Continual Online Backward-Compatible Learning for LiDAR Place Recognition in Adverse Weather

Binhong Liu[1], Qiaoyu Xu[1], Yangwang Fang[1], Zhi Yan[2] and Tao Yang[1*]

***Abstract*— LiDAR place recognition plays a critical role in map-based localization systems. However, its performance degrades significantly under adverse weather conditions due to severe domain shifts in point cloud observations. While recent works explore online or continual adaptation to address such shifts, they typically require updating the entire database or recomputing historical descriptors, which is impractical for long-term deployment. In this paper, we propose an online backward-compatible learning framework for LiDAR place recognition under adverse weather. The key idea is to adapt a dynamic query encoder to align with a fixed map embedding space constructed by a frozen model, thereby avoiding costly database updates. To this end, we formulate an asymmetric learning paradigm where query descriptors are continuously optimized against a static map representation. A reliability gate is introduced to filter noisy samples during online adaptation, and a map anchor loss is designed to constrain representation drift and preserve backward compatibility. We evaluate our method in a continual adaptation setting across multiple large-scale datasets (KITTI → NCLT → Boreas). Experimental results demonstrate that the proposed approach consistently improves online retrieval performance under severe domain shifts, while maintaining compatibility with pre-built map descriptors without backfilling. These results highlight the effectiveness of our framework for practical long-term LiDAR-based localization systems.**

I. INTRODUCTION

In many GNSS-denied or GNSS-degraded environments, autonomous robots primarily rely on onboard sensors for localization. Existing systems mainly fall into two categories: SLAM and map-based localization using pre-built maps. In both paradigms, place recognition is a fundamental component, providing coarse pose hypotheses by retrieving locations in the map that are similar to the current observation.

However, real-world deployments often occur under adverse weather conditions, such as rain, snow, and fog, which can degrade LiDAR measurements to different extents. These conditions introduce noise, sparsity, and structural distortions, leading to significant domain shifts between the pre-built map/reference environment and online observations. As a result, the discriminative ability of learned descriptors deteriorates, causing notable performance degradation. Existing approaches attempt to mitigate this issue through point cloud denoising [1], but they often remove useful structural information and struggle under severe degradation.

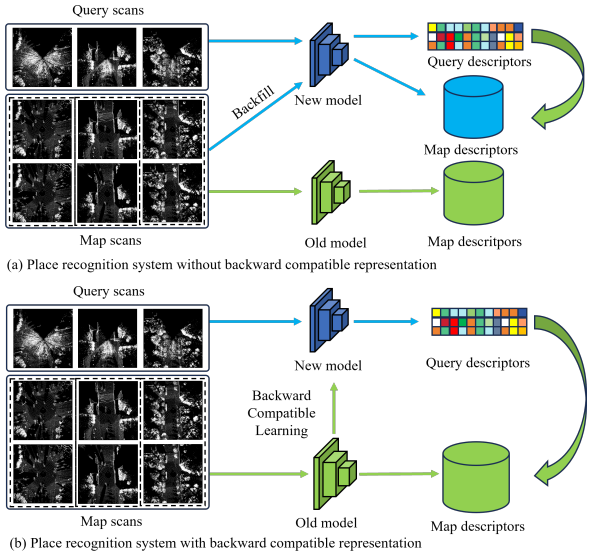

Fig. 1. Comparison of place recognition systems with and without backward-compatible representation. Without backward-compatible representation, updating the place recognition model requires re-extracting descriptors for all previously built maps, since descriptors generated by different model versions are not directly comparable. With backward-compatible learning, query descriptors produced by the updated model can be directly matched with existing map descriptors, eliminating the need for map re-indexing and enabling seamless model upgrades.

A natural solution is to adapt the model during deployment. However, existing adaptation methods [2], [3], [4] mainly focus on improving target-domain performance, but are not explicitly designed to preserve compatibility with previously stored map descriptors. Consequently, model updates may shift the embedding space, making newly extracted query descriptors mismatched with historical map descriptors and requiring costly descriptor recomputation or database re-indexing. As illustrated in Fig. 1(a), this mismatch makes descriptors produced by different model versions not directly comparable, requiring costly recomputation of all historical map features (i.e., backfilling). In contrast, Fig. 1(b) shows that backward-compatible learning enables the updated model to remain aligned with the original embedding space, allowing direct matching with existing descriptors without re-indexing.

To address this limitation, we propose a continual online backward-compatible learning framework for LiDAR place recognition under adverse weather. Our method performs self-supervised online adaptation by leveraging spatial-

[1]Binhong Liu, Qiaoyu Xu, 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.

temporal consistency in the observation stream and pose estimates from a LiDAR localization system to construct supervisory signals. Meanwhile, we explicitly enforce backward compatibility by constraining the updated model to remain consistent with a fixed map embedding space. This design enables the robot to improve place recognition performance online while maintaining compatibility with existing map descriptors, eliminating the need for database re-indexing and enabling a practical *learn-while-deploying* paradigm for long-term operation. The main contributions of this work are summarized as follows:

- We propose a continual online adaptation framework for LiDAR place recognition under adverse weather without requiring ground-truth annotations.
- We introduce a backward-compatible learning paradigm that avoids costly descriptor backfilling during model updates.
- Extensive experiments demonstrate consistent performance improvements under severe domain shifts while preserving compatibility with pre-built maps.

II. Related Work

A. Learning-based LiDAR Place Recognition

Learning-based LiDAR place recognition methods can be broadly categorized into point-based [5], [6], voxel-based [7], [8], and projection-based approaches [9], [10]. Point-based methods, such as PointNetVLAD [5], directly process raw point clouds to extract global descriptors. Voxel-based methods [7], [8] leverage voxelization and 3D convolutions to capture local geometric structures efficiently. Projection-based methods transform point clouds into 2D representations (e.g., BEV or range images) and apply mature CNN architectures, achieving a favorable balance between efficiency and robustness [9], [10], [11]. Despite their effectiveness, these methods generally assume consistent data distributions and struggle under severe domain shifts caused by adverse weather.

B. LiDAR Perception under Adverse Weather

Adverse weather introduces noise, sparsity, and structural degradation in LiDAR data, leading to significant domain shifts [12], [13]. Existing solutions mainly include denoising-based and domain adaptation-based methods. Denoising approaches aim to filter out corrupted measurements but often remove useful structural information, limiting downstream performance. Domain adaptation has been widely studied in 3D perception tasks such as semantic segmentation [14], [15] and object detection [16], especially under sim-to-real gaps [17]. For place recognition, some works explore continual learning to handle multi-domain scenarios [18], [19], [20], while GeoAdapt [4] leverages geometric consistency for self-supervised adaptation, but assumes relatively stable structures and may degrade under severe weather.

C. Backward-Compatible Learning

Backward-compatible learning aims to ensure that updated models remain compatible with previously generated features, avoiding costly descriptor recomputation (backfilling). This problem was first formalized in [21] via the BCT framework, and later extended to more general settings [22]. Some works attempt to relax the compatibility-performance trade-off by expanding embedding dimensions [23], but this often introduces additional computational overhead without guaranteeing effectiveness.

Most existing studies focus on large-scale retrieval tasks with static databases, whereas place recognition systems operate in a continual setting where both models and environments evolve over time. Consequently, model updates can easily break compatibility with historical map descriptors, making repeated feature extraction unavoidable. How to achieve online adaptation while preserving cross-version compatibility remains an open problem, which we address in this work.

III. Problem Formalization and Preliminaries

A. Problem Formalization

In real-world deployments, LiDAR-based place recognition systems are required to operate across multiple environments with continuously evolving conditions, such as rain, snow, and fog. These adverse weather conditions severely degrade point cloud quality, resulting in significant distribution shifts between the pre-training data and online observations. Consequently, a pre-trained model often suffers substantial performance degradation when directly applied to new environments.

To address this challenge, we consider a continual place recognition setting over a sequence of environments $\{\mathcal{E}^n\}_{n=1}^{N}$, where each environment $\mathcal{E}^n$ provides a stream of LiDAR scans $\{\mathbf{S}_t^n\}$. Let f_θ^n denote the model trained on the n-th environment. When entering a new environment $\mathcal{E}^{n+1}$, the model is initialized from f_θ^n and updated online.

For clarity, we denote the model used to construct the map database as the *frozen model* $f_{\text{old}} = f_\theta^n$, and the model being updated online as the *dynamic model* $f_t = f_{\theta_t}^{n+1}$. The frozen model remains fixed and is used to extract map descriptors, while the dynamic model is continuously updated and used to encode query observations.

Each incoming scan $\mathbf{S}_t$ is first converted into a BEV image $\mathbf{I}_t$, and then encoded into a global descriptor:

$$\mathbf{z}_t = f_t(\mathbf{I}_t). \tag{1}$$

In practical systems, a map database $\mathcal{D}$ is constructed offline using the frozen model f_{old} and remains fixed during deployment. During online operation, the dynamic model f_t is used to extract query descriptors. To avoid recomputing all map descriptors (i.e., backfilling), the dynamic model is required to maintain backward compatibility with the frozen model, such that descriptors produced by different model versions remain directly comparable.

B. Online Evaluation for Backward Compatibility

We consider the dynamic model to be backward compatible with the frozen model if the following condition holds:

$$\mathrm{M}\big(f_t, f_{\text{old}}; \mathcal{Q}, \mathcal{D}\big) > \mathrm{M}\big(f_{\text{old}}, f_{\text{old}}; \mathcal{Q}, \mathcal{D}\big), \tag{2}$$

where $\mathcal{Q}$ and $\mathcal{D}$ denote the query set and the map set, respectively. Query descriptors are extracted using the dynamic model f_t, while the map descriptors remain fixed and are computed by the frozen model f_{old}. Here, $\mathrm{M}(\cdot)$ denotes the retrieval performance metric, for which we adopt Recall@1 as the primary evaluation criterion. This criterion indicates that, under fixed query and map sets, the updated model achieves superior retrieval performance without recomputing the map descriptors. Consequently, the new model can directly match against historical map features, eliminating the need for costly backfilling.

IV. Proposed Method

A. Overview

We propose a backward-compatible learning framework for LiDAR place recognition that enables lifelong deployment under adverse weather conditions. The key objective is to continuously adapt the query encoder to new environments while maintaining compatibility with a pre-built database, thereby avoiding costly database re-indexing.

Given a global map, a frozen encoder f_{old} is first used to extract descriptors for all database scans, forming a static descriptor database. During deployment, a dynamic encoder f_t is updated online using streaming LiDAR observations. To ensure stable retrieval, the updated features are constrained to remain aligned with the frozen feature space.

B. Localization Process

Given incoming LiDAR scans and IMU measurements, the system performs localization in a tracking-by-relocalization manner, which consists of odometry estimation, scan-to-map matching, and descriptor-based relocalization.

Odometry and Scan-to-Map Tracking. For each incoming scan $\mathbf{S}_t^n$, we first apply DROR [24] to suppress isolated outliers and estimate the motion using FAST-LIO2 [25]. The estimated odometry pose $\hat{\mathbf{T}}_t^n \in SE(3)$ is then used as the initial guess for NDT-based scan-to-map registration, following the strategy in hdl_localization. Specifically, the refined pose is obtained by solving

$$\mathbf{T}_t^{n*} = \arg\min_{\mathbf{T}\in SE(3)} \mathcal{E}_{\text{NDT}}\left(\mathbf{T}, \mathbf{S}_t^n, \mathcal{M}\right), \tag{3}$$

where $\mathcal{M}$ denotes the global map and $\mathcal{E}_{\text{NDT}}(\cdot)$ is the NDT registration cost between the transformed scan and the map. The optimized transformation $\mathbf{T}_t^{n*}$ provides the refined pose $\mathbf{p}_t^n$, and the final registration cost is used as the matching error:

$$s_t = \mathcal{E}_{\text{NDT}}\left(\mathbf{T}_t^{n*}, \mathbf{S}_t^n, \mathcal{M}\right). \tag{4}$$

A smaller s_t indicates better scan-to-map alignment.

Descriptor Generation. Given a point cloud $\mathbf{S}$, the descriptor generation process consists of two stages: BEV image construction and descriptor extraction. For BEV construction, we first crop a cubic region of size $D \times D \times D$ m centered at the LiDAR origin to focus on the local spatial context. The cropped point cloud is then downsampled using voxelization with resolution r. Next, the space is projected onto the ground plane and discretized into $\frac{2D}{r} \times \frac{2D}{r}$ pillars. For each pillar located at (u, v), the normalized point density is computed as the pixel intensity:

$$\mathbf{I}_{\text{density}}(u, v) = \frac{\min(N_g, N_m)}{N_m}, \tag{5}$$

where N_g denotes the number of points falling into the corresponding pillar, and N_m is a predefined normalization factor to prevent saturation. This results in a density-based BEV image $\mathbf{I}_{\text{density}}$.

The BEV image is then fed into the dynamic model to extract a global descriptor:

$$\mathbf{z}_t = f_t(\mathbf{I}_t), \tag{6}$$

which is used for place recognition and relocalization.

Relocalization Trigger. The system operates in a tracking mode when the scan-to-map matching is reliable. Specifically, if the matching error satisfies $s_t < \tau_{\text{reloc}}$, the estimated pose $\mathbf{p}_t^n$ is accepted as the current pose. Otherwise, tracking is considered unreliable, and the system switches to relocalization mode.

Descriptor-based Relocalization. In relocalization mode, the descriptor $\mathbf{z}_t$ is used to retrieve candidate locations from the map database $\mathcal{D}^n$, which is constructed using the frozen model. The retrieved candidates provide coarse pose hypotheses, which are further refined through geometric verification (e.g., scan-to-map alignment), yielding a relocalized pose estimate.

Discussion. This localization pipeline combines geometric tracking and descriptor-based global retrieval. The matching error provides a principled criterion to switch between tracking and relocalization, ensuring robustness under challenging conditions such as severe noise or adverse weather. The relocalization module relies on descriptor matching against a fixed map database, which motivates the need for backward compatibility to maintain consistent retrieval performance during online model updates.

C. Online Training Process

1) Sample filter: To mitigate noisy updates during test-time adaptation, we introduce a reliability gate for each incoming query frame. For each query, the current model retrieves top-2 candidates from a fixed map database, and the retrieval confidence is quantified by the distance margin between the first and second nearest neighbors. A query is considered reliable only if this margin exceeds a predefined threshold.

In addition, the retrieved positive candidate is required to satisfy a spatial consistency constraint, i.e., its positional distance to the query must be smaller than a predefined threshold:

$$\text{reliable} \iff (d_2 - d_1 \geq \tau_m) \wedge (\|\mathbf{p}_{\text{p}} - \mathbf{p}_q\|_2 \leq r), \tag{7}$$

where d_1 and d_2 denote the distances to the first and second nearest neighbors, respectively. $\mathbf{p}_q$ and $\mathbf{p}_p$ denote the poses of the query and the retrieved positive candidate, respectively.

Only reliable queries are used for gradient updates, while unreliable ones are discarded. This mechanism suppresses

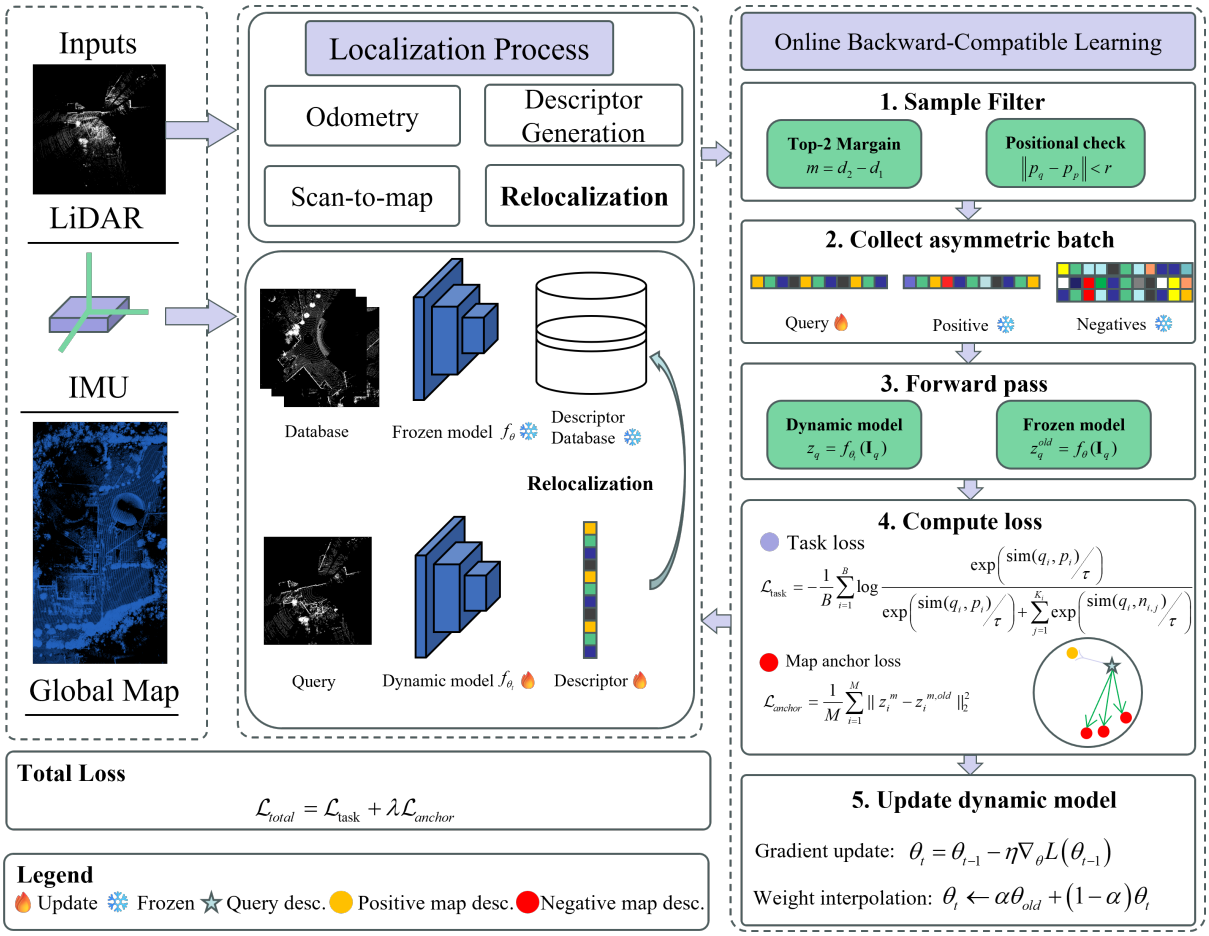

Fig. 2. Overview of the proposed online backward-compatible LiDAR place recognition framework. Given LiDAR and IMU inputs with a pre-built global map, the system first performs localization via odometry, descriptor generation, and scan-to-map matching. A frozen model is used to build a fixed descriptor database for relocalization, while a dynamic model is continuously updated online. During online training, incoming query samples are first filtered using a reliability gate based on top-2 margin and positional consistency. Reliable samples are then used to construct asymmetric triplet batches, where queries are encoded by the dynamic model and positive/negatives are retrieved from the fixed database. The model is optimized using a combination of task loss (InfoNCE) and map anchor loss to ensure discriminative representation while maintaining backward compatibility with the database feature space. Finally, the dynamic model is updated via gradient descent with weight interpolation, and the updated model is fed back into the relocalization process.

low-confidence pseudo-supervision, thereby stabilizing online adaptation and improving robustness under severe domain shifts.

2) Asymmetric Contrastive Learning: To maintain backward compatibility while adapting to new environments, we adopt an asymmetric contrastive learning strategy, as illustrated in Fig. 3. Conventional contrastive learning follows a symmetric formulation (Fig. 3(a)), where query, positive, and negative samples are encoded by the same model, enforcing mutual feature alignment and separation. However, such symmetric optimization may distort the feature space after model updates, breaking compatibility with previously constructed map descriptors.

To address this issue, we propose an asymmetric contrastive learning scheme. The query descriptor is extracted using the dynamic model, while the positive and negative descriptors are retrieved from a fixed map database encoded by the frozen model. As a result, only the query representation is updated, and the map feature space remains unchanged.

Under this formulation, we optimize the following contrastive objective:

$$\mathcal{L}_{\text{task}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(q_i^\top p_i/\tau)}{\exp(q_i^\top p_i/\tau)+\sum_{j=1}^{K_i}\exp(q_i^\top n_{i,j}/\tau)}. \tag{8}$$

Here, $q_i = f_t(\mathbf{I}_i^q)$ denotes the query descriptor extracted by the dynamic model, while $p_i = f_{\text{old}}(\mathbf{I}_i^p)$ and $n_{i,j} = f_{\text{old}}(\mathbf{I}_{i,j}^n)$ are the positive and negative descriptors extracted by the frozen model. The positive sample corresponds to a spatially consistent location, and negatives are sampled from spatially distant locations. K_i is the number of negatives, B denotes the mini-batch size, and τ is a temperature parameter.

Unlike the symmetric formulation, the proposed method performs unidirectional optimization, where only the query features are updated while the map descriptors remain fixed. As shown in Fig. 3(b), the query features are progressively aligned with the fixed map feature space, preserving backward compatibility while improving discriminability under new environmental conditions.

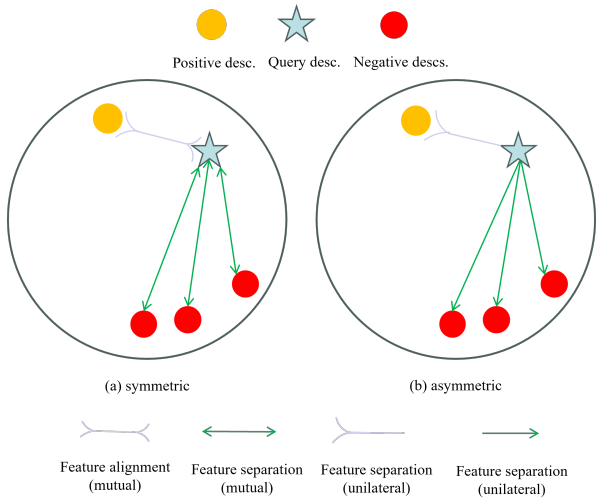

Fig. 3. Comparison between symmetric and asymmetric feature learning paradigms. In the symmetric setting, both query and map descriptors are jointly updated, leading to mutual feature alignment and separation. In contrast, the proposed asymmetric design updates only the query encoder while keeping map descriptors fixed, enforcing unilateral alignment and separation to preserve backward compatibility.

Algorithm 1 Online Localization and Relocalization Process

Input: Incoming LiDAR scan $\mathbf{S}_t$, IMU data, map database $\mathcal{D}$, threshold τ_{reloc}
Output: Reliable pose $\mathbf{p}_t$
Remove noisy LiDAR points from $\mathbf{S}_t$ using DROR [24]
Estimate odometry using FAST-LIO2
Perform scan-to-map matching to obtain pose $\mathbf{p}_t$ and matching error s_t
if $s_t < \tau_{reloc}$ **then**
    Accept $\mathbf{p}_t$ as the current reliable pose
else
    Convert $\mathbf{S}_t$ into a BEV image $\mathbf{I}_t$
    Extract query descriptor $\mathbf{z}_t = f_t(\mathbf{I}_t)$
    Retrieve candidate locations from $\mathcal{D}$ using $\mathbf{z}_t$
    Perform relocalization based on the retrieved candidates
    Update $\mathbf{p}_t$ with the relocalized pose
end if
return $\mathbf{p}_t$

3) Map Anchor Loss: We introduce a map anchor loss to stabilize online adaptation and constrain representation drift. The map anchor loss regularizes the model on map-domain inputs:

$$\mathcal{L}_{\text{anchor}} = \frac{1}{M}\sum_{i=1}^{M}\left\|\mathbf{z}_i^m - \mathbf{z}_i^{m,\text{old}}\right\|_2^2. \tag{9}$$

Since map descriptors are fixed, this term does not modify the reference space but constrains the dynamic model to remain consistent with the frozen model on the map manifold.

From a unified perspective, the map anchor loss acts as a representation-level regularizer that limits deviation from the original embedding manifold. It constrains the dynamic model on map data, thereby balancing adaptation and backward compatibility for robust online learning.

Algorithm 2 Environment-wise Online Backward-Compatible Adaptation

Input: Pretrained model f_θ^1, environment sequence $\{\mathcal{E}^n\}_{n=2}^N$
Parameters: batch size B, thresholds τ_m, r, weight λ
Initialize frozen model $f_{\text{old}} \leftarrow f_\theta^1$
for $n = 2$ to N **do**
    Initialize dynamic model $f_t \leftarrow f_\theta^{n-1}$
    Build map database $\mathcal{D}^n$ using frozen model f_{old}
    Initialize batch $\mathcal{B} \leftarrow \emptyset$
    for each incoming LiDAR scan and IMU data in $\mathcal{E}^n$ **do**
        Estimate pose through Algorithm 1
        Apply reliability gate via Eq. 7
        if reliable **then**
            Sample $(\mathbf{I}_t^q, \mathbf{I}_t^p, \{\mathbf{I}_{t,j}^{neg}\})$ from $\mathcal{D}^n$
            Add sample to $\mathcal{B}$
        end if
        if $|\mathcal{B}| = B$ **then**
            Encode query descriptors using the dynamic model f_t
            Retrieve positive and negative descriptors from $\mathcal{D}^n$
            Compute loss via Eq. 11
            Update f_t using gradient descent
            Apply weight interpolation via Eq. 10
            Reset $\mathcal{B} \leftarrow \emptyset$
        end if
    end for
    $f_\theta^n \leftarrow f_t$ ▷ adapted model
    $f_{\text{old}} \leftarrow f_\theta^n$ ▷ update frozen model
end for
return f_θ^N

4) Weight Interpolation: To mitigate parameter drift during online adaptation, we adopt a weight interpolation strategy that explicitly ties the updated model to a frozen reference model. Given the current parameters θ_t and the reference parameters θ_{old}, the update is defined as:

$$\theta_t \leftarrow (1-\alpha)\theta_t + \alpha\theta_{old}. \tag{10}$$

Unlike conventional continual learning approaches that rely on loss-level regularization (e.g., knowledge distillation [26]), this operation directly constrains the parameter trajectory, preventing the model from deviating excessively from the previously learned solution. As a result, the updated model remains in the vicinity of the original parameter space, which is crucial for maintaining compatibility with precomputed map descriptors.

By applying this interpolation at each update step, we obtain a stable online adaptation process that balances plasticity and stability, while preserving backward compatibility without requiring explicit feature alignment.

5) Overall Objective: The overall objective consists of a task loss and a regularization term:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{anchor}}, \tag{11}$$

where λ controls the strength of regularization.

The task loss $\mathcal{L}_{\text{task}}$ aligns query descriptors produced by the dynamic model with the fixed map descriptors extracted by the frozen model, enabling effective retrieval under domain shifts. In contrast, L_{anchor} acts as a representation-level regularizer, constraining the updated model to remain consistent with the frozen model on map-domain inputs.

Overall, the objective can be interpreted as learning a domain-adapted query encoder that projects new observations into a fixed map embedding space. This asymmetric formulation avoids updating map descriptors while maintaining backward compatibility, thereby enabling efficient online continual learning without costly database recomputation.

V. Experimental Settings

A. Baselines

We compare our method with representative domain adaptation and backward-compatible learning approaches, including AdaBN [3], CoTTA [2], and MixBCT [27].

AdaBN adapts a pre-trained model to the target domain by updating Batch Normalization statistics using target data. While simple and efficient, it only adjusts feature statistics and lacks the ability to handle large domain shifts or maintain discriminative structure.

CoTTA is a continual test-time adaptation method that performs online self-training using augmentation-averaged predictions and employs stochastic parameter restoration to mitigate forgetting. In our setting, we replace classification outputs with retrieval distributions over a fixed legacy database. However, CoTTA relies on pseudo-label quality and is prone to error accumulation under severe domain shifts.

MixBCT is a backward-compatible learning framework that aligns features from old and new models to avoid descriptor recomputation. We adapt it by introducing a compatibility regularization on query embeddings based on legacy feature neighborhoods. Nevertheless, it is primarily designed for offline training and does not explicitly address online adaptation under streaming data.

B. Datasets

We evaluate our method on three large-scale LiDAR datasets: KITTI [28], NCLT [29], and Boreas [30].

KITTI is a standard benchmark for LiDAR-based place recognition and localization, collected in urban, residential, and highway environments using a vehicle-mounted sensor suite. Following [11], we use the first 3000 frames of sequence "00" to train the initial model.

NCLT is a long-term dataset with significant seasonal and environmental variations. We use "2013-02-23" as the query sequence and "2012-01-15" as the map sequence. Following [31], adverse weather is simulated by injecting LiDAR noise and outliers using [32].

Boreas captures large-scale real-world seasonal and weather changes. We use "2021-03-30" (clear weather) as the map sequence and "2021-01-26" (heavy snow) as the query sequence. In the query sequence "2021-01-26", the LiDAR scans inherently exclude approximately 80° of the forward field of view, which further increases the difficulty of place recognition.

C. Backbone

In order to meet the online learning requirements for robotics, we choose LighterBEV [11] 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.

D. Training and Online Evaluation Protocols

Training Protocol. We first train the model offline on the KITTI dataset and then continuously perform online adaptation across the remaining datasets. Specifically, we adopt a 3-step protocol: KITTI $\rightarrow$ NCLT $\rightarrow$ Boreas. When entering a new environment, the model is initialized from the one trained on the previous environment. The frozen model is then used to extract map descriptors for the current environment, which remain fixed during subsequent online adaptation.

Online Evaluation Protocol. During deployment, LiDAR frames are processed sequentially in a streaming manner. At each time step t, the incoming scan is treated as a query and first evaluated before any model update.

Specifically, the query descriptor is extracted by the current dynamic model f_t and matched against the fixed map descriptors precomputed by the frozen model f_{old}. This forms a backward-compatible retrieval setting where query and map descriptors are produced by different models. The retrieval is considered correct if the top-1 candidate lies within a predefined distance threshold r, and is recorded as a true positive (TP).

By accumulating the true positives (TP) and false negatives (FN) over the sequence, the Recall@1 is defined as

$$\text{Recall@1} = \frac{TP}{TP + FN}, \tag{12}$$

where both TP and FN are computed in a streaming manner. Specifically, at each time step t, the retrieval result is evaluated using the dynamic model f_t, and the corresponding TP or FN is recorded. All methods are evaluated under the same online adaptation protocol and share identical supervision signals for fair comparison.

E. Implementation details

We adopt the LighterBEV [11] as our backbone. During online optimization, we use the Adam optimizer with an initial learning rate of 1×10^{-4}, and reset the optimizer at each environment switch to avoid momentum accumulation across domains. We select $K = 3$ negatives per query, with a positive radius $r = 5\,\text{m}$ and a negative radius $d = 10\,\text{m}$.

TABLE I

RESULTS ON CONTINUAL ONLINE ADAPTATION FROM KITTI TO NCLT AND BOREAS. "BACKFILL" DENOTES THE PERFORMANCE WHEN MAP DESCRIPTORS ARE RECOMPUTED USING THE UPDATED MODEL.

Method	NCLT				Boreas			
	frozen	dynamic	gain	backfill	frozen	dynamic	gain	backfill
AdaBN [3]	0.5331	0.5383	+0.0052	0.4405	0.4888	0.0112	-0.4777	0.8628
CoTTA [2]	0.5331	0.1690	-0.3641	0.4058	0.5699	0.0092	-0.5607	0.6963
mixBCT [27]	0.5331	0.7173	+0.1841	0.6315	0.5407	0.4071	-0.1336	0.8565
Ours	0.5331	**0.7749**	**+0.2418**	**0.7561**	**0.5890**	**0.7012**	**+0.1123**	**0.8854**

TABLE II

ABLATION STUDY OF DIFFERENT COMPONENTS IN THE PROPOSED FRAMEWORK ON CONTINUAL ONLINE ADAPTATION FROM KITTI TO NCLT AND BOREAS.

	Configuration	NCLT				Boreas			
		frozen	dynamic	gain	backfill	frozen	dynamic	gain	backfill
(a)	Full pipeline	0.5331	**0.7749**	**+0.2418**	**0.7561**	**0.5890**	**0.7012**	+0.1123	0.8854
(b)	No Sample Filter	0.5331	0.7710	+0.2379	0.6875	0.4649	0.6133	**+0.1484**	**0.8900**
(c)	No Weight Interpolation	0.5331	0.7089	+0.1758	0.6789	0.2814	0.3943	+0.1129	0.8795
(d)	No Map Anchor Loss	0.5331	0.7454	+0.2123	0.6669	0.4718	0.4202	-0.0515	0.8677
(e)	Asymmetric only	0.5331	0.7324	+0.1993	0.5636	0.4990	0.2896	-0.2095	0.5811

VI. EXPERIMENTAL RESULTS

A. Continual Online Adaptation Results

Table I reports the results of continual online adaptation from KITTI to NCLT and Boreas. We evaluate Recall@1 under four settings: *frozen* (before adaptation), *dynamic* (during online adaptation), *gain*, and *backfill*. Here, *backfill* denotes the performance obtained by recomputing map descriptors with the updated model, removing the backward compatibility constraint at the cost of significant computational overhead.

Overall, AdaBN provides only marginal improvement on NCLT and degrades severely on Boreas, indicating that adapting only feature statistics is insufficient under large domain shifts. CoTTA exhibits unstable behavior across both datasets, with significant performance drops in the online setting (-0.3641 on NCLT and -0.5607 on Boreas), due to the lack of backward compatibility constraints and resulting feature drift. MixBCT achieves noticeable improvement on NCLT (+0.1841), showing the benefit of compatibility constraints, but fails to maintain performance on Boreas (-0.1336), as it is primarily designed for offline training.

In contrast, our method consistently achieves the best performance across all settings, with substantial gains in the dynamic setting (+0.2418 on NCLT and +0.1123 on Boreas), and is the only method that maintains positive improvement under strong domain shifts. These gains are obtained in a fully online setting, where the model is continuously updated during deployment. Our method also achieves the highest backfill performance (0.7561 on NCLT and 0.8854 on Boreas), indicating strong representation capability. Notably, on NCLT, the dynamic performance slightly exceeds the backfill result, which may be attributed to the asymmetric contrastive learning mechanism that explicitly aligns query descriptors with the fixed map feature space.

Although backfilling improves performance, it incurs significant computational cost. In contrast, our method achieves competitive dynamic performance without reindexing. Compared to backfilling, our method avoids recomputing all map descriptors, significantly reducing computational cost during deployment, demonstrating its practicality for long-term systems.

B. Ablation Study

Table II presents ablation results under different configurations. The full model (a) achieves the best performance across both datasets, validating the effectiveness of the complete design.

Removing the sample filter (b) slightly degrades performance on NCLT while improving gain and backfill on Boreas, suggesting that although more samples provide stronger short-term signals, the lack of filtering introduces noisy updates and reduces stability. Removing weight interpolation (c) leads to noticeable degradation on both datasets, indicating its importance in stabilizing online updates and mitigating model drift. Removing the map anchor loss (d) significantly degrades performance on Boreas and results in negative gain, highlighting the importance of backward compatibility under strong domain shifts. Finally, using only asymmetric learning (e) further degrades performance, especially on Boreas, showing that additional regularization is necessary for stable adaptation.

Overall, each component contributes to a different aspect of adaptation: the sample filter improves robustness, weight interpolation stabilizes learning, and the map anchor loss enforces backward compatibility. Their combination enables effective and stable online adaptation under challenging cross-domain conditions.

VII. CONCLUSIONS

In conclusion, we propose a continual online backward-compatible learning framework for LiDAR place recognition under adverse weather. By combining asymmetric learning and map anchor regularization, the model adapts online while remaining compatible with pre-built map descriptors. The proposed framework is tightly integrated with a LiDAR-based localization system, where pose estimates are leveraged to construct reliable self-supervised signals for online adaptation. Experiments demonstrate improved performance without backfilling, enabling practical long-term deployment.

REFERENCES

[1] X. Zhao, C. Wen, X. Zhu, Y. Wang, H. Bai, and W. Dou, "Triplemixer: A triple-domain mixing model for point cloud denoising under adverse weather," *IEEE Transactions on Image Processing*, vol. 34, pp. 7712–7727, 2025.

[2] Q. Wang, O. Fink, L. Van Gool, and D. Dai, "Continual test-time domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7201–7211.

[3] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting batch normalization for practical domain adaptation," *arXiv preprint arXiv:1603.04779*, 2016.

[4] J. Knights, S. Hausler, S. Sridharan, C. Fookes, and P. Moghadam, "Geoadapt: Self-supervised test-time adaptation in lidar place recognition using geometric priors," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 915–922, 2024.

[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] 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.

[10] 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.

[11] B. Liu, T. Yang, H. Cao, S. Fu, Y. Fang, and Z. Yan, "Lighterbev: Lidar global localization meets online learning," *IEEE Robotics and Automation Letters*, vol. 11, no. 2, pp. 1170–1177, 2026.

[12] 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.

[13] J. Chang, R. Hu, F. Huang, D. Xu, and L.-T. Hsu, "Lidar-based ndt matching performance evaluation for positioning in adverse weather conditions," *IEEE Sensors Journal*, vol. 23, no. 20, pp. 25 346–25 355, 2023.

[14] L. Yi, B. Gong, and T. Funkhouser, "Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15 363–15 373.

[15] C. Saltori, F. Galasso, G. Fiameni, N. Sebe, E. Ricci, and F. Poiesi, "Cosmix: Compositional semantic mix for domain adaptation in 3d lidar segmentation," in *European conference on computer vision*. Springer, 2022, pp. 586–602.

[16] J. Yang, S. Shi, Z. Wang, H. Li, and X. Qi, "St3d: Self-training for unsupervised domain adaptation on 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 10 368–10 378.

[17] S. e. a. Huch, "Quantifying the lidar sim-to-real domain shift: A detailed investigation using object detectors and analyzing point clouds at target-level," *IEEE Transactions on Intelligent Vehicles*, 2023.

[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] 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.

[20] 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.

[21] Y. Shen, Y. Xiong, W. Xia, and S. Soatto, "Towards backward-compatible representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6368–6377.

[22] B. Zhang, Y. Ge, Y. Shen, S. Su, F. Wu, C. Yuan, X. Xu, Y. Wang, and Y. Shan, "Towards universal backward-compatible representation learning," *arXiv preprint arXiv:2203.01583*, 2022.

[23] Y. Zhou, Z. Li, A. Shrivastava, H. Zhao, A. Torralba, T. Tian, and S.-N. Lim, "Bt^ 2: Backward-compatible training with basis transformation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 11 229–11 238.

[24] M. H. Prio, S. Patel, and G. Koley, "Implementation of dynamic radius outlier removal (dror) algorithm on lidar point cloud data with arbitrary white noise addition," in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, 2022, pp. 1–7.

[25] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[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] Y. Liang, Y. Zhang, S. Zhang, Y. Wang, S. Xiao, R. Xiao, and X. Wang, "Mixbct: Towards self-adapting backward-compatible training," *arXiv preprint arXiv:2308.06948*, 2023.

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

[29] 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.

[30] K. Burnett, D. J. Yoon, Y. Wu, A. Z. Li, H. Zhang, S. Lu, J. Qian, W.-K. Tseng, A. Lambert, K. Y. Leung, *et al.*, "Boreas: A multi-season autonomous driving dataset," *The International Journal of Robotics Research*, vol. 42, no. 1-2, pp. 33–42, 2023.

[31] W. Kuang, X. Zhao, Y. Shen, C. Wen, H. Lu, Z. Zhou, and X. Chen, "Reslpr: A lidar data restoration network and benchmark for robust place recognition against weather corruptions," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025, pp. 12 494–12 501.

[32] M. Hahner, C. Sakaridis, D. Dai, and L. Van Gool, "Fog Simulation on Real LiDAR Point Clouds for 3D Object Detection in Adverse Weather," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.