels [50], our OPAL achieves the best results (74.68% R@10m), followed by predicted labels from Cylinder3D [49] (66.4%) and Rangenet++ [48] (60.30%). The 14.38% performance gap between Rangenet++ and GT annotations underscores the potential reliance of the framework's accuracy on the precision of semantic labels within the point cloud.

Table 5: Effect of semantic label in point cloud.

Semantic Label	R@1	R@5	R@10
Rangenet++ [48]	18.92	59.08	60.30
Cylinder3D [49]	21.82	65.78	66.40
Ground Truth [50]	25.37	73.99	74.68

5 Conclusion

In this work, we presented OPAL, a novel single-frame P2O place recognition framework. The proposed method introduces the visibility-aware mask to resolve the cross-modality occlusion, coupled with the adaptive radial fusion module for effectively and robustly global descriptor aggregation. Experiments on the KITTI and KITTI-360 datasets demonstrate that OPAL consistently outperforms state-of-the-art baseline methods across diverse challenging scenarios, significantly improving accuracy and computational efficiency.

6 Limitation

The localization accuracy of our OPAL heavily depends on the quality and distinctiveness of the surrounding objects in the point cloud. Fig. 6 shows some failure cases under different conditions. (a)-(b) show ambiguous scenarios at road crossings with limited distinctive features, leading to top-1 retrieval errors due to the geometric similarity between the retrieved and ground-truth locations. As shown in (c)-(d), cross-modal discrepancies occur when roadside vegetation and buildings detected in LiDAR scans are missed in the OSM data. Furthermore, as discussed in Sec. 4.3, the localization accuracy is strongly influenced by the precision of semantic labels assigned to the point cloud. To address these limitations, we plan to: 1) extend OPAL to sequential point cloud-based place recognition, which could leverage temporal and geometric consistency to improve the reliability and accuracy; 2) incorporate orientation priors, text, or images to reduce dependence on point cloud semantic labels.

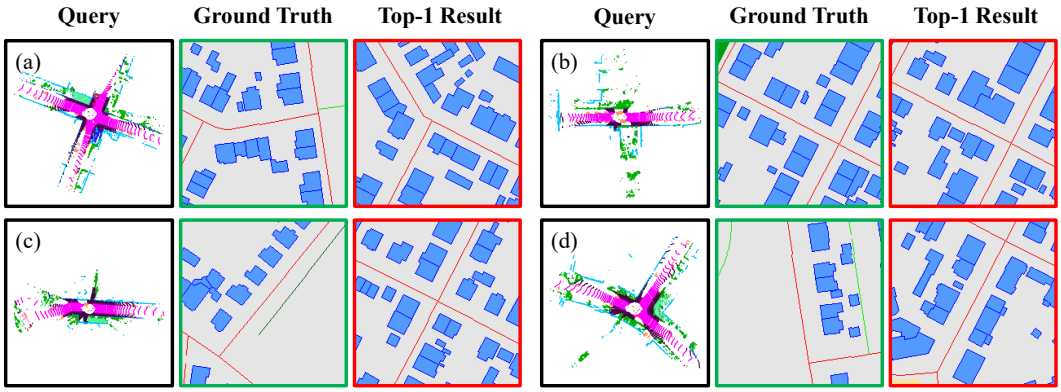


Figure 6: Failure cases. The red rectangle □ represents the wrong retrieved top-1 result and the green rectangle represents □ the GT OSM tile.

References

[1] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[2] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[3] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14141–14152, 2021.

[4] S. Zhu, L. Yang, C. Chen, M. Shah, X. Shen, and H. Wang. R2former: Unified retrieval and reranking transformer for place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19370–19380, 2023.

[5] J. Yu, H. Ye, J. Jiao, P. Tan, and H. Zhang. Gv-bench: Benchmarking local feature matching for geometric verification of long-term loop closure detection. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7922–7928. IEEE, 2024.

[6] Y. Xia, Y. Xu, C. Wang, and U. Stilla. Vpc-net: Completion of 3d vehicles from mls point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 174:166–181, 2021.

[7] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, and C. Stachniss. Overlapnet: Loop closing for lidar-based slam. *Robotics: Science and Systems XVI*, 2020.

[8] M. A. Uy and G. H. Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4470–4479, 2018.

[9] Z. Liu, S. Zhou, C. Suo, P. Yin, W. Chen, H. Wang, H. Li, and Y.-H. Liu. Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2831–2840, 2019.

[10] Y. Xia, Y. Xu, S. Li, R. Wang, J. Du, D. Cremers, and U. Stilla. Soe-net: A self-attention and orientation encoding network for point cloud based place recognition. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11348–11357, 2021.

[11] L. Luo, S.-Y. Cao, X. Li, J. Xu, R. Ai, Z. Yu, and X. Chen. Bevplace++: Fast, robust, and lightweight lidar global localization for unmanned ground vehicles. *IEEE Transactions on Robotics*, 2025.

[12] T. Y. Tang, D. De Martini, and P. Newman. Get to the point: Learning lidar place recognition and metric localisation using overhead imagery. *Proceedings of Robotics: Science and Systems, 2021*, 2021.

[13] H. Fan, A. Zipf, Q. Fu, and P. Neis. Quality assessment for building footprints data on openstreetmap. *International Journal of Geographical Information Science*, 28(4):700–719, 2014.

[14] P.-E. Sarlin, D. DeTone, T.-Y. Yang, A. Avetisyan, J. Straub, T. Malisiewicz, S. R. Bulo, R. Newcombe, P. Kontschieder, and V. Balntas. Orienternet: Visual localization in 2d public maps with neural matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21632–21642, 2023.

[15] Y. Liao, X. Chen, S. Kang, J. Li, Z. Dong, H. Fan, and B. Yang. Osmloc: Single image-based visual localization in openstreetmap with geometric and semantic guidances. *arXiv preprint arXiv:2411.08665*, 2024.

[16] Y. Cho, G. Kim, S. Lee, and J.-H. Ryu. Openstreetmap-based lidar global localization in urban environment without a prior lidar map. *IEEE Robotics and Automation Letters*, 7(2): 4999–5006, 2022.

[17] S. Lee and J.-H. Ryu. Autonomous vehicle localization without prior high-definition map. *IEEE Transactions on Robotics*, 2024.

[18] Y. Liao, J. Xie, and A. Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022.

[19] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[20] Z. Fan, Z. Song, H. Liu, Z. Lu, J. He, and X. Du. Svt-net: Super light-weight sparse voxel transformer for large scale place recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 551–560, 2022.

[21] W. Zhang, H. Zhou, Z. Dong, Q. Yan, and C. Xiao. Rank-pointretrieval: Reranking point cloud retrieval via a visually consistent registration evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[22] J. Ma, J. Zhang, J. Xu, R. Ai, W. Gu, and X. Chen. Overlaptransformer: An efficient and yaw-angle-invariant transformer network for lidar-based place recognition. *IEEE Robotics and Automation Letters*, 7(3):6958–6965, 2022.

[23] J. Komorowski. Minkloc3d: Point cloud based large-scale place recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1790–1799, 2021.

[24] Y. Xia, M. Gladkova, R. Wang, Q. Li, U. Stilla, J. F. Henriques, and D. Cremers. Casspr: Cross attention single scan place recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8461–8472, 2023.

[25] D. Cattaneo, M. Vaghi, S. Fontana, A. L. Ballardini, and D. G. Sorrenti. Global visual localization in lidar-maps through shared 2d-3d embedding space. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4365–4371. IEEE, 2020.

[26] Y.-J. Li, M. Gladkova, Y. Xia, R. Wang, and D. Cremers. Vxp: Voxel-cross-pixel large-scale image-lidar place recognition. In *2025 International Conference on 3D Vision (3DV)*, 2025.

[27] H. Xu, H. Liu, S. Huang, and Y. Sun. C2l-pr: Cross-modal camera-to-lidar place recognition via modality alignment and orientation voting. *IEEE Transactions on Intelligent Vehicles*, 2024.

[28] M. Kolmet, Q. Zhou, A. Ošep, and L. Leal-Taixé. Text2pos: Text-to-point-cloud cross-modal localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6687–6696, 2022.

[29] Y. Xia, L. Shi, Z. Ding, J. F. Henriques, and D. Cremers. Text2loc: 3d point cloud localization from natural language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14958–14967, 2024.

[30] Y. Xia, Z. Li, Y.-J. Li, L. Shi, H. Cao, J. F. Henriques, and D. Cremers. Uniloc: Towards universal place recognition using any single modality. *arXiv preprint arXiv:2412.12079*, 2024.

[31] G. Floros, B. Van Der Zander, and B. Leibe. Openstreetslam: Global vehicle localization using openstreetmaps. In *2013 IEEE international conference on robotics and automation (ICRA)*, pages 1054–1059. IEEE, 2013.

[32] P. Ruchti, B. Steder, M. Ruhnke, and W. Burgard. Localization on openstreetmap data using a 3d laser scanner. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 5260–5265. IEEE, 2015.

[33] O. Vysotska and C. Stachniss. Exploiting building information from publicly available maps in graph-based slam. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4511–4516. IEEE, 2016.

[34] B. Suger and W. Burgard. Global outer-urban navigation with openstreetmap. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1417–1422. IEEE, 2017.

[35] F. Yan, O. Vysotska, and C. Stachniss. Global localization on openstreetmap using 4-bit semantic descriptors. In *2019 European conference on mobile robots (ECMR)*, pages 1–7. IEEE, 2019.

[36] A. Bieringer, O. Wysocki, S. Tuttas, L. Hoegner, and C. Holst. Analyzing the impact of semantic LoD3 building models on image-based vehicle localization. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10:55–62, 2024.

[37] Z. Li, Y. Wang, R. Zhang, F. Ding, C. Wei, and J.-G. Lu. A lidar-openstreetmap matching method for vehicle global position initialization based on boundary directional feature extraction. *IEEE Transactions on Intelligent Vehicles*, 2024.

[38] M. Fan, M. Chen, C. Hu, and S. Zhou. Occˆ 2net: Robust image matching based on 3d occupancy estimation for occluded regions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9652–9662, 2023.

[39] O. Wysocki, Y. Xia, M. Wysocki, E. Grilli, L. Hoegner, D. Cremers, and U. Stilla. Scan2LoD3: Reconstructing semantic 3D building models at LoD3 using ray casting and Bayesian networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 6547–6557, 2023.

[40] Y. Zhang, Z. Zhou, P. David, X. Yue, Z. Xi, B. Gong, and H. Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9601–9610, 2020.

[41] X. Xu, S. Lu, J. Wu, H. Lu, Q. Zhu, Y. Liao, R. Xiong, and Y. Wang. Ring++: Roto-translation invariant gram for global localization on a sparse scan map. *IEEE Transactions on Robotics*, 39(6):4616–4635, 2023.

[42] S. Lu, X. Xu, L. Tang, R. Xiong, and Y. Wang. Deepring: Learning roto-translation invariant representation for lidar based place recognition. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1904–1911. IEEE, 2023.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[44] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[45] A. Ali-Bey, B. Chaib-draa, and P. Giguère. Boq: A place is worth a bag of learnable queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17794–17803, 2024.

[46] G. Kim and A. Kim. Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4802–4809. IEEE, 2018.

[47] A. Ali-Bey, B. Chaib-Draa, and P. Giguere. Mixvpr: Feature mixing for visual place recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2998–3007, 2023.

[48] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4213–4220. IEEE, 2019.

[49] X. Zhu, H. Zhou, T. Wang, F. Hong, W. Li, Y. Ma, H. Li, R. Yang, and D. Lin. Cylindrical and asymmetrical 3d convolution networks for lidar-based perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6807–6822, 2021.

[50] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307, 2019.

[51] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6398–6407, 2020.

A Complementary Datasets

The KITTI [1] dataset contains LiDAR scans collected from urban driving trajectories in Karlsruhe, with poses provided by integrated GPS/IMU systems. To avoid overlap between training and testing regions, we use sequences 01, 02, 04, 05, 06, and 08 for training, and sequences 00 and 07 for testing. Sequence 03 is excluded due to unavailable GPS information.

The KITTI-360 [18] extends the KITTI dataset with longer suburban routes. Following the former practice [16], synchronized sequences 00, 05, 06, and 09 are utilized for testing. Statistics of query point cloud frames, osm tiles and trajectory length are shown in Tab. 6.

Table 6: Statistics of test sets in KITTI and KITTI-360.

Dataset	KITTI		KITTI-360			
Sequence	00	07	00	05	06	09
Point Cloud Frames	4541	1101	10514	6291	9186	13247
OSM Tiles	8782	3332	12491	15080	12730	13060
Trajectory Length (m)	8478	3226	11612	14541	12201	12570

For the OSM tile sampling strategy, we follow the settings of Cho et al. [16]. During training, OSM tiles are sampled to align with the centers of the corresponding point cloud scans. In contrast, during testing, OSM tiles are uniformly sampled at 1 m interval on the 'highway' layer in the OSM data.

B Implementation Details

B.1 Data Pre-processing

For both KITTI [1] and KITTI-360 [18] datasets, we use Cylinder3D [49] pretrained on KITTI to predict 19-class semantic labels (following SemanticKITTI [50]) for each query point cloud.

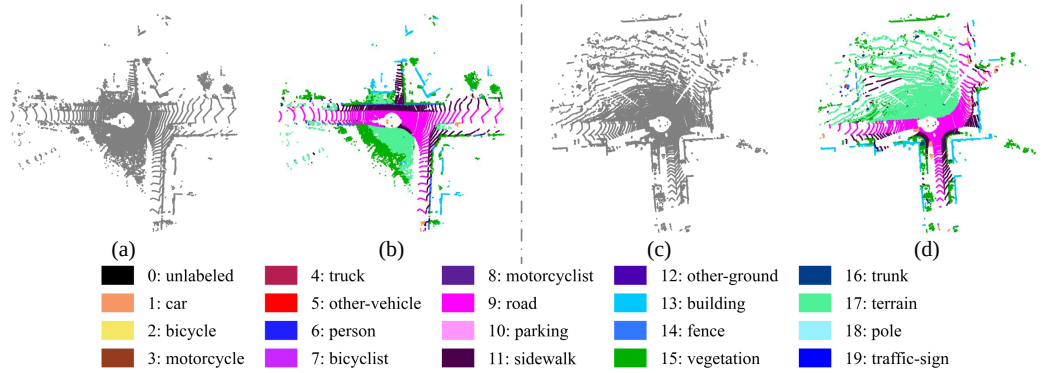


Figure 7: Details of semantic point cloud. Figures (a) and (c) display the raw point clouds, while (b) and (d) render them with semantic coloring.

The OSM data is processed through a structured pipeline to generate georeferenced semantic representations. The raw OSM data, comprising various entities, is first categorized into areas, ways and nodes classes according to the hierarchical classification detailed in Tab. 7. Each class is projected onto a local East-North coordinate frame and rasterized into a Cartesian grid with a fixed resolution of Δ_o = 50 cm/pixel. As shown in Fig. 8, the OSM tiles preserve the semantic and geographic information.

B.2 Loss Function

Our OPAL employs the circle loss [51] for optimization. During training, for each query point cloud $\mathcal{P}$ in a mini-batch, we consider its geographically matching OSM tile as the positive anchor $\mathcal{O}_{pos}$, whereas all other tiles are treated as negative samples $\mathbb{O}_{neg}$. In the shared feature space of the global descriptor, the query point cloud should be close to the positive anchor and far from all the negative

Table 7: Details of OSM elements.

Type	Element
Areas	building, parking, playground, grass, park, forest, water
Ways	Fence, wall, hedge, kerb, building outline, cycleway, path, road, busway, tree row
Nodes	parking entrance, street lamp, junction, traffic signal, stop sign, give way sign, bus stop, stop area, crossing, gate, bollard, gas station, bicycle parking, charging station, shop, restaurant, bar, vending machine, pharmacy, tree, stone, ATM, toilets, water fountain, bench, waste basket, post box, artwork, recycling station, clock, fire hydrant, pole, street cabinet

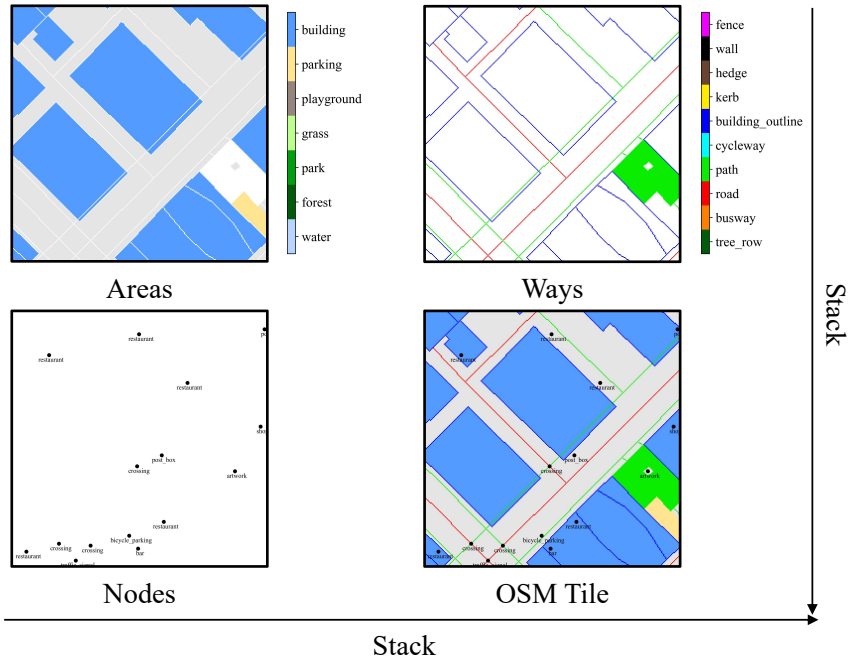


Figure 8: Illustration of areas, ways, nodes channels and full OSM tile.

anchors. The similarity between a query point cloud descriptor $\boldsymbol{d}_P$ and an OSM tile descriptor $\boldsymbol{d}_O$ is measured through cosine similarity:

$$s = \frac{< \boldsymbol{d}_P, \boldsymbol{d}_O >}{\|\boldsymbol{d}_P\| \|\boldsymbol{d}_O\|}. \tag{9}$$

The optimization objective simultaneously maximizes the similarity s_{pos} between queries and their positive anchors while minimizing similarities s_{neg} with negative anchors. The optimization objective is defined as:

$$\mathcal{L} = \log \left[1 + \sum_{i=1}^{|\mathbb{O}_{neg}|} \exp\left(\gamma \alpha_{neg}^{i} (s_{neg}^{i} - \Delta_{neg})\right) \cdot \exp\left(-\gamma \alpha_{pos} (s_{pos} - \Delta_{pos})\right) \right] \tag{10}$$

where $\alpha_{neg}^{i} = \max\left(0, s_{neg}^{i} + \Delta_{neg}\right)$ and $\alpha_{pos} = \max\left(0, 1 + \Delta_{pos} - s_{pos}\right)$ are dynamic weights for negative and positive mining respectively. The hyperparameters Δ_{pos} and Δ_{neg} establish safe margins in the embedding space, while γ is a scaling factor controlling gradient sensitivity.

B.3 Parameters Setting

Point clouds are filtered to retain points within a range of 3 m to 50 m, and OSM tiles are of size $H \times W = 100\,\text{m} \times 100\,\text{m}$. The polar representation consists of $R = 480$ rings and $P = 360$ sectors. By default, C_{oem} and C_{pem} are set to 16 and 64, respectively, consistent with the configurations

adopted in OrienterNet [14] and PolarNet [40]. In the loss function, positive margin Δ_{pos}, negative margin Δ_{neg} and scale factor γ in the loss function are set to 0.2, 1.8 and 10, respectively.

C Additional Results

We provide extensive qualitative results on the KITTI and KITTI-360 datasets, as shown in Fig. 9. Compared with both hand-crafted methods [16, 46] and learning-based methods [27], the proposed OPAL achieves more accurate and robust performance in various scenarios.

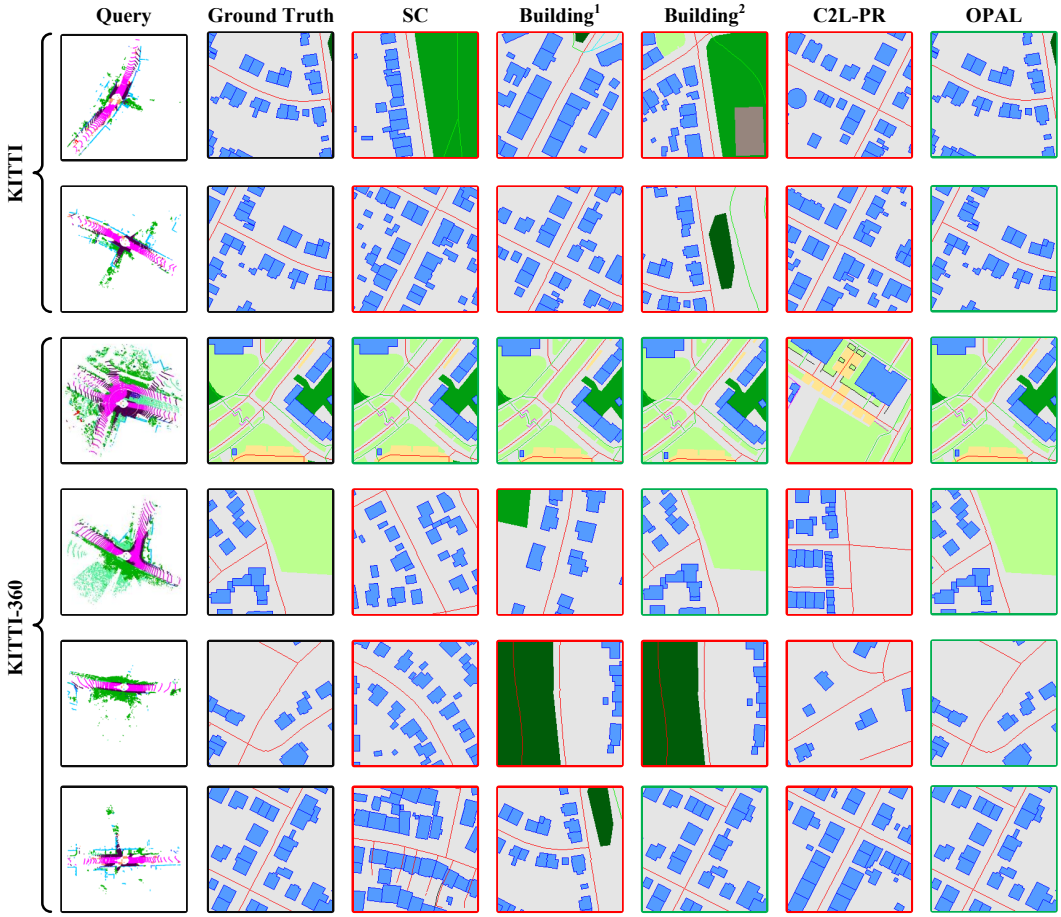


Figure 9: Examples of LiDAR queries and their top-1 retrieved matches on KITTI and KITTI-360 datasets. Red rectangles □ represent the wrong retrieved results and green rectangles □ represent the correct retrieved results.

D Extended Ablation Study

Influence of dynamic objects: Since OSM data contains only static map elements, whereas query point clouds include numerous dynamic objects (e.g., pedestrians and moving vehicles), this discrepancy may influence localization performance. To examine this effect, we remove dynamic objects from the KITTI dataset using ground-truth semantic labels in the test set and re-evaluate our model. As reported in Tab. 8, OPAL achieves comparable performance with and without dynamic objects, highlighting its robustness to such distractions.

Influence of noise in OSM data: To investigate the effect of noise in OSM data, we introduce random offsets in $[0\text{m}, 0.5\text{m}]$ to simulate localization noise, as shown in the last row of Tab. 8. Despite the degraded OSM quality, our method maintains robustness and delivers comparable performance.

Table 8: Ablation studies on the influence of dynamic objects and random noise in OSM data (Top-1 Recall@Km retrieved results on the KITTI dataset).

Setting	KITTI 00			KITTI 07		
	R@1	R@5	R@10	R@1	R@5	R@10
OPAL	21.82	65.78	66.40	45.41	69.85	70.30
OPAL w/o Mov. Obj.	21.80	65.78	66.42	45.14	69.66	70.21
OPAL w/ Random Noise	21.45	65.16	65.67	43.96	68.85	68.94

Descriptor dimension. Localization accuracy is highly dependent on the dimensionality of the global descriptor. Our method employs 2048-dimensional (2048-D) global descriptors. For handcrafted methods, we follow the official implementations: SC [46] uses a 1200-D descriptor, while Building [16] adopts 10-D and 360-D descriptors in the first and second stages, respectively. For the learning-based approach, C2L-PR [27] utilizes 3240-D (point clouds) and 2160-D (OSM) descriptors in stage one, followed by a 288-D descriptor in stage two. As shown in Tab. 9, we additionally report C2L-PR results with a 2048-D descriptor in stage two for a fair comparison. Under the same descriptor dimensionality, our method achieves a significant performance gain over C2L-PR.

Effect of visibility mask. To provide a comprehensive analysis of the visibility mask, we report the performance of OPAL with and without the mask across all test sequences of the KITTI and KITTI-360 datasets, as shown in the last two rows of Tab. 9. The results indicate that incorporating the visibility mask consistently improves performance on every sequence, highlighting its effectiveness.

Table 9: Ablation studies on global descriptor dimension (Top-1 Recall@5 m on KITTI (K-) and KITTI-360 (K360-) datasets).

Method	Dim	K-00	K-07	K360-00	K360-05	K360-06	K360-09
C2L-PR	288	9.69	13.17	8.23	4.72	3.35	9.59
C2L-PR	2048	28.52	26.79	18.97	9.12	8.01	22.13
OPAL w/o mask	2048	62.21	57.49	26.15	22.62	26.98	53.90
OPAL w/ mask	2048	**65.78**	**69.85**	**42.82**	**30.49**	**36.38**	**60.96**

Computational overhead: We use the $fvcore$ toolbox to measure the computational overhead of the learning-based baseline, C2L-PR, and our method. C2L-PR requires a total of 535.79 ms, comprising 0.122 M parameters, 0.694 M FLOPs, 1.42 ms for the learnable second stage, and 534.37 ms for the handcrafted first stage. In contrast, our method runs in 7.05 ms, with 88.18 M parameters and 30.59 G FLOPs for the whole process.