HeLiOS: Heterogeneous LiDAR Place Recognition via Overlap-based Learning and Local Spherical Transformer

Minwoo Jung[1], Sangwoo Jung[1], Hyeonjae Gil[1] and Ayoung Kim[1*]

***Abstract*— LiDAR place recognition is a crucial module in localization that matches the current location with previously observed environments. Most existing approaches in LiDAR place recognition dominantly focus on the spinning type LiDAR to exploit its large FOV for matching. However, with the recent emergence of various LiDAR types, the importance of matching data across different LiDAR types has grown significantly—a challenge that has been largely overlooked for many years. To address these challenges, we introduce HeLiOS, a deep network tailored for heterogeneous LiDAR place recognition, which utilizes small local windows with spherical transformers and optimal transport-based cluster assignment for robust global descriptors. Our overlap-based data mining and guided-triplet loss overcome the limitations of traditional distance-based mining and discrete class constraints. HeLiOS is validated on public datasets, demonstrating performance in heterogeneous LiDAR place recognition while including an evaluation for long-term recognition, showcasing its ability to handle unseen LiDAR types. We release the HeLiOS code as an open source for the robotics community at https://github.com/minwoo0611/HeLiOS.**

I. Introduction

LiDAR Place Recognition (LPR) identifies whether a location was previously visited by comparing current scans to past ones from a pair of Light Detection and Ranging (LiDAR) scans. Among many LiDAR types, high-resolution spinning LiDARs have been the most popular choice [1–4] to handle occlusions and deliver extensive data from their 360-degree coverage and comprehensive information. However, the reliance on high-resolution systems has constrained the generalized solutions across various LiDAR types.

In this paper, we shift the focus from this popular 360° scanning LiDAR to a diverse range of different LiDAR types, highlighting challenges in heterogeneous LPR. For example, limited data from narrow field of view (FOV) solid-state LiDARs [5–7] reintroduce complexity to the problem. Resolution difference among sensors raised an issue as sparsity caused the same structure to appear differently, making it challenging to use 2D convolution [8, 9] or methods that are adapted from Visual Place Recognition (VPR) [10]. The new scanning patterns introduced by Livox and Robosense [11] capture the surroundings in a completely different manner. As a result, the data varies significantly from sensor to sensor, even when viewed from the same location. These disparities pose significant challenges for heterogeneous LPR [12].

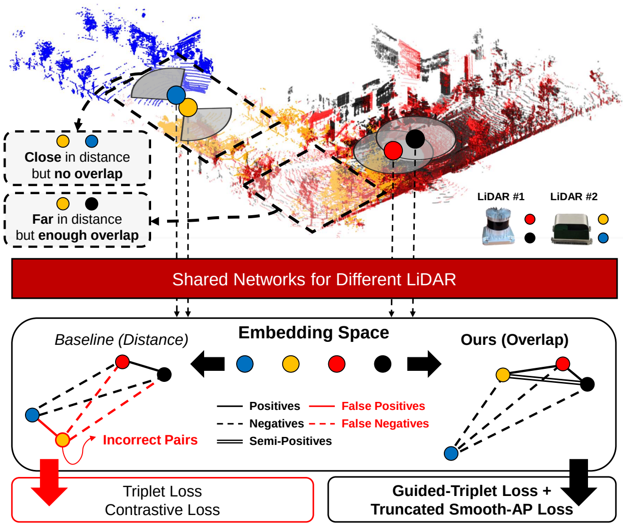


Fig. 1: HeLiOS utilizes overlap for mining and the loss function. Conventional distance-based mining might lead to incorrect pairings, such as blue-yellow circles (close distance, no overlap) or black-yellow circles (far apart, overlap). Incorrect pairings can negatively impact both the training process and overall performance.

Recently, transformer-based methods [8, 9] boosted performance to tackle similar challenges such as Ground-Aerial LiDAR [13] or Camera-LiDAR place recognition [14, 15]. However, applying transformers to heterogenous LiDARs often fails to encode data into a common embedding space due to different distributions, while using individual transformers for each source remains a limitation of generalizability. Handcrafted algorithms [5–7] offer viable solutions, but they require multiple scans and specific poses. Furthermore, their applicability for heterogeneous LiDARs is still limited as the validation is only done within homogeneous LiDARs.

In this paper, we present a novel network for single scan heterogeneous LPR, that addresses variations in FOV, resolution, and scanning patterns. We extend the spherical transformer [16] along the radial direction to create smaller patches to better capture the point cloud's local distributions. Global descriptors are generated using optimal transport-based cluster assignment [17], which filters out uninformative features while preserving the original distribution of local features. Moreover, our descriptor offers a flexible dimension range, allowing customization based on the user's objectives and requirements. An overlap-based data mining and Truncated Smooth-AP (TSAP) loss with guided-triplet loss are introduced to optimize the training process. The overlap-based approach addresses the inaccuracies associated with distance-based methods, providing more consistent con-

[†] This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT)(No. RS-2024-00461409), and in part by Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea Government (MSIT)(No.2022-0-00480).

[1] M. Jung, S. Jung, H. Gil and A. Kim are with the Dept. of Mechanical Engineering, SNU, Seoul, S. Korea [moonshot, dan0130, h.gil, ayoungk]@snu.ac.kr

straints through a semi-positive class, as illustrated in Fig. 1.

Our main contributions are as follows:

- We introduce HeLiOS, a deep network to overcome the major challenges of heterogeneous LiDARs: diverse scanning patterns, FOVs, and resolutions. Using the sparse convolution and spherical transformer with a local window, HeLiOS captures both low-level and high-level information. To our knowledge, this is the first method tailored for heterogeneous LiDAR systems.
- We propose overlap-based data mining and guided-triplet loss to capture the relationships between LiDAR descriptors, overcoming the limitations of discrete classes in traditional triplet loss and reducing wrong classes in distance-based mining. Our semi-positives ensure comprehensive constraints across various labels.
- HeLiOS is validated on public datasets, exhibiting superior performance in inter-LiDAR and inter-session place recognition compared to state-of-the-art (SOTA) methods. We open-source HeLiOS for LPR's community.

II. Related Work

A. Deep Learning in LiDAR Place Recognition

A seminal work in the learning-based LPR, PointNetVLAD [18] used PointNet [19] for feature extraction. Traditional methods [20, 21] relied on Multi-Layer Perception (MLP), enhancing global descriptors by improving local contextual relationships and minimizing information loss. To reduce computational costs and better encode point cloud, OverlapTransformer [9] and CVTNet [8] chose 2D convolutions on images of projected point cloud, but it struggles on various image formats of heterogeneous LiDARs. Other methods [22–24] applied sparse convolution for efficient 3D computation, but still requiring performance improvements.

Recent studies employed transformer [8, 24] to enhance LPR performance. While these approaches show improvement for homogeneous LiDARs, they struggle with heterogeneous LPR due to varying data distributions, complicating the effective learning of attention mechanisms. For example, SALSA [25] utilized SphereFormer [16] for feature extraction with multi-head attention applied within spherical windows. However, the windows are too large to accurately capture the distinct distributions of different LiDARs, limiting their effectiveness in heterogeneous LPRs.

Our model leverages sparse convolution and transformer but focuses on adaptation for heterogeneous LiDARs. We divide the spherical window into smaller regions based on spherical coordinates (r, θ, ϕ), allowing the transformer to learn local distributions. Additionally, our model uses a shared encoder to generate descriptors that are effective across different LiDARs, distinguishing it from methods focused on multi-modal place recognition [14, 15, 26].

B. Data Mining Strategies and Losses for Place Recognition

Traditional LPR relied on distance-based sampling to generate positive and negative samples for data mining. However, this approach faced challenges with narrow FOV LiDARs, as scans from the same location may not overlap. Leyva-Vallina et al. [27] used overlap for data mining, but their method, tailored for images and dense maps, is unsuitable for sparse LiDARs. Similarly, OverlapNet [28] requires the height and width of the range image, making it hard to decide the common format for different LiDARs. In contrast, our approach calculates overlap directly in 3D space considering scan density. We classify data into positive, semi-positive, and negative to position descriptors within embedding space to be more suitable for heterogeneous LPR.

Moreover, LPR traditionally used triplet loss [29] and contrastive loss [30], which are designed for discrete class tasks like image classification [31, 32], limiting their effectiveness in LPR. Leyva-Vallina et al. [27] improved this by multiplying overlap with contrastive loss, but sample with small overlap can still be embedded near the negatives. OverlapNet [28] utilized overlap in loss without affecting descriptor distribution. LoGG3D-Net [23] introduced local consistency loss for better feature similarity but struggles with varying scanning patterns and density. MinkLoc3Dv2 [22] used TSAP loss to improve average precision for top k positives but lacks explicit distance constraints. We combine TSAP loss with guided-triplet loss, adding overlap-based margin constraints to regulate descriptor distances better.

III. Methodology

A. Problem Definition

LPR aims to generate a global descriptor from a point cloud $\mathbf{P} \in \mathbb{R}^{N\times 3}$, where the N points are defined by its spatial coordinates (x, y, z). To encode the point cloud into a descriptor, a mapping function $\Omega = h(f(\cdot))$ is employed, where the feature extraction function $f(\cdot) : \mathbb{R}^{N\times 3} \rightarrow \mathbb{R}^{n\times d}$ derives local embeddings from N points, and the aggregation function $h(\cdot) : \mathbb{R}^{n\times d} \rightarrow \mathbb{R}^{e}$ compresses these embeddings into a global descriptor $g \in \mathbb{R}^{e}$ of fixed dimensions. The goal is to optimize Ω to meet the following conditions:

$$\mathcal{D}(\mathbf{x}_q, \mathbf{x}_i)\ \mathcal{D}(\mathbf{x}_q, \mathbf{x}_j) \implies d_g(g_q, g_i) < d_g(g_q, g_j), \qquad (1)$$

while $\mathbf{x}$ denotes the locations of the point cloud, $\mathcal{D}(\cdot)$ represents the distance between the locations, and $d_g(\cdot)$ signifies the distance within the embedding space. Optimization of Ω is achieved through metric learning, applying a loss function to the global descriptor derived from the training set.

B. Locality-aware Feature Extraction Network

To encode local features from point clouds, we utilize a U-Net style architecture with sparse convolution [33]. Point clouds are voxelized along the (x, y, z) axes, ensuring uniform resolution in 3D space. While sparse convolution effectively captures local information within each voxel, the variations in coverage and scanning patterns across heterogeneous LiDARs present significant challenges. The differing distribution of heterogeneous LiDARs disturbs their application by potentially causing divergence during training, even with transformers or self-attention that provide a viable solution for embedding point clouds into a common space.

To focus on local distribution rather than global distribution, we divide the space with a spherical window and

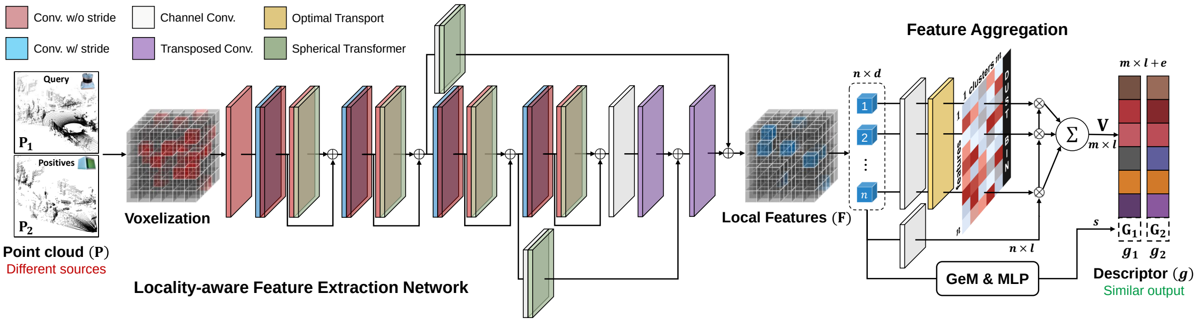


Fig. 2: The overall pipeline of HeLiOS. HeLiOS voxelizes point clouds $\mathbf{P}$ from heterogeneous LiDARs, which are then processed through a shared feature extraction network with sparse convolution and a spherical transformer. Local features $\mathbf{F}$ are aggregated into global descriptors using GeM and SALAD to produce similar descriptors, which are subsequently utilized for training and evaluation.

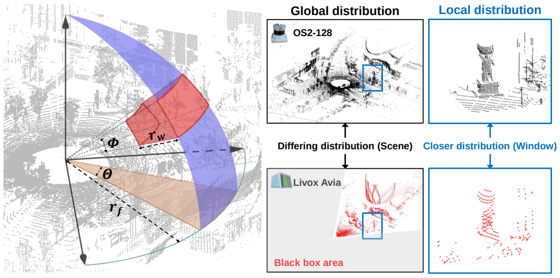


Fig. 3: Local spherical window for applying multi-head attention in heterogeneous LPR. Due to the differing global distribution when scanning the entire scene with different LiDARs, training attention is challenging. Distribution is closer within smaller, localized windows, enabling effective attention learning.

leverage multi-head attention as Fig. 3. Inspired by vision transformer [34] that segment images into patches, we apply multi-head attention to 3D voxels within spherical windows defined by spherical coordinates (r, θ, ϕ). As the volume of spherical windows increases with r for the same θ and ϕ, multi-head attention using a consistent cubic window is also applied to complement local information. Consequently, each half of the multi-head attention output is derived from cubic and spherical windows. This approach differs from SphereFormer [16], which uses full-radius spherical windows.

We apply the spherical transformer only where skip connections exist, allowing attention to be processed while preserving the output from the sparse convolution layer. Additionally, as the voxels are progressively downsampled through the model, some windows may not contain enough voxels. To address this, when each time sparse convolution with stride is applied, the spherical window scale is expanded by 1.5, while the cubic window scale remains constant. Our feature extraction network pipeline is illustrated in Fig. 2.

C. Feature Aggregation with optimal transport

Different LiDARs generate varying numbers of local features $\mathbf{F} \in \mathbb{R}^{n \times d}$, complicating feature aggregation. To address this, we adapt a clustering-based approach that is less sensitive to these variations. Specifically, we adapt SALAD [17] from vision tasks, where image patches are used as input. We consider voxels with local features as patches, enabling us to apply a method designed for a different task.

$\mathbf{F}$ is processed through a channel-wise convolution layer to predict the score matrix, $\mathbf{S} \in \mathbb{R}^{n \times m}$, where m is the cluster number. To manage non-informative points, a dustbin column is added, modifying the score matrix to $\bar{\mathbf{S}} \in \mathbb{R}^{n \times (m+1)}$. The Sinkhorn algorithm [35] is then applied to optimize the feature-to-cluster assignment, creating a refined score matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$ obtained by iteratively normalizing the rows and columns of $\exp(\bar{\mathbf{S}})$ and dropping the dustbin. To align features with the cluster space, $\mathbf{F}$ is projected to a lower-dimensional $\bar{\mathbf{F}} \in \mathbb{R}^{n \times l}$. Finally, the aggregated feature matrix $\mathbf{V} \in \mathbb{R}^{m \times l}$ is computed, where $V_{j,k}$ is:

$$V_{j,k} = \sum_{i=1}^{n} R_{i,k} \cdot \bar{F}_{i,j} \tag{2}$$

To address the loss of global context that may occur during the clustering process, we also employ GeM pooling [36] combined with MLP layers. This produces a compact global representation $\mathbf{G} \in \mathbb{R}^{e}$. The final descriptor is created by concatenating the flattened global features $\mathbf{G}$ with the aggregated features $\mathbf{V}$, ensuring a comprehensive representation with minimal dimensional increase. The descriptor $g \in \mathbb{R}^{m \times l + e}$ is then utilized for both training and evaluation.

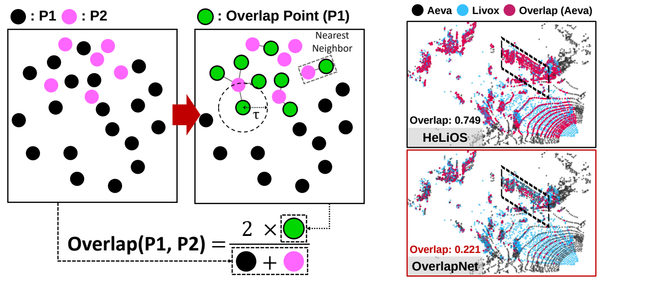
Unlike conventional place recognition that utilizes 256 or 512 descriptor dimension, HeLiOS exploits the various dimension based on m, l and e. This approach offers flexibility in adjusting the dimension, allowing users to balance computational requirements and performance by customizing the trade-off between time complexity and accuracy.

D. Overlap Guided Metric Learning

1) Overlap-Based Data Mining: Traditional metric learning for place recognition often relies on distance-based sampling, which can fail with heterogeneous LiDARs as Fig. 1. The naive distance-based sampling may result in unrelated positives or negatives, requiring a more refined approach considering LiDAR-specific characteristics. To address this, we employ an overlap-based data mining method, where defining the overlap between point clouds, P_1 and P_2, as:

$$\hat{\mathrm{O}}(P_1, P_2) = \frac{2 \times \sum_{i=1}^{N_1} \mathbb{1}\left(\mathrm{NN}(P_1^i, P_2) < \tau\right)}{N_1 + N_2}, \tag{3}$$

where NN returns the distance to the nearest neighbor in the other point cloud, and $\mathbb{1}(\cdot)$ is an indicator function. The overall process can be found in Fig. 4(a). To reduce the computationally expensive $n \times n$ overlap matrix calculation, where n is the number of samples, we truncate the overlap calculation if the distance exceeds twice the maximum scan range. Furthermore, point clouds are voxelized with size δ to reduce computational costs and standardize resolution,



(a) Overlap Calculation (b) Real-World Example

Fig. 4: (a) Overlap calculation to illustrate (3) as a diagram. (b) HeLiOS computes overlap for different LiDARs in 3D space. In contrast, OverlapNet misrepresents real-world overlap even if the LiDARs are in the same location, as their overlap occurs only when each point falls into the same pixel in a range image.

preventing the large overlap in dense regions. We set $\tau = 1.5\delta$ to ensure robustness to minor misalignment.

The use of the maximum overlap value ensures stable and consistent learning, keeping the overlap constant between the point clouds in a reversed relation. Final overlap is computed as $\mathrm{O}(P_1, P_2) = \max(\hat{\mathrm{O}}(P_1, P_2), \hat{\mathrm{O}}(P_2, P_1))$ with a maximum threshold of 1. Our method addresses challenges in determining overlap with range images of different LiDARs, such as varying resolutions [28] and the sparsity of LiDAR in 3D space [27], providing a reliable overlap measurement as Fig. 4(b). Scans are categorized as positive if overlap is over 0.5, semi-positive between 0 and 0.5, and negative if zero, enhancing robustness and generalization in metric learning.

2) Guided-Triplet Loss: In LPR, the balance of ranking relevant scans and controlling their distances in the embedding space is crucial for robust model. The TSAP loss $\mathcal{L}_{\mathcal{TSAP}}$ [22] ranks the top k positives but lacks explicit control over distances between positive (p) and negative (n) from the query (q). This leads to a dispersed distribution of descriptors with poorly defined boundaries, reducing the discriminative power and causing instability during training.

To address these limitations, we introduce a combined loss function that incorporates both $\mathcal{L}_{\mathcal{TSAP}}$ and a guided-triplet loss $\mathcal{L}_{\mathcal{GT}}$. Unlike triplet loss $\mathcal{L}_{\mathcal{T}}$ with a fixed margin to separate two classes, guided-triplet loss employs adaptive margins based on the overlap to allow more general regulation of distances. This reflects varying degrees of similarity between scans and ensures that the embeddings are more distributed and distances are effectively controlled. Additionally, we incorporate two $\mathcal{L}_{\mathcal{GT}}$ based on the relationship with semi-positives (s) and the others. Semi-positive should be closer to positives but still separated from them and positioned far from negatives. Guided-triplet loss $\mathcal{L}_{\mathcal{GT}}$ is formulated as:

$$\mathcal{L}_{\mathcal{GT}}(q, u, v) = \max(d_g(q, u) - d_g(q, v) + \alpha_{uv}, 0), \qquad (4)$$

where $d_g(\,\cdot\,)$ are distances in the embedding space, and α_{uv} is the adaptive margin based on overlap. The adaptive margin α_{ps} for positive and semi-positive pair, and α_{sn} for semi-positive and negative pair are defined as:

$$\alpha_{uv} = \begin{cases} m_1 \cdot (\mathrm{OV}(q,p) - \mathrm{OV}(q,s)) & \text{if } (u = p, v = s) \\ m_2 \cdot (\mathrm{OV}(q,s) - \mathrm{OV}(q,n) + 1) & \text{if } (u = s, v = n) \end{cases}$$

$$\mathrm{OV}(q,u) \triangleq \log(\beta \cdot \mathrm{O}(q,u) + 1), \qquad (5)$$

where m_1, m_2, and β are the scale factors, and logarithm regularizes the effect of overlap, ensuring large overlaps yield similar values while amplifying differences of small overlaps. In (5), we add $\mathrm{OV}(q, n)$ for the readability, even if it is always zero. To divide the semi-positive and negative more distinctly, an additional distance of m_2 is provided for α_{sn}.

By combining $\mathcal{L}_{\mathcal{TSAP}}$ with $\mathcal{L}_{\mathcal{GT}}$, our total loss function not only focuses on ranking the most relevant scans but also maintains appropriate distances between positives, semi-positives, and negatives. The total loss function is given as:

$$\mathcal{L} = \mathcal{L}_{\mathcal{TSAP}} + \omega_1 \cdot \mathcal{L}_{\mathcal{GT}}(q,p,s) + \omega_2 \cdot \mathcal{L}_{\mathcal{GT}}(q,s,n), \qquad (6)$$

which enhances model generalization by ensuring well organization of global descriptors in the embedding space.

IV. Experiment

A. Implementation Details

We trained HeLiOS on a GeForce RTX 3090 for 80 epochs using a MultiStepLR scheduler with an initial learning rate of 0.001. The maximum range is limited to $100\mathrm{m}$, and the 8192 points from a single scan are normalized within $[-1, 1]$. The spherical windows are set to $(10\mathrm{m}, 1.8°, 1.8°)$, and the voxel size d for overlap calculation is 4m. For the guided-triplet loss, the weights ω_1 and ω_2 are set to 0.1, and the scaling factors (m_1, m_2, β) are configured as $(0.02, 0.19, e - 1)$.

B. Datasets and Evaluation Metric

We evaluated HeLiOS on three public datasets: NCLT (HDL-32E) [37], MulRan (OS1-64) [38], and HeLiPR (OS2-128, Livox Avia, Aeva Aeries II, and VLP-16C) [12]. We compared HeLiOS against several methods, including PointNetVLAD [18], LoGG3D-Net [23], CASSPR [24], CROSSLOC3D [13], MinkLoc3Dv2 [22], and handcrafted descriptor SOLID [7], applying our overlap criteria across all benchmarks for a fair comparison.

We chose Average Recall@k (AR@k) for evaluation. A retrieval is correct if the overlap between the query and retrieval exceeds 0.5. HeLiOS with parameters $(m, l, e) = (64, 128, 256)$ is evaluated to demonstrate its complex descriptor capability. For fairness, a smaller architecture, HeLiOS-S, is also configured with lightweight parameters $(m, l, e) = (8, 32, 0)$ for a descriptor dimension of 256.

C. Heterogeneous LiDAR Place Recognition

The HeLiPR dataset is used for heterogeneous LPR. Training is conducted on `DCC04-06`, `KAIST04-06`, and `Riverside04-06` with four LiDAR types, sampled at 5m intervals, totaling $41k$ samples. Testing is conducted on `Roundabout01-03`, `Town01-03`, `Bridge01` (paired with `04`), and `Bridge02` (paired with `03`) to ensure sufficient overlap, with $96k$ samples also sampled at 5m.

Queries are classified by grouping Aeva and Livox as "Narrow" and Ouster and Velodyne as "Wide". The first sequences of Aeva and Ouster are chosen as databases, while all sequences and LiDARs at each location are queried against them. For example, to evaluate Ouster and "Wide" in `Roundabout`, the database is `Roundabout01-Ouster`,

TABLE I: Performance Comparison with Heterogeneous LiDARs (**Red**: Best, Blue: Second Best)

DB	Methods	Roundabout				Town				Bridge01				Bridge02			
		Narrow		Wide		Narrow		Wide		Narrow		Wide		Narrow		Wide	
		AR@1	AR@5	AR@1	AR@5	AR@1	AR@5	AR@1	AR@5	AR@1	AR@5	AR@1	AR@5	AR@1	AR@5	AR@1	AR@5
Ouster	SOLID [7]	0.054	0.146	0.278	0.386	0.124	0.300	0.292	0.404	0.027	0.073	0.321	0.394	0.038	0.098	0.293	0.371
	PointNetVLAD [18]	0.297	0.518	0.586	0.744	0.291	0.530	0.474	0.664	0.329	0.495	0.704	0.823	0.362	0.558	0.646	0.754
	LoGG3D-Net [23]	0.012	0.038	0.151	0.295	0.011	0.049	0.098	0.241	0.027	0.032	0.219	0.320	0.012	0.047	0.186	0.373
	CASSPR [24]	0.182	0.407	0.478	0.703	0.178	0.418	0.376	0.598	0.220	0.423	0.548	0.702	0.234	0.482	0.488	0.634
	CROSSLOC3D [13]	0.499	0.785	0.796	0.876	0.565	0.785	0.739	0.876	0.547	0.745	0.810	0.906	0.555	0.756	0.781	0.891
	MinkLoc3Dv2 [22]	0.620	0.790	0.870	0.928	0.660	0.838	0.817	0.919	0.650	0.818	0.876	0.938	0.631	0.819	0.833	0.910
	HeLiOS-S	0.637	0.801	0.880	0.929	0.686	0.862	0.828	0.918	0.660	0.828	0.880	0.950	0.649	0.827	0.831	0.921
	HeLiOS	**0.700**	**0.852**	**0.912**	**0.946**	**0.753**	**0.903**	**0.871**	**0.937**	**0.693**	**0.853**	**0.912**	**0.969**	**0.681**	**0.849**	**0.874**	**0.950**
Aeva	SOLID [7]	0.241	0.442	0.018	0.129	0.200	0.346	0.048	0.195	0.236	0.332	0.013	0.048	0.234	0.340	0.017	0.058
	PointNetVLAD [18]	0.477	0.624	0.338	0.591	0.408	0.588	0.313	0.573	0.678	0.821	0.366	0.598	0.625	0.777	0.365	0.609
	LoGG3D-Net [23]	0.022	0.094	0.029	0.096	0.033	0.116	0.026	0.101	0.015	0.062	0.011	0.052	0.026	0.098	0.025	0.249
	CASSPR [24]	0.300	0.524	0.160	0.427	0.264	0.479	0.154	0.416	0.505	0.733	0.181	0.411	0.475	0.718	0.205	0.469
	CROSSLOC3D [13]	0.711	0.836	0.634	0.846	0.635	0.800	0.581	0.833	0.790	0.907	0.665	0.865	0.738	0.876	0.665	0.848
	MinkLoc3Dv2 [22]	0.750	0.846	0.722	0.882	0.664	0.806	0.620	0.831	0.853	0.931	0.742	0.896	0.801	0.895	0.737	0.885
	HeLiOS-S	0.767	0.867	0.765	0.912	0.682	0.815	0.646	0.824	0.850	0.934	0.785	0.919	0.808	0.905	0.740	0.892
	HeLiOS	**0.806**	**0.885**	**0.849**	**0.940**	**0.744**	**0.850**	**0.737**	**0.883**	**0.886**	**0.950**	**0.818**	**0.930**	**0.857**	**0.936**	**0.773**	**0.903**

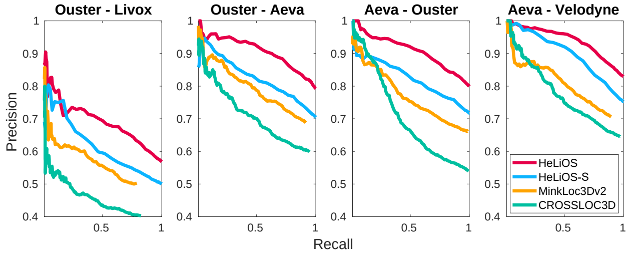


Fig. 5: PR curves with heterogeneous LiDARs. The title of each curve represents the database from `Roundabout01` and the query from `Roundabout02`. HeLiOS surpasses other methods regardless of the size of the descriptor.

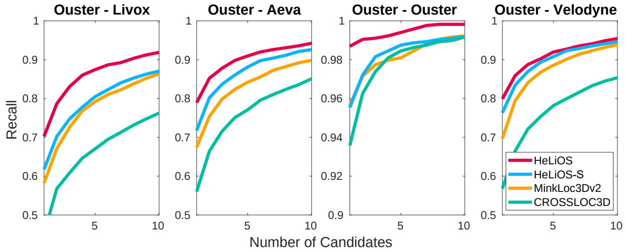


Fig. 6: R@N curves on `Town03` with heterogeneous LiDARs are shown, where each curve title indicates the database from `Town01` and the query from `Town03`. HeLiOS consistently outperforms other SOTA methods when retrieving the top 10 neighbors.

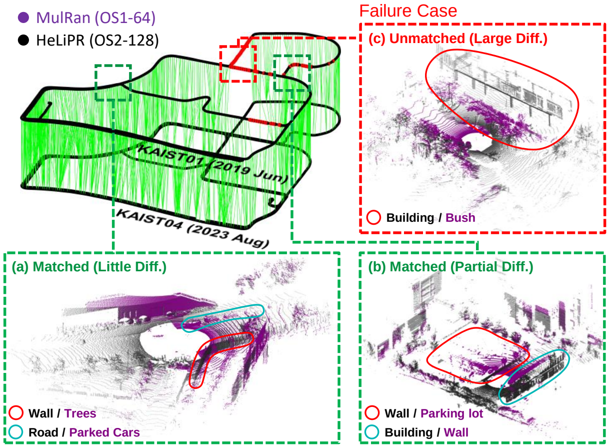


Fig. 7: Four-year long-term heterogeneous LPR between MulRan (purple) and HeLiPR (black). Red and cyan indicate significant differences between point clouds. (a, b) HeLiOS matches most queries despite long-term changes, demonstrating robustness to scene variations. (c) A building (black) replacing bushes (purple) leads to failure due to the drastic change in the scene's appearance.

while results are averaged over the recalls of six cases (3 sequences × 2 LiDARs) to cover both intra-session and inter-session. Scans within 30 seconds of the query are excluded to prevent obvious matching occurs in intra-session.

As seen in Table. I, even with lightweight 256 dimensions, HeLiOS-S surpasses others in most cases. Thanks to the spherical transformer and overlap-based metric learning, HeLiOS significantly outperforms all methods across all places. Interestingly, MinkLoc3Dv2 achieves the third-best results despite using only convolutional layers. In contrast, CASSPR and CROSSLOC3D demonstrate weaker performance as transformers applied to entire point clouds struggle with the varying distributions of heterogeneous LiDARs. LoGG3D-Net delivers inferior results, as its local consistency loss fails to accommodate differing LiDAR distributions. SOLID exhibits limitations in its descriptors across different FOVs.

We present the PR and AR@N curves for the top four methods in Fig. 5 and Fig. 6. The PR curves display results for Ouster with "Narrow" and Aeva with "Wide". HeLiOS outperforms others, achieving high recall and precision. The top 10 neighbors and their recall are plotted in Fig. 6, where HeLiOS excels in retrieving top candidates, with HeLiOS-S yielding better results even for Ouster. HeLiOS on homogeneous LiDAR is further discussed in §IV-E.

D. Long-term Place Recognition with Heterogeneous LiDAR

Long-term place recognition between HeLiPR and MulRan is evaluated. We set `KAIST01` from MulRan as the query and `KAIST04-Ouster` from HeLiPR as the database. Compared to `KAIST04`, a sequence from training, `KAIST01` is an unseen dataset captured with a different system and LiDAR (OS1-64) with different FOV and occlusion. Despite a four-year gap and scene changes, HeLiOS successfully matches almost every query, as shown in Fig. 7. Although some areas fail to retrieve, this occurs from entire scene differences between the query and database, resulting in distinct descriptors that understandably do not match. This demonstrates HeLiOS's capability for long-term place recognition and retrieving scans from unseen datasets and LiDARs in partial differences.

E. Homogeneous LiDAR Place Recognition

Differing from §IV-C, the networks are trained and evaluated with homogeneous LiDAR data. For the NCLT dataset, networks are trained on 8 sequences totaling $9.0k$ samples and tested with data from `2012-11-16` as the database and `2012-12-01` as the query with $1.8k$ samples. For the HeLiPR dataset, training is done separately on Livox and

TABLE II: Performance Comparison with Homogeneous LiDAR

Method	NCLT		R01-03 (Aeva)		R01-03 (Livox)	
	AR@1	AR@5	AR@1	AR@5	AR@1	AR@5
SOLID	0.217	0.409	0.423	0.537	0.331	0.479
PointNetVLAD	0.893	0.966	0.661	0.700	0.768	0.830
LoGG3D-Net	0.403	0.670	0.368	0.368	0.536	0.730
CASSPR	**0.961**	0.993	0.696	0.775	0.809	0.877
CROSSLOC3D	0.953	**0.996**	0.759	0.837	0.835	0.908
MinkLoc3Dv2	0.941	0.996	0.755	0.820	**0.853**	**0.922**
HeLiOS-S	0.953	0.992	**0.777**	**0.842**	0.831	0.900

TABLE III: Ablation Study with Loss for Each Class

Loss for class			Aeva (DB)		Ouster (DB)	
(p,n)	(s,n)	(p,s)	AR@1	AR@5	AR@1	AR@5
$\mathcal{L}_{\mathcal{TSAP}}$	$\mathcal{L}_{\mathcal{TSAP}}$	$\mathcal{L}_{\mathcal{TSAP}}$	0.701	0.847	0.722	0.854
	-	-	0.722	0.857	0.756	0.877
	$\mathcal{L}_{\mathcal{T}}$	-	0.771	0.885	0.793	0.903
	-	$\mathcal{L}_{\mathcal{T}}$	0.778	0.887	0.795	0.905
	$\mathcal{L}_{\mathcal{T}}$	$\mathcal{L}_{\mathcal{T}}$	0.778	0.885	0.800	0.903
	$\mathcal{L}_{\mathcal{GT}}$	$\mathcal{L}_{\mathcal{GT}}$	**0.784**	**0.890**	**0.809**	**0.910**

Aeva data, each totaling $10.3k$ samples. Place recognition is performed with Roundabout01 as the database (R01) and Roundabout03 as the query (R03), resulting in $3.4k$ test samples for each LiDAR. Retrieval is considered correct if the overlap exceeds 0.8, as homogeneous LiDARs typically yield higher overlap than heterogeneous ones. As Table. II, HeLiOS-S achieves comparable performance with others despite not being designed for homogeneous LiDAR.

F. Ablation Studies

We conducted ablation studies to highlight the performance differences and verify the impact of our proposed modules on HeLiOS. Training setup follows §IV-C, while results are average recall of both "Narrow" and "Wide" cases for Roundabout and Town.

Effect of Loss Function: We assessed the effect of $\mathcal{L}_{\mathcal{GT}}$ by keeping $\mathcal{L}_{\mathcal{TSAP}}$ fixed for positive and negative pairs while varying the loss for (s,n) and (p,s) pairs, with ω_1 and ω_2 set to 0.1. As Table. III, applying $\mathcal{L}_{\mathcal{TSAP}}$ to both class pairs only pushes the embeddings further apart, similar to pushing query-negative pairs away from query-positive pairs while leading to inaccuracy than applying only to (p,n) pairs. Conversely, using $\mathcal{L}_{\mathcal{T}}$ for (s,n) and (p,s) allows for additional performance improvements by providing distance control not achievable with $\mathcal{L}_{\mathcal{TSAP}}$ alone. Proposed $\mathcal{L}_{\mathcal{GT}}$ with overlap-based adaptive margins outperforms $\mathcal{L}_{\mathcal{T}}$ with fixed margins. This shows that the adaptive margin of $\mathcal{L}_{\mathcal{GT}}$ enhances the traditional discrete class separation, allowing the embedding process to reflect the real-world better.

Variation in Spherical Transformer: We evaluated the effect of spherical transformer under different configurations: without the transformer, varying radial window size, and applying expansion. Table. IV shows result with r_f and r_w of 100m and 10m, and an expansion factor of 1.5. Larger windows (r_f) result in slight performance improvements over the no transformer case, suggesting larger windows dilute attention on local patterns due to varying distributions of heterogeneous LiDARs. Conversely, smaller windows (r_w) lead to significant performance gains as the network better captures local features while reducing the impact of differing distribution. Gradual expansion ensures local windows contain enough points for attention even in deeper layers, enhancing spatial information encoding of the network.

TABLE IV: Ablation Study for Sphere Transformer Variations

Method	Aeva (DB)		Ouster (DB)	
	AR@1	AR@5	AR@1	AR@5
w/o S.T.	0.772	0.883	0.797	0.904
S.T. w/ r_f	0.778	0.888	0.795	0.901
S.T. w/ r_w	0.778	**0.891**	0.801	0.906
S.T. w/ r_w (Expanding)	**0.784**	0.890	**0.809**	**0.910**

TABLE V: Recall and Runtime with Various Dimension

Dimension	Aeva (DB)		Ouster (DB)		Runtime (ms)
(m,l,e)	AR@1	AR@5	AR@1	AR@5	
(8, 32, 0)	0.715	0.854	0.758	0.877	26.4 + 0.7
(16, 32, 0)	0.751	0.874	0.773	0.890	26.4 + 1.2
(32, 64, 0)	0.758	0.878	0.781	0.894	26.4 + 5.0
(32, 64, 256)	0.772	0.886	0.792	0.901	26.5 + 5.5
(64, 128, 0)	0.778	0.886	0.796	0.902	26.5 + 19.6
(64, 128, 256)	**0.784**	**0.890**	**0.809**	**0.910**	26.5 + 19.8

Multi Scalability: As Table. V, the effect of descriptor dimensions on performance, time complexity, and using GeM in feature aggregation are examined. Runtime is split into inference time (including preprocessing) and retrieval time based on a database of approximately $1.6k$ samples. A larger dimension improves recall while inference time remains almost constant as descriptor expansion is managed within the aggregator; however, retrieval time increases due to distance calculations. All configures achieve real-time speed below the 10Hz LiDAR frequency. $e = 256$ configuration, including GeM, yields notable gains with minimal dimension expansion. Notably, (32, 64, 256) performs similarly to (64, 128, 0), and even higher dimensions like (64, 128, 0) is benefited from additional concatenation. This indicates that combining GeM and MLP effectively compensates for information loss during clustering assignments as the class token in the vision transformer, enhancing the global descriptor with the addition of smaller dimensions.

V. Conclusion

In this paper, we present HeLiOS, the first deep network for heterogeneous LPR. HeLiOS utilizes a local spherical transformer to learn the local distribution from each LiDAR and optimal transport-based clustering to aggregate the local features. Our overlap-based data mining and guided-triplet loss address the limitations of distance-based mining and fixed-margin triplet loss, resulting in more effective embeddings. Evaluations with public datasets show HeLiOS outperforms existing methods and demonstrates robustness in long-term place recognition with unseen LiDARs. Ablation studies validate the impact of proposing loss functions, model architecture, and descriptor dimensions. As the first heterogeneous LPR framework, HeLiOS opens up new opportunities for future work, including reranking tasks to enhance performance or integration with LiDAR simultaneous localization and mapping (SLAM) and multi-robot applications.

References

[1] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2018, pp. 4802–4809.

[2] G. Kim, S. Choi, and A. Kim, "Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments," *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1856–1874, 2021.

[3] 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 Trans. Robot.*, 2023.

[4] L. Luo, S. Zheng, Y. Li, Y. Fan, B. Yu, S.-Y. Cao, J. Li, and H.-L. Shen, "Bevplace: Learning lidar-based place recognition using bird's eye view images," in *Proc. IEEE Intl. Conf. on Comput. Vision*, 2023.

[5] C. Yuan, J. Lin, Z. Zou, X. Hong, and F. Zhang, "Std: Stable triangle descriptor for 3d place recognition," in *Proc. IEEE Intl. Conf. on Robot. and Automat.* IEEE, 2023, pp. 1897–1903.

[6] C. Yuan, J. Lin, Z. Liu, H. Wei, X. Hong, and F. Zhang, "Btc: A binary and triangle combined descriptor for 3d place recognition," *IEEE Trans. Robot.*, 2024.

[7] H. Kim, J. Choi, T. Sim, G. Kim, and Y. Cho, "Narrowing your fov with solid: Spatially organized and lightweight global descriptor for fov-constrained lidar place recognition," *IEEE Robot. and Automat. Lett.*, 2024.

[8] J. Ma, G. Xiong, J. Xu, and X. Chen, "Cvtnet: A cross-view transformer network for lidar-based place recognition in autonomous driving environments," *IEEE Trans. Ind. Informatics*, 2023.

[9] J. Ma, J. Zhang, J. Xu, R. Ai, W. Gu, and X. Chen, "Overlap-transformer: An efficient and yaw-angle-invariant transformer network for lidar-based place recognition," *IEEE Robot. and Automat. Lett.*, vol. 7, no. 3, pp. 6958–6965, 2022.

[10] T. Shan, B. Englot, F. Duarte, C. Ratti, and D. Rus, "Robust place recognition using an imaging lidar," in *Proc. IEEE Intl. Conf. on Robot. and Automat.* IEEE, 2021, pp. 5469–5475.

[11] J. Lin, X. Liu, and F. Zhang, "A decentralized framework for simultaneous calibration, localization and mapping with multiple lidars," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2020, pp. 4870–4877.

[12] M. Jung, W. Yang, D. Lee, H. Gil, G. Kim, and A. Kim, "Helipr: Heterogeneous lidar dataset for inter-lidar place recognition under spatiotemporal variations," *Intl. J. of Robot. Research*, p. 02783649241242136, 2023.

[13] T. Guan, A. Muthuselvam, M. Hoover, X. Wang, J. Liang, A. J. Sathyamoorthy, D. Conover, and D. Manocha, "Crossloc3d: Aerial-ground cross-source 3d place recognition," in *Proc. IEEE Intl. Conf. on Comput. Vision*, 2023, pp. 11 335–11 344.

[14] A. J. Lee, S. Song, H. Lim, W. Lee, and H. Myung, "$(lc)^2$: Lidar-camera loop constraints for cross-modal place recognition," *IEEE Robot. and Automat. Lett.*, vol. 8, no. 6, pp. 3589–3596, 2023.

[15] J. Komorowski, M. Wysoczańska, and T. Trzcinski, "Minkloc++: lidar and monocular image fusion for place recognition," in *Intl. Joint Conf. on Neural Networks*, 2021, pp. 1–8.

[16] X. Lai, Y. Chen, F. Lu, J. Liu, and J. Jia, "Spherical transformer for lidar-based 3d recognition," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, pp. 17 545–17 555.

[17] S. Izquierdo and J. Civera, "Optimal transport aggregation for visual place recognition," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2024, pp. 17 658–17 668.

[18] M. A. Uy and G. H. Lee, "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2018, pp. 4470–4479.

[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*, 2017, pp. 652–660.

[20] 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 *Proc. IEEE Intl. Conf. on Comput. Vision*, 2019, pp. 2831–2840.

[21] J. Guo, P. V. Borges, C. Park, and A. Gawel, "Local descriptor for robust place recognition using lidar intensity," *IEEE Robot. and Automat. Lett.*, vol. 4, no. 2, pp. 1470–1477, 2019.

[22] J. Komorowski, "Improving point cloud based place recognition with ranking-based loss and large batch training," in *Proc. Intl. Conf. Pattern Recog.*, 2022, pp. 3699–3705.

[23] K. Vidanapathirana, M. Ramezani, P. Moghadam, S. Sridharan, and C. Fookes, "Logg3d-net: Locally guided global descriptor learning for 3d place recognition," in *Proc. IEEE Intl. Conf. on Robot. and Automat.*, 2022, pp. 2215–2221.

[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 *Proc. IEEE Intl. Conf. on Comput. Vision*, pp. 8461–8472.

[25] R. G. Goswami, N. Patel, P. Krishnamurthy, and F. Khorrami, "Salsa: Swift adaptive lightweight self-attention for enhanced lidar place recognition," *IEEE Robot. and Automat. Lett.*, 2024.

[26] Z. Zhou, J. Xu, G. Xiong, and J. Ma, "Lcpr: A multi-scale attention-based lidar-camera fusion network for place recognition," *IEEE Robot. and Automat. Lett.*, 2023.

[27] M. Leyva-Vallina, N. Strisciuglio, and N. Petkov, "Generalized contrastive optimization of siamese networks for place recognition," 2023.

[28] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, and C. Stachniss, "OverlapNet: Loop Closing for LiDAR-based SLAM," in *Proc. Robot.: Science & Sys. Conf.*, 2020.

[29] X. Dong and J. Shen, "Triplet loss in siamese network for object tracking," in *Proc. European Conf. on Comput. Vision*, 2018, pp. 459–474.

[30] F. Wang and H. Liu, "Understanding the behaviour of contrastive loss," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2021, pp. 2495–2504.

[31] W. Ge, "Deep metric learning with hierarchical triplet loss," in *Proc. European Conf. on Comput. Vision*, September 2018.

[32] S. Lee, S. Lee, H. Seong, and E. Kim, "Revisiting self-similarity: Structural embedding for image retrieval," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, June 2023.

[33] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2019, pp. 3075–3084.

[34] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu *et al.*, "A survey on vision transformer," *IEEE Trans. Pattern Analysis and Machine Intell.*, vol. 45, no. 1, pp. 87–110, 2022.

[35] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in Neural Information Processing Sys. Conf.*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013.

[36] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning cnn image retrieval with no human annotation," *IEEE Trans. Pattern Analysis and Machine Intell.*, vol. 41, no. 7, pp. 1655–1668, 2018.

[37] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of michigan north campus long-term vision and lidar dataset," *Intl. J. of Robot. Research*, vol. 35, no. 9, pp.

1023–1035, 2016.

[38] G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, “Mulran: Multimodal range dataset for urban place recognition,” in *Proc. IEEE Intl. Conf. on Robot. and Automat.*, 2020, pp. 6246–6253.