Person Re-Identification for Robot Person Following with Online Continual Learning

Hanjing Ye[1], Jieting Zhao[1], Yu Zhan[1], Weinan Chen[2], Li He[1] and Hong Zhang[1*], *Fellow IEEE*

***Abstract*— Robot person following (RPF) is a crucial capability in human-robot interaction (HRI) applications, allowing a robot to persistently follow a designated person. In practical RPF scenarios, the person can often be occluded by other objects or people. Consequently, it is necessary to re-identify the person when he/she reappears within the robot's field of view. Previous person re-identification (ReID) approaches to person following rely on a fixed feature extractor. Such an approach often fails to generalize to different viewpoints and lighting conditions in practical RPF environments. In other words, it suffers from the so-called domain shift problem where it cannot re-identify the person when his re-appearance is out of the domain modeled by the fixed feature extractor. To mitigate this problem, we propose a ReID framework for RPF where we use a feature extractor that is optimized online with both short-term and long-term experiences (i.e., recently and previously observed samples during RPF) using the online continual learning (OCL) framework. The long-term experiences are maintained by a memory manager to enable OCL to update the feature extractor. Our experiments demonstrate that even in the presence of severe appearance changes and distractions from visually similar people, the proposed method can still re-identify the person more accurately than the state-of-the-art methods.**

I. INTRODUCTION

Robot person following (RPF) [1] serves as an essential function in many HRI applications, enabling a robot to follow a specified person autonomously. However, the person being followed may become occluded in various situations, such as when other objects or people obstruct the view of the robot in the working environment. Therefore, it is crucial to re-identify the person when he re-appears in the view.

Existing RPF systems can be achieved through two steps: *identify* and *follow*. In the *identify* step, the system performs tracking and possibly ReID to locate the target person, while the *follow* step involves planning and executing the control of the robot to maintain the desired relative position with the target person. In this paper, we focus on the ReID aspect, specifically re-identifying the target person after occlusion. Existing ReID methods for RPF describe a person's appearance either with hand-crafted features [2], [3] or with learned features [4]. However, these methods may experience poor generalization when the features are not sufficiently discriminative for re-identifying the person. Some methods [5], [6] update the tracker online with newly acquired observations of the target person to distinguish the person from the background and other distracting individuals. Such solutions usually do not consider the appearance of a person explicitly, leading to suboptimal ReID performance. To improve the generalization ability, one possible solution is to train the feature extractor online with the most recently observed samples, i.e., short-term experiences. However, we found this would result in limited discriminative ability when the re-appearance of the target person is out of the learned domain represented by the short-term experiences. All these problems are commonly known as domain drift [7].

Fig. 1. Robot person following with online continual learning. To this end, long-term and short-term experiences are utilized to optimize the feature extractor online to represent the discriminative appearance of the target person.

To solve the above problems, we propose to utilize long-term experiences in addition to short-term ones to optimize the feature extractor online for representing the target person's discriminative appearance. Specifically, we approach the person ReID in RPF as a problem of online continual learning (OCL) [7], which aims to learn the newest knowledge without forgetting long-term experiences using a size-limited long-term memory. This idea has shown promising results in existing works on dense mapping [8] and place recognition [9]. For example, IMap [8] incrementally learns a NeRF-based dense map by replaying images and poses from a sparse keyframe set, where camera poses are estimated through the tracking process. Similarly, BioSLAM [9] constructs a discriminative long-term memory to replay point clouds and positions for learning a life-long place recognition network, where positions are obtained via LiDAR odometry.

To develop a ReID framework for RPF that capitalizes on long-term experiences, we established a long-term memory module designed to archive key historical samples, chosen via a loss-guided keyframe selection method. By integrating these long-term samples with short-term data, we optimize the feature extractor online to maintain a comprehensive understanding of the target person, bridging past and present knowledge. Additionally, we apply these optimized features

[1]Hanjing Ye, Jieting Zhao, Yu Zhan, Li He and Hong Zhang are with Shenzhen Key Laboratory of Robotics and Computer Vision, Southern University of Science and Technology. [2]Weinan Chen is with Guangdong University of Technology. *corresponding author (hzhang@sustech.edu.cn).

Our code, video and appendix are available at https://sites.google.com/view/oclrpf.

to train a ridge-regression-based classifier for accurate target recognition. Lastly, a ReID lifecycle management is implemented to form a complete ReID solution. In our experiments, the RPF system with our ReID method can reliably re-identify and follow the target person even in situations with visually similar distracting people and different appearances after occlusion.

II. Related Work

A. Person ReID in Robot Person Following

Person ReID is crucial for RPF, which helps re-identify the target person after occlusion. Existing ReID methods in RPF usually describe the appearance of the target person with hand-crafted features [2], [3] or learned features [4]. Examples of hand-crafted features include geometric attributes [2], and characteristics like height, gait and clothing color [3]. Alternatively, ReID can rely on features learned from a ReID dataset. For example, [4] trains a convolutional neural network (CNN) using a custom-built, small-scale ReID dataset and then extracts features from the low-level response maps of the CNN. Often, these features are further utilized to construct a target classifier with short-term experiences.

The above methods, however, often fail to re-identify the person in complicated RPF situations because features from a fixed feature extractor have a limited capability to generalize to different viewpoints and lighting conditions in practical RPF environments. To mitigate this generalization problem, we optimize the feature extractor online with the short-term experiences used to construct the target classifier in the above methods. However, optimized features are still not discriminative enough to recognize the target person, especially when the re-appearance of the target person is out of the learned domain. These are commonly referred to as domain drift [7]. To mitigate these problems, we propose to utilize OCL techniques to collect valuable long-term experiences. These experiences, in addition to short-term ones, are used to optimize ReID features, thereby improving the ReID performance of the RPF system.

B. Person ReID in Computer Vision

Person ReID has been a prominent research area in computer vision, primarily identifying individuals in video surveillance systems [10]. Various methods have been proposed to solve the ReID problem. For instance, [11] introduces a hand-crafted feature that combines eight color channels (RGB, HSV, and YCbCr) and 19 texture channels to achieve viewpoint invariance. Another approach [12] involves using attribute-based features to achieve competitive ReID performance. However, in recent years, with the advancement of deep learning techniques, learned features [13] have become dominant in ReID research due to their end-to-end nature and excellent generalization. Notably, [14] proposes a CNN-based ReID method that effectively models complex photometric and geometric transformations. However, ReID with a global CNN feature can introduce distractive information in case of occlusion, posing a challenge in real-world scenarios.

To address the issue of occlusion, researchers have introduced ReID methods [15], [16] that leverage pre-defined or learned part masks to match features defined with respect to parts of a target person. Considering that an occluded human body is frequently encountered in RPF scenarios, one can use part-guided ReID features to describe a person's appearance. Still, as mentioned before, these features from a fixed feature extractor have a limited generalization ability in practical RPF environments with different viewpoints and lighting conditions. To solve the generalization problem, a similar approach to ours is the memory-based ReID [17]–[19], which tackles unsupervised domain adaptation by transferring knowledge from a labeled source domain to an unlabeled target domain. However, deploying these methods poses significant challenges in the context of RPF due to the demands for extensive iterative training and substantial memory storage. In contrast, our approach enables the optimization of the feature extractor and the re-identification of the target person in real time, even on onboard devices. In this paper, we are the first to explore pre-trained deep features from the computer vision community for forming a complete RPF-task-driven target-ReID framework.

C. Online Continual Learning

OCL addresses the challenge of learning from a non-independent and identically distributed (Non-IID) stream of data in an online manner, with the objective of preserving and extending historical knowledge [7]. The Non-IID data setting aligns with the observation scenario of our RPF system, in which the appearance of an observed individual significantly varies due to complex backgrounds and the motion of the robot and target.

Recent works in OCL can be categorized into three main families: regularization-based, parameter-isolation-based and memory-replay-based methods. Regularization-based methods [20], [21] preserve knowledge by adding history-related constraints to the loss function during current task training, thereby balancing the loss gradient direction for old and new knowledge. However, these methods face challenges in finding the desired global optima, making it difficult to strike a balance between both types of knowledge. Parameter-isolation-based methods [22], [23] retain old knowledge by freezing the related parts of the model and only allowing the remaining parts to learn new knowledge. However, these methods are limited by the initial model capacity and require significant training time to achieve good performance. Memory-replay-based methods [24]–[26] utilize memory replays to learn old knowledge incrementally. Examples include Reservoir [26], which randomly forgets samples based on a distribution related to observation times, MIR [24], which randomly updates the memory and retrieves "the hardest" samples for model updating, and ASER [25], which utilizes an Adversarial Shapley value scoring method for memory retrieval to preserve latent decision boundaries for previously observed samples.

Recently, the benefits of memory-replay-based OCL have been demonstrated in several works [8], [9] to enhance the

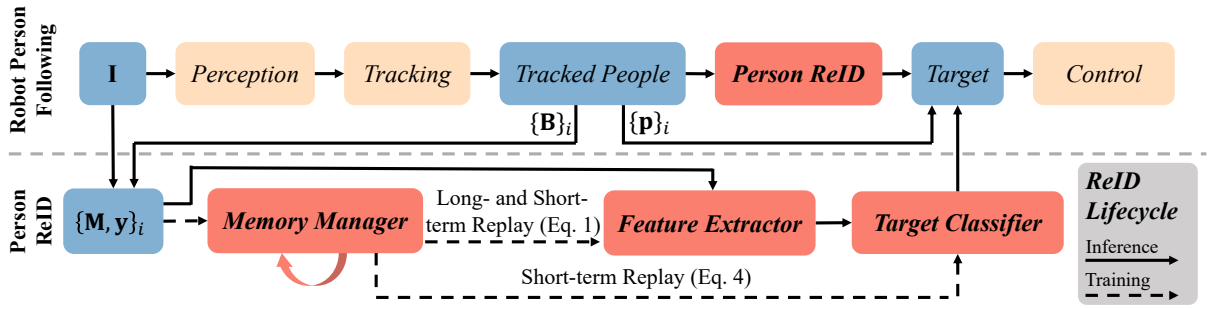


Fig. 2. The top part is the pipeline of our RPF system and the bottom part is the proposed person ReID framework. We obtain image patches $\{\mathbf{M}\}_i$ of the tracked people using the current image $\mathbf{I}$ and their bounding boxes $\{\mathbf{B}\}_i$. When the target person is consistently tracked, his label $\mathbf{y}$ represents positives and other people are negatives. Afterward, we add $\{\mathbf{M}, \mathbf{y}\}_i$ to the *memory manager* for memorization. Additionally, these patches are fed into the *feature extractor* to extract ReID features. These features are utilized by the *target classifier* to estimate the target confidence. If the target confidence is greater than a threshold, the corresponding position $\mathbf{p}$ is designated as the target position. In addition to the inference above process, the *memory manager* simultaneously replays long-term and short-term experiences to train the *feature extractor*. Meanwhile, the *target classifier* is trained with short-term experiences. If the target person is not found among the tracked individuals, the training process pauses, and all observations $\{\mathbf{M}, \mathbf{y}\}_i$ become candidates for re-identification. The above training and inference processes are managed by the *ReID lifecycle*.

perception ability of robot systems. Therefore, we adopt a memory-replay-based algorithm in the implementation of our RPF system, although our solution is not limited to any particular OCL algorithm. To the best of our knowledge, we are the first to integrate the OCL concept into an RPF system to optimize the feature extractor incrementally from both long-term and short-term experiences.

III. Method

A. Problem Statement and Overview

Our RPF system is an extension of our previous work [27], represented by the top half of Fig. 2. Our previous RPF system allows for accurate tracking of individuals, even in scenarios with partial occlusion. It first tracks multiple people and then identifies the target person to follow by selecting the corresponding identity (ID). However, when the target person undergoes occlusion and disappears from the camera view, his ID may be removed because no observation is associated with the ID. Therefore, re-identifying the target person after occlusion, either momentarily or over a long time, becomes crucial. To solve this problem in our current work, we introduce a person ReID process, which is performed by the module in the lower half of Fig. 2. In this ReID module, the *feature extractor* and the *target classifier* are optimized when the target person can be correctly identified from tracked people. Later, if and when a long-time occlusion occurs, the optimized models are utilized to re-identify the target person among all the tracked people.

In each ReID period, we capture image patches $\{\mathbf{M}\}_i$ of the tracked individuals using the current image $\mathbf{I}$ and their corresponding bounding boxes $\{\mathbf{B}\}_i$. When the target person is consistently tracked, his label $\mathbf{y}$ represents a positive sample, while labels for other people are negatives. Subsequently, these patches $\{\mathbf{M}\}_i$ are fed into the *feature extractor* for extracting ReID features (Sec. III-B) and these features are further utilized by the *target classifier* to estimate the target confidence (Sec. III-C). If the target confidence is greater than a threshold, the corresponding position $\mathbf{p}$ is designated as the target position.

In addition to the inference process mentioned above, we add $\{\mathbf{M}, \mathbf{y}\}_i$ to the *memory manager* (Sec. III-D) for performing memory-replay-based OCL. Specifically, the *feature extractor* is incrementally optimized with both long-term and short-term experiences ($m_{\text{lt}} \cup m_{\text{st}}$) in an OCL manner through Eq. 1. Besides, the *target classifier* is trained with short-term experiences m_{st} through Eq. 4. If the target person is not found among the tracked people, the training process pauses, and all observations $\{\mathbf{M}, \mathbf{y}\}_i$ become candidates for re-identification. The above training and inference processes are managed by the *ReID lifecycle* detailed in Algorithm I in APPENDIX-D. Except for the memory-replay-based OCL, we name this ReID framework as **RPF-ReID**, which is a complete RPF-task-driven target-ReID module based on pre-trained deep features from the computer vision community.

B. Feature Extractor

We use a feature-based neural network to extract a person's appearance features. Given an image $\mathbf{I}$ and a person's bounding box $\mathbf{B}$, we extract his image patch, denoted as $\mathbf{M}$. Subsequently, we fine-tune a feature extractor f, a ResNet pre-trained on MOT16 ReID [28], which extracts local features from $\mathbf{M}$. To represent partially visible human bodies, we further transform these local features into features associated with the body parts [27]. These features are denoted as $\mathbf{F} \in \mathbb{R}^{N \times C}$, where N represents the number of body parts and C is the size of the feature dimension. Besides, a visibility indicator v_i with $i \in \{1, \cdots, N\}$ is defined and set to 1 if the i_{th} body part is visible and 0 otherwise.

In previous RPF works [2]–[4], the feature extractor is trained offline and fixed under the assumption of independent and identically distributed (IID) observations, i.e., the training and testing scenarios are assumed to be IID. However, this assumption may not be valid in an application such as our RPF. For instance, it may not hold when the target person's appearance is non-discriminative in the predefined feature space. One possible solution is to utilize short-term experiences to fine-tune the feature extractor online. However, optimized features are still not discriminative

enough to recognize the target person. These two problems are commonly referred to as domain drift [7] and can be observed from Fig. 3 (a) and (b), respectively. Due to domain drift problems, the resulting ReID features fail to distinguish the target person from others across the observed samples in the sequence.

To address these problems, we adopt the concept of OCL [7]. Instead of utilizing a fixed feature extractor, we continually fine-tune the feature extractor with both long-term and short-term experiences. Due to the requirement of efficient learning, OCL demands that the model is trained with only one limited batch at a time, and other batches are not included. In addition, OCL requires that the batch should contain current and historical samples. Therefore, we typically maintain a long-term memory, denoted as $\mathbb{L}$, to store a subset of historical samples. In every ReID period, $\mathbb{L}$ replays only one batch, denoted as $m_{\mathrm{lt}} \subset \mathbb{L}$. Besides, the most recent observed K samples, denoted as m_{st}, are included to represent the current knowledge. Our OCL formulation thus can be represented as follows:

$$\arg\min_{\theta_f} \sum_{(\mathbf{M},\mathbf{y})\in\{m_{\mathrm{st}}\cup m_{\mathrm{lt}}\}} \mathbb{E}_{(\mathbf{M},\mathbf{y})}[\mathcal{L}_F(f(\mathbf{M};\theta_f),\mathbf{y})], \quad (1)$$

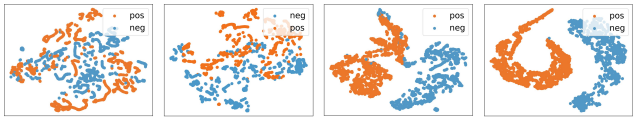
where $\mathbf{M}$ represents a person's image patch and $\mathbf{y}$ for label. f is the feature extractor to be learned, θ_f is the parameter of f, and $\mathcal{L}_F$ is the loss function. In this work, inspired by [15], a representation that is robust to occlusions is learned by employing a mixed loss approach. This approach combines cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$ with part triplet loss $\mathcal{L}^{\mathrm{parts}}_{\mathrm{triplet}}$, formulated as follows:

$$\mathcal{L}_F = \sum_{i\in\{g,c\}} \mathcal{L}_{\mathrm{CE}}(h_i(\mathbf{F}),\mathbf{y}) + \mathcal{L}^{\mathrm{parts}}_{\mathrm{triplet}}(\mathbf{F},\mathbf{y}), \quad (2)$$

where $\mathbf{F}$ denotes the part features. The cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$ focuses on optimizing the feature extractor to accurately predict the person's identity $\mathbf{y}$ from each holistic feature h_g, h_c. The global feature $h_g \in \mathbb{R}^C$ is obtained through global average pooling, while the concatenated feature $h_c \in \mathbb{R}^{(C\cdot N)}$ is derived by concatenating N part features. Moreover, the part triplet loss $\mathcal{L}^{\mathrm{parts}}_{\mathrm{triplet}}$ considers a triplet comprising a query sample, the hardest positive, and the hardest negative. The hardest positive is identified as the positive sample that is furthest from the query, based on the average distance across part features, calculated as $d^{ij}_{\mathrm{parts}} = \frac{1}{N}\sum_{k=1}^{N} ||\mathbf{F}^i_k - \mathbf{F}^j_k||_2$. At the same time, the hardest negative is selected as the nearest negative sample to the query.

By continually learning from these experiences ($m_{\mathrm{lt}} \cup m_{\mathrm{st}}$), the feature extractor incrementally acquires current knowledge while retaining previous experiences. This can be demonstrated by Fig. 3 (c), which shows that training the feature extractor in an OCL manner leads to the target person's features being distinguishable from others throughout the observed samples in the sequence. This incremental learning ability enables the robot to re-identify the target person if their re-appearance exists in the previous experiences.

After feature extraction, previous works [13]–[15] usually achieve ReID by averaging the similarities of features across all query-gallery pairs, assuming that the query feature and the gallery features are strictly in the same feature space. This requires one to re-extract features with the latest feature extractor from all samples in the memory buffer. However, for the purpose of effective RPF, this approach is not feasible due to the large size of our long-term memory. Therefore, to ensure efficient ReID processing, we leverage short-term experiences to train a classifier (Sec. III-C).



(a) W/o S&L (b) W/ S only (c) W/ S&L (d) Ideal

Fig. 3. Feature distribution of the target person (positive) and other distracting people (negative) across all observed samples at the end of the sequence. "S" represents short-term experiences and "L" for long-term ones. (a) Pre-trained features without any online optimization. (b) Trained features with online optimization using short-term experiences only. (c) Trained features using both short-term and long-term experiences within our framework. (d) Ideal feature distribution where features are optimized offline through extensive iterative training.

C. Target Classifier

We train a target classifier g using short-term experiences m_{st}, representing the latest knowledge about the target person. Here, we employ the ridge regression (RR) model with L2 regularization as our classifier, although any other classifiers that are capable of efficient optimization and inference can also be employed. Specifically, we train N RR models where each model is represented as $\mathbf{W}_i \in \mathbb{R}^{1\times C}$ corresponding to a part-level classifier. The target confidence s is estimated by averaging the outputs from all part-level classifiers:

$$s = \frac{\sum_{i=1}^{N} v_i \mathbf{W}_i \mathbf{F}_i^T}{\sum_{i=1}^{N} v_i}, \quad (3)$$

where v_i is the visible indicator and $\mathbf{F}_i \in \mathbb{R}^{1\times C}$ represents the i_{th} part feature of $\mathbf{F}$ where $\mathbf{F} = \bar{f}(\mathbf{M})$. Each RR model $\mathbf{W}_i$ is optimized with the most recent K features extracted from m_{st}:

$$\arg\min_{\mathbf{W}_i} \left\|\mathbf{W}_i\mathbf{X}_i^T - \mathbf{y}\right\|_2^2 + \lambda\left\|\mathbf{W}_i\right\|_2^2, \quad (4)$$

where $\mathbf{X}_i = \{\mathbf{F}_i^1, \mathbf{F}_i^2, \ldots, \mathbf{F}_i^K\} \in \mathbb{R}^{K\times C}$ represents the features of the i_{th} part. $\mathbf{y}$ indicates the labels and λ is a regularization parameter. The optimal solution, which is obtained using linear least squares, is given by $\mathbf{W}_i^* = (\mathbf{X}_i^T\mathbf{X}_i + \lambda\mathbf{I})^{-1}\mathbf{X}_i^T\mathbf{y}$.

This formulation can efficiently regress the classification boundary since the sizes of both the short-term memory and feature dimensions are small. Furthermore, it can also generalize to distinguish historical samples, although the classifier is trained on short-term experiences only. This is because optimized features are discriminative enough to establish a clear classification boundary with a few samples. This can be observed from Fig. 3 (c) and further verified in the experiments.

D. Memory Manager

To leverage long-term experiences to mitigate domain drift problems, we establish a long-term memory denoted as $\mathbb{L}$, responsible for storing valuable samples, i.e., pairs of image patches and labels. When presented with a new sample, the memory manager employs a *keyframe selection* strategy to decide whether to add this sample to the memory buffer. Once the buffer reaches its capacity, *memory consolidation* takes effect to create space by purging certain samples. In addition to the sample insertion and removal, the process of selecting samples for replay during model optimization (Eq. 1) is equally important and is overseen by the *memory replay* mechanism. In the following, we will introduce our *keyframe selection* strategy, as well as the *memory replay and consolidation* processes.

1) Keyframe Selection: Adding the newest sample directly to $\mathbb{L}$ may not be appropriate because the appearance of the target person in adjacent frames is often similar, and therefore, it may not provide additional information. Since information in images is temporally correlated and therefore highly redundant, we insert a keyframe to $\mathbb{L}$ only if it is informative. To this end, inspired by [8], we employ a *loss-guided* keyframe selection strategy to assess the significance of the incoming sample. Specifically, every time a new target sample is added to $\mathbb{L}$, and the feature extractor is optimized, we save a duplicate of the latest feature extractor f and record the loss from this optimization as l_t. The subsequent sample $\{\mathbf{M}, \mathbf{y}\}_{id}$ will then be used to optimize the duplicated f. If the optimization loss is larger than the previous loss l_t by a margin, this sample will be added to $\mathbb{L}$. This process can be expressed as:

$$\delta = \mathcal{L}_F(f(\mathbf{M}), \mathbf{y}) - l_t, \tag{5}$$

if $\delta > \delta_l$, the sample is added to $\mathbb{L}$, indicating that the forthcoming sample contributes additional information to the learned feature extractor.

2) Memory Replay and Consolidation: To preserve valuable experiences, we follow the standard technique of memory replay in OCL to replay samples for our feature extractor learning or consolidate the memory by removing non-informative samples. Specifically, the consolidation process is triggered when $\mathbb{L}$ is full. For example, Reservoir [26] adds a sample to $\mathbb{L}$ with a probability of $|\mathbb{L}|/n$, where $|\mathbb{L}|$ is the size of the long-term memory and n is the total number of observed samples. This is executed as follows: $\mathbb{L}[i] \leftarrow \{\mathbf{M}, \mathbf{y}\}$ if $i < |\mathbb{L}|$, with $i = \text{randint}(0, n)$. This approach inherently reduces the likelihood of later samples being sampled. Differing from a selection based on sequential observation, BioSLAM [9] chooses to remove non-discriminative samples from long-term memory via online clustering. In this work, we demonstrate that by leveraging existing OCL techniques, we can effectively address the issue of forgetting and enhance the person ReID capability. Besides, the memory management module itself does not solve the domain drift problem but is axillary to the feature extractor that directly tackles the domain drift problem.

Our framework optimizes the feature extractor and the classifier upon successful identification of the target person, identification recognized when the target id exists within the tracked individuals and the target confidence s surpasses the threshold δ_{sw}. When the target person is lost, the algorithm re-identifies him from all observed individuals. An individual is considered the target person if his estimated confidence has surpassed a threshold δ_{sw} for consecutive ζ_{reid} frames.

IV. Experiments

A. Experimental Setup

To ensure fair and consistent evaluation in terms of target-person-tracking ability for RPF, previous RPF works (such as [4], [5], [27]) typically assess person-following performance by evaluating person-tracking performance on a robot-centric following dataset.

1) Dataset: We conduct experiments on a public dataset [5] and a custom-built dataset. Both datasets consist of image sequences with the ground truth provided in the form of bounding boxes around the target person. The public dataset includes challenging scenarios such as quick multi-people-crossing, illumination changes, and appearance variations. However, this public dataset lacks scenarios that require person ReID, such as occlusion and similar appearances of distracting people. To address this limitation, we created a custom dataset that includes these challenging scenarios. The custom dataset comprises four sequences named *corridor1*, *corridor2*, *lab-corridor*, and *room*.

2) Baselines: To verify the effectiveness of the proposed RPF framework in terms of target-person-tracking ability, we first compare it with some popular one-stage baselines. With frame input, one-stage methods directly output the target person's bounding box (e.g., SiamRPN++ [30] and STARK [31]) or camera movement (e.g., Zhong's Method [29]). Moreover, we compare it with some people trackers (e.g., SORT [32], OC-SORT [33], and ByteTrack [34]), which can track people's bounding boxes based on motion models. However, these trackers alone cannot handle the target-person-tracking situation and are auxiliary to a target-ReID module for re-identifying the target person after long-term occlusion. Therefore, a complete RPF system is formed by combining each of these people trackers with our ReID module. Our complete RPF system includes ByteTrack [34] and the OCL-assisted RPF-ReID module, labeled as "**ByteTrack + RPF-ReID + OCL**" as shown in Table I.

To investigate the impact of different methods of memory consolidation (as illustrated in Sec. III-D) on person ReID ability, we conduct experiments involving three methods: BioSLAM [9], MIR [24], and Reservoir [26]. These methods are employed to assess whether any form of memory consolidation can enhance the performance of person ReID. Experimental analysis is shown in Sec. IV-C.

B. Evaluation of our RPF system

1) Metric: The evaluation metric of person tracking relies on those employed in previous RPF studies [4], [5], [27]. We assess tracking performance in the image space using

TABLE I. Success rate of person tracking (%) of the baseline and our method in the custom-built dataset† and the public dataset [5]. Our complete RPF system achieves the highest success rate due to the effective ReID performance of our OCL-assisted RPF-ReID module.

Methods	**Success Rate (%)**				
	corridor1†	***corridor2***†	***lab-corridor***†	***room***†	***public dataset*** [5]
Zhong's Method [29]	63.8	66.8	75.8	44.7	75.8
SiamRPN++ [30]	44.8	55.9	46.1	42.6	93.6
STARK [31]	44.3	83.8	73.1	65.8	96.5
SORT [32] + RPF-ReID	67.3	37.9	31.1	82.4	96.1
OC-SORT [33] + RPF-ReID	67.3	37.9	31.1	82.4	96.1
ByteTrack [34] + RPF-ReID	69.1	20.2	54.2	82.4	96.3
ByteTrack + RPF-ReID + OCL	**93.5**	**94.9**	**96.0**	**96.8**	**97.0**

the success rate of person tracking as the evaluation metric, which is calculated as $\frac{1}{N}\sum_{i=0}^{N} a_i$, where N represents the number of frames within a sequence and a_i is a binary indicator. It equals one if the distance between the recognized and ground-truth bounding boxes is less than 50 pixels and zero otherwise. As for Zhong's method [29], which is a reinforcement-learning-based tracker outputting the action directly, we compare it in the action space (detailed settings are explained in APPENDIX-A).

2) Experimental Results: The results are shown in Table I. It can be observed that a people tracker with our RPF-ReID can achieve better performance than one-stage methods in *corridor1* and *room*. For example, "ByteTrack [34] + RPF-ReID" achieves 69.1% and 82.4% respectively. However, this combination achieves a lower success rate with 20.2% and 54.2% in *corridor2* and *lab-corridor*, respectively. These two datasets contain significantly more long-term variations, with more than 5,000 frames recorded. Pre-trained deep features cannot generalize well to these practical RPF scenarios, which include different viewpoints and lighting conditions, suffering from the so-called domain drift problem.

When the feature extractor is fine-tuned online using our OCL-based memory manager ("ByteTrack + RPF-ReID + OCL"), it achieves the best performance, with success rates above 93.5% in all sequences of the custom-built dataset and 97.0% in the public dataset. This indicates the effectiveness of our proposed OCL-based memory manager in leveraging online collected experiences to optimize the feature extractor and mitigate domain drift. In this way, the resulting ReID features capture more knowledge about the target person, enabling successful ReID even in challenging RPF scenarios.

C. Online Continual Learning Evaluation

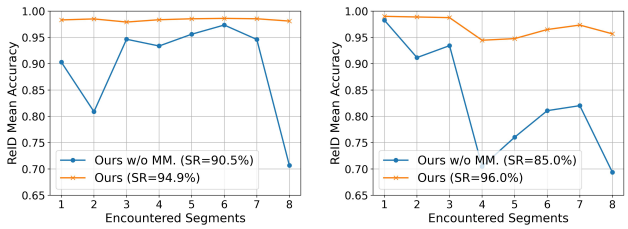
1) Metric: The evaluation of OCL [7] aims to assess how well the model remembers previous knowledge, which is essential for person ReID in RPF, as previous knowledge contains potentially matching experiences for future ReID. Additionally, incrementally remembering previous knowledge might result in a more generalized feature extractor. Specifically, we treat the OCL evaluation for person ReID as a classification task, where we assume that the true identity of the target person is known in each frame, and the model incrementally learns with known labels. For evaluation purposes, we divide each sequence into eight segments, each representing different levels of distribution drift. During incremental learning, after each segment is learned, the model is evaluated on previously seen segments. Similar to [7], we use the ReID mean accuracy at the end of training (r-mEAcc) as our OCL evaluation metric: $\frac{1}{8}\sum_{j=0}^{8} a_{8,j}$, where $a_{8,j}$ represents the average accuracy on the j_{th} segment, with the model learned from all eight segments. Higher r-mEAcc values indicate that the model retains more of the previous knowledge during incremental learning.

TABLE II. Experiments on *corridor2* and *lab-corridor* are conducted to evaluate the ReID mean accuracy at the end of training (r-mEAcc, %) and success rate (SR, %). All r-mEAcc values are averaged across three runs.

Methods	**corridor2**		**lab-corridor**	
	r-mEAcc ↑	***SR*** ↑	***r-mEAcc***	***SR***
ByteTrack [34] + RPF-ReID	59.2 ± 0.0	20.2	31.7 ± 0.0	54.2
ByteTrack + RPF-ReID + OCL, based on BioSLAM [9]	94.9 ± 2.0	94.9	79.0 ± 22.5	93.8
ByteTrack + RPF-ReID + OCL, based on MIR [24]	94.7 ± 0.8	95.4	86.1 ± 14.3	96.1
ByteTrack + RPF-ReID + OCL, based on Reservoir [26]	96.5 ± 0.4	94.9	94.0 ± 0.7	96.0

2) Experimental Results: The results are shown in Table II. The row of "ByteTrack [34] + RPF-ReID" utilizes a pre-trained and fixed feature extractor. In comparison to the original setup "ByteTrack + RPF-ReID + OCL, based on Reservoir [26]", it experiences a significant decline in performance, with reductions of 37.3% and 74.7% in r-mEAcc and success rate, respectively, on *corridor2*. This underscores the importance of online optimization of the feature extractor with collected experiences to combat domain drift and enhance the system's ReID performance. Moreover, this approach to discriminative ReID modeling markedly improves person tracking efficacy, as evidenced by higher success rate.

We also verify the necessity of our memory management (introduced in Sec. III-D) for mitigating domain drift. We use newly observed samples only to fine-tune the feature extractor without memory management. As shown in Fig. 4, this naive strategy ("Ours w/o MM.") leads to significant domain drift, resulting in a notable decrease in ReID accuracy across different segments. Such drift significantly undermines the RPF system's tracking efficiency, manifesting as success rate reductions of 4.4% and 11.0% on *lab-corridor* and *corridor2*, respectively. This outcome highlights the value of our mem-



(a) corridor2 (b) lab-corridor

Fig. 4. Plots of ReID mean accuracy w.r.t. encountered segments. "Ours w/o MM." indicates fine-tuning the feature extractor without memory management (introduced in Sec. III-D), using newly observed samples only. After fine-tuning from a new segment, the model is evaluated on segments it has encountered previously to determine its mean accuracy. This metric indicates the model's ability to retain knowledge from segments learned earlier.

ory manager in preserving valuable long-term experiences. By replaying these experiences to mitigate domain drift, we ensure that our ReID features remain robust, thereby enhancing the RPF system's consistent tracking performance.

In summary, the above experiments demonstrate the effectiveness of optimizing the feature extractor online using collected experiences managed by our memory manager. This strategy effectively addresses domain drift, resulting in enhanced ReID performance within the RPF system. Another observation is that although the OCL ability of BioSLAM [9] is worse than Reservoir [26] with r-mEAcc of 79.0% vs. 94.0% on *lab-corridor*, its tracking accuracy only drops by 2.2%. This indicates that not all historical knowledge needs to be memorized for person ReID in some situations. However, we claim that maximizing the enhancement of ReID ability at a long-term scale is still necessary as it ensures a discriminative appearance representation for dealing with complex ReID situations.

D. Runtime Analysis

We conducted a runtime analysis on two computer setups: a high-end PC and an onboard NUC. For our proposed ReID module, we tested the following configuration, which showed the best performance in our experiments: Reservoir-based [26] memory consolidation, a ResNet18-based feature extractor where only the layers after *conv3* (including *conv3*) are trainable, and the RR-based part classifiers.

In the experiment, we ran a separate thread that encompasses memory management and feature extractor fine-tuning, operating independently from the main thread. The results are shown in Table III. On the high-end PC, the main thread runs at 35.1 Hz, and the separate thread runs at 22.2 Hz. On the onboard NUC, the threads run at 18.8 Hz and 6.6 Hz, respectively. Therefore, we conclude that our RPF system can follow a target in real time, as also demonstrated in the supplementary video.

E. Implementation Details

For all experiments, we set the following default parameters: memory sizes $|\mathbb{S}| = 64$ and $|\mathbb{L}| = 512$, a batch size of 64 for each replay including long-term and short-term relays, a regularization parameter $\lambda = 1.0$ for RR,

TABLE III. Runtime analysis of our RPF framework. Two computer setups are evaluated, including a high-end PC (**Setup 1**) and an onboard NUC (**Setup 2 for real-world deployment**). We record the average time cost (**ms**) per frame. **Perception** includes bounding-box, human-joint and human-orientation detections. **Tracking** denotes the motion tracker. **ReID** indicates the re-identification process, including feature extraction and target estimation. The above three processes run in the **Main Thread**. A **Separate Thread** handles the OCL process, including memory management and replay, as well as the online continual learning of the feature extractor. For detailed experimental settings, refer to the supplementary materials.

Setup	Perception	Tracking	ReID	Total (Main Thread)	OCL (Separate Thread)
1	18.1	1.7	8.7	28.5	45.1
2	33.4	2.4	12.9	53.2	152.0

a keyframe selection threshold $\delta_l = 0.02$, an id switch threshold $\delta_{\text{sw}} = 0.35$, a ReID threshold $\delta_{\text{reid}} = 0.7$ and a number of consecutive frames $\zeta_{\text{reid}} = 5$. In this paper, for representing the part-level features, we define ten parts: $\{\text{front, back}\}\times\{\text{head, torso, legs, feet, whole}\}$.

For orientation estimation, we employ MonoLoco [35] to infer the orientation using detected joint positions from AlphaPose [36]. These joint positions are also utilized to estimate the visible parts. We utilize YOLOX-S [37] for bounding-box detection. For the tracking module, we utilize ByteTrack as our tracking method. For our proposed ReID module, we use ResNet18 as our feature extractor, pre-trained on the MOT16 dataset [28]. During OCL for the ResNet18, only the layers after *conv3* are trainable (including *conv3*).

All evaluations are conducted on both a high-end PC and an onboard NUC. The high-end PC includes an Intel® Core™ i9-12900K CPU and NVIDIA GeForce RTX 3090. The onboard NUC is an Intel NUC 11 mini PC powered by a Core i7-1165G7 CPU and NVIDIA GeForce RTX2060-laptop GPU. This NUC is mounted on a Unitree Go1 quadruped robot to perform robot person following in the real world as shown in Fig. 1 and the submitted video. Besides the computer, a dual-fisheye Ricoh camera is mounted on the robot, providing cropped perspective images with a resolution of 640×480 and a frequency of 30Hz.

V. CONCLUSION

We approach person ReID in RPF as a problem of online continual learning for mitigating the domain drift problems. This enables the RPF system to learn incrementally from online collected experiences. As a result, the framework achieves a discriminative appearance representation, allowing for effective ReID even in challenging scenarios, such as frequent appearance changes, occlusion, and distracting people with similar appearances. Compared to existing baselines, our target-ReID framework achieves state-of-the-art performance in person ReID within RPF scenarios.

For future work, i) we will explore methods to consolidate valuable samples, aiming to maximize the learning of appearance representations while preventing the forgetting of previous knowledge. Additionally, strategies for balancing efficient ReID with incremental memorization in crowded environments will be investigated. ii) We will create an

application-driven dataset containing practical RPF scenarios to support the development of person-tracking algorithms for RPF. These efforts will enhance the robustness and effectiveness of the OCLReID framework in real-world applications. Examples include a trolley-cart following system in airports [38] and shopping-cart assistance [39], which is designed to aid elderly individuals in dynamic environments.

REFERENCES

[1] M. J. Islam, J. Hong, and J. Sattar, "Person-following by autonomous robots: A categorical overview," *Int. J. Robot. Res. (IJRR)*, vol. 38, no. 14, pp. 1581–1618, 2019.

[2] X. Chen, J. Liu, J. Wu, C. Wang, and R. Song, "Lopf: An online lidar-only person-following framework," *IEEE Trans. Instrum. and Meas.*, vol. 71, pp. 1–13, 2022.

[3] K. Koide and J. Miura, "Identification of a specific person using color, height, and gait features for a person following robot," *Robot. and Auton. Syst.*, vol. 84, pp. 76–87, 2016.

[4] K. Koide, J. Miura, and E. Menegatti, "Monocular person tracking and identification with on-line deep feature selection for person following robots," *Robot. and Auton. Syst.*, vol. 124, p. 103348, 2020.

[5] B. X. Chen, R. Sahdev, and J. K. Tsotsos, "Integrating stereo vision with a cnn tracker for a person-following robot," in *Int. Conf. Comput. Vis. Syst.* Springer, 2017, pp. 300–313.

[6] ——, "Person following robot using selected online ada-boosting with stereo camera," in *Conf. Comput. Rob. Vis. (CRV)*, 2017, pp. 48–55.

[7] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, and S. Sanner, "Online continual learning in image classification: An empirical survey," *Neurocomputing*, vol. 469, pp. 28–51, 2022.

[8] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 6229–6238.

[9] P. Yin, A. Abuduweili, S. Zhao, L. Xu, C. Liu, and S. Scherer, "Bioslam: A bioinspired lifelong memory system for general place recognition," *IEEE Trans. Robot.*, vol. 39, no. 6, pp. 4855–4874, 2023.

[10] Q. Leng, M. Ye, and Q. Tian, "A survey of open-world person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 4, pp. 1092–1108, 2019.

[11] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features." *Proc. Eur. Conf. Comput. Vis.*, vol. 2008, pp. 262–275, 2008.

[12] Z. Shi, T. M. Hospedales, and T. Xiang, "Transferring a semantic representation for person re-identification and search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2015.

[13] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. Hoi, "Deep learning for person re-identification: A survey and outlook," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 2872–2893, 2021.

[14] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2014.

[15] V. Somers, C. De Vleeschouwer, and A. Alahi, "Body part-based representation learning for occluded person re-identification," in *Proc. IEEE Winter Conf. App. Comput. Vis. (WACV)*, 2023, pp. 1613–1623.

[16] Y. Li, J. He, T. Zhang, X. Liu, Y. Zhang, and F. Wu, "Diverse part discovery: Occluded person re-identification with part-aware transformer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2898–2907.

[17] Z. Zhong, L. Zheng, Z. Luo, S. Li, and Y. Yang, "Invariance matters: Exemplar memory for domain adaptive person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 598–607.

[18] ——, "Learning to adapt invariance in memory for person re-identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2723–2738, 2020.

[19] Y. Zhao, Z. Zhong, F. Yang, Z. Luo, Y. Lin, S. Li, and N. Sebe, "Learning to generalize unseen domains via memory-based multi-source meta-learning for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6277–6286.

[20] 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," *Proc. National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[21] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell, "Progress & compress: A scalable framework for continual learning," in *Int. Conf. Mach. Learn.* PMLR, 2018, pp. 4528–4537.

[22] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7765–7773.

[23] S. Lee, J. Ha, D. Zhang, and G. Kim, "A neural dirichlet process mixture model for task-free continual learning," in *Int. Conf. Learn. Representat.*, 2020.

[24] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, and L. Page-Caccia, "Online continual learning with maximal interfered retrieval," in *Adv. Neural Inf. Process. Syst.*, 2019, vol. 32, pp. 11 849–11 860.

[25] D. Shim, Z. Mai, J. Jeong, S. Sanner, H. Kim, and J. Jang, "Online class-incremental continual learning with adversarial shapley value," in *AAAI Conf. Artif. Intell.*, vol. 35, no. 11, 2021, pp. 9630–9638.

[26] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. Torr, and M. Ranzato, "Continual learning with tiny episodic memories," in *Int. Conf. Mach. Learn.* PMLR, 2019.

[27] H. Ye, J. Zhao, Y. Pan, W. Cherr, L. He, and H. Zhang, "Robot person following under partial occlusion," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2023, pp. 7591–7597.

[28] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," 2016.

[29] F. Zhong, P. Sun, W. Luo, T. Yan, and Y. Wang, "Towards distraction-robust active visual tracking," in *Int. Conf. Mach. Learn.* PMLR, 2021, pp. 12 782–12 792.

[30] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4282–4291.

[31] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10 448–10 457.

[32] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *IEEE Int. Conf. Inf. Process.*, 2017, pp. 3645–3649.

[33] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric sort: Rethinking sort for robust multi-object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 9686–9696.

[34] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," in *Proc. Eur. Conf. Comput. Vis.*, 2022.

[35] L. Bertoni, S. Kreiss, and A. Alahi, "Monoloco: Monocular 3d pedestrian localization and uncertainty estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, October 2019.

[36] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2353–2362.

[37] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[38] P. Xie, B. Xia, A. Hu, Z. Zhao, L. Meng, Z. Sun, X. Gao, J. Wang, and M. Q.-H. Meng, "Autonomous multiple-trolley collection system with nonholonomic robots: Design, control, and implementation," *J. Field Robot.*, 2024.

[39] N. Doering, S. Poeschl, H.-M. Gross, A. Bley, C. Martin, and H.-J. Boehme, "User-centered design and evaluation of a mobile shopping robot," *Int. J. Social Robot. (IJSR)*, vol. 7, pp. 203–225, 2015.

APPENDIX

A. Comparison with Active Object Tracking Methods

Active Object Tracking (AOT) methods, as described in [29], utilize an end-to-end approach through reinforcement learning. These methods process raw video frames as input and generate camera movement actions as output. According to [29], there are seven discrete actions: *move-forward/backward, turn-left/right, move-forward-and-turn-left/right*, and *no-op*. However, existing person-following datasets, such as Chen's dataset [5] and our own dataset, only provide ground truth bounding boxes of the target person. For a valid comparison, we need to map these bounding boxes to the action space. As noted in [29], a two-stage method (combining single object tracking with a PID controller) can achieve a 100% success rate if the estimated target bounding box (bbox) is accurate. Therefore, we can deduce the ground truth actions from the ground truth target bboxes. This approach allows us to evaluate AOT methods within person-following datasets. According to [29], the objective of the camera action is to keep the target person's bbox centered in the image, maintaining the same size as initially observed. The ground truth action is generated according to the horizontal error $X_{err} = \frac{X_b - W/2}{W/2}$ and the size error $S_{err} = \frac{W_b \times H_b - W_{exp} \times H_{exp}}{W_{exp} \times H_{exp}}$ shown as in Fig. 5. According to [29], we have the following ground truth action mappings:

(1) *Move forward* if $\mathrm{abs}(X_{err}) \leq 0.1$ and $S_{err} \leq -0.2$;
(2) *Move backward* if $\mathrm{abs}(X_{err}) \leq 0.1$ and $S_{err} \geqslant 0.2$;
(3) *No-op* if $\mathrm{abs}(X_{err}) < 0.1$ and $\mathrm{abs}(S_{err}) < 0.2$;
(4) *Move forward and turn right* if $0.1 \leq X_{err} \leq 0.3$;
(5) *Turn right* if $X_{err} > 0.3$;
(6) *Move forward and turn left* if $-0.3 \leq X_{err} \leq -0.1$;
(7) *Turn left* if $X_{err} < -0.3$.

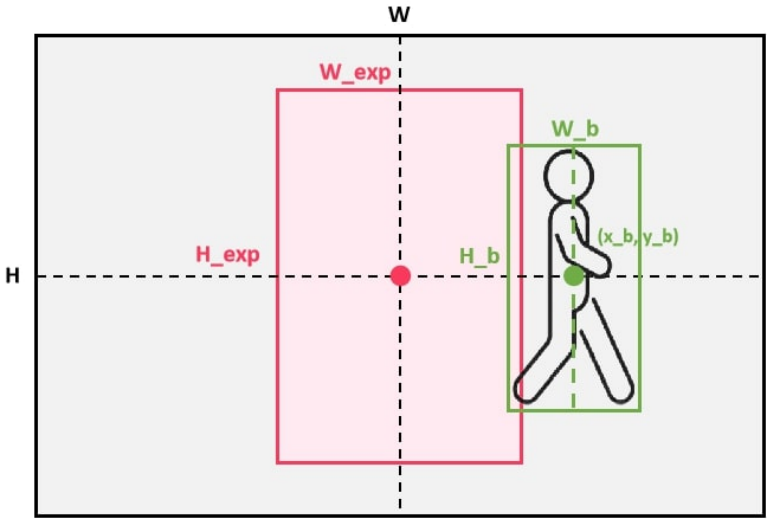
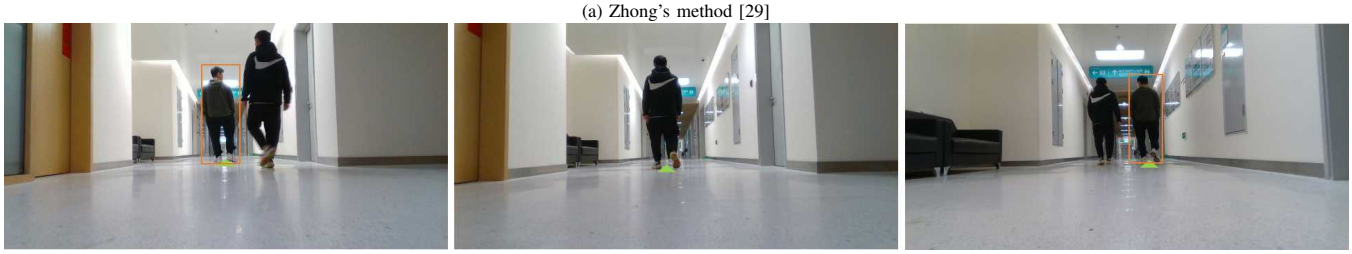


Fig. 5. An example to illustrate errors is as follows: The goal of a correct action is to move the target person closer to the center of the image. Here, (W, H) represents the image's width and height. (W_{exp}, H_{exp}) denotes the size of the expected centered bounding box, initialized by the first bounding box of the target person. (W_b, H_b) represents the target person's bounding box size in the current frame, and (x_b, y_b) is its center point.

This paper aims to evaluate the algorithms' tracking performance, meaning true target person identification is the first priority. Therefore, besides evaluating accurate action estimation, we reduce the matching standards. Specifically, we consider an action to be true if this action tries to move the target person to the center of the image. For example, if the bbox of the target person is on the left of the image, *move backward, turn left*, and *move forward and turn left* are all considered as true actions. The results are reported in Table. I. An example is shown in Fig. 6. We observe that after a long occlusion by a visually similar person, Zhong's method outputs *move forward and turn left*, failing to re-identify the target person. In contrast, our method reliably re-identifies the target person even after long-term occlusion.

(a) Zhong's method [29]



(b) Ours

Fig. 6. An example comparing Zhong's method [29] and ours. From left to right, the sequence represents observations where a long-term occlusion occurs. (a) Zhong's method outputs *move forward and turn left* due to a failed re-identification of the target person. (b) Our method reliably re-identifies the target person even after long-term occlusion.

B. Memory Examples

Several replayed examples of our short-term and long-term memories are shown in Fig. 7. For more visual examples of short-term and long-term memories, please refer to the supplementary video.

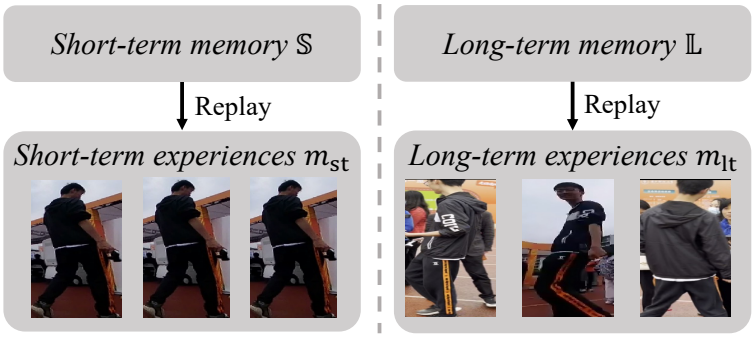


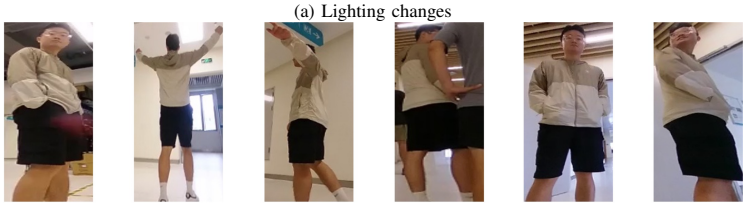
Fig. 7. Both short-term and long-term experiences $(m_{\mathrm{st}} \cup m_{\mathrm{lt}})$ are responsible for training the feature extractor with m_{st} and m_{lt} being sampled from short-term memory $\mathbb{S}$ and long-term memory $\mathbb{L}$, respectively. $\mathbb{L}$ contains sparse yet valuable historical samples, maintained by the *memory manager*. Besides, the target classifier is trained with m_{st} sampled from $\mathbb{S}$, which stores the most recently observed samples, representing the latest knowledge.

C. Visual Examples during RPF Task

Sampled images are shown in Fig. 8 consisting of the observation of the target person during the robot-person-following task. We can observe significant changes in the appearance of the target person from left to right including lighting and viewpoint changes. In such situations, previous methods relying short-term experiences would fail to re-identify the target person after occlusion. For example, as shown in Fig. 8 (b), short-term experiences capture the latest observation, i.e., the back view of the target person. Consequently, when the target person reappears with a front view, these short-term experiences fail to re-identify them.

In contrast, our method utilizes both short-term and long-term experiences. Specifically, we fine-tune the feature extractor within our OCL-assisted RPF-ReID module using both types of experiences. This approach constructs a complete representation of the target person, considering long-term experiences (e.g., the target person's front view) and short-term experiences (e.g., the target person's back view). As a result, our method can re-identify the target person even when they reappear with a front view.

(a) Lighting changes



(b) Viewpoint changes

Fig. 8. Sampled images of the target person during the robot-person-following task. We can observe significant changes in the appearance of the target person from left to right. (a) Lighting changes: bright light in the corridor, dim light in a corner, and bright light again in the elevator. (b) Viewpoint changes: the person's front view, back view, occlusion, and front view again.

D. Target-ReID Lifecycle

Algorithm 1: Target-ReID Lifecycle

Input: Current image $\mathbf{I}$ and tracked people $\{\mathbf{B}, \mathbf{p}\}_i$ representing bounding boxes and positions, target person's identity id, target confidence s, short-term memory $\mathbb{S}$, long-term memory $\mathbb{L}$, feature extractor f and target classifier g

Output: Target person's position $\{\mathbf{p}\}_{id}$ in the current frame

Extract image patches $\mathbf{M}$ from $\mathbf{I}$ and $\mathbf{B}$;
Construct the observation set $\{\mathbf{M}, \mathbf{y}\}_i$ where $\mathbf{y} = 1$ if $i == id$, otherwise $\mathbf{y} = 0$;
Extract features $\mathbf{F}$ from $\mathbf{M}$ with f;
if $id \in \{i\}$ **then**
    Estimate s of the target person based on Eq. 3;
    if $s > \delta_{\text{sw}}$ **then**
        Consider $\{\bar{i}\}$ as identities of negative tracks;
        $\{\mathbf{M}, \mathbf{y}\}_{id} \rightarrow \mathbb{S}$, $\{\mathbf{M}, \mathbf{y}\}_{\bar{i}} \rightarrow \mathbb{S}$ based on FILO rule;
        Sample m_{st} from $\mathbb{S}$;
        Train g with m_{st} based on Eq. 4;
        ### Separate Thread ###
        $\{\mathbf{M}, \mathbf{y}\}_{\bar{i}} \rightarrow \mathbb{L}$ based on FILO rule;
        $\{\mathbf{M}, \mathbf{y}\}_{id} \rightarrow \mathbb{L}$ if it is a keyframe based on Eq. 5;
        Consolidate $\mathbb{L}$ with OCL techniques if $\mathbb{L}$ is full;
        Sample m_{lt} from $\mathbb{L}$;
        Train f with m_{st} and m_{lt} based on Eq. 1;
        ### Separate Thread ###
        Return target position $\{\mathbf{p}\}_{id}$;
    else
        Let $id = -1$, indicates id switch between the target person and other people;
        Return ø;
else
    Estimate s of the i_{th} person based on Eq. 3;
    if $s > \delta_{\text{reid}}$ for consecutive ζ_{reid} frames **then**
        Let $id = i$, indicates successful target person ReID;
        Return target person's position $\{\mathbf{p}\}_{id}$;
    else
        Return ø;