2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

OCLPlace: Online Continual Learning on LiDAR Streams for Place Recognition

Binhong Liu[1], Kaixiao Ye[1], Yangwang Fang[1], Zhi Yan[2] and Tao Yang[1*]

***Abstract*— LiDAR place recognition is a critical component of LiDAR-based localization pipelines, tasked with identifying previously visited places across diverse environments and temporal conditions. A growing body of deep learning–based approaches has recently tackled this problem. However, their performance often degrades when the models are deployed in unseen environments. Although offline fine-tuning can partly recover performance, it is prone to catastrophic forgetting of previously acquired knowledge and cannot respond quickly enough to rapidly changing data distributions. In this paper, we introduce OCLPlace, an online continual learning framework that learns directly from highly temporally correlated LiDAR streams and strikes a trade-off between rapid domain adaptation and resistance to catastrophic forgetting. To the best of our knowledge, OCLPlace is the first LiDAR place-recognition approach enhanced by online continual learning that can automatically adapt to new environments while mitigating catastrophic forgetting. Experimental results on six large-scale datasets, which cover both ground-view and aerial-view scenarios, demonstrate the effectiveness and robustness of our method. The source code will be publicly available at: `https://github.com/npu-ius-lab/OCLPlace`.**

I. INTRODUCTION

Place recognition enables a robot to identify the most similar location in a pre-built map, serving as a foundation for global localization. Recently, deep learning-based LiDAR place recognition has achieved remarkable performance on large-scale benchmark datasets. However, most of these methods are developed and evaluated under single-type environments, where conditions remain consistent. In real-world navigation, robots are likely to encounter a diverse range of environments such as urban areas, campuses, and forests, or even cross-view scenarios (e.g., transitions from ground to aerial perspectives). Under such conditions, existing approaches often experience substantial performance degradation due to domain shifts in the operating environments, thereby limiting the deployment of source-domain pre-trained models in real-world applications. A straightforward solution is to incrementally collect training data from the working environment and fine-tune the model offline through multiple iterations. However, this method is often ineffective in the face of rapid and abrupt environmental changes. With these challenges in mind, the lack of rapid adaptability to changing environments remains a major obstacle preventing existing place recognition methods from effectively supporting long-term localization tasks.

For learning-based methods, it is crucial to update model parameters promptly to cope with continuous environmental changes, especially when domain shifts are present. However, when the test-time distribution diverges significantly from the source domain, long-term adaptation may lead to catastrophic forgetting of previously acquired knowledge. Online continual learning (OCL) has emerged as a principled framework to address this challenge, enabling models to adapt incrementally after deployment while alleviating forgetting. Recent studies have explored OCL in image classification [1], 3D object detection [2], person re-identification [3], etc., demonstrating promising results. Nevertheless, its application to LiDAR place recognition remains largely unexplored, leaving a gap between the demand for robust long-term localization and the current state of research.

In this paper, we explore LiDAR place recognition under the OCL paradigm. Using a lightweight, robust, and generalizable backbone [4], we introduce a dual-memory module that jointly leverages short-term and long-term memories for on-the-fly adaptation and durable knowledge. This architecture enables the model to quickly adapt to new LiDAR data distributions while preserving knowledge from previously LiDAR settings and visited environments, thereby providing a reliable solution for long-term localization in diverse real-world scenarios. The main contributions of this paper are as follows:

- We introduce an online continual learning paradigm for LiDAR place recognition that achieves fast in-situ adaptation while mitigating catastrophic forgetting.
- We propose OCLPlace, an online dual-memory framework where a short-term memory enables fast consolidation and a long-term memory preserves mined hard triplets for durable replay.
- We benchmark existing online continual learning methods for challenging LiDAR place recognition tasks across six large-scale datasets, covering both ground and aerial scenarios. The code and benchmarks will be released.

II. RELATED WORK

A. Learning-based LiDAR Place Recognition

The existing learning-based LiDAR place recognition methods can be broadly divided into three groups: point-based methods [5], [6], voxel-based methods [7], [8], and

[1]Binhong Liu, Kaixiao Ye, Yangwang Fang and Tao Yang are with Unmanned System Research Institute, National Key Laboratory of Unmanned Aerial Vehicle Technology, Integrated Research and Development Platform of Unmanned Aerial Vehicle Technology, Northwestern Polytechnical University, 710072 Xi'an, China. *Corresponding author: yangtao@nwpu.edu.cn

[2]Zhi Yan is with U2IS, ENSTA, Institut Polytechnique de Paris, France. zhi.yan@ensta.fr.

projection-based methods [4], [9]. Point-based methods directly process raw 3D points to exploit the inherent spatial structure of LiDAR data. A representative example is PointNetVLAD [5], the pioneering learning-based method for point cloud place recognition, which extracts point-wise features using PointNet [10] and aggregates them into a compact global descriptor via the NetVLAD module. Voxel-based methods discretize point clouds into voxel grids and employ sparse or dense 3D convolutions to learn discriminative global descriptors. These methods, such as MinkLoc3D [7] and EgoNN [8], leverage voxelization to preserve local geometric structures while enabling efficient large-scale retrieval. Projection-based methods transform point clouds into 2D representations, such as range images or bird's-eye view (BEV) maps, and then apply mature 2D convolutional networks for feature learning. Typical examples include OverlapNet[9] and BEVPlace++[4], which achieve high efficiency and robustness to viewpoint changes through effective projection designs.

B. Online Learning for Robotics

Online learning in robotics focuses on enabling robots to automatically adapt to changes in domain distributions by updating their models in real time from streaming data [11]. Unlike offline learning, where adaptation is performed after data collection, online learning continuously incorporates incoming observations from the environment or human interactions to maintain task performance under domain shift conditions. A major challenge for such methods lies in acquiring reliable supervision signals during deployment, so that the model can progressively adapt to the environment while being used, without relying on densely annotated target-domain data. In the field of deep visual-based odometry, self-supervised VO [12] addresses this issue by deriving supervision directly from sequential frames, using appearance consistency (e.g., photometric and structural similarity) and geometric consistency between adjacent views to guide model updates without external labels. Beyond purely visual self-supervision, Adaptive VIO [13] incorporates IMU measurements to provide high-frequency motion priors, and performs immediate model updates via joint optimization of visual consistency losses and inertial constraints, enabling continual adaptation to domain shifts without offline retraining or manual annotations. In the field of LiDAR-based 3D object detection, OCL3D [2] introduces an efficient online transfer learning framework that transfers knowledge from a pre-trained 2D visual detector to an online 3D detector, enabling rapid adaptation without extensive 3D annotations.

C. Continual Learning for Robotics

Continual learning is characterized by the ability to acquire new knowledge from sequentially arriving data while retaining previously learned information [14], [15]. In the field of visual place recognition, AirLoop [16] is the first to introduce a lifelong learning approach for loop closure detection, which mitigates catastrophic forgetting by combining experience replay with feature regularization based on Euclidean distance. VIPeR [17] introduces a continual visual place recognition method that combines adaptive mining for metric learning, a multi-stage memory bank for rehearsal, and probabilistic knowledge distillation for regularization. For LiDAR place recognition, InCloud [18] combined a random initialized memory buffer and a structure-aware loss function, to distill the angular relationship between embedded triplets. CCL [14] combines momentum contrast to help the model learn new environmental knowledge, along with replaying experience and KL divergence-based constraints to mitigate catastrophic forgetting. MICL [15] maximizes mutual information between input data and its representations, as well as between representations across multiple environments.

III. Problem Formalization and Preliminaries

A. BEV Image-based Place Recognition

Given a LiDAR scan $\mathbf{S}$ and its corresponding position $\mathbf{p}$, we first convert $\mathbf{S}$ into a BEV image $\mathbf{I}$. The feature extraction backbone f_θ then encodes $\mathbf{I}$ into a global descriptor $\mathbf{z}$:

$$\mathbf{z} = f_\theta(\mathbf{I}), \quad \mathbf{z} \in \mathcal{Z} \subset \mathbb{R}^d \tag{1}$$

where θ denotes the parameters of the backbone. The global descriptor $\mathbf{z}$ captures the distinctive features of the place at position $\mathbf{p}$. Ideally, the principle of place recognition requires that, for any point cloud triplet $\mathcal{T} = \{\mathbf{S_i}, \mathbf{S_j}, \mathbf{S_k}\}$, the following condition holds:

$$D_g(\mathbf{S_i}, \mathbf{S_j}) < D_g(\mathbf{S_i}, \mathbf{S_k}) \quad \Longrightarrow \quad \|\mathbf{z}_i - \mathbf{z}_j\|_2 < \|\mathbf{z}_i - \mathbf{z}_k\|_2, \tag{2}$$

where $D_g(\mathbf{S_i}, \mathbf{S_j})$ denotes the geographical distance between two point clouds, and $\|\cdot\|_2$ represents the L_2-norm in the embedding space $\mathcal{Z}$.

B. Online Continual Learning for Place Recognition

Online continual learning for place recognition targets practical scenarios where a pre-trained model can automatically adapt to domain shifts while alleviating catastrophic forgetting. In the OCL setting, we consider a sequence of training datasets $\mathbf{D}^{1:t}$, each of which was collected from distinct and disjoint environments. We further assume access to a small memory buffer $\mathbf{M}^t$, which is a subset of $\mathbf{M}^{1:t}$. Unlike offline continual learning methods, which typically assume repeated access to each environment in an offline manner, our approach builds and updates the memory $\mathbf{M}^t$ in an online fashion as the data stream arrives.

IV. Proposed Method

In this section, we present OCLPlace, an efficient online continual learning framework for LiDAR-based place recognition. The overall architecture is illustrated in Fig. 1. We first introduce two strategies for constructing BEV images, tailored for ground-vehicle and aerial-UAV perspectives, respectively. We then describe our online dual-memory system, which consists of both short-term and long-term memories designed to balance rapid adaptation and knowledge retention.

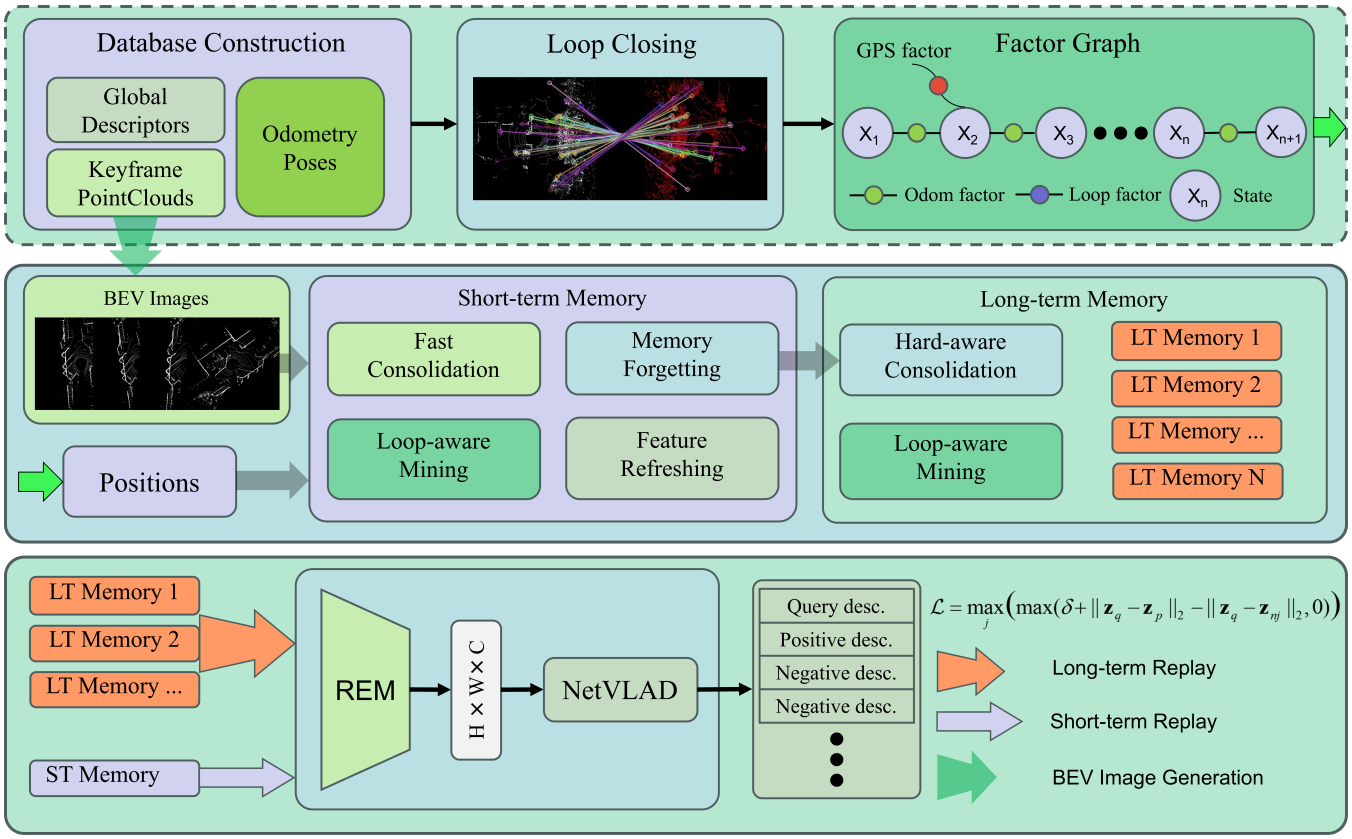

Fig. 1. The architecture of OCLPlace. The framework operates online with a dual-memory design: the short-term memory (STM) performs fast consolidation, loop-aware mining, feature refreshing, and forgetting, while hard triplets mined in the STM are consolidated into the long-term memory (LTM). Both memories are replayed to train a lightweight backbone, i.e., BEVPlace++ [4], using a lazy triplet loss. The proposed method can be seamlessly integrated into SLAM systems (dashed region), thereby enhancing online loop-closure detection while mitigating catastrophic forgetting. The information flow is represented by the shape and color of the arrows.

A. BEV Image Generation

BEV Density Images. Given a point cloud $\mathbf{S}$, we first crop a cubic region of size $D \times D \times D$ m centered at the LiDAR origin. We then apply a voxel-based downsampling with a resolution of r. The cropped cube is divided into vertical $\frac{2D}{r} \times \frac{2D}{r}$ pillars in the ground plane. For each pillar, we compute the normalized point density as the pixel value of the BEV image, which can be expressed as:

$$\mathbf{I}_{\text{density}}(u,v) = \frac{\min(N_g, N_m)}{N_m}, \tag{3}$$

where N_g is the number of points within the pillar at position (u,v), and N_m is the normalization factor, set to the maximum point density in the point cloud.

BEV Height Images. Different from ground-view LiDAR scans, we propose to use height-based BEV images for aerial-view LiDAR data. In ground-view LiDAR, variations in point density often capture object contours and scene structures. By contrast, aerial-view LiDAR mainly observes the rooftops and upper surfaces of objects, where density information provides limited distinction between ground and object surfaces. Instead, height becomes a more discriminative cue in the aerial perspective. To construct the height-based BEV image, we first shift all point cloud heights such that the minimum height is aligned to zero:

$$h_k = h_k^{\text{raw}} - \min_j \left(h_j^{\text{raw}}\right), \tag{4}$$

where h_k^{raw} denotes the raw height of the k-th point.

For each pillar at position (u,v), the pixel value of the height-based BEV image is defined as the normalized average height:

$$\mathbf{I}_{\text{height}}(u,v) = \frac{\bar{h}(u,v)}{H_{\max}}, \quad \bar{h}(u,v) = \frac{1}{N_g(u,v)} \sum_{k=1}^{N_g(u,v)} h_k, \tag{5}$$

where $N_g(u,v)$ is the number of points that fall into pillar (u,v), and

$$H_{\max} = \min\left(\max_{u,v} \bar{h}(u,v),\ 20\right) \tag{6}$$

is the normalization factor with a clipping threshold of 20 m. Fig. 2 illustrates the results of the two methods on the AMtown02 [19] sequence. It can be observed that the height-based representation is more suitable for aerial-view LiDAR scans than the density-based representation.

B. Short-term Memory

1) Fast Consolidation: We implement the short-term memory (STM) using reservoir sampling [20], a streaming-

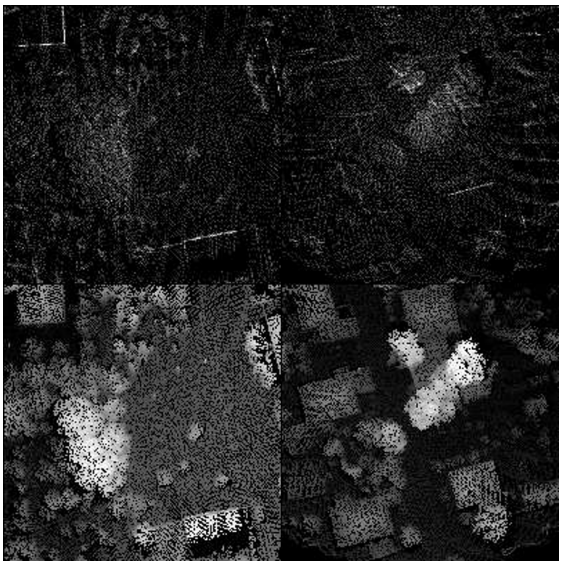

Fig. 2. Comparison of density-based and height-based BEV images on the MARS-LVIG AMtown02 sequence. **Top:** Density-based BEV images, where the boundaries between buildings, trees, and ground are difficult to distinguish. **Bottom:** Height-based BEV images, where trees, buildings, and ground are clearly separated.

friendly mechanism widely adopted in online continual learning. Since place recognition requires query–positive pairs rather than single labeled samples, we treat each query–positive pair as the atomic unit stored in the STM. Each frame is represented as $(\mathbf{I}, \mathbf{p}, \mathbf{z})$, where $\mathbf{I}$ is the BEV image, $\mathbf{p} \in \mathbb{R}^3$ the position, and $\mathbf{z} \in \mathbb{R}^D$ the global descriptor. We retain only pairs satisfying a spatial constraint, e.g., $\tau_{\min} \leq \|\mathbf{p}_q - \mathbf{p}_p\|_2 \leq \tau_+$, ensuring valid correspondences for metric learning.

Given a buffer capacity of M, the first $k = \frac{M}{2}$ pairs are directly inserted. For the n-th incoming pair ($n > k$), a random index $j \sim \mathcal{U}(0, n)$ is drawn; if $j \leq k$, the pair at index j is replaced. This yields an unbiased, time-adaptive subsample of the stream with $\mathcal{O}(1)$ update complexity, enabling efficient online consolidation without growth in computational cost.

2) Memory Forgetting: While the STM supports fast consolidation via probabilistic replacement when the buffer is full, pure reservoir sampling treats all arrivals uniformly and does not account for hardness. For LiDAR place recognition, however, we prefer to retain samples that constrain the decision boundary (i.e., "hard" cases) rather than trivially satisfied pairs. We therefore introduce a hardness-aware forgetting policy that periodically prunes easy pairs and frees capacity for informative ones.

Hardness measure. Let the flattened STM comprise $2N$ items (two views per pair: query and positive). Given the high dimensionality of the global descriptors ($D = 8192$), we first project all descriptors with PCA to a compact subspace $D' \ll D$ to accelerate distance computations while preserving most of the variance:

$$\tilde{\mathbf{Z}}^s = \mathrm{PCA}(\mathbf{Z}^s) \in \mathbb{R}^{2N \times D'}. \tag{7}$$

All subsequent feature space distances are computed in the PCA space $\tilde{\mathbf{Z}}^s$. We then compute the pairwise feature and spatial distance on all items:

$$(\mathbf{D}_f)_{ij} = \left\|\tilde{\mathbf{z}}_i^s - \tilde{\mathbf{z}}_j^s\right\|_2^2, \qquad (\mathbf{D}_p)_{ij} = \left\|\mathbf{p}_i^s - \mathbf{p}_j^s\right\|_2^2. \tag{8}$$

We obtain the per query-positive feature distance vector $\mathbf{d}_{\text{pos}} \in \mathbb{R}^{2N \times 1}$ by treating each item in the STM as the query and its paired counterpart as the positive, i.e., $(\mathbf{d}_{\text{pos}})_i = (\mathbf{D}_f)_{i,\, i \oplus 1}$ where $\oplus$ denotes XOR with 1 (thus $0 \leftrightarrow 1$, $2 \leftrightarrow 3$, $\dots$). We then mask out all negative indices using the spatial constraint $(\mathbf{D}_p)_{ij} > \tau_-^2$ for all queries. Define the binary negative mask:

$$\mathcal{M}_{ij} = \begin{cases} 1, & (\mathbf{D}_p)_{ij} > \tau_-^2, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

The masked negative distance matrix is then

$$\mathbf{D}_{\text{neg}} = \mathbf{D}_f \odot \mathcal{M} + (+\infty)\left(\mathbf{1}\mathbf{1}^\top - \mathcal{M}\right), \tag{10}$$

where $\odot$ denotes the Hadamard product and $\mathbf{1} \in \mathbb{R}^{2N \times 1}$ is the all-ones vector, so that $\mathbf{1}\mathbf{1}^\top \in \mathbb{R}^{2N \times 2N}$. Finally, the score matrix is defined as $\mathbf{S}$:

$$\mathbf{S} = \delta\, \mathbf{1}\mathbf{1}^\top + \mathbf{d}_{\text{pos}}\mathbf{1}^\top - \mathbf{D}_{\text{neg}} \in \mathbb{R}^{2N \times 2N}. \tag{11}$$

Algorithm 1 Memory Forgetting and hardness-aware Consolidation pipeline

Input: Short-term memory M^s, Long-term memory M^l, margin δ.
Data: N query-positive pairs $\mathbf{P}$. ▷ $N \leq k$
Parameters: global descriptor dimensionality D; PCA target dimension D', positive threshold τ_+, negative threshold τ_-, threshold τ_{hard} for hard triplets selection.
Output: Updated Short-term Memory M^s, Updated Long-term Memory M^l.
Step 1: From the pair set $\mathbf{P}$, assemble the descriptor matrix $\mathbf{Z}^s$ and the position matrix $\mathbf{P}^s$. ▷ $\mathbf{Z}^s \in \mathbb{R}^{2N \times D}$, $\mathbf{P}^s \in \mathbb{R}^{2N \times 3}$
Step 2: $\mathbf{Z}^s \leftarrow \mathrm{PCA}(\mathbf{Z}^s)$ ▷ $\mathbf{Z}^s \in \mathbb{R}^{2N \times D'}$
Step 3: Compute the pairwise feature distance matrix $\mathbf{D}_f$ with entries $(\mathbf{D}_f)_{ij} = \|\mathbf{z}_i^s - \mathbf{z}_j^s\|_2^2$. ▷ $\mathbf{D}_f \in \mathbb{R}^{2N \times 2N}$
Step 4: Compute the pairwise positional distance matrix $\mathbf{D}_p$ with entries $(\mathbf{D}_p)_{ij} = \|\mathbf{p}_i^s - \mathbf{p}_j^s\|_2^2$. ▷ $\mathbf{D}_p \in \mathbb{R}^{2N \times 2N}$
Step 5: Calculate pairwise query-positive feature distance $\mathbf{d}_{\text{pos}}$ based $\mathbf{D}_f$. ▷ $\mathbf{d}_{\text{pos}} \in \mathbb{R}^{2N \times 1}$.
Step 6: Calculate query-negatives feature distance $\mathbf{d}_{\text{neg}}$ based $\mathbf{D}_f$ and $\mathbf{D}_p$ with τ_+ and τ_-. ▷ $\mathbf{d}_{\text{neg}} \in \mathbb{R}^{2N \times 2N}$.
Step 7: Calculate hardness score $\mathbf{h}$ with $\mathbf{d}_{\text{pos}}$, $\mathbf{d}_{\text{neg}}$ and margin δ based on Eq. 11 and Eq. 12. ▷ $\mathbf{h} \in \mathbb{R}^{2N \times 1}$.
Prune from $\mathbf{P}$ all pairs whose hardness score $\mathbf{h}$ is non-positive (≤ 0). ▷ Short-term memory forgetting
Collect the selected hard triplets $\mathcal{T}$ whose hardness score $\mathbf{h}$ is larger than τ_{hard} and append them to M^l. ▷ Hard-aware consolidation
return Updated Short-term Memory M^s, Updated Long-term Memory M^l

And the hardness score $\mathbf{h}$ is obtained by taking the row-wise maximum:

$$\mathbf{h} \;=\; \max_j \mathbf{S} \;\in\; \mathbb{R}^{2N\times 1}, \tag{12}$$

The hardness score $\mathbf{h}$ reflects, for each query, the relative difficulty of distinguishing its positive pair from the most confusing negative. A larger score indicates that the closest negative lies closer to the query than its true positive, thus representing a harder training example. This measure allows us to prioritize difficult samples for consolidation into the long-term memory.

Forgetting rule. The hardness score $\mathbf{h}$ indicates how difficult each sample is. A negative value means that, for a given query, the closest valid negative is already farther away than its positive by at least the margin δ. In other words, the triplet constraint is satisfied with slack, and the sample provides little useful information for updating the decision boundary. Since the STM stores data in query-positive pairs, we evaluate forgetting at the pair level: for each pair, we take the larger hardness score of its two items as the pairwise difficulty. If this value is negative, the pair is considered uninformative and is directly removed from the STM. Discarding such pairs not only prevents redundant updates to the model but also frees memory space for harder and more informative samples, thereby improving the overall efficiency of the memory buffer. The complete forgetting procedure is summarized in Algorithm 1.

3) Feature Refreshing: In online continual learning, the backbone parameters θ are updated after each optimization step. As a result, the embedding function f_θ gradually drifts, causing the descriptors stored in memory to become outdated. For the same image $\mathbf{I}$, the previously cached descriptor

$$\mathbf{z}^{\text{old}} = f_{\theta_{\text{old}}}(\mathbf{I}) \;\neq\; \mathbf{z}^{\text{new}} = f_{\theta_{\text{new}}}(\mathbf{I}), \tag{13}$$

no longer matches the current feature representation, which misaligns feature-space distances used for mining hard examples. To ensure a consistent metric space, we periodically refresh the descriptors stored in the STM.

C. Long-term memory

1) hardness-aware Consolidation: Based on the hardness scores $\mathbf{h}$ computed in the STM (refer to Eq. 12), we form for each anchor i a candidate hard triplet $(i,\, i\oplus 1,\, j_i^\star)$, where $j_i^\star$ denotes the index of the hardest valid negative. Triplets with $h_i > \tau_{\text{hard}}$ are selected and inserted into the long-term memory (LTM) for future replay.

After completing training on an environment, the LTM performs a budgeted consolidation pass: it re-scores all triplets by hardness and retains only the highest-scoring subset, downsampling when necessary to ensure that the total number of stored items does not exceed the global budget.

2) Loop-aware Mining: To construct informative triplets for experience replay, we employ a loop-aware positive selection strategy combined with hard-negative mining for both STM and LTM. For a given query index i, we first collect spatial candidates within the positive radius τ_+:

$$\mathcal{C}_+(i) = \{\, j \neq i \mid (\mathbf{D}_p)_{ij} \le \tau_+^2 \,\}. \tag{14}$$

Among these, we prioritize loop-closure positives that are sufficiently separated in time:

$$\mathcal{L}(i) = \{\, j \in \mathcal{C}_+(i) \mid |t_j - t_i| \ge \Gamma \,\}, \tag{15}$$

where Γ denotes the minimum temporal gap. The final positive is selected as:

$$p^*(i) = \begin{cases} \arg\min\limits_{j\in\mathcal{L}(i)} (\mathbf{D}_p)_{ij}, & \text{if } \mathcal{L}(i)\neq\emptyset, \\ i\oplus 1, & \text{if } (i\oplus 1)\in\mathcal{C}_+(i), \\ \arg\min\limits_{j\in\mathcal{C}_+(i)} (\mathbf{D}_p)_{ij}, & \text{otherwise,} \end{cases} \tag{16}$$

where $i \oplus 1$ denotes the paired index in the stored query–positive pair (e.g., $0\leftrightarrow 1$, $2\leftrightarrow 3$, etc.).

Negative mining. For each query, we exclude its positive partner and any samples within the negative radius τ_-, so that only sufficiently distant items remain as valid negatives. Among these candidates, we sort them by feature-space distance to the query and select the K closest ones, which serve as the hardest negatives.

D. Loss Function

We train BEVPlace++ with the lazy triplet loss [4]. Given a query descriptor $\mathbf{z}_q$, its spatially verified positive $\mathbf{z}_p$, and a set of candidate negatives $\{\mathbf{z}_{n_j}\}_{j=1}^K$, the loss function is defined as:

$$\mathcal{L} = \max_j\big(\max(\delta + ||\mathbf{z}_q - \mathbf{z}_p||_2 - ||\mathbf{z}_q - \mathbf{z}_{nj}||_2, 0)\big), \tag{17}$$

where $\mathbf{z}_q$, $\mathbf{z}_p$ and $\mathbf{z}_{nj}$ are the global descriptors of the query sample, its positive sample, and the j-th negative sample, respectively. The tunable margin hyperparameter δ helps to enforce a minimum separation between the positive and negative pairs in the embedding space.

V. Experimental Settings

A. Datasets

We evaluate OCLPlace on six large-scale LiDAR datasets: KITTI [21], NCLT [22], HeLiPR [23], BotanicGarden [24], WildPlaces [25], and MARS-LVIG [19], spanning structured roads, natural scenes, and aerial surveys. KITTI is a standard place recognition benchmark covering country and highway driving scenarios. NCLT provides 27 long-term campus sessions across seasons and years, introducing substantial appearance change and dynamics. HeLiPR contains heterogeneous LiDAR from multiple sensors for cross-sensor generalization. We use KAIST sequences *04* and *06* spanning roughly five months. BotanicGarden captures 48,000 m^2 of dense vegetation, riversides, and grasslands, representing unstructured natural environments. WildPlaces comprises handheld-LiDAR runs in forest reserves with large viewpoint variation and uneven motion. MARS-LVIG offers UAV-borne Livox AVIA mapping over large open areas with non-repetitive, sparse scans, for which we aggregate the latest 10 frames to densify inputs. More details are summarized in Table I.

TABLE I

ONLINE CONTINUAL LEARNING DATASETS.

Dataset	KITTI [21]	NCLT [22]	HeLiPR [23]	BotanicGarden [24]	Wildplaces [25]	MARS-LVIG [19]
LiDAR	Velodyne HDL-64E	Velodyne HDL-32E	Velodyne VLP-16	Velodyne VLP-16	Velodyne VLP-16	Livox Avia
Scenes	Country, Highway	Campus	City, Campus	Natural, Unstructured	Natural, Wild	Aerial Surveying
Time span	Single days	Across 1 year	Several months	Single days	14 months	Single days
Train seq.	00	2013-02-23	KAIST 04	1005_00	V-01	AM02
Test seq.	02 & 08	2012-09-28 & 2012-11-16	KAIST 06	1006_01 & 1008_03	V-04 & K-04	AM01

V-01, V-04, and K-04 correspond to Venman "*01*", Venman "*04*" and Karawatha "*04*". AM01 and AM02 refer to AMtown 01 and AMtown02.

TABLE II

QUANTITATIVE RESULTS UNDER OCL PROTOCOLS

Item	M = 100		M = 200		M = 500		M = 1000	
	mF1 (%)↑	F (%)↓	mF1 (%)↑	F (%)↓	mF1 (%)↑	F (%)↓	mF1 (%)↑	F (%)↓
Fine-tuning	73.83 ± 2.79	11.88 ± 3.64	71.53 ± 1.89	13.70 ± 2.82	68.67 ± 2.80	17.16 ± 3.44	70.46 ± 2.91	14.40 ± 2.80
LwF [26]	68.42 ± 3.72	4.36 ± 2.03	66.88 ± 3.55	4.99 ± 1.43	67.11 ± 5.75	4.02 ± 2.32	68.48 ± 3.01	3.95 ± 1.04
EWC++ [27]	70.48 ± 4.91	3.90 ± 3.05	73.93 ± 4.77	5.30 ± 3.65	77.00 ± 2.28	2.90 ± 2.02	79.62 ± 5.40	2.92 ± 2.30
A-GEM [28]	74.52 ± 3.36	6.14 ± 3.48	72.11 ± 2.10	6.96 ± 3.55	74.57 ± 4.21	5.50 ± 2.83	75.43 ± 1.94	5.68 ± 2.20
ER [29]	74.91 ± 4.39	5.64 ± 1.99	79.38 ± 1.27	2.62 ± 1.25	81.40 ± 1.92	2.95 ± 0.47	83.33 ± 1.95	3.35 ± 2.34
InCloud [18]	76.99 ± 1.16	**2.24** ± 0.68	78.29 ± 2.22	**1.51** ± 0.73	81.15 ± 0.92	**1.19** ± 0.73	82.07 ± 1.50	**1.58** ± 0.89
OCLPlace (**ours**)	**79.76** ± 1.44	2.35 ± 1.42	**81.32** ± 2.72	2.12 ± 1.33	**83.96** ± 1.89	1.67 ± 1.44	**84.69** ± 1.02	2.10 ± 1.81

B. Backbone

In order to meet the real-time online continual learning requirements for robotics, we choose BEVPlace++ [4] as the backbone for our method for two main reasons: first, its lightweight nature allows for fast convergence, and second, its rotation invariance and strong generalization capabilities make it well-suited for the task.

C. Baselines

Fine-tuning. We only use short-term memory with reservoir sampling based consolidation with the common settings in online continual learning. We also enable the negative mining to sample hard triplets for training.

A-GEM [28]. An inequality-constrained replay method that updates on the current mini-batch while projecting the gradient to satisfy constraints induced by an episodic memory, thereby preventing the loss on replayed samples from increasing.

LwF [26]. A well-established continual learning method that does not require storing past data. It leverages the outputs of the previous model on new data as soft targets to regularize the current model, thereby alleviating catastrophic forgetting.

EWC++ [27]. An enhanced elastic weight consolidation [30] method for continual learning. It maintains a running estimate of the Fisher Information Matrix with an exponential moving average, allowing more stable and up-to-date importance weights for parameters. Additionally, it introduces normalization of the Fisher values to balance the regularization strength across tasks, achieving a better trade-off between stability and plasticity compared to standard EWC.

ER [29]. Experience Replay is a simple yet strong baseline for continual learning. It maintains a fixed-size memory buffer that stores a subset of samples from previous tasks. During training on a new task, the model is updated using both current task data and replayed samples from the buffer, mitigating catastrophic forgetting by explicitly rehearsing past knowledge. Despite its simplicity, ER often outperforms more sophisticated regularization-based methods, highlighting the importance of memory replay in continual learning.

InCloud [18]. Originally developed for offline continual LiDAR place recognition, we adapt InCloud to the online setting. We integrate its structure-aware loss with ER and use this variant as a baseline in our experiments.

D. Learning Protocols

We propose an OCL protocol where a model is first trained offline on the KITTI dataset and then continuously trained online across the remaining datasets. Specifically, we adopt a 6-step protocol: KITTI $\rightarrow$ NCLT $\rightarrow$ HeLiPR $\rightarrow$ BotanicGarden $\rightarrow$ WildPlaces $\rightarrow$ MARS-LVIG. During the online training process, each LiDAR scan is processed only once by the system. Since the data stream arrives sequentially over time, position information from time $t+n$ cannot be accessed at time t, which imposes a constraint that prevents offline selection of positive and negative samples.

E. Evaluation Metrics

We use the max F1 score to evaluate the loop closure detection performance. The F1 score can be expressed as:

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{18}$$

The Recall and Precision are calculate by the number of true positives(N_{TP}), false negatives(N_{FN}), and false

positives(N_{FP}), which are:

$$\text{Recall} = \frac{N_{\text{TP}}}{N_{\text{TP}} + N_{\text{FN}}} \tag{19}$$

$$\text{Precision} = \frac{N_{\text{TP}}}{N_{\text{TP}} + N_{\text{FP}}}. \tag{20}$$

We use average max F1 score (i.e., mF1) to evaluate the overall performance after finishing training on the last environment. It can be defined as:

$$\text{mF1} = \frac{1}{T}\sum_{t=1}^{T} \text{F1}_{T,t}, \tag{21}$$

where $\text{F1}_{i,j}$ represent the max F1 score on environment j after the model is trained from environment 1 to i. Forgetting score (i.e., F) to measure the performance of our framework in OCL. The Forgetting score metric is defined as:

$$\text{F} = \frac{1}{T-1}\max_{l\in 1\ldots t}\sum_{t=1}^{T}\{\text{F1}_{l,t}\} - \text{F1}_{T,t} \tag{22}$$

F. Implementation details

For each point cloud, we generate a BEV image of size 200×200. Following the default settings of BEVPlace++, we train on KITTI sequence "*00*". For online continual learning, we select $K = 3$ negatives per query, with a positive radius $\tau_+ = 5\,\text{m}$ and a negative radius $\tau_- = 10\,\text{m}$. By default, the short-term memory M^s has a maximum size of $M = 500$, corresponding to $k = 250$ query-positive pairs. Each long-term memory unit M_i^l is assigned the same capacity as M^s. Feature refreshing is performed every 200 samples. The margin δ is set to 0.3, and the hardness threshold τ_{hard} is set to 0.15. The minimum distance threshold $\tau_{\min}$ is set to 0.5 m to avoid adding overly similar BEV images into the memory.

VI. Experimental Results

In this section, we first demonstrate that models trained in a static setting fail to generalize to unseen environments, particularly when domain shifts are large, and highlight the impact of catastrophic forgetting. We then present quantitative results of OCLPlace compared with other baselines under the OCL protocols. To ensure fair evaluation, we use negative mining for all the methods. Each method is run five times independently, and we report the mean and standard deviation of the results.

A. Limited Generalizability and Catastrophic Forgetting

As shown in Fig. 3, the bottom part presents the performance of three methods across all datasets after training on the final dataset. The source-only represent the pre-trained model on KITTI dataset, performs poorly across most datasets. This highlights the limited generalization ability of models trained in static environments, where the learned representations are insufficient to cope with large variations in unseen domains.

Although simple fine-tuning improves the performance on the most recently trained environment, it often comes at the cost of degrading performance on previously learned domains, such as KITTI and NCLT. This phenomenon indicates that while fine-tuning enables rapid adaptation to the current domain, it simultaneously causes catastrophic forgetting of earlier knowledge, leading to unstable performance across diverse environments. In contrast, our proposed OCLPlace maintains stable performance on the KITTI dataset.

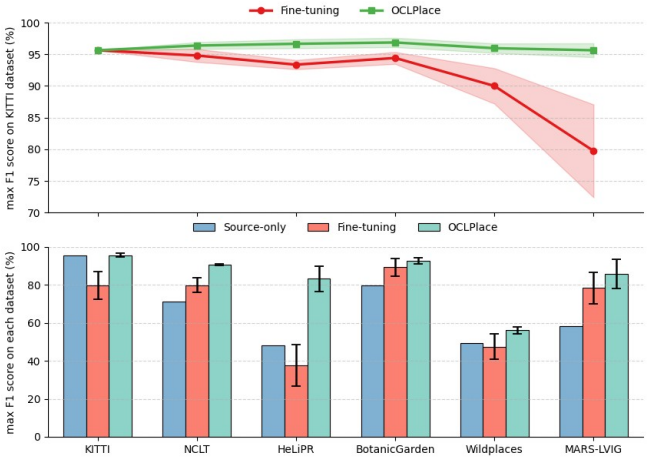

Fig. 3. Performance comparison across different continual learning strategies. **Top:** Evolution of the max F1 score on the KITTI dataset after sequential training on multiple environments. **Bottom:** Max F1 score on six datasets after training on the final environment.

B. Quantitative Results under OCL Protocols

Table II reports the quantitative results under different memory sizes $M = \{100, 200, 500, 1000\}$. Across all memory budgets, OCLPlace consistently achieves the highest mF1 scores. For instance, when $M = 100$, OCLPlace reaches an mF1 of 79.76%, outperforming strong baselines such as ER (74.91%) and InCloud (76.99%). These results suggest that our approach remains robust and effective under varying memory budgets, demonstrating strong scalability and reliability in leveraging replay memory.

In terms of forgetting (i.e., F), OCLPlace also exhibits competitive results. Although InCloud achieves slightly lower forgetting values in some settings, e.g., F = 1.19% at $M = 500$, it suffers from a noticeable drop in mF1 compared to our method. In contrast, OCLPlace achieves a better trade-off, simultaneously maintaining strong resistance to forgetting and superior overall accuracy. This indicates that our method effectively preserves past knowledge while adapting to new domains.

C. Ablation Study

In this section, we conduct ablation studies to evaluate the impact of each component in the proposed dual-memory system. Results are summarized in Table III. It can be seen from Table III that the baseline without dual-memory components ([a]) performs the worst, with lower mF and higher forgetting. Incorporating loop-aware mining ([b]) improves performance, and adding memory forgetting on top ([c]) brings further gains. With all three components enabled ([d]), the model achieves the highest mF (83.96%) and the

TABLE III
ABLATION STUDY FOR DUAL MEMORY

	Loop-aware Mining	Memory Forgetting	hardness-aware Consolidation	mF1 (%) ↑	F (%) ↓
[a]	×	×	×	81.27 ± 2.42	3.06 ± 1.97
[b]	√	×	×	82.50 ± 0.89	3.40 ± 0.91
[c]	√	√	×	83.05 ± 1.49	3.02 ± 1.01
[d]	√	√	√	**83.96** ± 1.89	**1.67** ± 1.44
[e]	√	×	√	83.77 ± 2.09	2.11 ± 1.38
[f]	×	√	√	82.26 ± 1.64	2.75 ± 2.16

lowest forgetting (1.67%), demonstrating their complementary effects. Removing any component ([e], [f]) degrades performance, confirming the necessity of the complete dual-memory design.

VII. CONCLUSIONS

In conclusion, we introduced OCLPlace, the first framework designed for online continual learning in LiDAR place recognition. By leveraging a dual-memory architecture, OCLPlace achieved both fast adaptation to new environments and mitigation of catastrophic forgetting. Extensive experiments across multiple large-scale datasets validated its effectiveness and robustness, establishing new baselines for future research. Future work will focus on extending OCLPlace to multi-modal settings, improving its scalability under long-term deployments, and integrating it more tightly with full SLAM systems for real-world robotic applications.

REFERENCES

[1] Y. Guo, B. Liu, and D. Zhao, "Online continual learning through mutual information maximization," in *International Conference on Machine Learning (ICML)*, 2022, pp. 8109–8126.

[2] B. Liu, D. Yao, R. Yang, Z. Yan, and T. Yang, "Semi-supervised online continual learning for 3d object detection in mobile robotics," *Journal of Intelligent & Robotic Systems*, vol. 110, no. 4, pp. 1–16, 2024.

[3] H. Ye, J. Zhao, Y. Zhan, W. Chen, L. He, and H. Zhang, "Person re-identification for robot person following with online continual learning," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 9151–9158, 2024.

[4] L. Luo, S. Cao, X. Li, J. Xu, R. Ai, Z. Yu, and X. Chen, "Bevplace++: Fast, robust, and lightweight lidar global localization for autonomous ground vehicles," *IEEE Transactions on Robotics*, vol. 41, pp. 4479–4498, 2025.

[5] M. A. Uy and G. H. Lee, "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4470–4479.

[6] D. Cattaneo, M. Vaghi, and A. Valada, "Lcdnet: Deep loop closure detection and point cloud registration for lidar slam," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2074–2093, Aug 2022.

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

[8] J. Komorowski, M. Wysoczanska, and T. Trzcinski, "Egonn: Egocentric neural network for point cloud based 6dof relocalization at the city scale," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 722–729, 2022.

[9] X. Chen, T. Läbe, A. Milioto, T. Röhling, J. Behley, and C. Stachniss, "Overlapnet: A siamese network for computing lidar scan similarity with applications to loop closing and localization," *Autonomous Robots*, pp. 1–21, 2022.

[10] 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 (CVPR)*, 2017, pp. 652–660.

[11] Z. Yan, *Robot Perception and Learning - A Human-aware Navigation and Long-term Autonomy Perspective*. Springer, 2025.

[12] S. Li, X. Wang, Y. Cao, F. Xue, Z. Yan, and H. Zha, "Self-supervised deep visual odometry with online adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (ICCV)*, 2020, pp. 6339–6348.

[13] Y. Pan, W. Zhou, Y. Cao, and H. Zha, "Adaptive VIO: deep visual-inertial odometry with online continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 18 019–18 028.

[14] J. Cui and X. Chen, "CCL: continual contrastive learning for lidar place recognition," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4433–4440, 2023.

[15] B. Liu, T. Yang, Y. Fang, and Z. Yan, "MICL: Mutual information guided continual learning for lidar place recognition," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 10 463–10 470, 2024.

[16] D. Gao, C. Wang, and S. Scherer, "Airloop: Lifelong loop closure detection," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 664–10 671.

[17] Y. Ming, M. Xu, X. Yang, W. Ye, W. Wang, Y. Peng, W. Dai, and W. Kong, "Viper: Visual incremental place recognition with adaptive mining and continual learning," *IEEE Robotics and Automation Letters*, 2025.

[18] J. Knights, P. Moghadam, M. Ramezani, S. Sridharan, and C. Fookes, "Incloud: Incremental learning for point cloud place recognition," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 8559–8566.

[19] H. Li, Y. Zou, N. Chen, J. Lin, X. Liu, W. Xu, C. Zheng, R. Li, D. He, F. Kong, *et al.*, "Mars-lvig dataset: A multi-sensor aerial robots slam dataset for lidar-visual-inertial-gnss fusion," *The International Journal of Robotics Research*, vol. 43, no. 8, pp. 1114–1127, 2024.

[20] J. S. Vitter, "Random sampling with a reservoir," *ACM Trans. Math. Softw.*, vol. 11, no. 1, p. 37–57, Mar. 1985.

[21] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *cvpr*, 2012, pp. 3354–3361.

[22] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of michigan north campus long-term vision and LiDAR dataset," *The International Journal of Robotics Research*, vol. 35, no. 9, pp. 1023–1035, 2016.

[23] 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," *The International Journal of Robotics Research*, vol. 43, no. 12, pp. 1867–1883, 2024.

[24] Y. Liu, Y. Fu, M. Qin, Y. Xu, B. Xu, F. Chen, B. Goossens, P. Z. Sun, H. Yu, C. Liu, *et al.*, "Botanicgarden: A high-quality dataset for robot navigation in unstructured natural environments," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2798–2805, 2024.

[25] J. Knights, K. Vidanapathirana, M. Ramezani, S. Sridharan, C. Fookes, and P. Moghadam, "Wild-places: A large-scale dataset for lidar place recognition in unstructured natural environments," in *2023 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2023, pp. 11 322–11 328.

[26] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[27] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 532–547.

[28] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with a-gem," in *ICLR*, 2019.

[29] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," *Advances in neural information processing systems*, vol. 32, 2019.

[30] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.