OLA-Place: Cross-Modal Place Recognition without Global Descriptors

Tianyi Shang
Purdue University, USA
shangt@purdue.edu

Zhenyu Li*
Qilu University of Technology
(Shandong Academy of Sciences), China
lizhenyu@qlu.edu.cn

Abstract

Cross-modal place recognition aims to retrieve a target 3D location from a spatial map using a natural language description. Most existing methods follow a global-descriptor matching paradigm, in which the text query and each 3D scene cell are independently compressed into single vectors and compared by global similarity. Although simple and widely adopted, this paradigm tends to discard the fine-grained object correspondences that are essential for language-guided localization. In this paper, we challenge the necessity of global descriptors and propose OLA-Place, an Object-Level Alignment framework for cross-modal place recognition without global descriptors. Instead of representing a query or a scene cell as a single holistic embedding, OLA-Place formulates place recognition as set-to-set semantic alignment between textual object mentions and 3D object instances. The framework contains three key modules: ObjectSet Encoder (OSE), which extracts object-level representations from both language descriptions and 3D scene cells; Object Message Encoder (OME), which injects intra-set object-context information while preserving object-level granularity; and Masked Max Alignment (MMA), which computes the query-cell matching score by aligning each textual object mention with its most similar valid 3D object. This formulation is permutation-invariant, naturally handles variable-size object sets, and requires no object-level correspondence annotations. Extensive experiments show that object-level alignment alone significantly outperforms global-descriptor-based methods, demonstrating that cross-modal place recognition is better understood as an object-level semantic correspondence problem rather than a global-embedding retrieval problem. Our code is available at: https://github.com/nuozimiaowu/OLA-Place.

1 Introduction

Cross-modal place recognition retrieves a target 3D location from a spatial map using a natural language query. Given a description such as "a red car is parked near a pole in front of a building", the system must find the corresponding 3D scene cell from a gallery. This problem is central to language-guided localization, embodied navigation, robotics, and autonomous driving. Most text-guided point-cloud localization pipelines adopt a coarse-to-fine strategy: a coarse module first retrieves candidate 3D cells, and a fine module then estimates the precise coordinate inside the retrieved cell. Since the fine stage is strongly constrained by the retrieved candidates, this paper focuses on the coarse cross-modal place recognition stage.

A common coarse retrieval paradigm is global descriptor matching. A text encoder compresses the whole query into a vector $\mathbf{z}_q = f_T(q)$, a 3D encoder compresses each cell into $\mathbf{z}_C = f_C(C)$, and retrieval is performed by $S(q, C) = \cos(\mathbf{z}_q, \mathbf{z}_C)$. This formulation is efficient, but it forces all objects,

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

colors, geometry, and spatial cues into a single embedding before matching. Such compression can obscure fine-grained correspondences, e.g., whether the textual mention "red car" is actually grounded by a car object in the candidate cell.

We argue that language-based place recognition is inherently object-centric. Natural language queries usually describe places through objects and their attributes, while 3D cells are also composed of object instances. Therefore, the shared semantic units between the two modalities are objects rather than whole-scene descriptors. This motivates a different formulation: instead of comparing two global vectors, we directly align textual object-level embeddings with 3D object-level embeddings.

We propose OLA-Place, a global-descriptor-free framework for cross-modal place recognition. OLA-Place represents a query as textual object features and a 3D cell as scene object features. It consists of three key components. First, OSE maps both modalities into object-indexed representations: textual units are encoded as language-side object features, while 3D object instances are encoded from point-cloud geometry, semantic attributes, color, and position cues. Second, OME refines scene object features by modeling intra-cell object context and relative geometry, while still preserving one feature vector for each object slot instead of pooling them into a global scene descriptor. Third, MMA compares the two object sets by computing pairwise similarities, masking invalid padded objects, and grounding each textual object feature to its most similar valid 3D object. The query-cell score is obtained by averaging these best-match similarities across textual features. No object-level correspondence labels are required; only query-cell pair supervision is used.

In other words, OSE provides a compact object-wise representation for each modality, OME enriches the 3D object set with local context without collapsing it into a single descriptor, and MMA turns the two sets into a retrieval score through masked one-to-many grounding. This division of labor keeps the overall pipeline object-centric from input encoding to final matching, which is precisely what enables our descriptor-free formulation.

The main contributions are:

- Object-centric retrieval paradigm: We move away from the traditional global-descriptor matching paradigm and propose an object-centric framework for cross-modal place recognition. By aligning textual descriptions with 3D object instances, we avoid the information loss inherent in compressing entire scenes and queries into single vectors.
- The OLA-Place framework: We introduce OLA-Place, which uses OSE for object-wise encoding, OME for contextual 3D object refinement, and MMA for masked grounding. This design enables fine-grained retrieval while remaining descriptor-free.

2 Related Work

2.1 Visual Place Recognition

Historically, 2D visual localization has depended on feature aggregation algorithms—most notably VLAD [1] and GeM—to construct global scene representations for direct image matching. To further elevate robustness, contemporary research investigates the structural correlations between cluster centroids and local features [11]. Concurrently, the community is rapidly adopting end-to-end Transformer models to generate highly discriminative global embeddings [13, 25]. Researchers have also tailored specialized architectures to mitigate the impact of severe environmental challenges, such as poor illumination [12] and extreme viewpoint variations [2]. Recent advancements have even scaled these visual positioning frameworks to operate effectively across continental distances [15]. As a complementary strategy, cross-modal techniques—including text-to-image [20] and point-cloud-to-image [28] retrieval—have gained traction, offering reliable alternatives when standard visual conditions are severely compromised.

2.2 LiDAR Place Recognition

In LiDAR-based localization, global signatures are typically formulated from either unstructured 3D points or Bird's-Eye View (BEV) projections [14]. Early milestone architectures like PointNetVLAD [21] successfully coupled NetVLAD with PointNet [18] to extract compact spatial vectors. Subsequent BEV-focused networks, including I2P-Rec [31] and BevPlace [17], have achieved enhanced rotational invariance by exploiting geometric transformations and multimodal projection strategies.

Another prominent branch of LPR research targets the inherent topology of 3D data via graph neural networks and 3D CNNs. For example, LPD-Net [16] leverages graph modules to encode local geometries, whereas architectures like SOENet [26] and DH3D [4] prioritize precise pose refinement. To maintain feature stability during continual learning phases, frameworks such as InCloud [8] have been proposed. The integration of Transformer networks has fundamentally advanced the modeling of long-range dependencies in 3D scenes; TransLoc3D [29] applies global self-attention, while models like PPT-Net [6] and NDT-Transformer [32] employ hierarchical paradigms to process multi-scale and NDT-transformed inputs.

To address the computational demands of large-scale mapping, sparse 3D convolution frameworks like MinkLoc3D [10] are widely utilized. Furthermore, cutting-edge methodologies incorporate optimal transport theory for superior feature matching (LCDNet [3]) and enforce local consistency constraints to guarantee viewpoint-invariant descriptors (LoGG3D-Net [22]).

2.3 Cross-Modal Localization

The paradigm of text-driven point-cloud retrieval was largely pioneered by Text2Pos [9], which established a multi-step coarse-to-fine localization strategy. This architecture was progressively refined by RET [23], incorporating attention modules, and Text2Loc [27], which integrated contrastive learning with an innovative matching-free fine-tuning stage. Recently, state space models (SSMs) have been successfully applied to this domain through MambaPlace [19]. Meanwhile, GOTPR [7] showcased how intermediate graph representations can substantially optimize the efficiency of the initial coarse retrieval phase.

Contemporary studies are increasingly focused on overcoming practical limitations, such as ambiguous cross-modal alignments and insufficient training data. IFRP-T2P [24], for instance, circumvents the need for explicit instance labeling by merging query extractors with RowColRPA/RPCA mechanisms to obtain robust spatial awareness. The management of matching uncertainty has also become a critical research direction; PMSH [5] tackles this by directly embedding uncertainty metrics into the retrieval process for partially overlapping submaps. Similarly, CMMLoc [30] mitigates the issue of incomplete text queries by employing a Cauchy-Mixture-Model alongside spatial consolidation and cardinal-direction heuristics.

3 Method

3.1 Task Formulation and Notation

OLA-Place targets coarse cross-modal retrieval. Given a query q_i, the goal is to retrieve the most relevant cell from a gallery $\mathcal{G} = \{C_j\}_{j=1}^{N_c}$ by maximizing a matching score $S(q_i, C_j)$:

$$\widetilde{C}_i = \arg\max_{C_j \in \mathcal{G}} S(q_i, C_j). \tag{1}$$

Unlike global-descriptor methods, OLA-Place computes $S(q_i, C_j)$ directly from object sets. For a mini-batch of size B, each query is represented by textual object features $\mathbf{T} \in \mathbb{R}^{B\times M\times d}$, and each cell by scene object features $\mathbf{O} \in \mathbb{R}^{B\times N\times d}$ together with an occupancy mask $\mathbf{M} \in \mathbb{R}^{B\times N\times 1}$. Here M is the number of textual object units, N is the fixed number of object slots per cell, and d is the embedding dimension. We write $\mathbf{t}_{im}$ and $\mathbf{o}_{jn}$ for the m-th textual feature of query q_i and the n-th object feature of cell C_j, respectively, and use $M_{jn} \in \{0, 1\}$ to denote whether a slot is valid.

Thus, the pipeline is simply

$$q_i \to \mathbf{T}_i, \qquad C_j \to (\mathbf{O}_j, \mathbf{M}_j), \tag{2}$$

followed by masked object-level alignment. This formulation keeps the retrieval stage object-centric from the outset.

3.2 OSE: Textual Object Encoding

OSE maps each query to a set of object-level language features. Its role is to preserve object-centric language cues rather than compress the query into a single vector. Given a query q_i, we split it into M textual units $\{u_{im}\}_{m=1}^{M}$ and encode each unit with a language backbone. Let

$$\mathbf{H}_{im} \in \mathbb{R}^{L_m\times d_T} \tag{3}$$

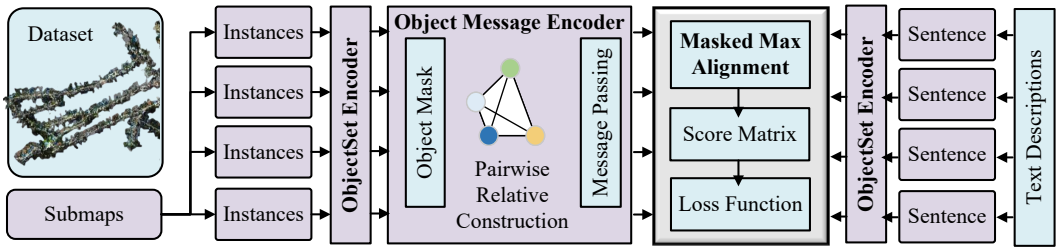


Figure 1: Overall architecture of the proposed OLA-Place framework.

be the token-level hidden states of unit u_{im}, where L_m is the token length and d_T is the language-model hidden dimension.

The encoder then applies token-level contextualization, pooling, and sentence-level aggregation to obtain a unit embedding

$$\mathbf{e}_{im}^{T} = \mathrm{InterEnc}\left(\psi_T\left(\mathrm{MaxPool}(\mathrm{IntraEnc}(\mathbf{H}_{im}))\right)\right)_m . \tag{4}$$

Stacking all units yields $\mathbf{E}_i^T \in \mathbb{R}^{M\times d}$. A final normalization produces the object-level textual set

$$\mathbf{T}_i = \mathrm{Normalize}(\mathbf{E}_i^T). \tag{5}$$

Thus, OSE returns one embedding per textual unit, which can later be matched to multiple scene objects in a permutation-invariant way.

3.3 OSE: 3D Object Encoding

OSE maps each 3D cell to a set of object-level scene features. Its purpose is to extract object evidence from geometry and attributes before any cross-modal matching occurs. Each cell C_j is represented as a set of object instances $\{x_{j1}, \ldots, x_{jN_j}\}$, where N_j is the number of objects before padding. Every object may contain semantic, geometric, and appearance cues.

For point-cloud features, a PointNet2-style encoder extracts an object geometry descriptor and projects it to the shared dimension d. Additional attributes, including RGB color, object center $\mathbf{p}_{jn} \in \mathbb{R}^3$, normalized point count ν_{jn}, and optional class/color embeddings, are encoded by lightweight MLPs or embedding tables. Let $\{\mathbf{a}_{jn}^k\}_{k\in\mathcal{F}}$ denote the selected feature components for object x_{jn}, where $\mathcal{F}$ is the enabled feature set. After normalization, the components are fused by an MLP:

$$\mathbf{e}_{jn}^{O} = \psi_{\text{merge}}\left(\mathrm{Concat}\{\mathrm{Norm}(\mathbf{a}_{jn}^k)\}_{k\in\mathcal{F}}\right). \tag{6}$$

OSE therefore produces a clean object-indexed scene representation. It does not yet encode cell-level context or perform query-cell matching; those roles are delegated to OME and MMA, respectively.

3.4 OME: Object Message Encoding

OME refines the object sets produced by OSE with intra-cell context. Its role is local and well-defined: it operates only within a scene cell, uses no language input, and preserves one feature vector per object slot. The goal is to enrich each object representation with neighboring objects and relative geometry before matching.

The flat object tensor is first scattered into a padded cell tensor

$$\mathbf{O}^{(0)} \in \mathbb{R}^{B\times N\times d}, \tag{7}$$

where N is the fixed number of object slots. Object centers are padded analogously into $\mathbf{P} \in \mathbb{R}^{B\times N\times 3}$. For cell C_j, the binary object mask is

$$M_{jn} = \begin{cases} 1, & 1 \le n \le \min(N_j, N), \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The relation mask is $R_{jab} = M_{ja}M_{jb}$.

OME then builds pairwise relative geometry $\mathbf{r}_{jab} = \mathbf{p}_{ja} - \mathbf{p}_{jb}$ and projects it to initial relation features $\mathbf{Q}^{(0)} \in \mathbb{R}^{B\times N\times N\times d}$. At each layer ℓ, the object stream is updated by a recurrent operator over object slots and masked:

$$\mathbf{O}^{(\ell)} = \mathrm{Rec}_O(\mathbf{O}^{(\ell-1)}) \odot \mathbf{M}. \tag{9}$$

The relation stream is updated from source objects, target objects, and previous relation features:

$$\mathbf{Q}^{(\ell)}_{jab} = \tanh\left(\mathbf{W}_r[\mathbf{o}^{(\ell-1)}_{ja}; \mathbf{o}^{(\ell-1)}_{jb}; \mathbf{q}^{(\ell-1)}_{jab}]\right) \odot R_{jab}. \tag{10}$$

After the message layers, the scene object features are normalized:

$$\mathbf{O}_j = F^P_{\mathrm{Object},j} = \mathrm{Normalize}(\mathbf{O}^{(L)}_j). \tag{11}$$

OME therefore acts as a context-refinement block: it strengthens object representations before alignment, but the retrieval rule itself remains unchanged.

3.5 MMA: Masked Max Alignment

MMA converts two object sets into a single retrieval score. For query q_i and cell C_j, let $\mathbf{T}_i = [\mathbf{t}_{i1}, \ldots, \mathbf{t}_{iM}] \in \mathbb{R}^{M\times d}$ and $\mathbf{O}_j = [\mathbf{o}_{j1}, \ldots, \mathbf{o}_{jN}] \in \mathbb{R}^{N\times d}$ denote the normalized textual and scene object tensors. The pairwise similarity between textual unit m and scene object slot n is

$$A^{mn}_{ij} = \mathbf{t}^\top_{im}\mathbf{o}_{jn}. \tag{12}$$

Because both features are normalized, this dot product is cosine similarity.

To handle varying object counts, MMA applies the occupancy mask $\mathbf{M}_j$ and suppresses padded slots before max pooling:

$$\widehat{A}^{mn}_{ij} = \begin{cases} A^{mn}_{ij}, & M_{jn} = 1, \\ -100, & M_{jn} = 0. \end{cases} \tag{13}$$

Each textual unit then selects its best valid object match, and the final score is the average of these best matches:

$$S^O_{ij} = S(q_i, C_j) = \frac{1}{M}\sum_{m=1}^{M} \max_{1\le n\le N} \widehat{A}^{mn}_{ij}. \tag{14}$$

This one-sided soft grounding is well suited to language-based place recognition: each textual cue searches over all scene objects, while the average over M units rewards cells that explain the full query. The score is permutation-invariant, robust to variable-size object sets, and does not require object correspondence labels.

For a mini-batch, the implementation vectorizes Eq. (14). It forms

$$\mathcal{A}_{ijmn} = \mathbf{t}^\top_{im}\mathbf{o}_{jn}, \qquad \mathcal{A} \in \mathbb{R}^{B\times B\times M\times N}, \tag{15}$$

applies the broadcasted object mask, and produces a score matrix $\mathbf{S}^O \in \mathbb{R}^{B\times B}$. Entry S^O_{ij} is the object-level alignment score between query q_i and cell C_j; diagonal entries are positives and off-diagonal entries are in-batch negatives.

3.6 Training Loss

The object branch is optimized by a score-based contrastive calibration loss, which receives the score matrix $\mathbf{S}^O$ and a distance weight matrix. It does not receive global descriptors. The score matrix is first converted to exponentiated scores $\exp(S^O_{ij}/\tau)$, where τ is the temperature. The row-normalized and column-normalized probabilities are

$$P^{q\to C}_{ij} = \frac{\exp(S^O_{ij}/\tau)}{\sum_{k=1}^{B}\exp(S^O_{ik}/\tau)}, \qquad P^{C\to q}_{ij} = \frac{\exp(S^O_{ij}/\tau)}{\sum_{k=1}^{B}\exp(S^O_{kj}/\tau)}. \tag{16}$$

The implementation removes diagonal positives and selects hard negatives by top-K over $P^{q\to C}$. If ratio is outside $(0, 1)$, $K = B - 1$; otherwise $K = \lfloor \texttt{ratio} \cdot B \rfloor$ clipped to $[1, B - 1]$. Let $H_{ij} \in \{0, 1\}$ denote the resulting hard-negative mask.

The distance weight matrix is computed outside the loss. The code averages object features over textual units and object slots, normalizes the means, and computes

$$D_{ij} = \left\| \mathrm{Norm}\left(\frac{1}{M} \sum_{m} \mathbf{t}_{im} \right) - \mathrm{Norm}\left(\frac{1}{N} \sum_{n} \mathbf{o}_{jn} \right) \right\|_2 . \tag{17}$$

The distance is normalized by its batch maximum and converted to

$$W_{ij} = \left(1 - \frac{D_{ij}}{\max_{a,b} D_{ab}} + 10^{-6} \right)^{1/\alpha} . \tag{18}$$

For the default logarithmic criterion $\varphi(x) = -\log(1 - x + \epsilon)$, the loss is

$$\mathcal{L}_{\mathrm{obj}} = \frac{1}{B} \sum_{i,j} H_{ij} W_{ij} \varphi(P_{ij}^{q \to C}) + \frac{1}{B} \sum_{i,j} H_{ij} W_{ji} \varphi(P_{ij}^{C \to q}). \tag{19}$$

This objective suppresses high-probability negative query-cell pairs according to the object-alignment score matrix. All supervision is at the query-cell level; no textual-unit to 3D-object correspondence label is used.

3.7 Inference

At inference time, database cells are encoded into object features and object masks. A query is encoded into textual object features. The same masked max-alignment score in Eq. (14) is computed between the query and every database cell, and the cells are ranked by descending score:

$$\mathrm{Rank}(q_i) = \mathrm{argsort}_{C_j \in \mathcal{G}}(-S_{ij}^{O}). \tag{20}$$

The inference path is therefore identical to training in its scoring rule. The only difference is that training scores are computed within a mini-batch, while inference scores are computed against the full gallery.

3.8 Fine Stage

For the final coordinate regression within the retrieved submap, we follow the fine-stage design of Text2Loc. Specifically, a cascaded cross-attention module fuses geometry-enhanced point-cloud features with aligned text features, and an MLP regressor predicts the relative offset to the submap anchor point. The absolute location is then recovered directly from this offset.

4 Experiment

4.1 Experiment Setup

The performance of the proposed OLA-Place framework is primarily assessed on the KITTI360Pose benchmark. Spanning a total area of 15.51 km^2 across nine city regions, this dataset is divided into three subsets: a training split with 28,689 queries from five zones, a validation split with 3,187 queries from one zone, and a test split with 11,505 queries from the remaining three zones. All experiments were conducted on a hardware platform equipped with a 128-core AMD EPYC processor and an NVIDIA RTX V100 GPU (32 GB). For the coarse retrieval stage, the network is trained for 20 epochs with a batch size of 64 and a learning rate of 5×10^{-4}, producing 256-dimensional feature embeddings. The subsequent fine-grained localization stage is trained for 100 epochs with a batch size of 32 and a learning rate of 3×10^{-4}. These hyperparameter settings follow the established evaluation protocol for KITTI360Pose.

To further verify the cross-domain generalization capability of our model, we introduce StreetLoc. This street-level text-to-point-cloud benchmark is designed to assess algorithmic adaptability in heterogeneous urban environments. The data collection covers a 4.7-kilometer driving trajectory, using a 64-channel LiDAR sensor paired with high-precision GPS. Following the data-generation pipeline of KITTI360Pose, we synthesize 1,600 query-submap pairs. In addition, to support finer analysis of semantic interactions, we provide manual annotations for scene objects spanning 19 semantic classes.

Table 1: Comparison between OLA-Place and existing state-of-the-art methods on the KITTI360Pose test set and the StreetLoc dataset. Each entry reports localization recall under 5m/10m error thresholds, and the metrics evaluate the full pipeline, including both coarse submap retrieval and fine localization.

Methods	Localization Recall ($\epsilon < 5/10m$) ↑					
	KITTI360Pose			StreetLoc		
	$k = 1$	$k = 5$	$k = 10$	$k = 1$	$k = 5$	$k = 10$
Text2Pos (CVPR'22)	0.13/0.20	0.33/0.42	0.43/0.61	0.14/0.17	0.27/0.31	0.36/0.52
RET (AAAI'23)	0.16/0.25	0.35/0.51	0.46/0.65	0.13/0.21	0.32/0.47	0.40/0.58
Text2Loc (CVPR'24)	0.33/0.48	0.60/0.75	0.70/0.84	0.29/0.42	0.47/0.66	0.55/0.72
IFRP-T2P (ACM MM'24)	0.22/0.40	0.47/0.68	0.58/0.78	0.20/0.33	0.45/0.63	0.53/0.68
MambaPlace (IROS'25)	0.38/0.52	0.66/0.79	0.76/0.87	0.32/0.42	0.50/0.70	0.58/0.76
CMMLoc (CVPR'25)	0.39/0.53	0.67/0.80	0.77/0.87	0.31/0.44	0.50/0.68	0.57/0.78
PMSH (ICCV'25)	0.39/0.55	0.68/0.81	0.78/0.89	0.30/0.43	0.52/0.70	0.59/0.77
OLA-Place	**0.50/0.71**	**0.76/0.91**	**0.82/0.95**	**0.40/0.53**	**0.65/0.78**	**0.74/0.87**

4.2 Comparison with SOTA Methods

Table 1 summarizes the overall localization performance of OLA-Place against existing state-of-the-art methods on both KITTI360Pose and StreetLoc. Across all recall settings, our method achieves the best results. On KITTI360Pose, OLA-Place reaches 0.50/0.71 at $k = 1$, 0.76/0.91 at $k = 5$, and 0.82/0.95 at $k = 10$, outperforming the strongest competing baseline by a clear margin. In particular, compared with PMSH, which is the best-performing prior method on KITTI360Pose, OLA-Place improves the strictest recall setting from 0.39/0.55 to 0.50/0.71, indicating that object-level alignment is especially effective when the retrieval budget is small and the correct candidate must be ranked near the top.

The gains are even more pronounced on StreetLoc, which is more challenging because of its cross-domain street-level setting and richer object diversity. OLA-Place achieves 0.40/0.53 at $k = 1$, 0.65/0.78 at $k = 5$, and 0.74/0.87 at $k = 10$, setting a new state of the art on this benchmark. Relative to the strongest baseline, the improvement at $k = 1$ is especially notable, suggesting that the proposed object-centric formulation generalizes better than global descriptor matching under domain shift. These results support our central claim that fine-grained object correspondences provide a more reliable retrieval signal than holistic embeddings.

Table 2 further isolates the coarse retrieval stage and confirms that the superiority of OLA-Place is not limited to the full localization pipeline. On KITTI360Pose, OLA-Place achieves recall of 0.49, 0.72, and 0.79 at $k = 1/3/5$, corresponding to absolute improvements of 15, 16, and 14 percentage points over the strongest baseline, respectively. On StreetLoc, the method reaches recalls of 0.41, 0.62, and 0.70, improving over the best competitor by 15, 14, and 13 percentage points.

Table 2: Coarse-stage submap retrieval recall on the KITTI360Pose and StreetLoc test sets.

Methods	KITTI360Pose (Recall ↑)			StreetLoc (Recall ↑)		
	k = 1	k = 3	k = 5	k = 1	k = 3	k = 5
Text2Pos	0.12	0.25	0.33	0.10	0.22	0.25
RET	0.15	0.29	0.37	0.14	0.23	0.30
Text2Loc	0.28	0.49	0.58	0.23	0.42	0.50
IFRP-T2P	0.23	0.39	0.48	0.19	0.31	0.40
MambaPlace	0.31	0.53	0.62	0.26	0.45	0.57
CMMLoc	0.32	0.53	0.63	0.25	0.47	0.55
PMSH	0.34	0.56	0.65	0.26	0.48	0.57
OLA-Place	**0.49**	**0.72**	**0.79**	**0.41**	**0.62**	**0.70**

Table 3: Ablation study on the key components of OLA-Place on the KITTI360Pose and StreetLoc test sets. ‘w/o OME‘ replaces the original OME with a two-layer MLP; ‘w/o MMA‘ replaces masked max alignment with direct pooled similarity; ‘w/o Mask‘ computes alignment without occupancy masking.

Methods	KITTI360Pose (Recall ↑)			StreetLoc (Recall ↑)		
	k = 1	k = 3	k = 5	k = 1	k = 3	k = 5
w/o OME	0.31	0.55	0.64	0.24	0.44	0.53
w/o MMA	0.40	0.62	0.71	0.31	0.52	0.60
w/o Mask	0.38	0.61	0.70	0.32	0.50	0.61
Full OLA-Place	**0.49**	**0.72**	**0.79**	**0.41**	**0.62**	**0.70**

4.3 Ablation Study

The ablation study in Table 3 verifies the role of each module in the proposed object-level retrieval pipeline and is consistent with the method design. Removing OME leads to the largest performance drop, reducing KITTI360Pose recall@1 from 0.49 to 0.31 and StreetLoc recall@1 from 0.41 to 0.24. This suggests that the contextual refinement performed by OME is not merely auxiliary but an important step for strengthening object representations before cross-modal matching. As described in the Method section, OME enriches each object slot with intra-cell context and relative geometry, so the final object features are no longer isolated local descriptors but context-aware embeddings. Without this message-passing stage, visually or semantically similar objects in the same submap become harder to distinguish, which is especially harmful when the query mentions only a subset of objects.

Replacing MMA with pooled similarity also causes a clear decline on both datasets. This result supports the design of masked max alignment, where each textual unit is grounded to its most similar valid scene object rather than being blended into a pooled scene summary. The max-over-object operation preserves fine-grained object correspondences and allows different query phrases to match different 3D instances, which is exactly the behavior we want for language-based place recognition. In contrast, pooled similarity collapses the alignment process too early and weakens the model's ability to exploit sparse but highly informative object matches, leading to lower recall across all retrieval ranks.

Removing the occupancy mask also degrades performance, which further confirms that the set-matching procedure must explicitly handle variable object counts. Since each cell is padded to a fixed number of slots, invalid slots can otherwise participate in similarity computation and introduce spurious high responses. The mask in MMA suppresses those padded positions before max pooling, ensuring that only real objects contribute to the final score. This is particularly important when the number of objects varies widely across scenes, because without masking the similarity matrix becomes contaminated by meaningless entries and the ranking quality deteriorates.

4.4 Computational Cost

We compare the efficiency of OLA-Place and the state-of-the-art method PMSH under identical configurations in the coarse stage. PMSH contains 372M parameters in total, including 336M backbone parameters and 36M non-backbone parameters, with an inference runtime of approximately 39 ms. In comparison, OLA-Place contains 392M parameters in total, where the backbone also accounts for 336M parameters and the non-backbone components account for 56M parameters, with an inference runtime of approximately 43 ms. In particular, OLA-Place achieves these gains with only a modest increase in non-backbone parameters and inference time compared with PMSH, indicating that the improved retrieval accuracy comes at a small computational cost.

4.5 Visualization Results

The qualitative examples show that OLA-Place ranks the target within the top three candidates in most cases. The strong retrieval performance indicates that instance-level descriptors bridge the

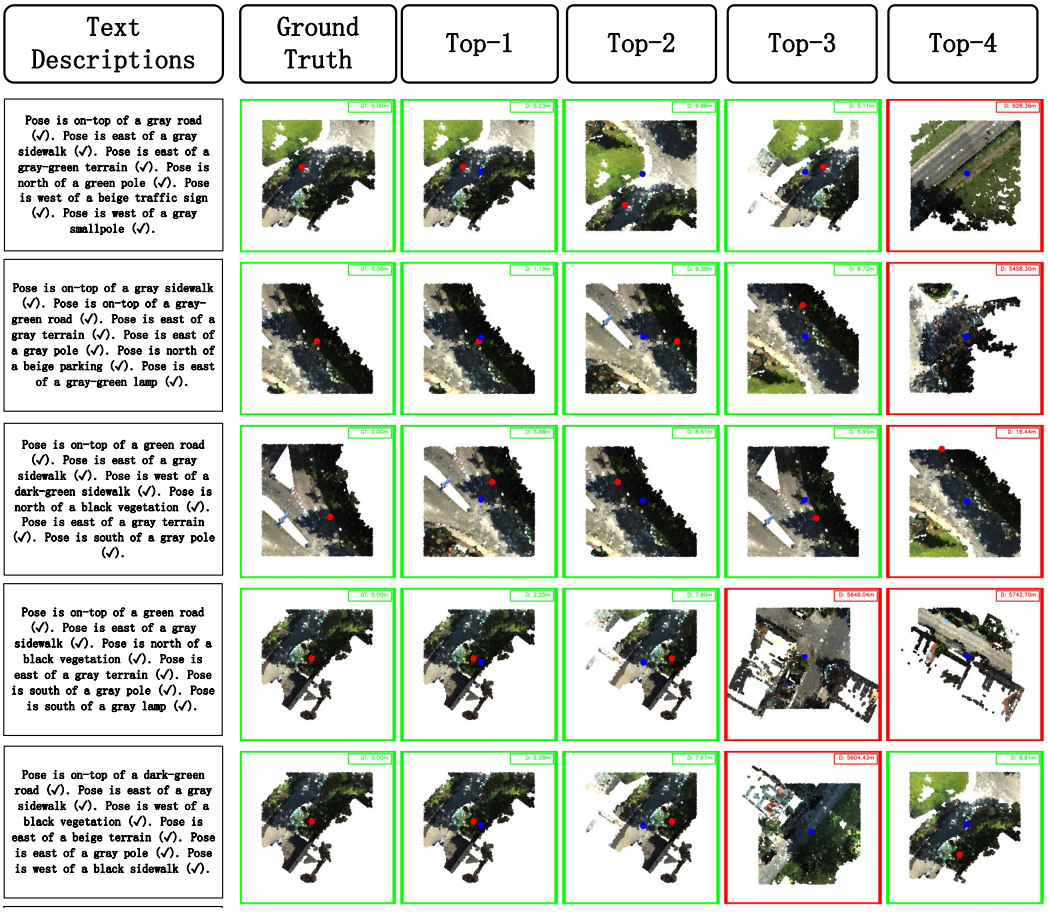


Figure 2: Qualitative localization results of OLA-Place. Red dots denote ground-truth coordinates and blue dots denote predicted points; green borders indicate successful localization with error below 10m, while red borders indicate failed localization.

cross-modal semantic gap between natural language and 3D point-cloud features more effectively than global descriptors. Analysis of failure cases reveals that localization errors mainly stem from perceptual aliasing: regions more than 1,000 meters apart but with highly similar geometric topology can still cause retrieval ambiguity. This observation indicates that the discriminative power of the coarse retrieval stage remains the key bottleneck in large-scale urban scenarios.

5 Conclusion

We presented OLA-Place, an object-centric framework for cross-modal place recognition that eliminates the need for global scene descriptors. By representing a language query and a 3D scene cell as sets of textual and object-level features, OLA-Place formulates retrieval as object-level semantic alignment rather than holistic embedding matching. The proposed design, built on OSE, OME, and MMA, preserves fine-grained correspondences, supports variable-size object sets, and provides an effective scoring mechanism without requiring explicit object-level correspondence supervision.

Experiments on KITTI360Pose and StreetLoc demonstrate that this formulation delivers consistent gains over existing methods in both coarse retrieval and end-to-end localization. These results suggest that object-level alignment offers a more effective retrieval signal than global descriptor matching for language-guided 3D place recognition. We hope this perspective will encourage future research on more flexible and semantically grounded cross-modal localization systems.

Acknowledgments and Disclosure of Funding

This work was supported by the Natural Science Foundation of Shandong Province (ZR2024QF284).

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef 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.

[2] Gabriele Berton, Gabriele Trivigno, Barbara Caputo, and Carlo Masone. Eigenplaces: Training viewpoint robust models for visual place recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11080–11090, 2023.

[3] Daniele Cattaneo, Matteo Vaghi, and Abhinav Valada. Lcdnet: Deep loop closure detection and point cloud registration for lidar slam. *IEEE Transactions on Robotics*, 38(4):2074–2093, 2022.

[4] Juan Du, Rui Wang, and Daniel Cremers. Dh3d: Deep hierarchical 3d descriptors for robust large-scale 6dof relocalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 744–762. Springer, 2020.

[5] Mingtao Feng, Longlong Mei, Zijie Wu, Jianqiao Luo, Fenghao Tian, Jie Feng, Weisheng Dong, and Yaonan Wang. Partially matching submap helps: Uncertainty modeling and propagation for text to point cloud localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8296–8305, 2025.

[6] Le Hui, Hang Yang, Mingmei Cheng, Jin Xie, and Jian Yang. Pyramid point cloud transformer for large-scale place recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6098–6107, 2021.

[7] Donghwi Jung, Keonwoo Kim, and Seong-Woo Kim. Gotpr: General outdoor text-based place recognition using scene graph retrieval with openstreetmap. *IEEE Robotics and Automation Letters*, 2025.

[8] Joshua Knights, Peyman Moghadam, Milad Ramezani, Sridha Sridharan, and Clinton Fookes. Incloud: Incremental learning for point cloud place recognition. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8559–8566. IEEE, 2022.

[9] Manuel Kolmet, Qunjie Zhou, Aljoša Ošep, and Laura 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.

[10] Jacek 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.

[11] Zhenyu Li and Tianyi Shang. a^2gc: A symmetric a ggregation with geometric constraints for locally aggregated descriptors. *arXiv preprint arXiv:2511.14109*, 2025.

[12] Zhenyu Li, Tianyi Shang, Pengjie Xu, Zhaojun Deng, and Ruirui Zhang. Toward robust visual place recognition for mobile robots with an end-to-end dark-enhanced net. *IEEE Transactions on Industrial Informatics*, 2024.

[13] Zhenyu Li, Pengjie Xu, Zhenbiao Dong, Ruirui Zhang, and Zhaojun Deng. Feature-level knowledge distillation for place recognition based on soft-hard labels teaching paradigm. *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[14] Baihong Lin, Zhengxia Zou, and Zhenwei Shi. Rsbev: Multi-view collaborative segmentation of 3d remote sensing scenes with bird's-eye-view representation. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[15] Philipp Lindenberger, Paul-Edouard Sarlin, Jan Hosang, Matteo Balice, Marc Pollefeys, Simon Lynen, and Eduard Trulls. Scaling image geo-localization to continent level. *arXiv preprint arXiv:2510.26795*, 2025.

[16] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yun-Hui 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.

[17] Lun Luo, Shuhang Zheng, Yixuan Li, Yongzhi Fan, Beinan Yu, Si-Yuan Cao, Junwei Li, and Hui-Liang Shen. Bevplace: Learning lidar-based place recognition using bird's eye view images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8700–8709, 2023.

[18] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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.

[19] Tianyi Shang, Zhenyu Li, Wenhao Pei, Pengjie Xu, ZhaoJun Deng, and Fanchen Kong. Mambaplace: Text-to-point-cloud cross-modal place recognition with attention mamba mechanisms. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

[20] Tianyi Shang, Zhenyu Li, Pengjie Xu, Jinwei Qiao, Gang Chen, Zihan Ruan, and Weijun Hu. Bridging text and vision: A multi-view text-vision registration approach for cross-modal place recognition. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

[21] Mikaela Angelina Uy and Gim Hee 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.

[22] Kavisha Vidanapathirana, Milad Ramezani, Peyman Moghadam, Sridha Sridharan, and Clinton Fookes. Logg3d-net: Locally guided global descriptor learning for 3d place recognition. In *International Conference on Robotics and Automation (ICRA)*, pages 2215–2221. IEEE, 2022.

[23] Guangzhi Wang, Hehe Fan, and Mohan Kankanhalli. Text to point cloud localization with relation-enhanced transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2501–2509, 2023.

[24] Lichao Wang, Zhihao Yuan, Jinke Ren, Shuguang Cui, and Zhen Li. Instance-free text to point cloud localization with relative position awareness. *arXiv preprint arXiv:2404.17845*, 2024.

[25] Ruotong Wang, Yanqing Shen, Weiliang Zuo, Sanping Zhou, and Nanning Zheng. Transvpr: Transformer-based place recognition with multi-level attention aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13648–13657, 2022.

[26] Yan Xia, Yusheng Xu, Shuang Li, Rui Wang, Juan Du, Daniel Cremers, and Uwe 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.

[27] Yan Xia, Letian Shi, Zifeng Ding, Joao F Henriques, and Daniel 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.

[28] Huaiyuan Xu, Huaping Liu, Shoudong Huang, and Yuxiang Sun. C2l-pr: Cross-modal camera-to-lidar place recognition via modality alignment and orientation voting. *IEEE Transactions on Intelligent Vehicles*, 2024.

[29] Tian-Xing Xu, Yuan-Chen Guo, Zhiqiang Li, Ge Yu, Yu-Kun Lai, and Song-Hai Zhang. Transloc3d: Point cloud based large-scale place recognition using adaptive receptive fields. *arXiv preprint arXiv:2105.11605*, 2021.

[30] Yanlong Xu, Haoxuan Qu, Jun Liu, Wenxiao Zhang, and Xun Yang. Cmmloc: Advancing text-to-pointcloud localization with cauchy-mixture-model based framework. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6637–6647, 2025.

[31] Shuhang Zheng, Yixuan Li, Zhu Yu, Beinan Yu, Si-Yuan Cao, Minhang Wang, Jintao Xu, Rui Ai, Weihao Gu, Lun Luo, et al. I2p-rec: Recognizing images on large-scale point cloud maps through bird's eye view projections. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1395–1400. IEEE, 2023.

[32] Zhicheng Zhou, Cheng Zhao, Daniel Adolfsson, Songzhi Su, Yang Gao, Tom Duckett, and Li Sun. Ndt-transformer: Large-scale 3d point cloud localisation using the normal distribution transform representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5654–5660. IEEE, 2021.

A Technical appendices and supplementary material

A.1 Limitations and Future Work

Despite the effectiveness of OLA-Place on established benchmarks, several limitations remain. At present, point-cloud instance extraction and the subsequent cross-modal matching process rely on a fixed set of semantic labels. Because real-world environments contain an open-ended variety of entities and human language is inherently flexible, this dependence on a closed-set vocabulary can be restrictive. In particular, the system may struggle when it encounters unfamiliar objects or linguistic expressions that deviate from the training distribution.

To address these limitations, future work will explore the integration of large language models (LLMs) with open-vocabulary 3D perception. By leveraging the contextual reasoning ability and broad semantic priors of LLMs, we aim to reduce the reliance on rigid category definitions. This direction could enable the grounding of unconstrained natural-language queries in novel 3D geometry, thereby improving the adaptability and robustness of the framework in diverse large-scale environments.

A.2 Broader Impacts

OLA-Place advances text-driven spatial localization in urban environments and may benefit applications such as autonomous driving, robotic assistance, and aerial reconnaissance. A primary advantage of this research is its potential to make navigation systems more accessible: by enabling systems to interpret natural language, it reduces the reliance on specialized sensor queries or prior visual familiarity. This shift toward more intuitive human–machine interaction may be particularly valuable for search-and-rescue operations and assistive technologies, where efficient communication can substantially improve operational effectiveness.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction are consistent with the proposed method and experimental evidence.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   [Yes]

   Justification: The paper includes a "Limitations and Future Works" section in Appendix A.1.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include formal theoretical results such as theorems, lemmas, or propositions that require proofs. The equations are used to define the proposed model components and training objective.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides the main model design, datasets, splits, training epochs, hardware, evaluation metrics in Section 4.1. We also provide open-source code.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The abstract provides an anonymized repository link. The repository includes the source code, data access and preparation instructions, environment configuration, and training/evaluation scripts for reproducing the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper reports the datasets, train/validation/test splits, evaluation metrics, training stages, number of epochs, hardware configuration, and baseline settings in Section 4.1. Additional implementation details are provided in the anonymized repository.

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The current paper reports point estimates but does not include error bars, confidence intervals, or statistical significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the workstation configuration, including a 128-core AMD EPYC CPU and an NVIDIA RTX V100 32GB GPU in Section 4.1. The Section 4.4 Computational Cost further reports model parameter counts and inference runtime for the coarse retrieval stage.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work does not involve human-subject experiments or high-risk release, and the research is intended for 3D localization and embodied AI. The authors should ensure the data collection and release comply with relevant laws and institutional requirements.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: A.2 in Appendix discusses potential broader impacts.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release high-risk generative models, pretrained language models, or scraped internet datasets. The released assets are intended for text-guided 3D localization research.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper uses existing benchmark datasets and methods with proper citations, including KITTI360Pose and prior text-to-point-cloud localization baselines such as Text2Pos, RET, Text2Loc, MambaPlace, CMMLoc, and PMSH. The datasets and baselines are credited through citations.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We have uploaded the details of all assets to our open-source repository. The repository provides documentation for data preparation, code usage, training, and evaluation to support reproducibility.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This paper is purely computational/theoretical and does not involve any human subjects, or crowdsourcing experiments.

Guidelines:

- The answer [N/A] means that the paper does not involve
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not use LLMs as an important, original, or non-standard component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.