A Memorizing and Generalizing Framework for Lifelong Person Re-Identification

Nan Pu, Zhun Zhong, Nicu Sebe, Michael S. Lew

Abstract—In this paper, we introduce a challenging yet practical setting for person re-identification (ReID) task, named lifelong person re-identification (LReID), which aims to continuously train a ReID model across multiple domains and the trained model is required to generalize well on both seen and unseen domains. It is therefore critical to learn a ReID model that can learn a generalized representation without forgetting knowledge of seen domains. In this paper, we propose a new MEmorizing and GEneralizing framework (MEGE) for LReID, which can jointly prevent the model from forgetting and improve its generalization ability. Specifically, our MEGE is composed of two novel modules, *i.e.*, Adaptive Knowledge Accumulation (AKA) and differentiable Ranking Consistency Distillation (RCD). Taking inspiration from the cognitive processes in the human brain, we endow AKA with two special capacities, knowledge representation and knowledge operation by graph convolution networks. AKA can effectively mitigate catastrophic forgetting on seen domains while improving the generalization ability to unseen domains. By considering the ranking factor that is specifically important in ReID, RCD is designed to distill the ranking knowledge in a differentiable manner, which can further prevent the catastrophic forgetting. To supporting the study of LReID, we build a new and large-scale benchmark with two practical evaluation protocols that consider the metrics of non-forgetting and generalization. Experiments demonstrate that 1) our MEGE framework can effectively improve the performance on seen and unseen domains under the domain-incremental learning constraint, and that 2) the proposed MEGE outperforms state-of-the-art competitors by large margins. The LReID benchmark and source code are publicly available at https://LifelongReID.github.io.

Index Terms—Person Re-Identification, Lifelong Learning, Knowledge Accumulation, Ranking Distillation

1 Introduction

Person re-identification (ReID), which aims at retrieving instances of the persons across disjoint camera views, has received increasing attention in the computer vision community [1], [2]. Although the advanced deep learning methods [3], [4], [5], [6], [7], [8] have shown a powerful feature generalization ability in ReID [9], [10], their training process heavily limited by the fixed and stationary datasets [11], [12], [13], which means that the all data need to be always accessible during the training process. However, this strict condition is hardly satisfied in many practical scenarios where the data are continuously increasing from different domains. For instance, in the smart surveillance systems that are deployed over a mass of crossroads, millions of new images are captured every day. To handle the newly incoming data, the systems are required to possess the ability of incremental or lifelong learning.

To meet the real-world requirements, we propose a challenging yet practical ReID setting, called *lifelong person re-identification* (LReID). In LReID, the model is required to incrementally learn and accumulate the informative knowledge from a stream of seen domains, and then the trained model needs to be evaluated on the test data of both seen and unseen domains (see Fig. 1). Thus, memorizing the informative knowledge of seen domains and obtaining generalized representation are both important during the training process. Compared with conventional lifelong learning tasks and existing ReID settings, our LReID has four differences that make it more challenging and practical. 1) Unlike the existing lifelong classification tasks [14], [15] that mainly focus on reducing the forgetting rate on the seen classes, LReID additionally concentrates on improving the discrimination of the model on unseen classes that never appear during the training stage. This is because, as a retrieval task, ReID typically assumes that the training and testing sets are from non-overlapped identities/classes. 2) Existing lifelong learning tasks commonly assume that all the data belong to the same domain. In contrast, in LReID, there are large domain shifts between training data of different steps, and the testing data are composed of both seen and unseen domains. The existence of domain gap largely rises the difficulty of the LReID. 3) LReID is a more challenging since the intra-class appearance variations in ReID are significantly subtler than those in traditional classification tasks (*e.g.*, CIFAR [16] and ImageNet [17]). This particularly increases the challenges of lifelong learning, as the model has to learn a discriminative representation that is robust to unseen classes/identities across multiple learning steps. 4) Compared with the existing ReID settings summarized in Tab. 1, LReID allows the model to incrementally accumulate the knowledge of already-trained (seen) domains and improve the model's generalization ability on unseen domains in the ever-changing real-world environment. A recent work [18] introduces a continual representation learning (CRL) setting for bio-metric identification, which shares a similar motivation with our LReID. However, CRL overlooks the practical aspect of domain-incremental data collection, which is commonly encountered in real-world ReID systems. This renders the CRL setting impractical and

- *Nan Pu and Michael S. Lew are with Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands.*
- *Z. Zhong and N. Sebe are with the Department of Information Engineering and Computer Science, University of Trento, Trento 38100, Italy.*
- *Corresponding author: Zhun Zhong (zhunzhong007@gmail.com)*

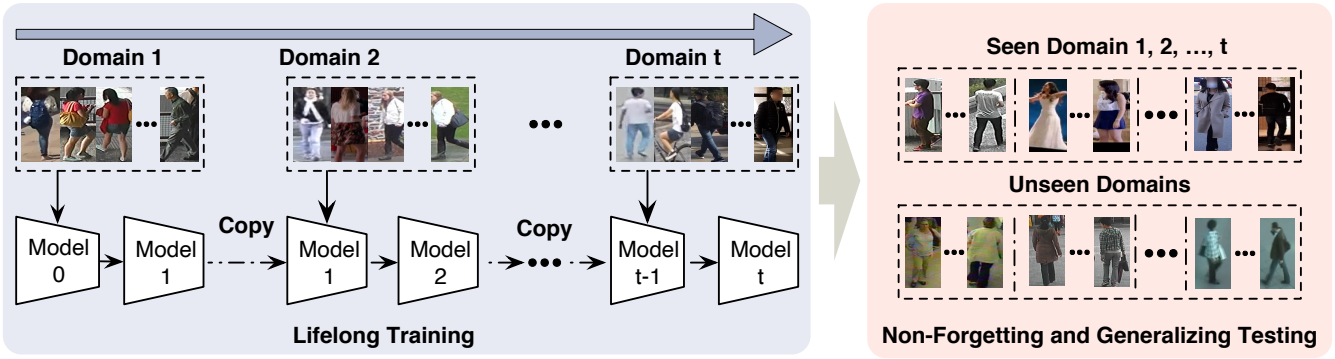


Fig. 1: Pipeline of lifelong person re-identification (LReID). The model is trained in multiple steps, each of which includes images of new identities from a new domain. The data of previous domains are not available in the following steps. During testing, the model is required to be evaluated on testing images of both seen and unseen domains.

reduces the associated challenges, as the models under CRL are less susceptible to the issue of catastrophic forgetting.

To this end, we propose a novel Memorizing and Generalizing framework (MEGE) to solve the challenges in LReID. Our MEGE consists of two novel components, *Adaptive Knowledge Accumulation* (AKA) and differentiable *Ranking Consistency Distillation* (RCD). They cooperatively help the model to learn the generalized representation without forgetting knowledge from seen data. Concretely, AKA is designed to adaptively extract the underlying and transferable knowledge from old domains and leverage this knowledge to facilitate learning representations with a robust generalization performance on unseen domains. The mechanism of AKA is inspired by the cognitive processes in the human brain. As discovered by [19], [20], when a visual cognitive process starts, the human brain retrieves relevant representational content (knowledge) from high-dimensional memories based on similarity or familiarity. Then, the human brain summarizes the captured information, and updates relevant knowledge or allocates new memory. Such cognitive processes can be decomposed into *"representations"* and *"operations"* sub-processes [19]. Motivated by this, we attempt to mimic the cognitive processes during LReID and endow AKA with lifelong learning capabilities by separately accomplishing *knowledge representation* and *knowledge operation*. Specifically, we first represent the learned knowledge by an accumulated knowledge graph (AKG). Then, given mini-batch samples, we temporally construct an instance similarity graph (ISG) based on their relationships. Next, AKA establishes cross-graph links between the AKG and the ISG, and executes a graph convolution for information query and propagation. Such *operations* enable the AKG to transfer the previous knowledge to each current instance. Meanwhile, AKG is updated through summarizing the relationships among current instances. Furthermore, we integrate *plasticity loss* and *stability loss* into the AKA, which encourages AKG to learn the generalized representation without forgetting in a balanced manner.

In our previous work [21], we directly employ a classical Logit-based Knowledge Distillation (LKD) technique [22] to improve the anti-forgetting ability. However, this approach ignores the underlying adjacent relations between samples that are vital in ReID tasks. In other words, as a retrieval task, ReID aims to learn discriminative representations based on inter-sample ranking relations rather than classification probability of each sample. In light of this, as a notable extension of our previous work [21], we propose a differentiable Ranking Consistency Distillation (RCD) approach to enforce the model to explicitly consider the knowledge of relations between samples during the distillation process, thereby promoting the lifelong learning capability of the ReID model. RCD is built upon the classical Spearman's footrule distance (SFD) [23], enabling us to measure the discrepancies of affinity relationship of samples between the teacher and the student models. However, since the ranking function is discontinuous, SFD cannot be used to optimize the model with back-propagation. To address this issue, we propose to approximate SFD by a hyperbolic tangent function, allowing our RCD to be differentiable for model optimization. In addition, considering the importance of each ranking position, we propose to dynamically learn the position-wise weights during the distillation process, which encourages the model to automatically focus on informative ranking knowledge and thus further improves the anti-forgetting ability.

In summary, our contributions are featured as follows:

- We propose the LReID setting, which places ReID problem under a lifelong learning scenario. The LReID is challenging but practical, raising a new perspective toward the real-world ReID application.
- We build a large-scale benchmark along with two evaluation protocols for supporting the study of LReID.
- We introduce a human-like approach, Adaptive Knowledge Accumulation (AKA) approach, for LReID, which can adaptively update previous knowledge and learn the generalized knowledge by a learnable knowledge graph.
- We present a Ranking Consistency Distillation (RCD), which explicitly distills the ranking knowledge in a differentiable and weight-dynamic manner.
- We design the Memorizing and Generalizing framework (MEGE) that derives the mutual benefits of the proposed AKA and RCD. Extensive experiments demonstrate the effectiveness of our MEGE in learning a generalized representation without forgetting previous knowledge. Our MEGE outperforms state-of-the-art methods by a large margin under our built LReID benchmark.

TABLE 1: The comparison of different settings. $\mathcal{S}$ and $\mathcal{T}$ indicate source domain and target domain, respectively. "tr" and "te" represent the train split and test split, respectively. "# Training Steps" indicates the number of continuous training steps. For the one stage case, the training data are provided all at once during training.

ReID Setting	# Training Steps	Train Data	Train Label	Test Data	Domain Shift
Fully-Supervised	One	$\mathcal{S}^{tr}$	$\mathcal{S}^{tr}$	$\mathcal{S}^{te}$	✗
Unsupervised Domain Adaptation	One	$\mathcal{S}^{tr}$ and $\mathcal{T}^{tr}$	$\mathcal{S}^{tr}$	$\mathcal{T}^{te}$	✓
Pure Unsupervised	One	$\mathcal{S}^{tr}$	None	$\mathcal{S}^{te}$	✗
Domain Generalization	One	$\mathcal{S}_1^{tr}, \mathcal{S}_2^{tr}, ..., \mathcal{S}_n^{tr}$	$\mathcal{S}_1^{tr}, \mathcal{S}_2^{tr}, ..., \mathcal{S}_n^{tr}$	$\mathcal{T}^{te}$	✓
Our Lifelong Learning	Multiple	$\mathcal{S}_t^{tr}, 1 \leq i \leq t$	$\mathcal{S}_i^{tr}, 1 \leq i \leq t$	$\mathcal{S}_{1,...,t}^{te}$ and $\mathcal{T}^{te}$	✓

2 Related Work

2.1 Person Re-identification

Person ReID has been widely studied in the last decade. As summarized in Tab. 1, the existing works are mainly conducted on four settings. 1) In the *Fully-Supervised* setting, the training data are fully labeled, and the test data share the same distribution with the training data. Existing fully-supervised methods mainly focus on investigating and exploiting different network structures (*e.g.*, omni-scale network [24], part-based network [25], pyramid network [26]) and loss functions (*e.g.*, softmax-based losses [27], triplet-based losses [28], and other kinds of losses [29], [30]).

2) In the *Unsupervised Domain Adaptation* setting [3], [31], we are given a labeled source domain and an unlabeled target domain. The goal is to mitigate the domain gaps between source and target domains and thus to learn a model that is robust to target testing data. 3) The objective of the *Pure Unsupervised* ReID [32] is to learn a discriminative ReID model with only unlabeled training data. In general, the model is trained by a clustering strategy and the test data are assumed to be sampled from the same distribution as training data. 4) Under the *Domain generalization* setting [9], [33], we are provided with labeled data captured from one domain or multiple domains and the trained model is evaluated on unseen target domains.

Although these explorations have narrowed the gaps between ReID algorithms and real applications, they ignore the important lifelong learning scenario that is commonly encountered in practice. Recently, the one-pass person ReID setting [34] and the continual bio-metric representation learning (CRL) setting [18] were introduced. However, CRL neglects the domain-incremental data collection manner that pervasively exists in practical ReID applications so that they wrongly think that lifelong ReID models hardly encounters catastrophic forgetting problems. On the other hand, due to the distinct distribution discrepancies between the training datasets, the model in our LReID setting is harder to continuously accumulate knowledge, compared with that in the CRL setting. We show experimental evidences in Tab. 6. Hence, this paper proposes a more practical and challenging LReID setting for real-world person ReID. Note that in this paper, our main focus is on the conventional ReID task, where individuals maintain consistent clothing appearances. However, we acknowledge that this assumption does not always hold in real-world scenarios, where persons wear different clothes, as introduced in cloth-changing ReID studies [35], [36], [37], [38], [39]. Therefore, including cloth-changing ReID scenarios in our LReID setting would provide a more challenging yet practical study for the community.

2.2 Lifelong Learning

Lifelong learning [40] is also named continual learning [14], [15], incremental learning [41] or sequential learning [42]. The study of it can be dated back to several decades. Thanks to the impressive progresses in deep neural networks, lifelong learning has regained the spotlight and is widely employed in various vision and learning tasks, such as object recognition [15], [43], object detection [44], image generation [45], reinforcement learning [46], [47], unsupervised learning [48] and zero-shot learning [49]. In lifelong learning, the model is required to have the ability to learn from a sequence of tasks and to transfer knowledge obtained from earlier tasks to a later one. The key challenge for lifelong learning is *catastrophic forgetting*, in which the model will encounter a significant performance degradation on previous tasks after training on new tasks. Existing methods can be divided into three categories, including knowledge distillation by the teacher-student structure [22], regularizing the parameter updates [50] when training with new tasks, and learning with stored or generated image samples of previous tasks [15].

Despite the effectiveness of the above mentioned methods, most of them are not well suitable for LReID due to the following four reasons. 1) The number of classes in ReID is much larger than that in conventional lifelong learning tasks. Specifically, the popular benchmarks for conventional lifelong learning tasks include MNIST [51], CORe50 [52], CIFAR-100 [16], CUB [53] and ImageNet [17]. Except for ImageNet, other benchmarks are small-scale in terms of classes numbers. In contrast, the commonly used ReID datasets include more than 1,000 classes/identities for each, *e.g.*, Market-1501 [11], MSMT17 [13], and CUHK03 [54]. 2) ReID datasets are more imbalanced because the number of samples per class ranges from 2 to 100 [55]. Since model degradation typically happens when learning from tail classes, LReID also raises a few-shot learning challenge. 3) Similar with the fine-grained retrieval task [56], the inter-class appearance variations in ReID are significantly subtler than in generic classification tasks, which further increases the difficulty of lifelong learning. 4) Existing lifelong learning works assume that the training and testing data have the same classes, while the testing data are always from unseen classes in ReID. The above four factors make LReID very different from traditional lifelong learning tasks and thus bring unique challenges for LReID.

2.3 Graph Convolutional Networks

Recently, graph-based deep learning methods have received more and more attention from researchers. Inspired by convolutional neural networks (CNNs) in computer vision,

many graph-based neural networks (GNN) have been designed, such as Graph Convolutional Network (GCN) [57] and graph attention networks (GATs) [58]. The techniques of GNN are applied to various tasks, such as semi-supervised classification [57], visual question answering [59], image captioning [60], shape completion [61] and point cloud segmentation [62]. Moreover, due to the advantage of GNN in reasoning and aggregating graph data, some works apply GNN to solve various ReID applications, *e.g.* , positive pair prediction [31] for unsupervised domain adaptation and spatial-temporal GCN [63] for video-based ReID. Different from them, in this paper, we explore GNN in lifelong ReID setting, in which two different graph structures are proposed to learn informative knowledge through a cross-graph communication manner instead of an intra-graph propagation way.

2.4 Knowledge Distillation

Knowledge distillation (KD) is a technique to enable the student model to learn richer information from the teacher, which has become a popular and effective way to retain the learned knowledge devoid of forgetting in incremental tasks [64]. The most two popular methods are logit-based knowledge distillation (LKD) [22] and feature-based knowledge distillation (FKD) [44], which constrain the discrepancies of teacher and student models on the logit-level and feature-level respectively. Many metrics can be used to measure the teacher-student discrepancy, such as cross-entropy [65], l_1-distance [66], l_2-distance [67], Gramian matrix [68], Kullback-Leibler (KL) divergence [69], and Maximum Mean Discrepancy (MMD) [70]. Some recent methods [71], [72], [73] also consider additional inter-instance relationships during distillation, such Similarity-Preserving knowledge Distillation (SPD) [72] and Correlation Distillation (CD) [73]. Different from these methods, we propose a Ranking Consistency Distillation (RCD) method that is tailor-made for the ReID task. Our RCD considers the ranking information during the distillation process and optimizes the network in a differentiable manner.

3 Lifelong Person Re-Identification

3.1 Problem Definition

In this section, we introduce the setting definition and the experimental setup of lifelong person re-identification (LReID). LReID aims at learning one unified model from T domains in an incremental fashion. Suppose we have a stream of datasets $\mathcal{S} = \{\mathcal{S}_t\}_{t=1}^{T}$. The dataset of the t-th domain is represented as $\mathcal{S}_t = \{\mathcal{S}_t^{tr}, \mathcal{S}_t^{te}\}$, where $\mathcal{S}_t^{tr}$ and $\mathcal{S}_t^{te}$ indicate the training set and testing set respectively. $\mathcal{S}_t^{tr} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{|\mathcal{S}_t^{tr}|}$ contains the training image set $\mathcal{X}_t^{tr}$ and the corresponding label set $\mathcal{Y}_t^{tr}$, where $|\mathcal{S}_t^{tr}|$ indicates the number of training samples. Similarly, $\mathcal{S}_t^{te} = \{\mathcal{X}_t^{te}, \mathcal{Y}_t^{te}\}$, which is only used for evaluation. The identities/classes of training and testing sets are disjoint, so that $\mathcal{Y}_t^{tr} \cap \mathcal{Y}_t^{te} = \varnothing$. In addition, the identities of different domains are totally different, we thus have $\mathcal{Y}_t \cap \mathcal{Y}_{\neq t} = \varnothing$. At the t-th training step, only $\mathcal{S}_t^{tr}$ is available while the training data from previous domains are *NOT* available any more. For evaluation, we estimate the retrieval performance on the testing sets of all encountered (seen) domains, *i.e.*, $\mathcal{S}_1^{te}, ..., \mathcal{S}_t^{te}$, respectively. Moreover, to verify the generalization ability, the trained model is also evaluated on a new testing set $\mathcal{T}^{te}$, which is composed of the testing sets of several unseen target domains. Commonly, there are significant domain shifts between different (both seen and unseen) domains, increasing the difficulties of training and testing stages. Since we mainly elaborate the training stages in the following, we will omit the superscript $\{tr, te\}$ for simplicity.

3.2 Baseline for LReID

A straightforward approach for LReID is continually finetuning a pre-trained model on the new domains. However, such simple finetuning strategy will cause two severe problems. 1) The trained model will forget the knowledge previously learned on old domains. That is, the performance on old domains will deteriorate drastically due to the well-known catastrophic forgetting [74]. 2) The trained model will be biased towards the training domain at hand. In this situation, the model cannot effectively refer to historical knowledge from old domains, hampering the generalization ability on both seen and unseen domains.

To deal with the above two challenges, we introduce a baseline solution based on knowledge distillation to address LReID. The training model of the baseline consists of a feature extractor $h(\cdot;\theta)$ with parameters θ and an identification classifier $g(\cdot;\phi)$ with parameters ϕ. The whole network $f(\cdot;\theta,\phi)$ is the mapping from the input space to confidence scores, which is defined as: $f(\cdot;\theta,\phi) = g(h(\cdot;\theta);\phi)$. At the beginning of the stage t, we initialize $f(\cdot;\theta,\phi)$ by the model obtained by the previous stage $t-1$, which is represented by $\hat{f}\left(\cdot;\hat{\theta},\hat{\phi}\right)$. Here, we omit the step indicator t for simplicity. In addition, the dimension of the classifier ϕ is extended to $\sum_{i=1}^{t}|\mathcal{Y}_i|$, where $|\mathcal{Y}_i|$ is the number of classes in domain i.[1] During training, the network $f(\cdot;\theta,\phi)$ is optimized by the traditional cross-entropy loss,

$$\mathcal{L}_c = -\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{S}} \mathbf{y}\log\left(\boldsymbol{\sigma}\left(f\left(\mathbf{x};\theta,\phi\right)\right)\right), \tag{1}$$

where $\boldsymbol{\sigma}$ is *softmax* function. $\mathbf{x}$ and $\mathbf{y}$ are the training sample and its identity label of the current domain at t-th training step, respectively. Note that, the *softmax* function is only applied on the outputs of the current domain.

In addition, we adopt the logit-based knowledge distillation (LKD) [22] technique for mitigating forgetting on previous $t-1$ domains. By introducing a teacher-student structure, the LKD technique considers the discrepancies between the outputs of the student and teacher models (i.e., the current model and the frozen model copied from the initial states of the current model before training on the current domain) in a probabilistic space for each instance. The loss function is defined as:

$$\mathcal{L}_d = -\sum_{\mathbf{x}\in\mathcal{S}}\sum_{j=1}^{n}\boldsymbol{\sigma}\left(f(\mathbf{x};\hat{\theta},\hat{\phi})\right)_j \log\left(\boldsymbol{\sigma}\left(f(\mathbf{x};\theta,\phi)\right)_j\right), \tag{2}$$

where $n = \sum_{i=1}^{t-1}|\mathcal{Y}_i|$ is the number of the classes of previous $t-1$ domains. Note that, the *softmax* function is only applied on the outputs of the previous $t-1$ domains.

1. At the first stage, θ is initialized by ImageNet [16] pretrained model and ϕ is randomly initialized with the dimension of $|\mathcal{Y}_1|$.

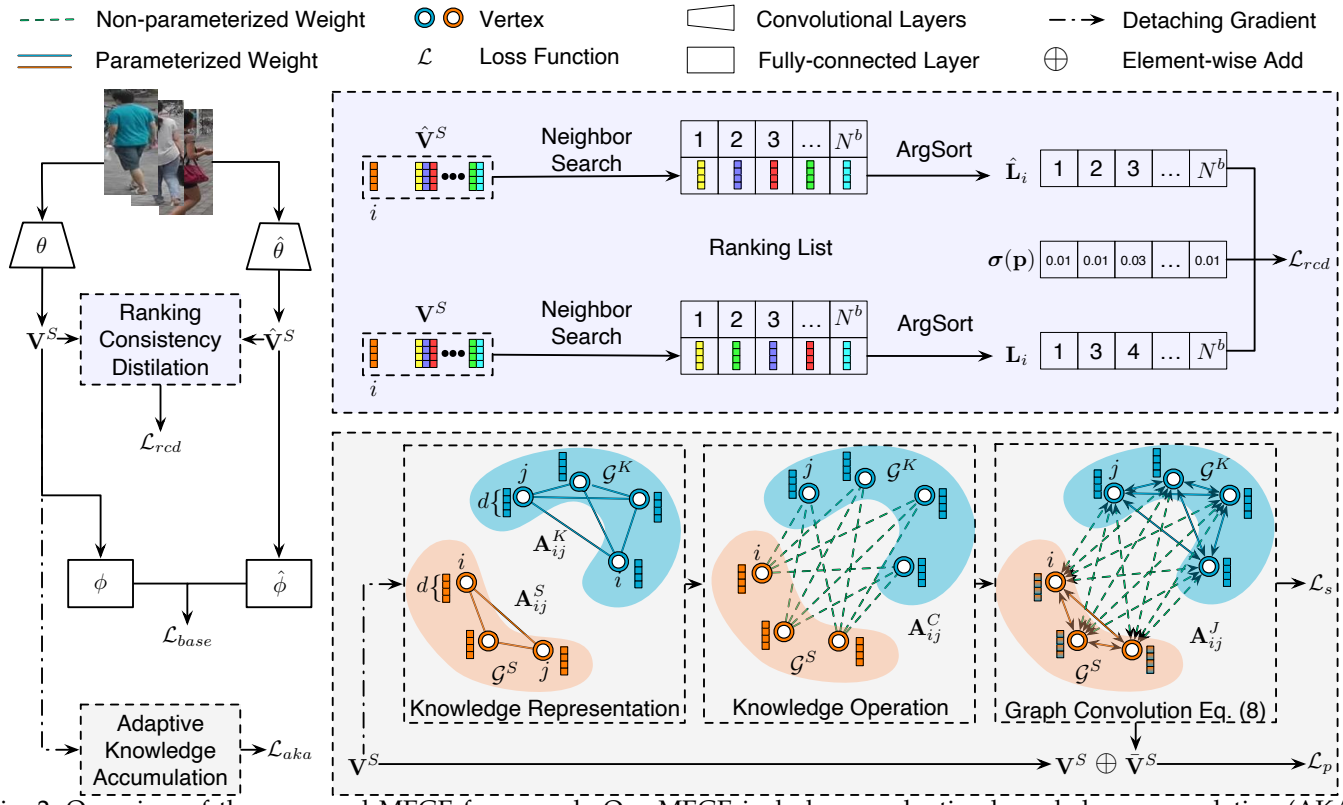


Fig. 2: Overview of the proposed MEGE framework. Our MEGE includes an adaptive knowledge accumulation (AKA) module and a ranking consistency distillation (RCD) module. The former encourages the model to obtain anti-forgetting and generalization abilities by learnable knowledge graphs. The latter enforces the model to maintain more informative knowledge from the previous domains by distilling ranking results. In addition, RCD is optimized in a differentiable and dynamically-weighting manner.

The total objective of the baseline method is formulated as:

$$\mathcal{L}_{base} = \mathcal{L}_c + \gamma \mathcal{L}_d, \tag{3}$$

where γ is the weight of the knowledge distillation loss. We set γ to 1 in our experiments, which achieves consistent well performance in all settings. Note that, only $f(\cdot;\theta,\phi)$ is optimized while $\hat{f}\left(\cdot;\hat{\theta},\hat{\phi}\right)$ is fixed during training.

4 Memorizing and Generalizing Framework

Although the proposed baseline approach is able to mitigate catastrophic forgetting in the LReID setting, the abilities of generalizing on unseen domains and flexibly learning informative knowledge of the current domain are still limited, resulting in a large margin to the up-bound performance of the model trained by all domains jointly. In this paper, we carefully consider the characteristics of LReID (discussed in Sec. 1) and further propose a novel Memorizing and Generalizing (MEGE) framework upon the baseline for facilitating lifelong learning. As shown in Fig. 2, our MEGE consists of an Adaptive Knowledge Accumulation (AKA) module and a differentiable Ranking Consistency Distillation (RCD) module, which collaboratively improve the abilities of generalizing and anti-forgetting. We will introduce AKA and RCD in the following sections.

4.1 Adaptive Knowledge Accumulation

In this section, we introduce the details of the proposed Adaptive Knowledge Accumulation (AKA). The goal of AKA is to improve the abilities of generalizing and anti-forgetting by learning transferable and informative knowledge. Referring to biological prior knowledge, AKA mimics the cognitive process of the human brain [19] to construct two relatively independent sub-processes: *knowledge representation* and *knowledge operation*. The former aims at establishing an informative knowledge bank by explicitly learning and storing knowledge representations. The latter enables the model to leverage the knowledge banks learned from previous domains to improve the generalization ability as well as to update the knowledge bank with less forgetting. The illustration of AKA is shown in the low-right part of Fig. 2. We next elaborate the *knowledge representation* and *knowledge operation*.

4.1.1 Knowledge Representation

AKA implements knowledge representation (KR) by constructing two different graph structures: *instance similarity graph* (ISG) and *accumulated knowledge graph* (AKG). Specifically, ISG is used for representing the potential knowledge in a mini-batch. On the other hand, AKG focuses on accumulating the transferable knowledge that is learned from already-trained domains.

Instance Similarity Graph. To mine and represent the structural knowledge contained in a mini-batch samples,

we construct a fully-connected graph based on similarities of these samples, called Instance Similarity Graph (ISG). Specifically, given a mini-batch with N^b samples from the current domain, the ISG is defined as $\mathcal{G}^S = (\mathbf{A}^S, \mathbf{V}^S)$, where $\mathbf{A}^S \in \mathbb{R}^{N^b \times N^b}$ is the edge set and $\mathbf{V}^S \in \mathbb{R}^{N^b \times d}$ is the vertex set. The vertex set $\mathbf{V}^S$ are the features for the mini-batch samples, which are obtained by $g(\mathbf{x}; \phi)$. The edge weight $\mathbf{A}^S_{ij}$ is measured by a learnable L_1-based distance between the corresponding vertices $\mathbf{V}^S_i$ and $\mathbf{V}^S_j$:

$$\mathbf{A}^S_{ij} = \boldsymbol{\rho}(\mathbf{W}^S \left| \mathbf{V}^S_i - \mathbf{V}^S_j \right| + \mathbf{b}^S), \tag{4}$$

where $\mathbf{W}^S$ and $\mathbf{b}^S$ represent learnable parameters, and $\boldsymbol{\rho}$ is *Sigmoid* function. That is, the edge weights are parameterized and learned from training processes. During each mini-batch training, our AKA temporarily constructs an ISG to mine proximity relationships between instances as well as provides a path to allow inter-instance information to flow mutually. This mechanism enables the model to learn generalized knowledge instead of overfitting on independent instances.

Accumulated Knowledge Graph. Unlike the ISG that is temporarily built for each mini-batch training, we construct a fixed-size Accumulated Knowledge Graph (AKG) and maintain the AKG during the whole lifelong training process, which stores and updates the accumulated knowledge learned across previous domains. Specifically, the AKG is denoted as $\mathcal{G}^K = (\mathbf{A}^K, \mathbf{V}^K)$. The $\mathbf{V}^K \in \mathbb{R}^{N^k \times d}$ is the vertex set, where d is the feature dimension and N^k is the number of the vertices of AKG. Correspondingly, the $\mathbf{A}^K \in \mathbb{R}^{N^k \times N^k}$ is the adjacent matrix of AKG. Analogous to the definition of ISG in Eq. (4), the edge weight between $\mathbf{V}^K_i$ and $\mathbf{V}^K_j$ is defined as:

$$\mathbf{A}^K_{ij} = \boldsymbol{\rho}(\mathbf{W}^K(\left| \mathbf{V}^K_i - \mathbf{V}^K_j \right|) + \mathbf{b}^K), \tag{5}$$

where $\mathbf{W}^K$ and $\mathbf{b}^K$ are learnable parameters. The design of AKG is based on the following considerations: 1) During domain-incremental training, domains arrive one after another in a sequence and the vertices of AKG are expected to be dynamically updated in a timely manner. Therefore, the vertex representations are parameterized and learned at the training time. 2) To encourage the diversity of knowledge encoded in the AKG, the vertex representations are randomly initialized. 3) The edge weights in the ISG and the AKG are calculated by independent learnable parameters, as the manners of knowledge organizations in two graphs have distinct differences. The former focuses on the relationship among current samples. The latter is required to consider both its own structure and efficient knowledge transformation that is elaborated in next section. This design is different from the graph matching network [75] where the two graphs share the same weights.

In this way, the vertices of AKG are encouraged to represent different types of knowledge (*e.g.*, the representative person appearance and structure) and the corresponding edges are automatically constructed to reflect the relationship between such knowledge. As a result, AKG tends to learn common meta-knowledge for generalizing on unseen domains well.

4.1.2 Knowledge Operation

Based on the recent discoveries in cognitive science [19], [20], our brains can continually learn new knowledge with less forgetting, which largely attribute to the relative independence between the *"knowledge operation"* and the *"knowledge representation"* in a complex cognitive process. Motivated by this, different from the proposed KR that employs parameterized edge weights to organize knowledge, we apply non-parameterized weights for implementing the knowledge operation (KO) with less domain dependence. Furthermore, we decompose the KO into *knowledge transfer* and *knowledge accumulation* stages: the former aims at extracting the knowledge of AKG accumulated from the previous learning processes and then transfers such knowledge to benefit the model's ability to generalize on unseen domains; the latter enables the AKG to self-update so as to adaptively accumulate the learned knowledge.

Knowledge Transfer. To selectively transfer knowledge from the AKG to the ISG, we propose a novel cross-graph communication (CGC) mechanism based on graph convolution networks (GCNs) [76]. Specifically, the proposed CGC can be divided into the following four steps.

The first step involves establishing inter-graph links based on vertex similarity. For any two vertices from different graphs $\mathbf{V}^S_i$ and $\mathbf{V}^K_j$, the weight of the cross-graph edge $\mathbf{A}^C_{ij}$ is calculated by:

$$\mathbf{A}^C_{ij} = \frac{\exp(-\frac{1}{2}\left\| \mathbf{V}^S_i - \mathbf{V}^K_j \right\|^2_2)}{\sum_{k=1}^{N^k} \exp(-\frac{1}{2}\left\| \mathbf{V}^S_i - \mathbf{V}^K_k \right\|^2_2)}. \tag{6}$$

Note that, unlike designing parameterized weights for knowledge representation, we use non-parameterized weights for knowledge operation. The reason will be explained in Sec. 4.2.

In the second step, a new fully-connected joint graph is constructed by considering both inter-graph and intra-graph structures to associate the AKG with the ISG. The joint graph $\mathcal{G}^J = (\mathbf{A}^J, \mathbf{V}^J)$ is defined by:

$$\mathbf{A}^J = \begin{bmatrix} \mathbf{A}^S & \mathbf{A}^C \\ (\mathbf{A}^C)^T & \mathbf{A}^K \end{bmatrix}, \mathbf{V}^J = \begin{bmatrix} \mathbf{V}^S \\ \mathbf{V}^K \end{bmatrix}, \tag{7}$$

where $\mathbf{A}^J \in \mathbb{R}^{(N^b+N^k)\times(N^b+N^k)}$ and $\mathbf{V}^J \in \mathbb{R}^{(N^b+N^k)\times d}$ are the adjacent matrix and vertex matrix of the joint graph, respectively.

After constructing the joint graph, the third step involves propagating the most related knowledge from the AKG to the ISG via a graph convolution, which is formulated as:

$$\mathbf{V}^G = \boldsymbol{\delta}(\mathbf{A}^J(\mathbf{V}^J\mathbf{W}^J)), \tag{8}$$

where $\mathbf{V}^G \in \mathbf{R}^{(N^b+N^k)\times d}$ is the vertex embedding after one-layer "message-passing" [77] and $\mathbf{W}^J$ is a learnable weight matrix of the GCN layer followed by a non-linear function $\boldsymbol{\delta}$, *e.g.*, ReLU [78]. Moreover, from the results in Tab. 9, we experimentally found that stacking more GCN layers cannot acquire significant improvements, even worse on the anti-forgetting evaluation. Thus, we employ one-layer GCN to accomplish information propagation for simplicity.

Finally, we obtain the information-propagated feature representation of ISG by passing features through the GCN, which is formulated as:

$$\bar{\mathbf{V}}^S = \{\mathbf{V}_i^G | i \in [1, N^b]\}. \tag{9}$$

In short, the main purposes of CGC are: 1) to query the relevant knowledge from the previous training experience in the AKG for promoting the training of a new domain; 2) to enable the intra- and inter-graph information to propagate mutually, thereby guiding models towards a better optimization.

Knowledge Accumulation. Maintaining a knowledge graph within limited storage resource during lifelong learning is inevitably expected to compact memorized knowledge and selectively update the knowledge graph. To achieve this goal during the optimization of AKG, we first consider the CGC mechanism as a knowledge retrieval process to extract the related knowledge contained in the AKG and leverage these feedback knowledge to complement the original features. Then, we propose a new stability-plasticity objective to force the AKG to learn transferable and generalized knowledge while reducing the manipulation of the previously-learned representations in the AKG. The whole process is elaborated in the following paragraphs.

To begin with, we utilize the vertices $\mathbf{V}^S$ of the ISG as query representations to retrieve pertinent knowledge from the AKG. Consequently, corresponding feedback representations $\bar{\mathbf{V}}^S$ are generated. As query representations primarily contain domain-specific information and feedback representations are extracted from multiple previous domains, these two types of representation are deemed complementary for composing generalized representations. To jointly optimize these representations, we aggregate $\mathbf{V}^S$ and $\bar{\mathbf{V}}^S$ by computing their sum, which is formulated as:

$$\mathbf{F} = \frac{1}{2}\left(\mathbf{V}^S + \bar{\mathbf{V}}^S\right). \tag{10}$$

In order to enhance the generalization capability of the fused representations, we introduce a plasticity objective:

$$\mathcal{L}_p = \frac{1}{N^b} \sum_{(a,p,n)} \ln\Big(1 + \exp\left(\Delta(\mathbf{F}_a, \mathbf{F}_p) - \Delta(\mathbf{F}_a, \mathbf{F}_n)\right)\Big), \tag{11}$$

where Δ denotes a distance function, *e.g.*, L_2 distance or cosine distance. a, p and n donate the anchor, positive and negative instances in a mini-batch respectively, which are selected by online hard-mining sampling strategy [28].

However, optimizing the AKG solely based on the plasticity objective $\mathcal{L}_p$ results in overfitting on the current domain and significant changes in the AKG's vertices. This exacerbates the issue of catastrophic forgetting. To solve this problem, we propose a stability objective to punish the large movements of AKG's vertices during the update process from the ending state $\hat{\mathbf{V}}^K$ of last training step to current state $\mathbf{V}^K$. The stability loss function is formulated as:

$$\mathcal{L}_s = \frac{1}{N^k} \sum_{i=1}^{N^k} \ln\Big(1 + \exp\left(\Delta(\mathbf{V}_i^K, \hat{\mathbf{V}}_i^K)\right)\Big). \tag{12}$$

Both Eq. (11) and Eq. (12) are used to optimize the parameters of AKG. However, their gradient flowing into the feature extractor $h(\cdot;\theta)$ is detached. We will discuss this design in Sec. 4.2. Through enforcing such stability-plasticity dilemma, the AKG accumulates more refine and general knowledge from comparison with previous knowledge and thus generates better representation for generalizable ReID.

During the training on the t-th domain, we use the data of $\mathcal{S}_t$ to train the feature extractor, classifier, ISG, and AKG, without accessing any data from previous domains. The loss function of the AKA framework is formulated as:

$$\mathcal{L}_{aka} = \lambda_p \mathcal{L}_p + \lambda_s \mathcal{L}_s, \tag{13}$$

where λ_s and λ_p are plasticity-stability weights. When λ_p is relatively larger than λ_s, the AKG focuses on learning new knowledge while paying few attentions on preserving previous knowledge. On the contrary, the AKG maintains approximately fixed knowledge representations and the model is benefited from the knowledge learned from only the first training domain instead of continuously accumulating knowledge across different domains. The optimal balance between these two terms not only ensures the stability of knowledge graph, but also endows AKG with a plasticity that allows new knowledge to be incorporated and accumulated.

4.2 Discussion

Q1: Why use parameterized edge weights for the knowledge representation but non-parameterized edge weights for the knowledge operation? In the sight of [79], the partial parameters of top layers favor becoming domain-specific during incremental training on different domains, which leads to severe performance degradation on previous domains. In addition, according to the biological inspiration [19], the representation and operation should be independent. To this end, when performing the knowledge transfer, a non-parameterized metric allows the model to treat different domains with less bias so that the knowledge transferred from the AKG can generalize on unseen domains well. In contrast, the knowledge representation that focuses on summarizing and updating knowledge requires the power of parameterization. Thus, we design the non-parameterized metrics for the knowledge representation. Furthermore, we conduct the experiments in Tab. 8 to verify our analyses. Our careful design achieves the best performance compared with other variants.

Q2: Why detach the gradient of GCN? Without detaching gradient, AKA will tend to learn relatively similar knowledge/representation as the feature extractor, which is caused by the degradation of GCN [80]. This largely limits the power of graph-guided structure and hampers AKA to learn more generalizable knowledge. Instead, detaching the gradient encourages AKA to independently learn diverse and generalizable knowledge across different domains, making AKA learn new knowledge that is different but complementary to the feature extractor. In Fig. 7, we experimentally demonstrate the above explanation by comparing the difference between the ISG representations before and after propagation ($\mathbf{V}^S$ and $\bar{\mathbf{V}}^S$) through training.

4.3 Ranking Consistency Distillation

ReID is a retrieval task, where modeling the inter-instance ranking relations during training is of importance in improving testing accuracy. However, in our AKA framework, we do not explicitly consider the inter-instance ranking relations during the lifelong learning process, which will lead the model to largely ignore such important knowledge and thus to have sub-optimal anti-forgetting ability. To solve this problem, we propose a novel Ranking Consistency Distillation (RCD) loss, which enables us to constrain the consistency of the ranking lists generated from the student and the teacher models and thus efficiently preserves the knowledge of previous domains. RCD is designed based on the classical Spearman's footrule distance (SFD) [23]. However, since SFD is a non-continuous ranking function, it cannot be directly used for optimization. To solve this challenge, we propose to use a differentiable surrogate function to make our RCD compatible with general optimizers (*e.g.*, SGD [81] and Adam [82]). Moreover, considering the varying importance of each position in a ranking list, we inject learnable position weights into RCD to further facilitate the training process. Next, we will first revisit SFD and then introduce our RCD in detail.

4.3.1 Revisit SFD in FKD

In general, SFD measures the l_1 distance between a pair of ranking lists or permutations. To formulate the SFD under the context of LReID, we first calculate the elements in ranking lists, and then derive the formula of SFD.

Given two feature sets generated from the student and the teacher models, $\mathbf{V}^S$ and $\hat{\mathbf{V}}^S \in \mathbb{R}^{N^b \times d}$, the cosine similarity matrices of them are defined as:

$$\begin{aligned} \mathbf{S} &= (\mathbf{V}^S / \left\|\mathbf{V}^S\right\|) \cdot (\mathbf{V}^S / \left\|\mathbf{V}^S\right\|)^T \in \mathbb{R}^{N^b \times N^b}, \\ \hat{\mathbf{S}} &= (\hat{\mathbf{V}}^S / \left\|\hat{\mathbf{V}}^S\right\|) \cdot (\hat{\mathbf{V}}^S / \left\|\hat{\mathbf{V}}^S\right\|)^T \in \mathbb{R}^{N^b \times N^b}, \end{aligned} \tag{14}$$

where $\cdot$ denotes matrix multiplication and $\|\cdot\|$ is l_2 normalization. Inspired by Bubble Sort, we formulate the ranking list for each instance i by:

$$\mathbf{L}_{ij} = 1 + \sum_{k=1, k\neq j}^{N^b} \mathbb{1}(\mathbf{S}_{ij} < \mathbf{S}_{ik}), \tag{15}$$

where j indicates the j^{th} element in the mini-batch. The indicator function $\mathbb{1}(\cdot)$ is defined as:

$$\mathbb{1}(condition) = \begin{cases} 1, & \text{if } condition \text{ is true,} \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

In this way, Eq. 15 indicates the ranking positions of each element corresponding to a query i in $\mathbf{V}$. Similarly, we can derive $\hat{\mathbf{L}} \in \mathbb{R}^{N^b \times N^b}$ for $\hat{\mathbf{V}}$. Given a mini-batch with N^b samples, the SFD-based knowledge distillation loss is:

$$\mathcal{L}_{sfd} = \frac{1}{N^b} \sum_{i=1}^{N^b} \sum_{j=1}^{N^b} \left| \mathbf{L}_{ij} - \hat{\mathbf{L}}_{ij} \right|, \tag{17}$$

where $|\cdot|$ denotes the absolute value function.

Limitation. Although SFD can well establish the distances between rankings, it depends on a discrete sort operation. In addition, it fails to take into account the importance of different positions in a ranked list. These two aspects induce two problems during the knowledge distillation of LReID. 1) Calculating the SFD is associated with a discontinuous optimization problem, which is unfavorable for gradient-based back-propagation optimization schemes. 2) Without considering the importance of each position in the ranking list, the model will treat each position equally. Although we can previously assign different fixed weights to enforce the importance of each position during distillation, it is uncertain which weights are suitable at different training stages. This hampers us distill informative ranking knowledge effectively and flexibly.

To address the above two limitations, we first derive a surrogate function as the differentiable approximation that enables the SFD-based loss function to be compatible with general deep neural networks. In addition, we extend SFD to an adaptive position-aware weighting variant that allows the model to learn how to transfer ranking knowledge in a dynamic way. The details are elaborated in the following sections.

4.3.2 Differentiable Argsorted Function

To make the SFD-based loss function differentiable, we straightforwardly derive a variant of the popular hyperbolic tangent function as the surrogate function to approximate the indicator function in Eq. (16). The surrogate function $\mathbb{S}$ is defined as:

$$\mathbb{1}(\mathbf{S}_{ij}, \mathbf{S}_{ik}) \approx \mathbb{S}_i = \frac{1}{2}(\tanh(\mathbf{S}_{ij} - \mathbf{S}_{ik}) + 1). \tag{18}$$

Correspondingly, the derivative is derived as following:

$$\frac{\partial \mathbb{S}_i}{\partial \mathbf{S}_{ij}} = 1 - \frac{1}{2}\Big(\tanh(\mathbf{S}_{ij} - \mathbf{S}_{ik})\Big)^2. \tag{19}$$

By replacing Eq. (16) by Eq. (18), the ranking list in Eq. (15) can be approximated by:

$$\mathbf{L}_{ij} = 1 + \sum_{k=1, k\neq j}^{N^b} \frac{1}{2}(\tanh(\mathbf{S}_{ij} - \mathbf{S}_{ik}) + 1). \tag{20}$$

Given the approximated ranking list sets $\mathbf{L}$ and $\hat{\mathbf{L}}$ generated by the student-teacher models, we apply a sort function to align them following descending order. This results in new sorted ranking lists. To simplify the notation, we still use $\mathbf{L}$ and $\hat{\mathbf{L}}$ to denote the new sorted ranking lists henceforth.

In this paper, we call the above-mentioned calculations as *Argsorted* function, which ensures the differentiability of SFD-based loss functions. Moreover, due to the negligible computational cost of Eq. (18) that mainly includes several mini-batch matrix multiplications, the proposed differentiable Argsorted function can be optimized efficiently.

4.3.3 Adaptive Position-Aware Weighting

To explicitly consider the varying importance of positions in the ranking list during the distillation process, we propose to dynamically learn the corresponding weights by regarding them as trainable parameters. This enables us not only soften the consistent ranking constraint but also dynamically modify these weights instead of depending on prior knowledge (*e.g.*, the closer and the more important in general).

TABLE 2: The statistics of ReID datasets involved in the Alpha-LReID benchmark. "*" denotes that we modify the original dataset by selecting samples according to the ground-truth bounding boxes.

Benchmark	Datasets Name	Scale	Balanced Protocol						Imbalanced Protocol					
			#Identities			#Images			#Identities			#Images		
			Train	Query	Gallery	Train	Query	Gallery	Train	Query	Gallery	Train	Query	Gallery
LReID-Seen	Market-1501 [11]	large	500	750	751	9,173	3,368	15,913	751	750	751	12,936	3,368	15,913
	CUHK-SYSU ReID* [83]	mid	500	2,900	2,900	2,180	2,900	5,447	5,532	2,900	2,900	15,088	2,900	5,447
	DukeMTMC-ReID [12]	large	500	702	1,110	11,027	2,228	17,661	702	702	1,110	16,522	2,228	17,661
	MSMT17_V2 [13]	large	500	3,060	3,060	13,212	11,659	82,161	1,041	3,060	3,060	30,248	11,659	82,161
	CUHK03 [54]	mid	500	700	700	4,867	1,400	5,332	767	700	700	7,365	1,400	5,332
LReID-Unseen	VIPeR [84]	small	-	316	316	-	316	316	-	316	316	-	316	316
	PRID [85]	small	-	100	649	-	100	649	-	100	649	-	100	649
	GRID [86]	small	-	125	126	-	125	900	-	125	126	-	125	900
	i-LIDS [87]	small	-	60	60	-	60	60	-	60	60	-	60	60
	CUHK01 [88]	small	-	486	486	-	972	972	-	486	486	-	972	972
	CUHK02 [89]	mid	-	239	239	-	478	478	-	239	239	-	478	478
	SenseReID [90]	mid	-	521	1,718	-	1,040	3,388	-	521	1,718	-	1,040	3,388

Specifically, we initialize a set of parameters $\mathbf{p} \in \mathbb{R}^{N^b}$ with identical values and employ the Softmax function $\boldsymbol{\sigma}$ to generate probabilistic weights. The position-weighted ranking consistency distillation loss is formulated as:

$$\mathcal{L}_{rcd} = \frac{1}{N^b} \sum_{i=1}^{N^b} \sum_{j=1}^{N^b} \boldsymbol{\sigma}(\mathbf{p})_j \left| \mathbf{L}_{ij} - \hat{\mathbf{L}}_{ij} \right|, \tag{21}$$

where $\mathbf{p}$ is dynamically learned to control the importance of each position.

4.4 Optimization

Overall, our MEGE framework consists of the baseline, the AKA and the RCD modules, which are optimized jointly. The baseline module adapts conventional lifelong learning approaches into the proposed LReID setting, which basically realizes learning without forgetting. On this basis, the proposed RCD module encourages the feature extractor to preventing catastrophic forgetting. This enables us to provide robust feature representations, facilitating the AKA module in organizing and accumulating knowledge. In turn, the AKA transfers generalizable knowledge to the feature extractor, which further improves the feature discrimination. During the optimization process, this mutual promotion mechanism guides the whole MEGE framework towards effective lifelong learning. The overall loss function is formulated as:

$$\mathcal{L}_{total} = \mathcal{L}_{base} + \mathcal{L}_{aka} + \lambda_{rcd}\mathcal{L}_{rcd}, \tag{22}$$

where λ_{rcd} controls the weigh of the RCD module.

5 Benchmark and Evaluation Protocol

5.1 A New Lifelong Person ReID Benchmark

To support the study of LReID, we propose a large-scale benchmark based on existing ReID datasets, which is composed of LReID-Seen and LReID-Unseen subsets. We call it as **Alpha-LReID** benchmark. The LReID-Seen subset is used to incrementally train LReID models and evaluate their anti-forgetting ability. The LReID-Unseen subset serves as unseen testing domains to verify generalization ability of the models. The training datasets are completely non-overlapping with the testing datasets.

LReID-Seen Subset. We select five relatively large-scale person ReID datasets: Market-1501 (MA) [11], CUHK-SYSU (SY) [83], DukeMTMC-ReID (DU) [12], MSMT17 (MS) [13], and CUHK03 (CU) [54], and use their original training sets to compose the lifelong ReID subset, called "LReID-Seen". Note that for the SY [83] dataset, we modify the original dataset by using the ground-truth person bounding box annotation, rather than using the original images which are originally used for person search evaluation. This process generates 942 training identities. For testing, we fix both query and gallery sets instead of using variable gallery sets. We select 2,900 query persons, where each query contains at least one image in the gallery. We call this variant as CUHK-SYSU ReID. As shown in Tab. 2, the LReID-Seen subset includes 82,159 images of the 8,793 identities in total. Their original testing sets are used to evaluate the performance of previous domains (anti-forgetting) and the performance on the current domain.

LReID-Unseen Subset. We merge the testing sets of 7 popular person ReID datasets: VIPeR [84], PRID [85], GRID [86], i-LIDS [87], CUHK01 [88], CUHK02 [89], and SenseReID [90] to form the unseen testing subset, named as "LReID-Unseen". Specifically, as reported in Tab. 2, LReID-Unseen includes 3,594 different identities with total 9,854 images, which is adopted to evaluate the generalization ability of the learned model.

Remarks. The proposed Alpha-LReID is different from existing lifelong learning benchmarks in three main aspects: 1) Alpha-LReID is specially designed for person re-identification that is the fine-grained retrieval task, while existing lifelong learning benchmarks mainly focus on general image classification; 2) The total number of classes in Alpha-LReID ($|Y| \approx$14K) is much larger than existing benchmarks ($\leq$ 1K); 3) In Alpha-LReID, we evaluate the model on novel identities captured from seen and unseen domains, while existing benchmarks commonly test the model on samples of known classes of seen domains.

5.2 Evaluation Protocols and Metrics

To comprehensively evaluate the model performance, we propose two evaluation protocols for Alpha-LReID: balanced evaluation protocol and imbalanced evaluation protocol.

Balanced Evaluation Protocol. We follow the configurations of lifelong classification benchmarks [16], [17], [51] to build the balanced evaluation protocol, where each training domain contains the uniform amount of classes/identities[2]. As

2. Note that for the SY [83] dataset, we only select the identities that include at least 4 samples for training.

TABLE 3: Seen-domain non-forgetting evaluation on Order-1. We test the model after sequentially training on seen domains.

Training Order		Market		SYSU		Duke		MSMT17		CUHK03		Average Seen $\bar{s}$	
Protocol	Method	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1
Balanced	SFT	16.9$_{\pm 0.3}$	39.6$_{\pm 0.4}$	57.1$_{\pm 0.2}$	60.7$_{\pm 0.2}$	7.9$_{\pm 0.3}$	15.7$_{\pm 0.4}$	1.8$_{\pm 0.3}$	5.7$_{\pm 0.5}$	**52.4**$_{\pm 0.2}$	**55.1**$_{\pm 0.3}$	27.2$_{\pm 0.3}$	35.4$_{\pm 0.3}$
	SPD [91]	30.6$_{\pm 0.2}$	53.2$_{\pm 0.2}$	65.3$_{\pm 0.1}$	68.4$_{\pm 0.2}$	12.6$_{\pm 0.3}$	22.5$_{\pm 0.3}$	2.9$_{\pm 0.3}$	9.1$_{\pm 0.4}$	42.8$_{\pm 0.1}$	44.2$_{\pm 0.3}$	31.3$_{\pm 0.3}$	40.2$_{\pm 0.4}$
	LwF [22]	34.5$_{\pm 0.3}$	54.2$_{\pm 0.2}$	69.2$_{\pm 0.2}$	72.2$_{\pm 0.2}$	15.6$_{\pm 0.2}$	26.7$_{\pm 0.3}$	2.8$_{\pm 0.3}$	8.4$_{\pm 0.4}$	30.2$_{\pm 0.3}$	30.9$_{\pm 0.3}$	30.4$_{\pm 0.3}$	38.5$_{\pm 0.3}$
	CRL [18]	35.2$_{\pm 0.2}$	55.1$_{\pm 0.2}$	70.2$_{\pm 0.1}$	73.7$_{\pm 0.2}$	15.9$_{\pm 0.3}$	27.5$_{\pm 0.3}$	3.5$_{\pm 0.4}$	10.5$_{\pm 0.5}$	31.6$_{\pm 0.1}$	31.8$_{\pm 0.3}$	31.3$_{\pm 0.2}$	39.7$_{\pm 0.3}$
	AKA [21]	37.0$_{\pm 0.2}$	59.6$_{\pm 0.2}$	70.5$_{\pm 0.3}$	73.8$_{\pm 0.3}$	16.3$_{\pm 0.2}$	28.1$_{\pm 0.2}$	3.6$_{\pm 0.4}$	10.8$_{\pm 0.4}$	36.3$_{\pm 0.2}$	36.9$_{\pm 0.3}$	32.7$_{\pm 0.3}$	41.8 $_{\pm 0.3}$
	MEGE	**39.0**$_{\pm 0.2}$	**61.6**$_{\pm 0.1}$	**73.3**$_{\pm 0.1}$	**76.6**$_{\pm 0.1}$	**16.9**$_{\pm 0.2}$	**30.3**$_{\pm 0.3}$	**4.6**$_{\pm 0.3}$	**13.4**$_{\pm 0.3}$	36.4$_{\pm 0.2}$	37.1$_{\pm 0.3}$	**34.0**$_{\pm 0.2}$	**43.8**$_{\pm 0.2}$
	Joint	70.6$_{\pm 0.1}$	87.4$_{\pm 0.2}$	74.3$_{\pm 0.2}$	77.3$_{\pm 0.2}$	63.4$_{\pm 0.2}$	79.3$_{\pm 0.2}$	29.3$_{\pm 0.1}$	49.5$_{\pm 0.2}$	44.4$_{\pm 0.3}$	46.0$_{\pm 0.3}$	56.4$_{\pm 0.2}$	67.9$_{\pm 0.2}$
Imbalanced	SFT	22.7$_{\pm 0.2}$	47.1$_{\pm 0.3}$	64.7$_{\pm 0.3}$	67.7$_{\pm 0.3}$	12.0$_{\pm 0.2}$	22.6$_{\pm 0.3}$	3.2$_{\pm 0.3}$	9.6$_{\pm 0.4}$	**62.2**$_{\pm 0.3}$	**65.0**$_{\pm 0.2}$	33.0$_{\pm 0.3}$	42.4$_{\pm 0.3}$
	SPD [91]	32.9$_{\pm 0.3}$	59.3$_{\pm 0.4}$	71.5$_{\pm 0.3}$	74.7$_{\pm 0.3}$	15.0$_{\pm 0.3}$	25.4$_{\pm 0.4}$	3.8$_{\pm 0.4}$	11.7$_{\pm 0.5}$	54.2$_{\pm 0.3}$	55.9$_{\pm 0.3}$	35.5$_{\pm 0.3}$	45.4$_{\pm 0.4}$
	LwF [22]	40.2$_{\pm 0.3}$	61.9$_{\pm 0.3}$	73.0$_{\pm 0.2}$	76.1$_{\pm 0.3}$	16.1$_{\pm 0.2}$	28.6$_{\pm 0.2}$	4.6$_{\pm 0.3}$	13.0$_{\pm 0.4}$	42.6$_{\pm 0.2}$	43.1$_{\pm 0.3}$	35.3$_{\pm 0.2}$	44.5$_{\pm 0.3}$
	CRL [18]	40.8$_{\pm 0.2}$	62.6$_{\pm 0.3}$	74.4$_{\pm 0.2}$	77.6$_{\pm 0.3}$	17.2$_{\pm 0.3}$	30.0$_{\pm 0.3}$	4.6$_{\pm 0.3}$	13.4$_{\pm 0.3}$	43.7$_{\pm 0.2}$	44.1$_{\pm 0.2}$	36.1$_{\pm 0.2}$	45.5$_{\pm 0.3}$
	AKA [21]	42.3$_{\pm 0.2}$	64.5$_{\pm 0.2}$	75.2$_{\pm 0.1}$	78.1$_{\pm 0.3}$	20.1$_{\pm 0.1}$	33.3$_{\pm 0.2}$	5.4$_{\pm 0.2}$	15.2$_{\pm 0.2}$	47.3$_{\pm 0.1}$	48.1$_{\pm 0.2}$	38.1$_{\pm 0.1}$	47.8$_{\pm 0.2}$
	MEGE	**46.6**$_{\pm 0.2}$	**67.6**$_{\pm 0.3}$	**77.2**$_{\pm 0.2}$	**79.8**$_{\pm 0.3}$	**21.8**$_{\pm 0.2}$	**36.1**$_{\pm 0.2}$	**6.7**$_{\pm 0.2}$	**18.4**$_{\pm 0.3}$	47.8$_{\pm 0.3}$	49.3$_{\pm 0.3}$	**40.0**$_{\pm 0.2}$	**50.2**$_{\pm 0.3}$
	Joint	75.9$_{\pm 0.1}$	89.3$_{\pm 0.2}$	90.4$_{\pm 0.2}$	91.7$_{\pm 0.3}$	66.7$_{\pm 0.1}$	80.2$_{\pm 0.2}$	35.6$_{\pm 0.2}$	58.8$_{\pm 0.3}$	51.5$_{\pm 0.2}$	52.4$_{\pm 0.2}$	64.0$_{\pm 0.2}$	74.5$_{\pm 0.2}$

shown in Tab. 2, we uniformly sample 500 identities from each training domain in the LReID-Seen subset for 5-step domain-incremental training. As a consequence, in total, 40,459 training images of the 2,500 identities are employed in balanced evaluation protocol.

Imbalanced Evaluation Protocol. Since the scale of each dataset varies largely in the wild, we further present an imbalanced evaluation protocol, which is more practical for LReID. Different from randomly choosing unified amount of identities in each domain [21], the model is trained on the whole training set of each domain, where the number of identities is different in each domain. As a result, the imbalanced evaluation protocol involves 82,159 images of the 8,793 identities.

Training Order. In practice, the order of input domains is agnostic. Thus, we evaluate models with two different training orders, *Order-1:* **MA**→**SY**→**DU**→**MS**→**CU** and *Order-2:* **DU**→**MS**→**MA**→**SY**→**CU**.

Evaluation Metrics: We use $\bar{s}$ (average performance on seen domains) to measure the capacity of retrieving incremental seen domains and $\bar{u}$ (average performance on unseen domains) to measure the generalization capacity on unseen domains. Note that the performance gap of $\bar{s}$ between joint training (upper bond) and a certain method indicates the method's ability to prevent forgetting. $\bar{u}$ and $\bar{s}$ are measured with mean average precision (mAP) and rank-1 (R-1) accuracy. These metrics are calculated after the last training step. Furthermore, inspired by the metrics used in lifelong zero-shot learning [49], we also introduce a harmonic mean of $\bar{u}$ and $\bar{s}$:

$$H = \frac{2 \times \bar{u} \times \bar{s}}{\bar{u} + \bar{s}}, \tag{23}$$

to measure model's comprehensive ability to balance anti-forgetting and generalization ability. In this paper, we call it H-metric.

6 Experiments

6.1 Implementation Details

Implementation of MEGE. We use the ResNet-50 [92] as the backbone, where we remove the last classification layer and use the retained layers as the feature extractor. Hence, the feature dimension is 2,048. All images are resized to 256 × 128. The AKA network consists of one GCN layer. During training, the batch size is set to 64. Following the popular person ReID training strategy, in each training batch, we randomly select 16 identities and sample 4 images for each identity. The Adam optimizer [82] with learning rate 1.75×10^{-4} is used. To determinate the number of training epochs, we follow a validation procedure. At each step, we create a validation set by randomly selecting 20% identities from the current training dataset. Within the validation set, we randomly sample one example from each identity as the query considered the remaining examples as the gallery. We then evaluate the training loss and performance on validation set during the training process. We find that the model achieves stable and nearly optimal performance around the 50^{th} epoch across all datasets. Therefore, we train the model for 50 epochs using all training data for all experiments. The learning rate is decreased by × 0.1 at the 25^{th} epoch and 35^{th} epoch. In this paper, we only use the *Order-1* with imbalanced setting to tune the hyperparameters. The selected hyperparameters are then directly applied in all experiments. We set γ, λ_p, λ_s, λ_{rcd} and N^K to 1, 1, 5×10^{-4}, 1.3 and 64 respectively, which achieve well performance in all settings.

During testing, we extract the summed representations in Eq. (10) of test samples following a random order and use the Euclidean distance to estimate the similarities between samples.

Compared Methods. We compare our MEGE with 5 methods. 1) Sequential fine-tuning (SFT): this is the simple baseline which fine-tunes the model with new datasets without distilling old knowledge. 2) Learning without forgetting (LwF): the baseline method [22] introduced in Sec. 3.2. 3) Similarity-preserving distillation (SPD): a competitor with advanced feature distillation [91]. 4) Continual representation learning (CRL) [18]: a state-of-the-art method for continual ReID. 5) Adaptive Knowledge Accumulation (AKA): the reduction of our MEGE method. For fair comparison, we apply these six methods to our Alpha-LReID benchmark using the same training settings as our MEGE.

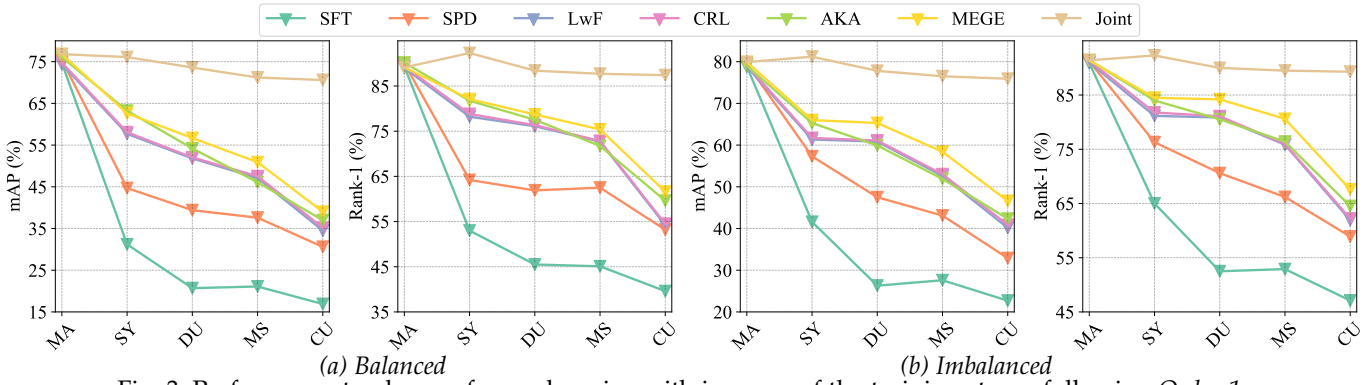
Upper Bound Method. We train the model jointly with the training data of all domains without the constraint of lifelong learning, which is regarded as the upper-bound method.

6.2 Seen-domain Non-forgetting Evaluation

We first evaluate the performance of our MEGE on seen domains, which reflects the ability of anti-forgetting. The comparisons between different methods are shown in Tab. 3 and Tab. 4 for two orders respectively. Clearly, our MEGE outperforms the compared methods regardless of the training order and evaluation protocol, demonstrating its large effectiveness for addressing the problem of LReID.

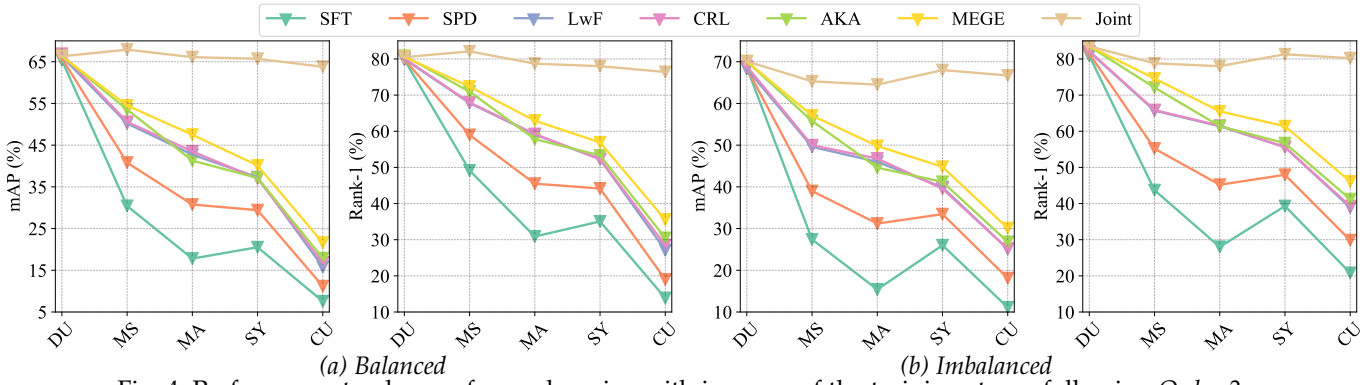
TABLE 4: Seen-domain non-forgetting evaluation on Order-2. We test the model after sequentially training on seen domains.

Training Order		Duke		MSMT17		Market		SYSU		CUHK03		Average Seen $\bar{s}$	
Protocol	Method	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1
Balanced	SFT	7.6$_{\pm 0.3}$	13.9$_{\pm 0.4}$	1.8$_{\pm 0.4}$	5.6$_{\pm 0.5}$	21.8$_{\pm 0.2}$	44.6$_{\pm 0.3}$	60.0$_{\pm 0.3}$	62.3$_{\pm 0.3}$	**49.4**$_{\pm 0.2}$	**51.4**$_{\pm 0.2}$	28.1$_{\pm 0.3}$	35.6$_{\pm 0.3}$
	SPD [91]	11.7$_{\pm 0.3}$	20.5$_{\pm 0.3}$	2.2$_{\pm 0.3}$	7.1$_{\pm 0.4}$	21.8$_{\pm 0.3}$	45.7$_{\pm 0.3}$	63.5$_{\pm 0.2}$	66.6$_{\pm 0.2}$	39.5$_{\pm 0.1}$	40.8$_{\pm 0.2}$	27.7$_{\pm 0.2}$	36.1$_{\pm 0.3}$
	LwF [22]	15.8$_{\pm 0.3}$	27.1$_{\pm 0.3}$	2.8$_{\pm 0.3}$	8.7$_{\pm 0.3}$	21.7$_{\pm 0.2}$	46.6$_{\pm 0.3}$	67.4$_{\pm 0.2}$	71.3$_{\pm 0.2}$	29.2$_{\pm 0.2}$	29.9$_{\pm 0.3}$	27.4$_{\pm 0.2}$	36.7$_{\pm 0.3}$
	CRL [18]	16.8$_{\pm 0.2}$	28.1$_{\pm 0.3}$	2.8$_{\pm 0.3}$	8.7$_{\pm 0.4}$	22.5$_{\pm 0.2}$	47.1$_{\pm 0.3}$	65.0$_{\pm 0.1}$	68.8$_{\pm 0.2}$	30.1$_{\pm 0.2}$	30.3$_{\pm 0.3}$	27.4$_{\pm 0.2}$	36.6$_{\pm 0.3}$
	AKA [21]	17.9$_{\pm 0.2}$	30.5$_{\pm 0.3}$	2.3$_{\pm 0.3}$	7.1$_{\pm 0.3}$	24.1$_{\pm 0.2}$	48.5$_{\pm 0.2}$	66.8$_{\pm 0.1}$	69.7$_{\pm 0.2}$	35.6$_{\pm 0.2}$	36.5$_{\pm 0.2}$	29.3$_{\pm 0.2}$	38.5$_{\pm 0.2}$
	MEGE	**21.6**$_{\pm 0.2}$	**35.5**$_{\pm 0.2}$	**3.0**$_{\pm 0.3}$	**9.3**$_{\pm 0.4}$	**25.0**$_{\pm 0.2}$	**49.8**$_{\pm 0.2}$	**69.9**$_{\pm 0.1}$	**73.1**$_{\pm 0.2}$	34.7$_{\pm 0.2}$	35.1$_{\pm 0.2}$	**30.8**$_{\pm 0.2}$	**40.6**$_{\pm 0.2}$
	Joint	63.4$_{\pm 0.2}$	79.3$_{\pm 0.2}$	29.3$_{\pm 0.1}$	49.5$_{\pm 0.2}$	70.6$_{\pm 0.2}$	87.4$_{\pm 0.2}$	74.3$_{\pm 0.2}$	77.3$_{\pm 0.2}$	44.4$_{\pm 0.1}$	46.0$_{\pm 0.1}$	56.4$_{\pm 0.2}$	67.9$_{\pm 0.2}$
Imbalanced	SFT	11.1$_{\pm 0.4}$	20.8$_{\pm 0.5}$	2.2$_{\pm 0.3}$	6.9$_{\pm 0.4}$	25.8$_{\pm 0.3}$	50.6$_{\pm 0.3}$	66.3$_{\pm 0.3}$	69.3$_{\pm 0.4}$	**64.5**$_{\pm 0.2}$	**67.9**$_{\pm 0.2}$	34.0$_{\pm 0.3}$	43.1$_{\pm 0.4}$
	SPD [91]	18.1$_{\pm 0.2}$	29.9$_{\pm 0.3}$	3.3$_{\pm 0.4}$	9.5$_{\pm 0.5}$	27.6$_{\pm 0.3}$	52.2$_{\pm 0.3}$	70.2$_{\pm 0.2}$	73.3$_{\pm 0.3}$	50.2$_{\pm 0.3}$	51.7$_{\pm 0.3}$	33.9$_{\pm 0.3}$	43.3$_{\pm 0.3}$
	LwF [22]	15.8$_{\pm 0.4}$	27.1$_{\pm 0.4}$	2.8$_{\pm 0.4}$	8.7$_{\pm 0.5}$	21.7$_{\pm 0.3}$	46.6$_{\pm 0.3}$	67.4$_{\pm 0.1}$	71.3$_{\pm 0.3}$	29.2$_{\pm 0.1}$	29.9$_{\pm 0.2}$	27.4$_{\pm 0.3}$	36.7$_{\pm 0.3}$
	CRL [18]	25.0$_{\pm 0.2}$	38.5$_{\pm 0.2}$	3.9$_{\pm 0.3}$	11.7$_{\pm 0.4}$	29.4$_{\pm 0.3}$	53.8$_{\pm 0.3}$	74.0$_{\pm 0.3}$	77.3$_{\pm 0.3}$	35.2$_{\pm 0.3}$	35.0$_{\pm 0.3}$	33.5$_{\pm 0.3}$	43.3$_{\pm 0.3}$
	AKA [21]	26.8$_{\pm 0.2}$	41.2$_{\pm 0.3}$	3.9$_{\pm 0.3}$	11.6$_{\pm 0.5}$	31.1$_{\pm 0.2}$	55.6$_{\pm 0.3}$	75.6$_{\pm 0.2}$	78.4$_{\pm 0.3}$	43.9$_{\pm 0.1}$	44.6$_{\pm 0.2}$	36.3$_{\pm 0.2}$	46.3$_{\pm 0.3}$
	MEGE	**30.1**$_{\pm 0.2}$	**46.1**$_{\pm 0.2}$	**5.7**$_{\pm 0.2}$	**16.4**$_{\pm 0.3}$	**33.1**$_{\pm 0.2}$	**56.5**$_{\pm 0.3}$	**77.6**$_{\pm 0.1}$	**80.5**$_{\pm 0.2}$	44.1$_{\pm 0.2}$	45.3$_{\pm 0.3}$	**38.1**$_{\pm 0.2}$	**49.1**$_{\pm 0.3}$
	Joint	66.7$_{\pm 0.1}$	80.2$_{\pm 0.3}$	35.6$_{\pm 0.2}$	58.8$_{\pm 0.2}$	75.9$_{\pm 0.2}$	89.3$_{\pm 0.2}$	90.4$_{\pm 0.2}$	91.7$_{\pm 0.3}$	51.5$_{\pm 0.1}$	52.4$_{\pm 0.3}$	64.0$_{\pm 0.2}$	74.5$_{\pm 0.3}$



(a) Balanced *(b) Imbalanced*

Fig. 3: Performance tendency of seen domains with increase of the training stages following *Order-1*.



(a) Balanced *(b) Imbalanced*

Fig. 4: Performance tendency of seen domains with increase of the training stages following *Order-2*.

Balanced Evaluation. For both orders, our MEGE achieves the best mAP and rank-1 accuracy on the first four training domains. Although SFT achieves high performance on the last domain, it produces poor performance on old domains. For the performance averaged on all seen domains, our MEGE significantly outperforms the compared methods, demonstrating that MEGE can effectively mitigate catastrophic forgetting. Specifically, MEGE is higher than CRL by 4.1% and 4.0% in average R-1 on the Order-1 and Order-2 respectively. On the other hand, we can find that there is still a large margin between our MEGE and the upper-bound method (Joint Training), especially on the early trained domains.

Imbalanced Evaluation. Compared with the balanced evaluation protocol, the imbalance evaluation protocol includes more training data. This leads all the methods achieve commonly higher performance on the seen domains. Nevertheless, our MEGE obtains a similar advantage as in the balanced evaluation protocol and achieves the best anti-forgetting performance. Concretely, our MEGE outperforms CRL by 4.7% and 4.8% in R-1 on the Order-1 and Order-2 respectively.

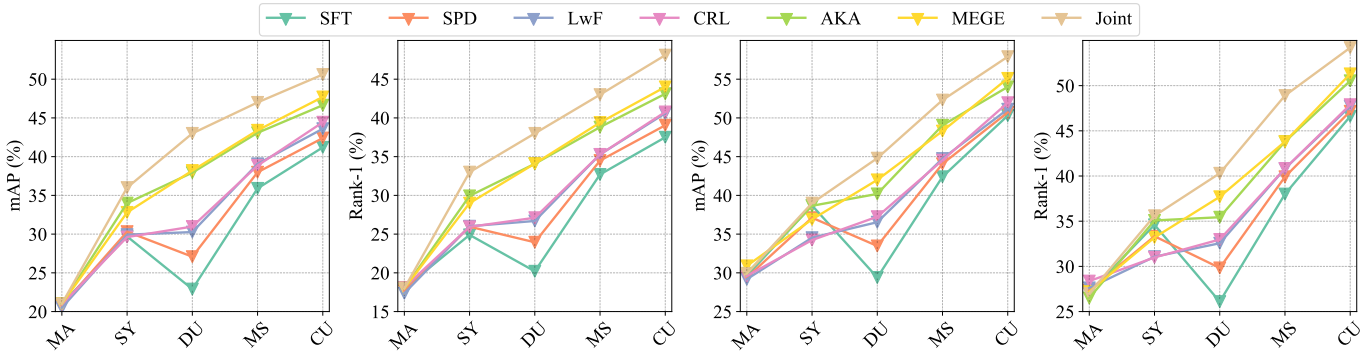
Forgetting Tendency. In Fig. 3 and Fig. 4, we track the performance of the first training domain with the incremental training stages. We can make the following observations. First, the results of all methods decrease with the training stages. Second, our MEGE consistently obtains higher performance than other methods through the training stages. Third, for the joint training method, the performance on the first domain could be improved by training with more datasets. These observations again verify the consistent anti-forgetting advantage of our MEGE and show the gap to the upper-bound method.

6.3 Unseen-domain Generalising Evaluation

To evaluate the generalization ability, we evaluate the results on unseen domains of our Alpha-LReID and the CRL-ReID setting [18].

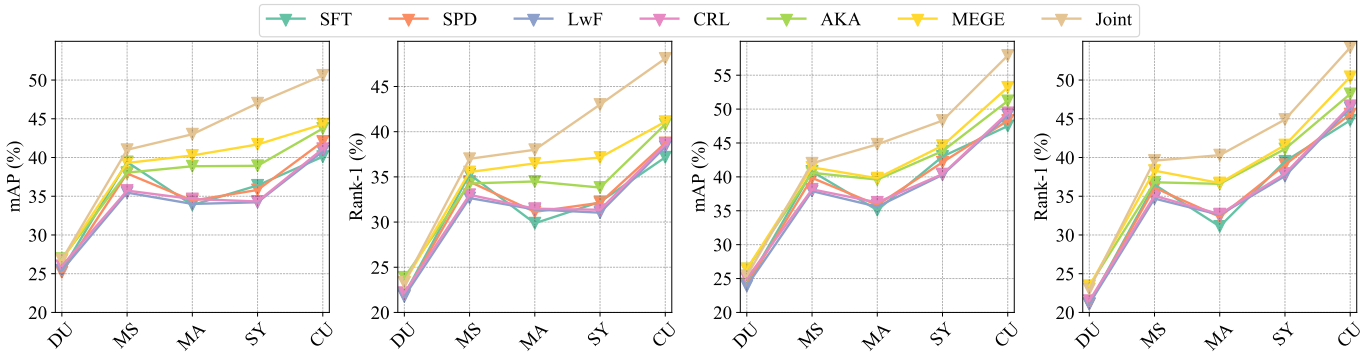
TABLE 5: Generalizing evaluation on unseen-domains.

Training Order	Protocol	Average Unseen $\bar{u}$	SFT	SPD [91]	LwF [22]	CRL [18]	AKA [21]	MEGE	Joint
Order-1	Balanced	mAP	$41.2_{\pm 0.3}$	$42.4_{\pm 0.3}$	$43.6_{\pm 0.2}$	$44.0_{\pm 0.2}$	$46.6_{\pm 0.3}$	$\mathbf{47.7}_{\pm 0.3}$	$50.6_{\pm 0.3}$
		R-1	$37.5_{\pm 0.3}$	$39.0_{\pm 0.3}$	$40.6_{\pm 0.3}$	$41.0_{\pm 0.2}$	$43.1_{\pm 0.2}$	$\mathbf{44.0}_{\pm 0.2}$	$48.1_{\pm 0.3}$
	Imbalanced	mAP	$50.3_{\pm 0.3}$	$50.7_{\pm 0.5}$	$51.1_{\pm 0.3}$	$51.5_{\pm 0.4}$	$54.0_{\pm 0.2}$	$\mathbf{55.1}_{\pm 0.2}$	$57.9_{\pm 0.3}$
		R-1	$46.6_{\pm 0.6}$	$47.2_{\pm 0.5}$	$47.7_{\pm 0.4}$	$48.1_{\pm 0.4}$	$50.5_{\pm 0.2}$	$\mathbf{51.3}_{\pm 0.2}$	$54.2_{\pm 0.3}$
Order-2	Balanced	mAP	$40.1_{\pm 0.3}$	$42.1_{\pm 0.3}$	$40.8_{\pm 0.3}$	$40.9_{\pm 0.2}$	$43.7_{\pm 0.2}$	$\mathbf{44.3}_{\pm 0.2}$	$50.6_{\pm 0.2}$
		R-1	$37.2_{\pm 0.4}$	$38.7_{\pm 0.5}$	$38.3_{\pm 0.3}$	$39.0_{\pm 0.3}$	$40.8_{\pm 0.2}$	$\mathbf{41.1}_{\pm 0.3}$	$48.1_{\pm 0.2}$
	Imbalanced	mAP	$47.5_{\pm 0.4}$	$48.3_{\pm 0.5}$	$49.1_{\pm 0.3}$	$49.2_{\pm 0.3}$	$51.2_{\pm 0.2}$	$\mathbf{53.2}_{\pm 0.3}$	$57.9_{\pm 0.2}$
		R-1	$44.8_{\pm 0.5}$	$45.6_{\pm 0.3}$	$46.4_{\pm 0.3}$	$47.0_{\pm 0.3}$	$48.2_{\pm 0.2}$	$\mathbf{50.4}_{\pm 0.3}$	$54.2_{\pm 0.2}$



(a) Balanced *(b) Imbalanced*

Fig. 5: Performance tendency of unseen domains with increase of the training stages following *Order-1*.



(a) Balanced *(b) Imbalanced*

Fig. 6: Performance tendency of unseen domains with increase of the training stages following *Order-2*.

Evaluation on Alpha-LReID. As shown in Tab. 5, our method consistently outperforms the state-of-the-art-methods regardless of training orders and evaluation protocols, which verifies the superiority of our method in improving generalization ability. Specifically, our MEGE outperforms the methods by a large margin, except for AKA. Although MEGE is the extension of AKA for improving the anti-forgetting performance, MEGE also achieves slightly better results than AKA on unseen domains.

Similar to the results on seen domains, a model trained under the imbalanced evaluation protocol obtains higher results than the balanced one. On the other hand, the gap between our MEGE and the upper-bound method is small on unseen domains, which is different from that of the seen domains.

Generalizing Tendency. In Fig. 5 and Fig. 6, we illustrate the trend of the performance on unseen domains with the incremental training stages. We can find the following observations. First, in most cases, the results of all methods are increased by training with more datasets. However, in both orders, LwF, SFT and SPD will encounter a performance degradation when training on a certain domain. For example, when training under the *Order-2*, the results of LwF, SFT and SPD decrease at the training stage of SY domain. Second, both our MEGE and AKA consistently improve the performance with the training stages. The above two phenomena further demonstrate the effectiveness of our MEGE and AKA in learning generalized representation in LReID.

Evaluation on CRL-ReID. We also evaluate our method under the CRL setting [18]. Results in Tab. 6 show that our MEGE outperforms all the compared methods by a large margin. In addition, by comparing between the unseen results produced in our setting and CRL setting that both undergone 5 learning steps, our MEGE achieves significantly higher results in the CRL-ReID (5-step). For example, the best mAP achieved in Tab. 5 is 55.1% in our Alpha-LReID setting, which is largely lower than the one (64.5% in Tab. 6) obtained in CRL-ReID setting (5-step). This verifies the difficulty of our Alpha-LReID setting.

TABLE 6: Generalization evaluation under the CRL setting in [18].

Banchmark	$\bar{u}$	SFT	SPD	LwF	CRL	MEGE	Joint
CRL-ReID (5-step)	mAP	44.2 ± 0.2	47.1 ± 0.2	48.7 ± 0.2	51.2 ± 0.2	**64.5** ± 0.1	64.8 ± 0.2
	R-1	53.4 ± 0.3	54.1 ± 0.4	59.6 ± 0.2	62.8 ± 0.3	**75.0** ± 0.2	75.3 ± 0.2
CRL-ReID (10-step)	mAP	31.7 ± 0.2	40.3 ± 0.3	42.8 ± 0.2	43.8 ± 0.3	**51.2** ± 0.2	64.8 ± 0.2
	Rank-1	40.3 ± 0.4	47.5 ± 0.4	51.7 ± 0.2	54.7 ± 0.4	**60.1** ± 0.2	75.3 ± 0.2

TABLE 7: Evaluation of the loss functions of MEGE in Order-1 under the imbalanced evaluation protocol. $\mathcal{L}_p$: plasticity loss, $\mathcal{L}_s$: stability loss , $\mathcal{L}_{sfd}$: SFD-knowledge distillation, $\mathcal{L}_{rcd}$: SFD-knowledge distillation with adaptive weighting.

#ID	Setting	Average Seen $\bar{s}$		Average Unseen $\bar{u}$		H-metric	
		mAP	Rank-1	mAP	Rank-1	mAP	Rank-1
1	Baseline (LwF [22])	35.3± 0.2	44.5± 0.3	51.1± 0.3	47.7± 0.4	41.8± 0.2	46.0± 0.3
2	Baseline + GCN	35.6± 0.2	44.6± 0.3	51.3 ± 0.1	47.6 ± 0.3	42.0± 0.2	46.1± 0.3
3	+ $\mathcal{L}_p$	35.2± 0.2	45.0± 0.2	53.4 ± 0.1	50.3 ± 0.3	42.4± 0.1	47.5± 0.2
4	+ $\mathcal{L}_p$ + $\mathcal{L}_s$ (Full AKA)	**38.1**± 0.1	**47.8**± 0.2	**54.0**± 0.2	**50.5**± 0.2	**44.7**± 0.2	**49.1**± 0.2
5	+ $\mathcal{L}_{sfd}$	37.7± 0.2	47.2± 0.2	46.9± 0.1	43.7± 0.2	41.8± 0.1	45.4± 0.2
6	+ $\mathcal{L}_{rcd}$ (Full RCD)	38.4± 0.2	49.0± 0.2	51.3 ± 0.2	47.1± 0.2	43.9± 0.2	48.0± 0.2
7	+ $\mathcal{L}_p$ + $\mathcal{L}_s$ + $\mathcal{L}_{rcd}$ (Our MEGE)	**40.0**± 0.2	**50.2**± 0.3	**55.1**± 0.2	**51.3** ± 0.2	**46.4** ± 0.2	**50.7**± 0.3

TABLE 8: Evaluation of the different designs of edge weight for ISG, AKG and KO in AKA. Experiments are conducted in Order-1 under the imbalanced evaluation protocol. PA: parameterized weight, N-PA: non-parameterized weight.

ISG	AKG	KO	Average Seen $\bar{s}$		Average Unseen $\bar{u}$		H-metric	
			mAP	R-1	mAP	R-1	mAP	R-1
PA	PA	PA	37.7± 0.3	48.3 ± 0.4	53.1± 0.3	50.0± 0.3	44.1± 0.3	49.1± 0.3
PA	PA	N-PA	**38.1**± 0.1	47.8± 0.2	**54.0**± 0.2	**50.5**± 0.2	**44.7**± 0.2	**49.1**± 0.2
PA	N-PA	N-PA	36.1 ± 0.2	46.9 ± 0.2	51.3± 0.3	48.0± 0.3	42.4± 0.3	47.4 ± 0.3
N-PA	PA	N-PA	37.6± 0.1	**48.5** ± 0.2	52.0± 0.2	48.5± 0.2	43.6± 0.2	48.5± 0.2
N-PA	N-PA	N-PA	36.9± 0.1	47.8 ± 0.1	51.6± 0.1	48.2± 0.2	43.0± 0.1	48.0± 0.2

6.4 Effectiveness Evaluation

In this section, we conduct extensive experiments to investigate the effectiveness of each component of MEGE. All the experiments are evaluated in *Order-1* under the imbalanced evaluation protocol. The baseline method is LwF [22], which uses logit-based knowledge distillation to prevent catastrophic forgetting.

Effectiveness of AKA. In Tab. 7, we report the results of adding different components of MEGE into the baseline. We first evaluate the effectiveness of AKA in the first four rows of Tab. 7. We consider building a straightforward KG-based baseline by adding a AKA module without any additional loss on the top of LwF method, namely "Baseline + GCN" in Tab. 7. Specifically, we directly feed the fused feature in Equ.(10) to the identification classifier $g(\cdot;\phi)$ and jointly optimize the backbone network and the graph convolution network. The table shows that without the proposed stability-plasticity loss, the AKA module cannot effectively improve the model's generalization ability, due to the lack of the constraint to learn knowledge selectively. Moreover, we can find that the plasticity loss ($\mathcal{L}_p$) is mainly beneficial for unseen domains. This indicates that AKG is encouraged to learn how to transfer positive knowledge to improve generalization. Adding the stability loss further improves the performance on both seen and unseen domains. This indicates that enforcing the stability of knowledge during training can largely preserve the knowledge of previous domains and thus remits the influence of catastrophic forgetting. Meanwhile, the improvement on unseen domains demonstrates that the stability loss can also improve the generalization ability of the model, due to effectively accumulating generalizable knowledge.

Effectiveness of RCD. In the row#4-row#5 of Tab. 7, we show the impact of two variants of RCD. We can find four observations. First, the two variants of RCD can consistently improve the performance on seen domains. This verifies the effectiveness of distilling ranking information for solving the catastrophic forgetting problem in LReID. Second, the fixed weighting version of RCD ($\mathcal{L}_{sfd}$) hampers the performs on unseen domains. Third, the dynamic weighting version ($\mathcal{L}_{rcd}$) can well address the above problem and further increases the performance on unseen domains over the baseline. This indicates that learning dynamic weights during ranking distillation can encourage the model learn more generalized representation instead of overfitting on seen domains. Fourth, the proposed AKA and RCD are complementary to each other. Combining them achieves the best results in seen domains and unseen domains.

TABLE 9: Effects of the number of GCN layers in AKA. Experiments are conducted in the Order-1 under the imbalanced evaluation protocol.

# GCN	Average Seen $\bar{s}$		Average Unseen $\bar{u}$		H-metric	
	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1
1	38.1	47.8	54.0	50.5	44.7	49.1
2	37.8	47.5	54.3	50.6	44.6	49.0
3	37.9	47.8	53.8	50.2	44.5	49.0

Evaluation of design of edge weight for AKA. In our AKA, we use different designs (parameterized or non-parameterized) of edge weight for ISG, AKG and KO. In Tab. 8, we conduct experiments to investigate the impact of using different designs. We can observe that: 1) using parameterized design for ISG and AKG leads to clearly higher results 2) while applying non-parameterized design for KO produces better performance especially for the H-metric that reflects the balance between anti-forgetting and generalizing abilities. These results verify the effectiveness and motivation of using different designs of edge weights for knowledge representation and knowledge operation as discussed in Sec. 4.2.

Effects of the number of GCN layers in AKA. In Tab. 9,

TABLE 10: Effects of using different weighting manners for RCD. Experiments are conducted in Order-1 under the imbalanced evaluation protocol. EW: qual weights, LDW: linear decrease weight, EDW: exponential decrease weight, LIW: linear increase weight (LIW), EIW: exponential increase weight, LPW: learned prior weight, DW: the proposed dynamic weight, RE: re-initializing weights at each domain.

Weighting Strategy	Average Seen $\bar{s}$		Average Unseen $\bar{u}$		H-metric	
	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1
EW	37.7	47.2	46.9	43.7	41.8	45.4
LDW	37.1	46.6	48.0	44.1	41.8	45.3
EDW	36.5	45.8	50.1	46.2	42.2	46.0
LIW	32.0	42.8	46.3	42.6	37.8	42.7
EIW	29.5	40.3	44.5	40.6	35.5	40.4
LPW	36.9	46.1	49.7	46.0	42.4	46.0
DW	37.2	46.8	50.9	46.8	43.0	46.8
DW + RE	**38.4**	**49.0**	**51.3**	**47.1**	**43.9**	**48.0**

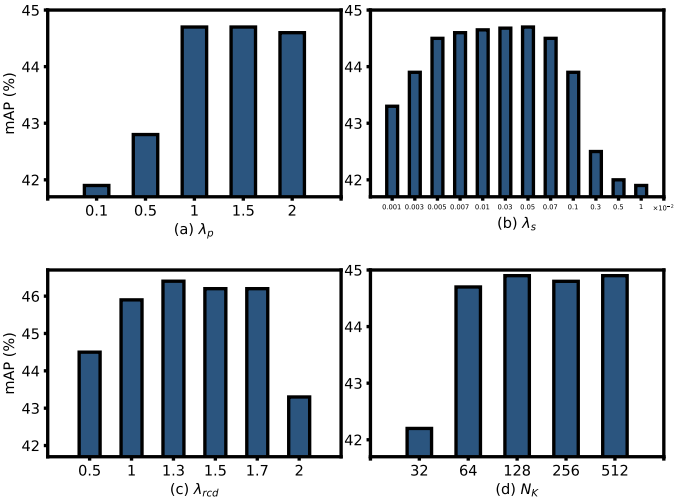


Fig. 7: Impact of hyper-parameters. H-metric is reported.

we analyze the impact of the number of GCN layers in AKA. We can observe that stacking more GCN layers does not achieve clear improvements and even reduces the anti-forgetting performance. Thus, we employ one-layer GCN in our AKA for simplicity and superiority.

Effects of different weighting manners for RCD. In our RCD, we adaptive learn position weights during training. To verify the effectiveness of this adaptive manner, we compare it with several variants that uses fixed position weights, including equal weights (EW), linear decrease weight (LDW), exponential decrease weight (EDW), linear increase weight (LIW), exponential increase weight (EIW), and learned prior weight (LPW). For EW, we use the same weight for all positions. For LDW, EDW, LIW, and EIW, the weights are linearly/exponentially changed with the increase/decrease order of positions. For LPW, we first learn the position weights of each training domain and obtained the prior weights by averaging them based on domains. Then, we use the fixed prior weights to train the model in a new training process. Results in Tab. 10 show that 1) using a proper fixed weighting strategy can improves the performance on unseen domains and that 2) the proposed learnable weighting strategy achieves better results than all fixed weighting strategies. These results demonstrate the advantage of our learnable weighting strategy. In addition, the proposed learnable weighting strategy is more flexible since it is automatically learned. On the other hand, we also show that re-initializing the weights instead of inheriting the weights obtained by the last domain leads to better performance. The main reason is that the importance of each position will be different at each training epoch and thus the weights should be re-initialized and re-learned at the beginning of each stage.

6.5 Hyper-Parameter Analysis

In this section, we discuss the impact of the hyper-parameters in our MEGE, including loss weights (λ_p and λ_s, λ_{rcd}) and the number of knowledge nodes (N^K). We adopt a harmonic mean of the average accuracy of seen and unseen domains as the performance metric, which reflects both anti-forgetting and generalization abilities.

Impact of weights. For evaluation of loss weights, we first select the optimal λ_p to achieve best $\bar{u}$, then we search the optimal λ_s based on the selected λ_p. We find that λ_s should be assigned with a small value and it is stable within a range of 5×10^{-5} to 7×10^{-4}. Finally, we choose the best λ_{rcd} with the selected λ_p and λ_s. The impact of different values of them are shown in Fig. 7(a-c). In this manner, our final model is obtained by using $\lambda_p = 1$, $\lambda_s = 5 \times 10^{-4}$ and $\lambda_{rcd} = 1.3$.

Impact of number of knowledge nodes. In a similar way, we study the influence of the number of knowledge nodes (N^K) on the hold-off validation data. We vary the value of N^K in the range of $\{32, 64, 128, 256, 512\}$. Results in Fig. 7(d) show that the performance increases from $N^K = 32$ to $N^K = 64$ and the performance is stable between 64 and 256. Considering the balance between memory consumption and performance, we thus set $N^K = 64$ in all experiments.

Impact of the size of mini-batch. We evaluate the impact of mini-batch size in Tab. 11. The results can be summarized as follows. 1) Our method is robust to the batch size and using a larger batch size commonly leads to slightly higher results. 2) With the increase of N^b, the training time of our method grows up fast, because its complexity is a quadratic function of batch size. Considering the balance between the comprehensive performance and training cost, we set N^b to 64 in all our experiments.

6.6 Evaluation of Training Cost

In this section, we conduct experiments to estimate and discuss the complexity of the different methods in terms of training time and GPU memory cost.

Comparison of the proposed modules and other state-of-the-art methods. Based on the experimental results in Tab. 12, we find that 1) the proposed AKA and MEGE enjoy a neglectable memory overhead compared to other methods while obtaining considerable improvement, especially on unseen domains; 2) although our MEGE costs relatively longer training time than CRL by 0.07s per iteration, MEGE significantly outperforms CRL on both seen and unseen domains.

Comparison of different differentiable ranking approaches. Since the proposed RCD is agnostic to the differentiable ranking function, we provide the comparison of

TABLE 11: Evaluation of running time and memory for varying sizes of mini-batch.

N^b	Training time (s/iter)	GPU memory (MB)	Average Seen mAP	Average Seen Rank-1	Average Unseen mAP	Average Unseen Rank-1	H-metric mAP	H-metric Rank-1
32	≈0.281	≈3921	$39.8_{\pm 0.2}$	$50.0_{\pm 0.2}$	$55.0_{\pm 0.2}$	$51.1_{\pm 0.4}$	$46.2_{\pm 0.2}$	$50.5_{\pm 0.3}$
64	≈0.509	≈5205	$40.0_{\pm 0.2}$	$50.2_{\pm 0.3}$	$55.1_{\pm 0.2}$	$51.3_{\pm 0.2}$	$46.4_{\pm 0.2}$	$50.7_{\pm 0.3}$
128	≈0.989	≈8285	$40.3_{\pm 0.1}$	$50.4_{\pm 0.1}$	$55.0_{\pm 0.1}$	$51.1_{\pm 0.2}$	$46.5_{\pm 0.1}$	$50.7_{\pm 0.2}$
256	≈1.722	≈11847	$40.4_{\pm 0.1}$	$50.6_{\pm 0.2}$	$54.7_{\pm 0.2}$	$51.0_{\pm 0.1}$	$46.5_{\pm 0.2}$	$50.8_{\pm 0.2}$
512	≈3.147	≈15789	$40.6_{\pm 0.2}$	$50.8_{\pm 0.2}$	$54.3_{\pm 0.1}$	$50.8_{\pm 0.2}$	$46.5_{\pm 0.2}$	$50.8_{\pm 0.2}$

TABLE 12: Evaluation of running time and memory for state-of-the-art methods when N_b=64.

Method	Training time (s/iter)	GPU memory (MB)	Average Seen mAP	Average Seen Rank-1	Average Unseen mAP	Average Unseen Rank-1	H-metric mAP	H-metric Rank-1
SFT	≈0.372	≈4151	$33.0_{\pm 0.3}$	$42.4_{\pm 0.3}$	$50.3_{\pm 0.3}$	$46.6_{\pm 0.6}$	$39.9_{\pm 0.3}$	$44.4_{\pm 0.4}$
LwF [22]	≈0.446	≈5177	$35.3_{\pm 0.2}$	$44.5_{\pm 0.3}$	$51.1_{\pm 0.3}$	$47.7_{\pm 0.4}$	$41.8_{\pm 0.2}$	$46.0_{\pm 0.3}$
CRL [18]	≈0.439	≈5175	$36.1_{\pm 0.2}$	$45.5_{\pm 0.3}$	$51.5_{\pm 0.4}$	$48.1_{\pm 0.4}$	$42.4_{\pm 0.3}$	$46.8_{\pm 0.3}$
AKA (LwF + $\mathcal{G}^S$ and $\mathcal{G}^S$)	≈0.461	≈5195	$38.1_{\pm 0.1}$	$47.8_{\pm 0.2}$	$54.0_{\pm 0.2}$	$50.5_{\pm 0.2}$	$44.7_{\pm 0.2}$	$49.1_{\pm 0.2}$
MEGE (AKA + RCD)	≈0.509	≈5205	$40.0_{\pm 0.2}$	$50.2_{\pm 0.3}$	$55.1_{\pm 0.2}$	$51.3_{\pm 0.2}$	$46.4_{\pm 0.2}$	$50.7_{\pm 0.3}$

TABLE 13: Evaluation of different differentiable ranking approaches.

Method	Training time (s/iter)	GPU memory (MB)	Average Seen mAP	Average Seen Rank-1	Average Unseen mAP	Average Unseen Rank-1
Ours	≈0.509	≈5205	$40.0_{\pm 0.2}$	$50.2_{\pm 0.3}$	$55.1_{\pm 0.2}$	$51.3_{\pm 0.2}$
FDSR [93]	≈0.495	≈5204	$39.9_{\pm 0.3}$	$50.0_{\pm 0.3}$	$54.8_{\pm 0.2}$	$51.2_{\pm 0.3}$

TABLE 14: Evaluation of detaching gradient in AKA. Experiments are conducted in the Order-1 under the imbalanced evaluation protocol.

Detaching	Average Seen $\bar{s}$ mAP	Average Seen $\bar{s}$ Rank-1	Average Unseen $\bar{u}$ mAP	Average Unseen $\bar{u}$ Rank-1	H-metric mAP	H-metric Rank-1
✓	**38.1**	**47.8**	**54.0**	**50.5**	**44.7**	**49.1**
✗	36.4	45.6	51.9	48.7	42.8	47.1

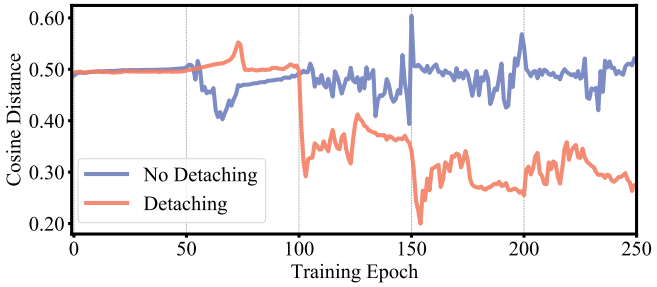


Fig. 8: Difference of ISG representations before and after propagation.

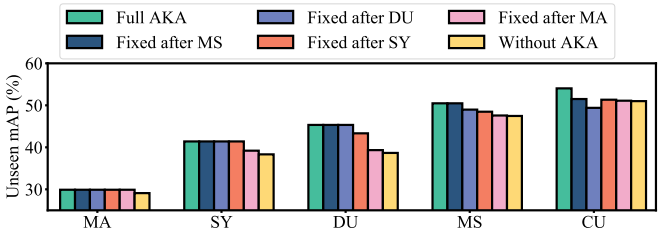


Fig. 9: Evaluation of the generalizability of the models by fixing AKG after a certain domain.

TABLE 15: Evaluation of the generalizability of the models with varying AKGs that is trained and saved on different domains. "S.A." indicates "save after".

Matric	S.A. MA	S.A. SY	S.A. DU	S.A. MS	S.A. CU
mAP	$49.5_{\pm 0.4}$	$51.7_{\pm 0.2}$	$52.0_{\pm 0.3}$	$53.1_{\pm 0.3}$	$54.0_{\pm 0.2}$
Rank-1	$46.5_{\pm 0.5}$	$47.8_{\pm 0.3}$	$48.3_{\pm 0.4}$	$49.6_{\pm 0.5}$	$50.5_{\pm 0.2}$

using the Argsorted Function or FDSR [93] to implement RCD. Tab. 13 shows that these two methods achieve similar results with similar computational costs, indicating that the proposed ranking consistency distillation loss is compatible with different differentiable function.

6.7 Further Investigation

In this section, we conduct four experiments to help us further understand the proposed AKA and RCD.

Investigation on gradient detaching in AKA. In our AKA, we detach the gradients from the graph networks. To verify the effectiveness of this strategy, we compare the results of using detaching and without using detaching in Tab. 14. It is clear that, detaching the gradient of AKA achieves higher performance on all metrics. To help us further understand the effectiveness of the detaching strategy, we compute the difference between the ISG representations before and after propagation through training. Fig. 8 shows that using detaching strategy encourages the AKA to learn different representations from the feature extractor at each training stages, which echos the discussion in Sec. 4.2.

Investigation on the knowledge extension of the AKG. Intuitively, we think that the more knowledge the AKG accumulates, the better the generalization ability the model acquires. To experimentally demonstrate that the knowledge contained in the AKG is extended with the increase of training domains, we conduct two groups of experiments to explore the effects of the knowledge transferred from the AKGs trained on different domains. Experiments are conducted on Order-1 with the imbalanced protocol.

1) We fix all the AKG's parameters after learning on one certain domain (*e.g.*, 1^{st}, 2^{ed}, 3^{rd} and 4^{th} domain), and then continue to train the model with the frozen AKG for the subsequent lifelong learning steps. During this process, we test the model's performance on unseen domains. Results are illustrated in Fig. 9. It is obvious that after fixing the AKG, the model's performance drops to different extents, indicating that the AKG is extended with more knowledge that is favorable for improving generalization ability.

TABLE 16: Evaluation of anti-forgetting and generalization ability on 5 different domain orders.

Protocol	Order	Seen Domains						Unseen Domains					
		mAP			Rank-1			mAP			Rank-1		
		CRL	AKA	MEGE	CRL	AKA	MEGE	CRL	AKA	MEGE	CRL	AKA	MEGE
Balanced	MS→SY→DU→MA→CU	32.9	33.6	**36.2**	41.5	43.7	**48.3**	45.6	48.3	**50.1**	42.4	44.8	**45.9**
	DU→MA→CU→MS→SY	33.2	34.7	**38.8**	44.3	45.8	**49.7**	38.1	40.6	**41.9**	35.6	37.6	**39.0**
	SY→DU→CU→MS→MA	36.3	37.0	**42.3**	44.9	45.1	**49.4**	37.4	38.6	**40.2**	35.0	35.7	**36.3**
	CU→MS→DU→MA→SY	34.4	34.7	**38.7**	43.8	44.6	**49.1**	40.9	43.5	**44.0**	38.3	39.7	**40.8**
	MA→MS→DU→SY→CU	30.3	31.1	**33.8**	39.9	41.5	**43.7**	43.4	45.6	**47.3**	39.9	43.0	**44.2**
	Average	33.4	34.2	**38.0**	42.9	44.1	**48.0**	41.1	43.3	**44.7**	38.2	40.2	**41.2**
Imbalanced	MS→SY→DU→MA→CU	39.0	40.8	**44.6**	43.5	44.9	**48.2**	50.3	54.1	**55.6**	47.5	51.2	**52.3**
	DU→MA→CU→MS→SY	39.7	41.7	**45.0**	44.2	45.1	**49.0**	46.4	50.2	**51.3**	44.4	47.3	**48.8**
	SY→DU→CU→MS→MA	45.1	46.4	**49.3**	48.5	49.8	**51.8**	44.0	45.7	**46.5**	44.2	46.5	**47.6**
	CU→MS→DU→MA→SY	43.5	45.1	**48.9**	46.2	47.1	**51.3**	48.3	52.9	**54.1**	46.0	50.5	**51.8**
	MA→MS→DU→SY→CU	38.6	40.1	**43.2**	47.9	48.5	**51.9**	46.8	51.0	**51.7**	46.5	49.0	**49.2**
	Average	41.2	42.8	**46.2**	46.1	47.1	**50.4**	47.2	50.8	**51.8**	45.7	48.9	**49.9**

2) We store AKGs at the end of each domain-incremental training. Then, we combine the different AKGs with the trained backbone network, which is evaluated on unseen domains. The results in Tab. 15 demonstrate that the AKG that experiences more domains can provide more beneficial knowledge for generalization evaluation.

Investigation on adaptive weight in RCD. To better understand the proposed adaptive weighting method, we track the variations of weights of each position during the whole training epochs. We observe an interesting phenomenon from Fig. 10. The learned weights follow a similar tendency through the training epochs at each domain. Specifically, the top and the bottom positions are gradually assigned with relatively small weights, while the middle positions are consistently assigned with large weights. This phenomenon is reasonable, since in person ReID, the model can well learn the pattern of easy position and negative samples that rank at top/bottom positions in the beginning of training. As the increase of training epochs, the model should pay more attention on hard position and negative samples that are ranked at the middle positions and are more important in learning informative patterns. As a result, our RCD learns an adaptive weighting manner that always assigning higher weights to hard samples during training. Importantly, as reported in Tab. 10, our RCD is more flexible and superior than manually fixed weighting strategies. This is because that the importance of each position is changed at each training epoch and each domain. For example, in the beginning of training epochs, the top and bottom positions should be assigned with high weights since the model have not learn too much from them. While with the increase of training epochs, these easy samples cannot contribute too much for training and should be assigned with lower weights. Our RCD can dynamically adapt the above tendency. However, a manual strategy commonly assigns fixed weighting for each position and thus fails to follow the above tendency.

Investigation on different training orders. To verify the robustness of the proposed methods on varying training orders, we conduct more experiments with different domain orders. The experimental results in Tab. 16 are summarized as: 1) Our MEGE and AKA significantly outperform CRL [18] for different orders on both balanced and imbalanced protocols; 2) Our MEGE achieves consistent improvement over AKA on all cases, especially for the anti-forgetting performance evaluated on seen domains.

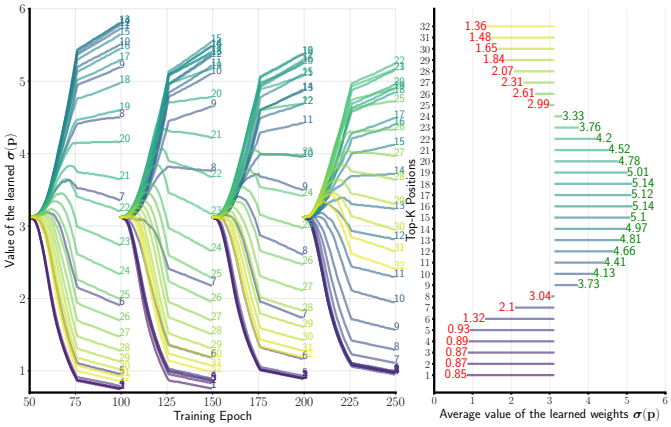


Fig. 10: Tendency of weights obtained by the proposed adaptive position weighting. Left: weight changes of each position during the incremental training process. Right: The average weights of different positions overall the whole training process.

7 Conclusion

In this paper, we introduce the challenging yet practical ReID setting, lifelong person re-identification (LReID). To solve this problem, we propose a new MEmorizing and GEneralizing framework (MEGE) by injecting an Adaptive Knowledge Accumulation (AKA) module and a Ranking Consistency Distillation (RCD) module into the LReID system. The AKA maintains a transferable knowledge graph to adaptively keep the previous knowledge as well as learn generalizable representation. The RCD encourages the model to inherit more informative knowledge of previous domains by distilling ranking results in a differentiable and dynamic manner. Extensive experiments demonstrate that our MEGE can significantly improve the model's anti-forgetting and generalization abilities and can outperform other competitors by large margins on our Alpha-LReID benchmark. Nevertheless, there is still a large margin to the performance of the upper-bound on seen domains, remaining a large room in improving the model's anti-forgetting ability in future study. In our future work, we aim to extend our LReID setting to include cloth-changing scenarios, which pose more challenges but are also more relevant to real-world ReID applications. In addition, we also plan to further investigate and develop a more appropriate strategy for tuning hyperparameters within the LReID setting.

REFERENCES

[1] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," *arXiv preprint arXiv:1610.02984*, 2016.

[2] M. Gou, Z. Wu, A. Rates-Borras, O. Camps, R. J. Radke *et al.*, "A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018.

[3] F. Zhao, S. Liao, G.-S. Xie, J. Zhao, K. Zhang, and L. Shao, "Unsupervised domain adaptation with noise resistible mutual-training for person re-identification," in *ECCV*, 2020.

[4] C. Ding, K. Wang, P. Wang, and D. Tao, "Multi-task learning with coarse priors for robust part-aware person re-identification," *IEEE TPAMI*, vol. 44, no. 3, pp. 1474–1488, 2022.

[5] J. Li, S. Zhang, Q. Tian, M. Wang, and W. Gao, "Pose-guided representation learning for person re-identification," *IEEE TPAMI*, vol. 44, no. 2, pp. 622–635, 2022.

[6] Y. Shen, T. Xiao, S. Yi, D. Chen, X. Wang, and H. Li, "Person re-identification with deep kronecker-product matching and group-shuffling random walk," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1649–1665, 2021.

[7] N. Pu, W. Chen, Y. Liu, E. M. Bakker, and M. S. Lew, "Dual gaussian-based variational subspace disentanglement for visible-infrared person re-identification," in *ACM MM*, 2020.

[8] Q. Yang, A. Wu, and W. Zheng, "Person re-identification by contour sketch under moderate clothing change," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 2029–2046, 2021.

[9] J. Song, Y. Yang, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Generalizable person re-identification by domain-invariant mapping network," in *CVPR*, 2019.

[10] X. Jin, C. Lan, W. Zeng, Z. Chen, and L. Zhang, "Style normalization and restitution for generalizable person re-identification," in *CVPR*, 2020.

[11] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *ICCV*, 2015.

[12] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," in *ICCV*, 2017.

[13] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person transfer gan to bridge domain gap for person re-identification," in *CVPR*, 2018.

[14] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*, 1989, pp. 109–165.

[15] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *CVPR*, 2017.

[16] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, 2015.

[18] B. Zhao, S. Tang, D. Chen, H. Bilen, and R. Zhao, "Continual representation learning for biometric identification," in *WACV*, 2021.

[19] R. A. Cowell, M. D. Barense, and P. S. Sadil, "A roadmap for understanding memory: Decomposing cognitive processes into operations and representations," *Eneuro*, 2019.

[20] W.-C. Wang, N. M. Brashier, E. A. Wing, E. J. Marsh, and R. Cabeza, "Knowledge supports memory retrieval through familiarity, not recollection," *Neuropsychologia*, 2018.

[21] N. Pu, W. Chen, Y. Liu, E. M. Bakker, and M. S. Lew, "Lifelong person re-identification via adaptive knowledge accumulation," in *CVPR*, 2021.

[22] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE TPAMI*, 2017.

[23] C. Spearman, "The proof and measurement of association between two things," *The American journal of psychology*, 1987.

[24] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Learning generalisable omni-scale representations for person re-identification," *IEEE TPAMI*, 2021.

[25] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang, "Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline)," in *ECCV*, 2018.

[26] L. He, Y. Wang, W. Liu, H. Zhao, Z. Sun, and J. Feng, "Foreground-aware pyramid reconstruction for alignment-free occluded person re-identification," in *ICCV*, 2019.

[27] K. Li, Z. Ding, K. Li, Y. Zhang, and Y. Fu, "Support neighbor loss for person re-identification," in *ACM MM*, 2018.

[28] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[29] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "Joint detection and identification feature learning for person search," in *CVPR*, 2017.

[30] D. Chen, D. Xu, H. Li, N. Sebe, and X. Wang, "Group consistent similarity learning via deep crf for person re-identification," in *CVPR*, 2018.

[31] Z. Zhong, L. Zheng, Z. Luo, S. Li, and Y. Yang, "Learning to adapt invariance in memory for person re-identification," *IEEE TPAMI*, 2020.

[32] Y. Lin, X. Dong, L. Zheng, Y. Yan, and Y. Yang, "A bottom-up clustering approach to unsupervised person re-identification," in *AAAI*, 2019.

[33] 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 *CVPR*, 2021.

[34] W.-H. Li, Z. Zhong, and W.-S. Zheng, "One-pass person re-identification by sketch online discriminant analysis," *Pattern Recognition*, 2019.

[35] P. Xu and X. Zhu, "Deepchange: A large long-term person re-identification benchmark with clothes change," *arXiv preprint arXiv:2105.14685*, 2021.

[36] M. Li, P. Xu, C.-G. Li, and J. Guo, "Maskcl: Semantic mask-driven contrastive learning for unsupervised person re-identification with clothes change," *arXiv preprint arXiv:2305.13600*, 2023.

[37] M. Li, P. Xu, X. Zhu, and J. Guo, "Unsupervised long-term person re-identification with clothes change," *arXiv preprint arXiv:2202.03087*, 2022.

[38] S. Yu, S. Li, D. Chen, R. Zhao, J. Yan, and Y. Qiao, "Cocas: A large-scale clothes changing person dataset for re-identification," in *CVPR*, 2020.

[39] X. Gu, H. Chang, B. Ma, S. Bai, S. Shan, and X. Chen, "Clothes-changing person re-identification with rgb modality only," in *CVPR*, 2022.

[40] A. Pentina and C. Lampert, "A pac-bayesian bound for lifelong learning," in *ICML*, 2014.

[41] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *CVPR*, 2019.

[42] R. Aljundi, M. Rohrbach, and T. Tuytelaars, "Selfless sequential learning," *arXiv preprint arXiv:1806.05421*, 2018.

[43] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *ECCV*, 2018.

[44] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental learning of object detectors without catastrophic forgetting," in *ICCV*, 2017.

[45] C. Wu, L. Herranz, X. Liu, J. van de Weijer, B. Raducanu *et al.*, "Memory replay gans: Learning to generate new categories without forgetting," in *NeurIPS*, 2018.

[46] D. Abel, D. Arumugam, L. Lehnert, and M. Littman, "State abstractions for lifelong reinforcement learning," in *ICML*, 2018.

[47] C. Kaplanis, M. Shanahan, and C. Clopath, "Continual reinforcement learning with complex synapses," in *ICML*, 2018.

[48] D. Rao, F. Visin, A. Rusu, R. Pascanu, Y. W. Teh, and R. Hadsell, "Continual unsupervised representation learning," in *NeurIPS*, 2019.

[49] K. Wei, C. Deng, and X. Yang, "Lifelong zero-shot learning," in *IJCAI*, 2020.

[50] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," *arXiv preprint arXiv:1708.01547*, 2017.

[51] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[52] V. Lomonaco and D. Maltoni, "Core50: a new dataset and benchmark for continuous object recognition," *arXiv preprint arXiv:1705.03550*, 2017.

[53] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

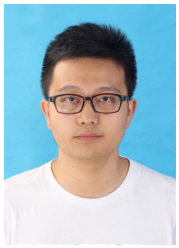
[54] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *CVPR*, 2014.

[55] J. Liu, Y. Sun, C. Han, Z. Dou, and W. Li, "Deep representation learning on long-tailed data: A learnable embedding augmentation perspective," in *CVPR*, 2020.

[56] W. Chen, Y. Liu, W. Wang, T. Tuytelaars, E. M. Bakker, and M. Lew, "On the exploration of incremental learning for fine-grained image retrieval," in *BMVC*, 2020.

[57] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

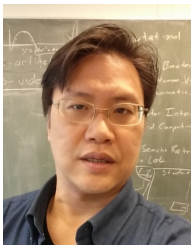
[58] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.
[59] M. Narasimhan, S. Lazebnik, and A. Schwing, "Out of the box: Reasoning with graph convolution nets for factual visual question answering," in *NeurIPS*, 2018.
[60] T. Yao, Y. Pan, Y. Li, and T. Mei, "Exploring visual relationship for image captioning," in *ECCV*, 2018.
[61] O. Litany, A. Bronstein, M. Bronstein, and A. Makadia, "Deformable shape completion with graph convolutional autoencoders," in *CVPR*, 2018.
[62] G. Te, W. Hu, A. Zheng, and Z. Guo, "Rgcnn: Regularized graph cnn for point cloud segmentation," in *ACM MM*, 2018.
[63] J. Yang, W.-S. Zheng, Q. Yang, Y.-C. Chen, and Q. Tian, "Spatial-temporal graph convolutional network for video-based person re-identification," in *CVPR*, 2020.
[64] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE TPAMI*, 2021.
[65] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.
[66] M. Zhai, L. Chen, F. Tung, J. He, M. Nawhal, and G. Mori, "Lifelong gan: Continual learning for conditional image generation," in *ICCV*, 2019.
[67] Z. Yu, L. Chen, Z. Cheng, and J. Luo, "Transmatch: A transfer-learning scheme for semi-supervised few-shot learning," in *CVPR*, 2020.
[68] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *CVPR*, 2017.
[69] Y. Chebotar and A. Waters, "Distilling knowledge from ensembles of neural networks for speech recognition." in *Interspeech*, 2016.
[70] Z. Huang and N. Wang, "Like what you like: Knowledge distill via neuron selectivity transfer," *arXiv preprint arXiv:1707.01219*, 2017.
[71] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *CVPR*, 2019.
[72] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *ICCV*, 2019.
[73] W. Chen, Y. Liu, N. Pu, W. Wang, L. Liu, and M. S. Lew, "Feature estimations based correlation distillation for incremental image retrieval," *IEEE TMM*, 2021.
[74] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, 1999.
[75] Y. Li, C. Gu, T. Dullien, O. Vinyals, and P. Kohli, "Graph matching networks for learning the similarity of graph structured objects," in *ICML*, 2019.
[76] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.
[77] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," *arXiv preprint arXiv:1704.01212*, 2017.
[78] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.
[79] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, 2017.
[80] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *arXiv preprint arXiv:2004.05439*, 2020.
[81] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks*, 1999.
[82] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.
[83] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "End-to-end deep learning for person search," in *ICCV*, 2017.
[84] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *ECCV*, 2008.
[85] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof, "Person re-identification by descriptive and discriminative classification," in *Scandinavian conference on Image analysis*, 2011.
[86] C. C. Loy, T. Xiang, and S. Gong, "Time-delayed correlation analysis for multi-camera activity understanding," *IJCV*, 2010.
[87] Z. Wei-Shi, G. Shaogang, and X. Tao, "Associating groups of people," in *BMVC*, 2009.
[88] W. Li, R. Zhao, and X. Wang, "Human reidentification with transferred metric learning," in *ACCV*, 2012.
[89] W. Li and X. Wang, "Locally aligned feature transforms across views," in *CVPR*, 2013.
[90] H. Zhao, M. Tian, S. Sun, J. Shao, J. Yan, S. Yi, X. Wang, and X. Tang, "Spindle net: Person re-identification with human body region guided feature decomposition and fusion," in *CVPR*, 2017.
[91] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *ICCV*, 2019.
[92] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
[93] M. Blondel, O. Teboul, Q. Berthet, and J. Djolonga, "Fast differentiable sorting and ranking," in *ICML*, 2020.



Nan Pu received his Ph.D. degree at Leiden University in the Netherlands in 2022. He is currently working as a postdoctoral researcher at the University of Trento, Italy. His research interests focus on lifelong learning and multi-modality learning with deep learning methods. He has published papers in international conferences and journals, including CVPR, AAAI, ACM MM, ICASSP and, IEEE TMM etc.

Zhun Zhong received his Ph.D. degree in the Department of Artificial Intelligence of Xiamen University, China, in 2019. He was also a joint Ph.D. student at University of Technology Sydney, Australia. He was a postdoc and is an Assistant Professor at University of Trento, Italy. His research interests include person re-identification, novel class discovery, data augmentation and domain adaptation.

Nicu Sebe is Professor in the University of Trento, Italy, where he is leading the research in the areas of multimedia analysis and human behavior understanding. He was the General Co-Chair of the IEEE FG 2008 and ACM Multimedia 2013. He was a program chair of ACM Multimedia 2011 and 2007, ECCV 2016, ICCV 2017 and ICPR 2020. He is a general chair of ACM Multimedia 2022 and a program chair of ECCV 2024. He is a fellow of IAPR.



Michael S. Lew is the head of the Deep Learning research group and a full Professor at Leiden University. He has published over a dozen books and 190 peer-reviewed scientific articles in the areas of deep learning, lifelong learning, multimedia retrieval and computer vision. He is also the founder and editor-in-chief of the International Journal of Multimedia Information Retrieval. Notably, he had the most cited paper in the ACM Transactions on Multimedia and one of the top 10 most cited articles in the history (out of more than 19,000 articles) of the ACM SIGMM. He was also a founding member of the advisory committee for the TRECVID video retrieval evaluation project, chair of the steering committee for the ACM International Conference on Multimedia Retrieval and a member of the ACM SIGMM Executive Committee.