This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

FlashMix: Fast Map-Free LiDAR Localization via Feature Mixing and Contrastive-Constrained Accelerated Training

Raktim Gautam Goswami[1], Naman Patel[1], Prashanth Krishnamurthy[1], Farshad Khorrami[1] *

Abstract

Map-free LiDAR localization systems accurately localize within known environments by predicting sensor position and orientation directly from raw point clouds, eliminating the need for large maps and descriptors. However, their long training times hinder rapid adaptation to new environments. To address this, we propose FlashMix, which uses a frozen, scene-agnostic backbone to extract local point descriptors, aggregated with an MLP mixer to predict sensor pose. A buffer of local descriptors is used to accelerate training by orders of magnitude, combined with metric learning or contrastive loss regularization of aggregated descriptors to improve performance and convergence. We evaluate FlashMix on various LiDAR localization benchmarks, examining different regularizations and aggregators, and demonstrating its effectiveness for rapid and accurate LiDAR localization in real-world scenarios. The code is available at https://github.com/raktimgg/FlashMix.

1. Introduction

Localization systems form the backbone of many modern technologies, from navigation to autonomous driving. These systems rely on sensors like LiDARs and cameras to determine an agent's position and orientation in a scene. LiDARs often prove more reliable, particularly in environments where appearances fluctuate. Conventional approaches for LiDAR localization use place recognition algorithms to retrieve a target point cloud from a database and perform registration to ascertain the query's pose [8, 16, 33, 34]. However, this strategy necessitates significant memory for storing map points and descriptors in addition to computationally intensive registration processes.

Recently, map-free LiDAR localization systems have shown great promise for pose estimation in known environments. Initially developed for camera images, these systems aim to predict sensor pose directly through regression, potentially reducing the need for memory-intensive maps and descriptors. These approaches either directly predict 6-DoF pose or estimate scene coordinates to determine pose using a Perspective-n-Point solver [7] within a Random Sample Consensus (RANSAC) [6] framework. While effective for small, room-sized, camera-captured scenes, these methods face challenges when scaling to large-scale environments.

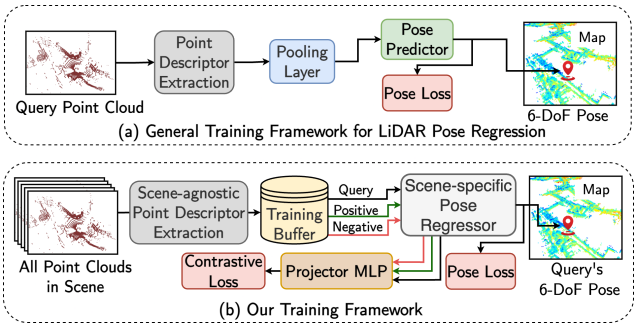


Figure 1. Comparision of LiDAR pose regression-based framework (top) with our fast map-free LiDAR localization system.

LiDAR-based map-free localization approaches were subsequently introduced to capitalize on the rich geometric information provided by LiDAR sensors [38]. Further developments [21, 37, 40, 42] demonstrated the ability to achieve low localization errors in diverse environments through improvements in training methodologies and architectures. Despite promising results across various LiDAR datasets, these approaches face a significant challenge: current pose regression networks typically require lengthy training periods, often lasting hours to days, due to the need for individual training in each scene. This limitation hinders their practicality for robotics applications such as navigation and manipulation, which rely on rapid adaptation to new scenes for subsequent tasks.

PosePN++ [42] introduced the concept of a universal feature encoder, showing that encoder weights trained on one scene could be transferred to another. While this approach reduced training time by requiring only decoder retraining for new scenes, it still faced limitations in quickly

*[1]Control/Robotics Research Laboratory (CRRL), Department of Electrical and Computer Engineering, NYU Tandon School of Engineering, Brooklyn, NY, 11201. E-mails: {rgg9769,nkp269,prashanth.krishnamurthy,khorrami}@nyu.edu.
This paper is supported in part by the Army Research Office under grant number W911NF-21-1-0155 and by the New York University Abu Dhabi (NYUAD) Center for Artificial Intelligence and Robotics, funded by Tamkeen under the NYUAD Research Institute Award CG010.

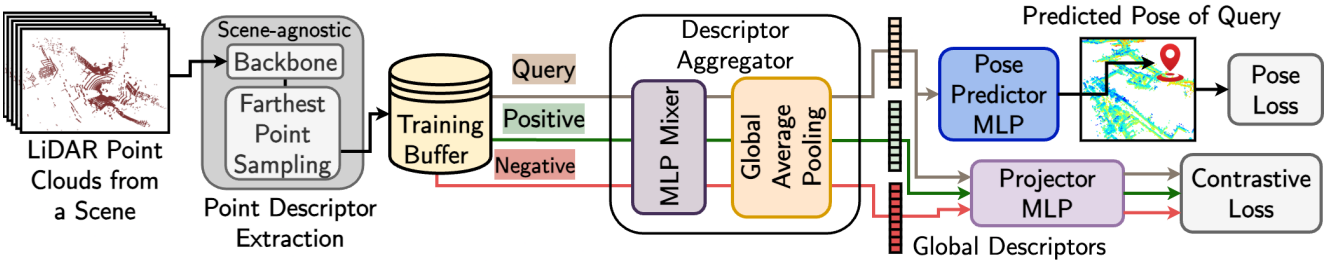


Figure 2. FlashMix framework: A scene-agnostic backbone extracts local descriptors from farthest point sampled point clouds to store in a training buffer. An MLP Mixer and global average pooled aggregate descriptor predict pose from trained pose and contrastive loss.

adapting to diverse environments without compromising accuracy. FlashMix, shown in Fig. 1, directly addresses these challenges, offering a solution that enables rapid adaptation to new scenes while maintaining high localization accuracy. By combining a pre-trained backbone with a scene-specific regressor and innovative training techniques, FlashMix significantly reduces training times for LiDAR localization systems without sacrificing performance. A pre-trained, scene-agnostic backbone is used to extract local point descriptors, which are then aggregated using an MLP mixer to estimate sensor pose. FlashMix creates a training buffer by extracting local descriptors for each point in the point cloud. This buffer is used to train a descriptor aggregator and a pose predictor, which together form the scene-specific pose regressor. This approach, shown in Fig. 2, accelerates training as only the pose regressor needs to be trained for each new scene. Moreover, the pose regressor's design allows the feature aggregator to be customized for specific scenes, accommodating their unique geometries and objects.

The aggregator incorporates an MLP-Mixer [32] layer for feature mixing, integrating global point relationships, followed by global pooling to generate a single descriptor for the point cloud. This descriptor is then processed by the pose predictor to determine the 6-DoF pose. The regressor is trained using a pose loss [14]. To ensure robust global descriptors, we implement a projector MLP, whose output is used for metric learning or contrastive loss regularization. This enhancement boosts performance while maintaining rapid training times. To our knowledge, FlashMix is the first to incorporate contrastive regularization in a LiDAR pose regression framework. This innovative approach enables swift adaptation to new environments, making it ideal for real-world scenarios requiring rapid deployment with reduced storage and communication requirements, making it suitable for single and multi-robot localization systems.

In summary, our contributions are:

- A novel map-free localization framework combining a pre-trained point encoder with a scene-specific pose regressor with feature buffer enabled rapid training.
- Integration of an MLP-Mixer as a descriptor aggregator, to fuse global point relationships by feature mixing for adapting to scene-specific geometries.
- Introduce metric learning and contrastive loss regularization, enhancing global descriptor quality for stable convergence while maintaining fast training times.
- Extensive experiments in outdoor and indoor environments, demonstrating rapid training and adaptation with competitive performance compared to existing map-free localization methods.

The rest of the paper is organized as follows: Sec. 2 covers related work, Sec. 3 presents the problem and our framework, Sec. 4 shows experimental results, and Sec. 5 concludes the paper.

2. Related Works

Traditional LiDAR relocalization systems employing maps typically use retrieval and matching-based approaches for pose estimation. These methods process inputs in the form of BEV projections [16, 24] or raw point clouds [4, 8, 18, 33, 35] to extract local descriptors. The extraction is achieved either through histogram/statistics-based techniques or learned from data. These frameworks then aggregate these local descriptors to generate a global descriptor, which is used to retrieve nearby point clouds from the map. Subsequently, local descriptors from these retrieved point clouds are matched to estimate pose through 3D registration. To enhance pose estimation accuracy, multiple candidate point clouds can be retrieved from the map for a given query. These candidates undergo reranking based on RANSAC registration-derived geometric fitness scores or through spectral methods [34]. This multi-step approach yields more robust and precise localization within the mapped environment, albeit at the cost of increased computational complexity and storage requirements.

Map-free localization approaches address these issues by predicting pose directly from the input image or point cloud using regression, avoiding the need for memory-intensive

databases and costly registration. Camera pose regression has been extensively researched, with various deep learning methods [11, 15, 36]. Notable works include HSC-Net [22], which uses regional classification and FiLM conditioning [27], and its extension SRCNet [5] for few-shot learning. ReCoLoc [30] incorporated region contrastive representation learning, while DSAC* [3] employed scene coordinate regression with differentiable RANSAC [6] for end-to-end pose prediction. Recent advancements accelerate training by using uniform [2] or guided [25] random shuffling of image patch features from a training buffer to decorrelate gradients for learning scene-specific MLP from scene-agnostic dense feature encoder. However, while these methods perform well on small, camera-captured scenes, they face challenges scaling to larger environments.

The pioneering map-free localization method, PointLoc [38], uses a PointNet++ [28] encoder with self-attention to directly estimate 6-DoF poses from LiDAR frames by minimizing pose loss [14]. Building on this, [42] PosePN, PosePN++, PoseSOE, and PoseMinkloc each with different encoders, showed that encoder weights for the same models could be transferred across datasets. STCLoc [41] incorporated spatio-temporal constraints to handle dynamic environments, while NIDALOC [40] implemented a Hebbian memory module to preserve historical information. Hypliloc [37] enhanced performance by fusing descriptors from 3D and spherical representations of point clouds using a hyperbolic fusion function. Departing from direct pose prediction, SGLoc [21] adopted scene coordinate regression, predicting 3D scene coordinates for each point cloud using Kabsch [13] algorithm in a RANSAC loop for pose estimation. Recent approaches have explored generative paradigms, with DiffLoc [20] proposing a multi-step inference process using stable diffusion-based denoising for pose prediction. LiSA [39] utilizes diffusion-based distillation from a 3D semantic segmentation model to learn a multi-scale feature extractor for scene coordinate regression and subsequent pose estimation. Although while SGLoc, DiffLoc, and LiSA have demonstrated promising results with high localization accuracy, they involve lengthy training times and/or computationally intensive evaluation processes, presenting challenges for rapid deployment.

Our method, FlashMix, improves upon existing approaches by using a scene-agnostic point encoder and a scene-specific pose regressor consisting of an MLP-Mixer-based descriptor aggregator and MLP pose predictor. This setup, enhanced with pose loss and metric or contrastive loss regularization, directly predicts the pose while incorporating global relationships through Mixer layers for improved aggregation. FlashMix accelerates training by using a training buffer of local point descriptors extracted from the scene-agnostic encoder, significantly reducing the computational overhead. This allows for rapid adaptation to new environments without compromising accuracy due to the scene-specific aggregator, resulting in a highly competitive performance with significantly reduced training time.

3. Methodology

3.1. Problem Statement

The objective of our map-free LiDAR localization framework is to determine the 6-DoF pose of a LiDAR sensor within a scene from a single LiDAR scan. The LiDAR pose is defined as a rigid transformation that maps coordinates from the LiDAR's local frame to the global scene frame. Our framework, therefore, learns this rigid transformation as a function $\Phi : \mathbb{R}^{N\times 3} \rightarrow \mathbb{R}^6$, which takes a point cloud $Q \in \mathbb{R}^{N\times 3}$ of N points as input and outputs its 6-DoF pose in the scene. This pose comprises a 3-dimensional position vector and a 3-dimensional orientation vector, the latter being represented as the logarithm of the unit quaternion.

In line with previous work, we formulate Φ as a composite function $\Phi = g(h(.))$. Here, $h : \mathbb{R}^{N\times 3} \rightarrow \mathbb{R}^{M\times d}$ is the feature encoder that encodes each point into a feature of dimension d and downsamples the point cloud from N to M points. The function $g : \mathbb{R}^{M\times d} \rightarrow \mathbb{R}^6$ is the regressor that predicts the 6-DoF pose as described above. In our approach, h is pre-trained on a large dataset, making it scene-agnostic and not specific to any particular scene. For each new scene, h is frozen, and only g is trained.

3.2. Scene-agnostic Backbone

FlashMix leverages the SALSA [8] backbone to encode the input point cloud into a higher-dimensional space, generating robust point descriptors. SALSA's backbone is a SphereFormer [19] that utilizes a U-Net [29] backbone with sparse 3D convolutions [9], and Spherical Transformer layers at each depth. The Spherical Transformer block combines cubic-window attention with radial-window attention, ensuring attention computation for distant points within the same radial window. SALSA was trained end-to-end on the extensive Mulran [17] and Apollo-Southbay [23] datasets for LiDAR place recognition. In FlashMix, we use the pre-trained weights of SALSA's backbone and keep it frozen while training the scene-specific pose regressor.

The input LiDAR point cloud is preprocessed by removing the ground plane and voxelized with a voxel size of 0.5m to get $Q \in \mathbb{R}^{N\times 3}$ which is processed through the backbone to generate descriptors for each point, resulting in an output $\hat{F} \in \mathbb{R}^{N\times d}$, where d represents the feature dimension. To manage the variability in the number of points in each point cloud, we apply farthest point sampling (FPS) on Q. This process selects a subset of points and their corresponding descriptors, producing $F \in \mathbb{R}^{M\times d}$. The FPS algorithm begins by randomly selecting an initial point, then iteratively chooses the point farthest from the already selected points

until M points are chosen. This method ensures better coverage of the entire point cloud compared to random sampling. In our experiments, for outdoor scenes, M is set to 1024, while for indoor scenes, M is set to 512.

3.3. Scene-specific Regressor

A global descriptor is formed by aggregating the point descriptors using the MLP-Mixer 3.3.1 architecture. This descriptor is subsequently processed through the Pose Predictor 3.3.2 to estimate the LiDAR's 6-DoF pose.

3.3.1 Descriptor Aggregator

We employ an MLP-Mixer [32] layer to incorporate global relationships within the point descriptors, followed by global average pooling to aggregate a single global descriptor per point cloud (Fig. 3). The MLP-Mixer architecture comprises point-mixing and feature-mixing MLP layers. The point descriptors for a point cloud (P_d of shape $M \times d$) are transposed to shape $d \times M$ and passed through the point-mixing MLP layer. In this layer, the point descriptors interact with each other, facilitating information sharing and enhancing the global representation. The output is then transposed back to shape $M \times d$ and processed through the feature-mixing MLP layer. Both the point-mixing and feature-mixing MLPs incorporate layer normalization and two linear layers with GELU [10] nonlinearity between them. The output of the feature-mixing MLP is projected to a higher dimension l using a linear layer with ReLU nonlinearity, followed by global average pooling to obtain an l-dimensional global descriptor for the point cloud.

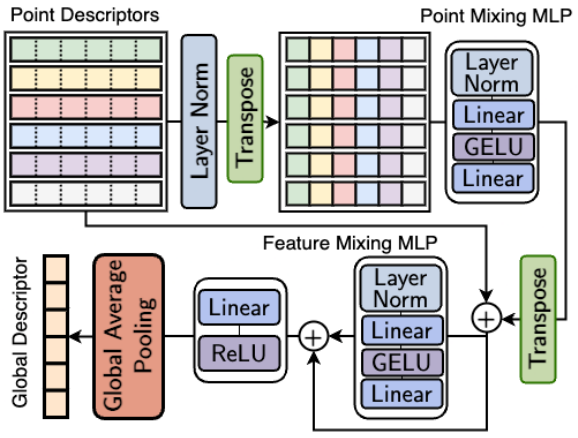


Figure 3. MLP-Mixer Aggregator that fuses local descriptor using point and channel mixing MLPs followed by average pooling.

3.3.2 Pose Predictor

The global descriptor $\in \mathbb{R}^l$ is subsequently passed through a series of MLP regressors, each consisting of a linear layer, followed by batch normalization and a ReLU activation. The output is then bifurcated into separate translation and rotation heads, each employing similar MLP structures. The translation head outputs $t_{pred} \in \mathbb{R}^3$, representing the x, y, z position of the LiDAR, while the rotation head outputs an orientation $q_{pred} \in \mathbb{R}^3$, represented as the logarithm of the unit quaternion to avoid singularities.

3.4. Training Objective

Our training strategy incorporates two key components: a pose loss for accurate position and orientation estimation, and a metric or contrastive loss-based regularization term to enhance performance by making global descriptors robust.

3.4.1 Pose Loss

Following a similar approach as previous work [40, 42], we implement a pose loss mechanism [14] that separately applies L_1 losses to the predicted position and orientation. The overall pose loss, $\mathcal{L}_{pose}$, is then computed as a weighted sum of these individual losses:

$$\mathcal{L}_{pose} = \|t_{pred} - t_{gt}\|_1 + \alpha \|q_{pred} - q_{gt}\|_1 \quad (1)$$

where $pred$ denotes the predicted values, gt denotes the ground truth values from the dataset [1, 17, 38], and α serves as a hyperparameter.

3.4.2 Contrastive Regularization

FlashMix incorporates a contrastive regularization component to enhance the robustness and discriminative power of global descriptors. This process begins by projecting global descriptors into a new embedding space using a projector MLP, consisting of two linear layers with a ReLU activation function between them. We apply the Barlow Twins [43] contrastive loss on the normalized embeddings. This loss is designed to minimize the distance between embeddings of geometrically close point clouds (positives) while maximizing the distance between embeddings of point clouds that are further apart (negatives). The Barlow Twins contrastive loss, hyperparameter μ (0.005), is formulated as:

$$\mathcal{L}_{C.L} = \sum_i (1 - C_{ii})^2 + \mu \sum_i \sum_{j \neq i} C_{ij}^2 \quad (2)$$

where C is the cross-correlation matrix between the descriptors of queries and positives in a batch, and given by

$$C_{i,j} = \frac{\sum_a l^q_{a,i} l^p_{a,j}}{\sqrt{\sum_a (l^q_{a,j})^2} \sqrt{\sum_a (l^p_{a,j})^2}} \quad (3)$$

where $l^q_{a,i}$ and $l^p_{a,i}$ are the values at index a of the projected embeddings of the query (l^q_i) and its positive counterpart (l^p_i), respectively. This encourages a high correlation between queries and their positives and a low correlation with negatives by minimizing feature redundancy while maximizing invariance to positives, effectively enhancing descriptor discrimination.

Alternatively, we propose employing triplet margin loss on the projected embeddings as a regularizer, replacing the Barlow Twins contrastive loss. Being typically used in representation learning tasks like place recognition, the triplet margin loss enhances model robustness by ensuring that the Euclidean distance between the embeddings of a query and its positive is smaller than that between the query and a negative. Specifically, for each set of query l^q_i, positive l^p_i, and negative l^n_i, the triplet margin loss is defined as:

$$\mathcal{L}_{M.L.} = \max\left\{\|l^q_i - l^p_i\|_2^2 - \|l^q_i - l^n_i\|_2^2 + m, 0\right\} \quad (4)$$

where the margin m is set to 0.05. Contrastive loss $\mathcal{L}_{C.L}$ or metric loss $\mathcal{L}_{M.L}$ is composited with the pose loss $\mathcal{L}_{pose}$.

3.5. FlashMix Training

The fast training of the pose regressor 3.3 for each new scene is facilitated by generating a training buffer of point descriptors. This is achieved using the pre-trained backbone, which iterates over the complete dataset of the scene. By employing farthest point sampling, we ensure uniformity in the number of points across all point clouds in the buffer, while also enabling large batch size training on a single GPU. For enhanced training efficiency, we implement mixed precision training and preload the training buffer directly onto the GPU, thereby eliminating communication overhead during data loading. The integration of the MLP-Mixer aggregator and contrastive regularization within FlashMix's training procedure accelerates training times while maintaining high accuracy levels.

4. Experiments

4.1. Dataset and Implementation Details

We test our framework on three public datasets: Oxford-Radar [1], Mulran DCC [17], and vReLoc [38]. Oxford-Radar and Mulran DCC are large-scale outdoor datasets, while vReLoc is a small indoor dataset. Oxford-Radar dataset captures over 32 traversals in central Oxford, containing sensor data collected over a time span of 1 year and a length span of 1000 km. Moreover, it covers various seasons and weather conditions. Similar to previous methods [37, 40, 42], we trained on the sequences named *2019-01-11-14-02-26, 2019-01-14-12-05-52, 2019-01-14-14-48-55, 2019-01-18-15-20-12* and tested on the sequences named *2019-01-15-13-06-37 (Full6), 2019-01-17-13-26-39 (Full7), 2019-01-17-14-03-00 (Full8), 2019-01-18-14-14-42 (Full9)*. For training our method for the Oxford-Radar dataset, we sample point clouds every 1 meter. Mulran DCC contains three trajectories of scans collected from an Ouster-64 LiDAR in South Korea. This dataset is more challenging due to multiple trajectory reversals and occlusions. We trained on DCC1 and DCC2 sequences and tested on the DCC3 sequence. vReLoc is an indoor dataset collected inside a room of area 4m x 5m. Several obstacles are laid in the scene. Similar to previous work, we trained on *seq-03, seq-12, seq-15, seq-16* and tested on *seq-05, seq-06, seq-07, seq-14*.

FlashMix was implemented in Pytorch [26], and experiments were run on an Nvidia RTX A4000 GPU with an Intel(R) i9 CPU and 128 GB RAM. We used the Adam optimizer with initial learning rates of 0.01 for Oxford-Radar and 0.001 for Mulran DCC and vReLoc datasets. A one-cycle policy with cosine annealing was employed, with final learning rates of 10^{-6} for Oxford-Radar and 10^{-5} for Mulran DCC and vReLoc. Batch sizes were 1024 for Oxford-Radar and Mulran DCC and 1280 for vReLoc.

4.2. Results

We compare metric learning (ML Reg.) and contrastive learning (CL Reg.) FlashMix with HypLiLoc [37], NIDALoc [40], and PosePN++ [42] on three datasets described previously. Additional comparisons are in Supp..

4.2.1 Training Time vs Relocalization Rate

Figure 4 shows relocalization success rates versus training time for three datasets. Relocalization is defined as the percentage of samples within 5 meters and 5° degrees for Oxford-Radar and Mulran DCC, and 0.25 meters and 5° for vReLoc. FlashMix achieves competitive results with significantly reduced training times across all datasets. It outperforms HypLiLoc on Oxford-Radar by 20% with only 1/12th of the training time. On Mulran DCC and vReLoc, FlashMix reaches 62.8% and 60.1% relocalization rates (compared to best rates of 65.1% and 63.1%), requiring only 20 and 5 minutes of training respectively.

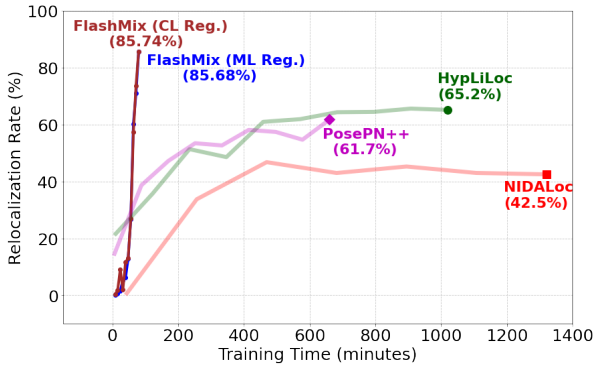
4.2.2 Translation and Rotation Errors

Tab. 1 and 2 show average translation and rotation errors with training times for Oxford-Radar and Mulran DCC datasets. FlashMix achieves lowest translation errors across all sequences, with FlashMix (CL Reg.) slightly outperforming (ML Reg.). On Oxford-Radar, FlashMix's rotation error (1.96°) is higher than HypLiLoc by 0.9° but trains in 80 minutes versus several hours.

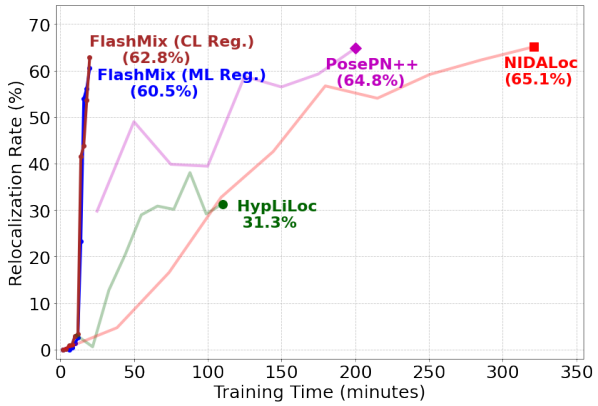
Table 3 shows median errors for vReLoc (500 scans/sequence, 5x4m room). FlashMix trains in 5 minutes

Methods	Training Time	Full6	Full7	Full8	Full9	Average
PosePN++	590 minutes	9.59, 1.92	10.66, 1.92	9.01, 1.51	8.44, 1.71	9.43, 1.77
NIDALoc	1200 minutes	6.71, 1.33	5.45, 1.40	6.68, 1.26	4.80, 1.18	5.91, 1.29
HypLiLoc	1020 minutes	6.00, **1.31**	6.88, **1.09**	5.82, **0.97**	3.45, **0.84**	5.54, **1.05**
FlashMix (ML Reg.)	**80 minutes**	3.15, 2.00	**4.07**, 1.88	**4.61**, 2.54	3.68, 1.79	3.88, 2.05
FlashMix (CL Reg.)	**80 minutes**	**3.05**, 1.96	4.55, 2.05	4.67, 2.05	**2.94**, 1.79	**3.80**, 1.96

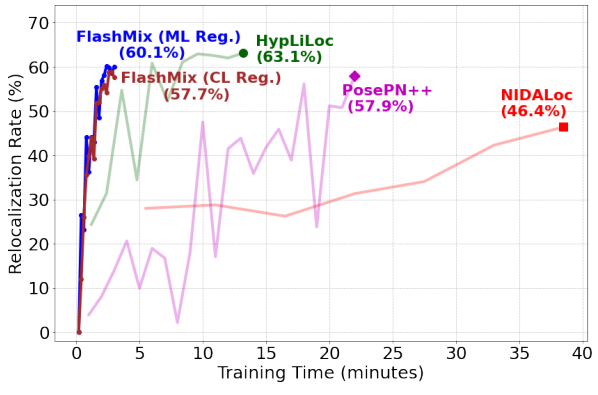
Table 1. Mean position (m) and orientation errors (°) on Oxford-Radar Dataset. Best performance is highlighted in **bold**, lower is better.



(a) Oxford-Radar



(b) Mulran DCC



(c) vReLoC

Figure 4. Analysis of relocalization rate as a function of train time.

Methods	Training Time	DCC3
PosePN++	200 mins.	6.64, 3.43
NIDALoc	321 mins.	5.87, 3.39
HypLiLoc	110 mins.	10.86, **2.88**
FlashMix (ML Reg.)	**20 mins.**	6.07, 4.17
FlashMix (CL Reg.)	**20 mins.**	**5.82**, 3.96

Table 2. Mean position (m) and orientation errors (°) on DCC.

with low errors, being worse than HypLiLoc by only 0.04m and 0.9° which requires double the training time.

Mulran DCC's multiple trajectory retraversals and occlusions create challenges for relocalization, as projecting point clouds into range images results in uninformative occluded pixels. Thus, HypLiLoc struggles with high translation errors whereas FlashMix avoids overfitting, achieving lowest translation error of 5.82 meters.

4.3. Qualitative Comparison

The predicted 2D positions by each method (red), along with the actual values (dark blue), are illustrated in Figure 5. We include plots for the Full8 trajectory of the Oxford-Radar dataset, the DCC3 trajectory of the Mulran DCC dataset, and the seq-06 trajectory for the vReLoc dataset. Our method consistently demonstrates high accuracy in prediction, with the predicted positions closely aligning with the ground truth. In contrast, other methods exhibit a greater dispersion of values, particularly in the outdoor datasets.

4.4. Ablation Studies

Contrastive Regulariazation: We conduct ablation studies with different contrastive losses as training regularizers. Namely, we compare the contrastive losses of SigLIP [44], NTXent [31], and Barlow Twins [43]. We also compare these with the metric learning Triplet Loss because of its use in common LiDAR place recognition frameworks. For the ablation studies, we use the Oxford-Radar dataset and present the relocalization rates in each of its sequences: Full6 (F6), Full7 (F7), Full8 (F8), and Full9 (F9). Further, the performance of the method without using any contrastive or metric loss regularization is shown for reference. As observed in Tab. 4, not using any regularization loss re-

Methods	Training Time	Seq-05	Seq-06	Seq-07	Seq-14	Average
PosePN	40 minutes	0.12, 4.38	0.09, 3.16	0.17, 3.94	**0.08**, 3.27	0.12, 3.69
PosePN++	22 minutes	0.15, 3.12	0.10, 3.31	0.15, 2.92	0.10, 2.80	0.13, 3.04
NIDALoc	38 minutes	0.18, 3.63	0.15, 4.09	0.21, 3.24	0.17, 3.98	0.18, 3.74
HypLiLoc	13 minutes	**0.09, 2.52**	**0.08, 2.58**	**0.13, 2.55**	0.09, **2.34**	**0.10, 2.50**
FlashMix (ML Reg.)	**5 minutes**	0.14, 3.03	0.12, 3.58	0.18, 3.70	0.11, 3.11	0.14, 3.34
FlashMix (CL Reg.)	**5 minutes**	0.16, 3.14	0.12, 3.30	0.18, 3.92	0.11, 3.32	0.14, 3.42

Table 3. Median translation and rotation errors (m/°) on the vReLoc dataset.

	F6	F7	F8	F9	Avg.
No Reg. Loss	88.92	78.15	76.32	89.72	82.77
SigLIP	88.14	81.01	79.43	90.87	84.48
NTXent	88.63	**82.29**	**81.58**	**92.56**	**85.92**
Triplet	**91.95**	82.13	78.80	91.80	85.69
Barlow Twins	91.82	81.56	80.42	90.92	85.74

Table 4. Ablation Study: Impact of Contrastive and Metric Loss regularization.

Aggregator	Time	F6	F7	F8	F9	Avg.
MLP + GAP	**70**	89.26	80.52	78.68	90.50	84.31
MHA + GAP	225	78.37	65.12	62.79	86.93	72.55
Mixer+SALAD	135	90.20	79.00	78.81	91.01	84.27
Mixer+GAP	80	**91.95**	**82.13**	**78.80**	**91.80**	**85.69**

Table 5. Ablation Study: Descriptor Aggregators with Triplet loss regularization. Here, time refers to training time in minutes.

sulted in the poorest performance. While NTXent offers higher performance, its quadratic scaling with batch size creates efficiency challenges. To balance performance and efficiency, we use Triplet loss and Barlow Twins loss.

Descriptor Aggregator: In this ablation study, we explored the impact of different aggregator blocks on FlashMix's performance. Namely, we experimented with Mixer+Global Average Pooling (GAP), MLP+GAP, Multi-headed Attention+GAP, and Mixer+SALAD [12]. The performance of these aggregators, when combined with Barlow Twins regularization and Triplet regularization, is detailed in Tab. 6 and 5, respectively. With Triplet regularization, Mixer+GAP achieved the best results while also requiring less training time. Under Barlow Twins regularization, although Mixer+SALAD performed the best, it required approximately 135 minutes of training, compared to the Mixer+GAP, which achieved slightly lower results but needed only 80 minutes. Consequently, we adopted the Mixer+GAP architecture for all our experiments.

Descriptor Dimension: Our findings indicated that increasing the descriptor dimension consistently enhanced performance but also lengthened the training time, as shown in Table 7. To balance performance gains with training efficiency, we settled on a descriptor dimension of 1024.

Aggregator	Time	F6	F7	F8	F9	Avg.
MLP + GAP	**70**	88.06	80.64	81.37	89.96	84.68
MHA + GAP	225	65.17	62.23	58.98	84.34	67.14
Mixer+SALAD	135	**93.67**	**82.50**	**83.47**	**93.56**	**87.86**
Mixer+GAP	80	91.82	81.56	80.42	90.92	85.74

Table 6. Ablation Study: Descriptor Aggregators with Barlow Twins regularization. Time refers to training time in minutes.

Desc. Dim.	Time	F6	F7	F8	F9	Avg.
256	**75**	88.13	75.62	71.2	87.64	80.03
512	80	88.57	77.85	76.02	89.45	82.47
1024	80	91.95	82.13	78.80	91.80	85.69
2048	110	**92.46**	**83.55**	**82.00**	**92.18**	**87.15**

Table 7. Ablation Study: Global Descriptor Dimension. Here, time refers to training time in minutes.

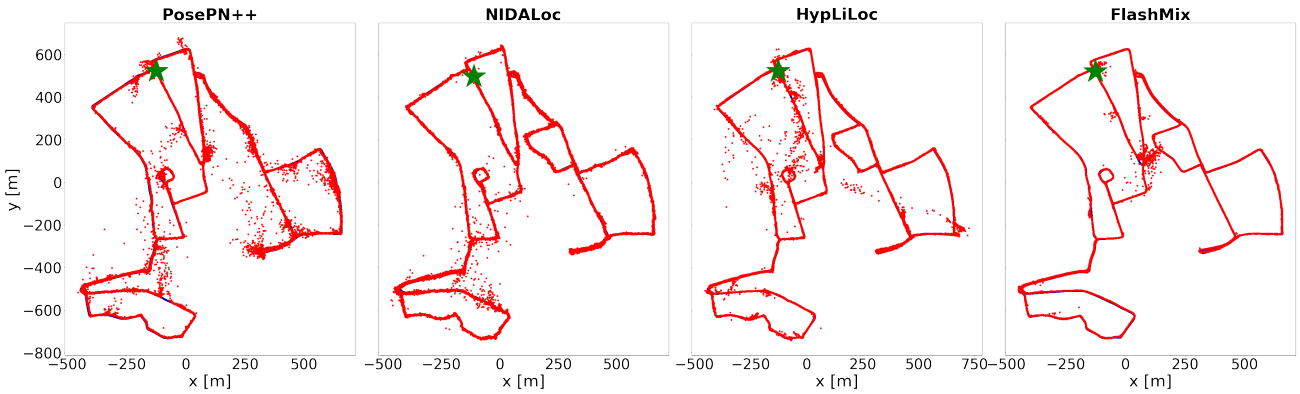
No. of Layers	Time	F6	F7	F8	F9	Avg.
1	**80**	**91.95**	82.13	78.80	**91.80**	85.69
2	90	91.60	82.28	80.11	90.49	85.70
3	100	91.83	82.07	**81.28**	90.90	86.11
4	115	91.60	**83.09**	80.6	91.21	**86.22**

Table 8. Ablation Study: Number of Mixer layers. Here, time refers to training time in minutes.

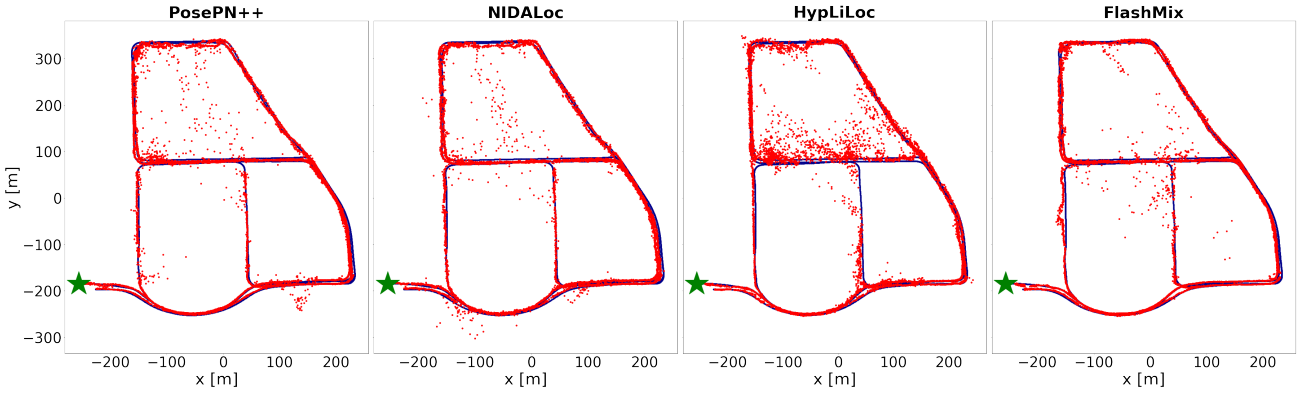
No. of Layers	Time	F6	F7	F8	F9	Avg.
4	80	84.67	76.1	74.25	87.48	80.17
6	80	**91.95**	82.13	78.8	91.80	85.69
8	80	91.76	**82.71**	**80.96**	**91.81**	**86.39**

Table 9. Ablation Study: Number of Pose predictor layers. Here, time refers to training time in minutes.

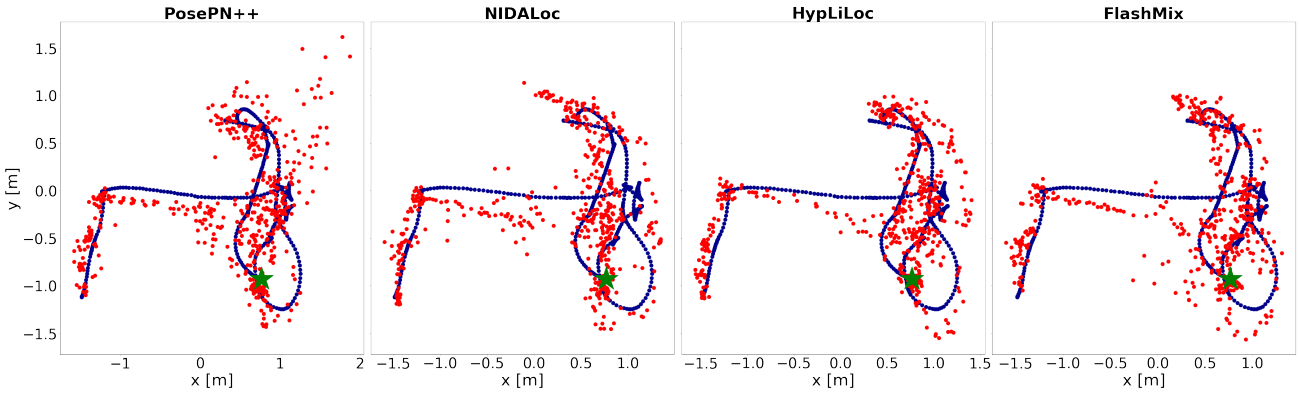
Aggregator and Pose Predictor layer depths: While adding more mixer layers generally improved performance, it also resulted in longer training times, as detailed in Table 8. Consequently, we opted for a single mixer layer in FlashMix to optimize efficiency. Additionally, we chose six layers for the pose predictor as the performance saturates



(a) Oxford-Radar Full8



(b) Mulran DCC DCC3



(c) vReLoC seq-05

Figure 5. Visualization of different methods on test trajectories from Oxford-Radar, DCC, and vReLoC dataset. Trajectory visualization: The ground truth and estimated positions are shown in dark blue and red dots, respectively. The star shows the starting position.

after about 6 layers (Table 9).

5. Conclusion

FlashMix addresses the challenge of long training times in map-free LiDAR localization systems while maintaining accuracy and studies the effect of contrastive/metric regularization on pose estimation performance. It utilizes a frozen, scene-agnostic backbone, a descriptor buffer, and an MLP mixer with contrastive or metric loss regularization to rapidly adapt to new scenes. Our extensive evaluations across various LiDAR localization benchmarks demonstrate FlashMix's effectiveness in delivering fast and accurate localization in real-world scenarios.

References

[1] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In Proceedings of the IEEE International Conference on Robotics and Automation, Paris, France, May 2020. 4, 5

[2] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5044–5053, Vancouver, Canada, June 2023. 3

[3] Eric Brachmann and Carsten Rother. Visual camera relocalization from rgb and rgb-d images using dsac. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(9):5847–5865, 2021. 3

[4] 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. 2

[5] Siyan Dong, Shuzhe Wang, Yixin Zhuang, Juho Kannala, Marc Pollefeys, and Baoquan Chen. Visual localization via few-shot scene region classification. In Proceedings of the International Conference on 3D Vision, pages 393–402, Prague, Czechia, September 2022. IEEE. 3

[6] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6):381–395, 1981. 1, 3

[7] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(8):930–943, 2003. 1

[8] Raktim Gautam Goswami, Naman Patel, Prashanth Krishnamurthy, and Farshad Khorrami. Salsa: Swift adaptive lightweight self-attention for enhanced lidar place recognition. IEEE Robotics and Automation Letters, 9:8242–8249, 2024. 1, 2, 3

[9] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D semantic segmentation with submanifold sparse convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9224–9232, Salt Lake City, UT, June 2018. 3

[10] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415, 2016. 4

[11] Joao F Henriques and Andrea Vedaldi. Mapnet: An allocentric spatial memory for mapping environments. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8476–8484, Salt Lake City, UT, June 2018. 3

[12] Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 17658–17668, Seattle, WA, June 2024. 7

[13] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography, 32(5):922–923, 1976. 3

[14] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5974–5983, Honolulu, HI, July 2017. 2, 3, 4

[15] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings of the IEEE International Conference on Computer Vision, pages 2938–2946, Boston, MA, June 2015. 3

[16] Giseop Kim and Ayoung Kim. Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4802–4809, Madrid, Spain, October 2018. IEEE. 1, 2

[17] Giseop Kim, Yeong Sang Park, Younghun Cho, Jinyong Jeong, and Ayoung Kim. Mulran: Multimodal range dataset for urban place recognition. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 6246–6253, Paris, France, May-August 2020. 3, 4, 5

[18] Jacek Komorowski, Monika Wysoczanska, and Tomasz Trzcinski. Egonn: Egocentric neural network for point cloud based 6dof relocalization at the city scale. IEEE Robotics and Automation Letters, 7(2):722–729, 2021. 2

[19] Xin Lai, Yukang Chen, Fanbin Lu, Jianhui Liu, and Jiaya Jia. Spherical transformer for lidar-based 3D recognition. In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, pages 17545–17555, Vancouver, Canada, June 2023. 3

[20] Wen Li, Yuyang Yang, Shangshu Yu, Guosheng Hu, Chenglu Wen, Ming Cheng, and Cheng Wang. Diffloc: Diffusion model for outdoor lidar localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 15045–15054, Seattle, WA, June 2024. 3

[21] Wen Li, Shangshu Yu, Cheng Wang, Guosheng Hu, Siqi Shen, and Chenglu Wen. Sgloc: Scene geometry encoding for outdoor lidar localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9286–9295, Vancouver, Canada, June 2023. 1, 3

[22] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11983–11992, Seattle, WA, June 2020. 3

[23] Weixin Lu, Yao Zhou, Guowei Wan, Shenhua Hou, and Shiyu Song. L3-Net: Towards learning based lidar localization for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6389–6398, Long Beach, CA, June 2019. 3

[24] Lun Luo, Si-Yuan Cao, Bin Han, Hui-Liang Shen, and Junwei Li. Bvmatch: Lidar-based place recognition using bird's-eye view images. IEEE Robotics and Automation Letters, 6(3):6076–6083, 2021. 2

[25] Son Tung Nguyen, Alejandro Fontan, Michael Milford, and Tobias Fischer. Focustune: Tuning visual localization through focus-guided sampling. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 3606–3615, Seattle, WA, June 2024. 3

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems, 32, 2019. 5

[27] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, New Orleans, LA, February 2018. 3

[28] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in Neural Information Processing Systems, 30, 2017. 3

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In Proc. of the Medical Image Computing and Computer-Assisted Intervention, pages 234–241, Munich, Germany, October 2015. 3

[30] Mehmet Sarıgül and Levent Karacan. Region contrastive camera localization. Pattern Recognition Letters, 169:110–117, 2023. 3

[31] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, Proceedings of the Advances in Neural Information Processing Systems, volume 29, Barcelona, Spain, December 2016. 6

[32] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. Advances in Neural Information Processing Systems, 34:24261–24272, 2021. 2, 4

[33] 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, Salt Lake City, UT, June 2018. 1, 2

[34] Kavisha Vidanapathirana, Peyman Moghadam, Sridha Sridharan, and Clinton Fookes. Spectral geometric verification: Re-ranking point cloud retrieval for metric localization. IEEE Robotics and Automation Letters, 8(5):2494–2501, 2023. 1, 2

[35] Kavisha Vidanapathirana, Milad Ramezani, Peyman Moghadam, Sridha Sridharan, and Clinton Fookes. Logg3d-net: Locally guided global descriptor learning for 3d place recognition. In 2022 International Conference on Robotics and Automation, pages 2215–2221, Philadelphia, PA, May 2022. IEEE. 2

[36] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 10393–10401, New York, NY, February 2020. 3

[37] Sijie Wang, Qiyu Kang, Rui She, Wei Wang, Kai Zhao, Yang Song, and Wee Peng Tay. Hypliloc: Towards effective lidar pose regression with hyperbolic fusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5176–5185, Vancouver, Canada, June 2023. 1, 3, 5

[38] Wei Wang, Bing Wang, Peijun Zhao, Changhao Chen, Ronald Clark, Bo Yang, Andrew Markham, and Niki Trigoni. Pointloc: Deep pose regressor for lidar point cloud localization. IEEE Sensors Journal, 22(1):959–968, 2021. 1, 3, 4, 5

[39] Bochun Yang, Zijun Li, Wen Li, Zhipeng Cai, Chenglu Wen, Yu Zang, Matthias Muller, and Cheng Wang. Lisa: Lidar localization with semantic awareness. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 15271–15280, Seattle, WA, June 2024. 3

[40] Shangshu Yu, Xiaotian Sun, Wen Li, Chenglu Wen, Yunuo Yang, Bailu Si, Guosheng Hu, and Cheng Wang. Nidaloc: Neurobiologically inspired deep lidar localization. IEEE Transactions on Intelligent Transportation Systems, 2023. 1, 3, 4, 5

[41] Shangshu Yu, Cheng Wang, Yitai Lin, Chenglu Wen, Ming Cheng, and Guosheng Hu. Stcloc: Deep lidar localization with spatio-temporal constraints. IEEE Transactions on Intelligent Transportation Systems, 24(1):489–500, 2022. 3

[42] Shangshu Yu, Cheng Wang, Chenglu Wen, Ming Cheng, Minghao Liu, Zhihong Zhang, and Xin Li. Lidar-based localization using universal encoding and memory-aware regression. Pattern Recognition, 128:108685, 2022. 1, 3, 4, 5

[43] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In Proceedings of the International Conference on Machine Learning, pages 12310–12320, Vienna, Austria, July 2021. 4, 6

[44] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 11975–11986, Paris, France, October 2023. 6