Information Fusion 115 (2025) 102730

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Enhancing few-shot lifelong learning through fusion of cross-domain knowledge

Yaoyue Zheng [a], Xuetao Zhang [a], Zhiqiang Tian [c,*], Shaoyi Du [b,a,**]

[a] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China
[b] Department of Ultrasound, the Second Affiliated Hospital of Xi'an Jiaotong University, Xi'an 710049, China
[c] School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China

ARTICLE INFO

Keywords:
Lifelong learning
Continual learning
Few-shot learning
Cross-domain knowledge

ABSTRACT

Humans can continually solve new problems with a few examples and enhance their learned knowledge by incorporating new ones. Few-shot lifelong learning (FSLL) has been presented to mimic human learning ability. However, they overlook the significance of cross-domain knowledge and little effort has been made to investigate it. In this paper, we explore the effects of cross-domain knowledge in FSLL and propose a new framework to enhance the model's ability by fusing cross-domain knowledge into the learning process. Moreover, we investigate the impact of both debiased and non-debiased models in the FSLL context for the first time. Compared with previous works, our setting presents a unique challenge: the model should continually learn new knowledge from cross-domain few-shot data and update its existing knowledge by fusing new knowledge throughout its lifelong learning process. To address this challenge, the proposed framework focuses on learning and updating while migrating the well-known issues of forgetting and overfitting. The framework comprises three key components designed for learning cross-domain knowledge: the Debiased Base Learning strategy, Knowledge Acquisition, and Knowledge Update. The superiority of the framework is validated on mini-ImageNet, CIFAR-100, OfficeHome, and Meta-Dataset. Experiments show that the proposed framework exhibits the capability to perform in cross-domain situations and also achieves state-of-the-art performance in the non-cross-domain situation.

1. Introduction

Human can continually learn new knowledge from a few examples without compromising existing knowledge. This ability is known as few-shot lifelong learning (FSLL) [1,2]. Further, a notable characteristic of human learning is its dynamic nature, allowing learned knowledge to be updated throughout the lifespan, as it is encountered in diverse situations. For example, a child can learn the concept of a "*dog*" from both cartoons and real-life scenarios at different times, and the knowledge from both situations enriches their understanding of what a dog is. Traditional machine learning methods, on the other hand, excel primarily in pre-defined or similar distributions, proving inadequate for learning in dynamic environments [3].

Many efforts have been made to mimic the lifelong learning ability of humans, such as task-incremental lifelong learning (TILL) [4,5] and class-incremental lifelong learning (CILL) [6,7]. TILL focuses on learning multiple tasks by developing separate classifiers for each task. CILL aims to learn new classes using a single model while avoiding forgetting old ones. For practical reasons, CILL has gained more research attention. However, it still relies on ample labeled samples, which may not always be available in realistic applications. To achieve the well-known few-shot learning ability [8,9] of humans, few-shot lifelong learning [2], also known as few-shot class-incremental learning (FSCIL) [1], is present to learn new classes with a few samples, representing a more realistic scenario to CILL. Despite their efforts, FSLL still lacks human-level performance as it overlooks a crucial underlying challenge in lifelong learning. That is, humans learn not only from unseen classes but also from unseen domains. *This observation assumes even greater significance when deploying a model in realistic scenarios*. For example, in clinical applications, a disease diagnosis model should not only identify new diseases but, more importantly, learn from ever-growing data on known diseases with diverse imaging conditions. This is crucial because prioritizing accuracy is more important than the sheer number of recognized disease classes. Medical data often show a

* Corresponding author.
** Corresponding author at: Department of Ultrasound, the Second Affiliated Hospital of Xi'an Jiaotong University, Xi'an 710049, China.
E-mail addresses: yaoyuezheng@stu.xjtu.edu.cn (Y. Zheng), xuetaozh@xjtu.edu.cn (X. Zhang), zhiqiangtian@xjtu.edu.cn (Z. Tian), dushaoyi@xjtu.edu.cn (S. Du).

https://doi.org/10.1016/j.inffus.2024.102730
Received 15 May 2024; Received in revised form 28 August 2024; Accepted 7 October 2024
Available online 11 October 2024

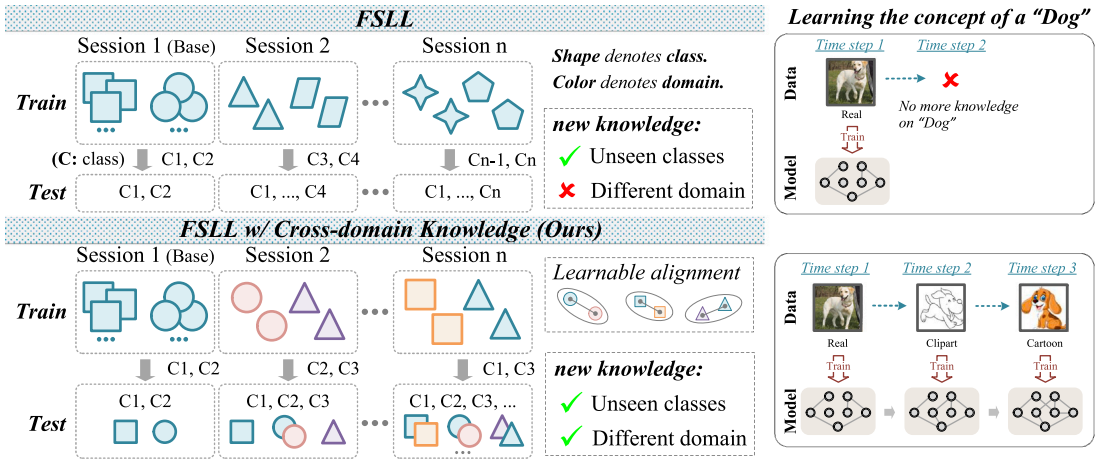

Fig. 1. Few-shot Lifelong Learning (FSLL) overlooks humans learn not only from unseen classes but also from unseen domains. In this paper, we conceptualize FSLL as an extension of the cross-domain problem and fuse cross-domain knowledge into the learning process to facilitate its application in realistic scenarios.

domain gap due to variations in devices or imaging conditions. Therefore, it is necessary to learn from cross-domain data to enhance the diagnostic accuracy. Many works have underscored the effectiveness of leveraging knowledge from multiple domains to enhance model performance [10–12]. Nonetheless, within the field of FSLL, a gap persists. Currently, works on FSLL mainly revolve around mitigating the catastrophic forgetting [13] of old classes. The significance of cross-domain knowledge tends to be underestimated, and little effort has been made to investigate it. A most related study is CDFSCIL [14], which aims to continually learn from different domains. However, they treat each domain as an independent task and do not consider the updating of previous knowledge to enhance overall performance. A more recent work by Yang [15] explores cross-domain TILL, focusing on task-specific predictions. Nevertheless, they demand a large number of samples to train a Siamese network for each task, presenting challenges in realistic applications.

In this paper, we conceptualize FSLL as an extension of the cross-domain problem, aiming to enhance its applicability in realistic scenarios by fusing cross-domain knowledge into the learning process. Fig. 1 illustrates the advantages of this perspective and its adaptability to a broader context. Compared with previous works, our setting presents a unique challenge: the model should continually learn new knowledge from cross-domain few-shot data and update its existing knowledge by fusing new knowledge throughout its lifelong learning process. To this end, we propose a new learning framework, that comprises three key components to facilitate the learning process: (1) Debiased Base Learning (DBL): We investigate the impact of both debiased and non-debiased models in cross-domain FSLL, and propose a debiased base learning strategy to benefit the learning on new sessions. (2) Knowledge Acquisition: We propose cross-domain alignment (CDA) and prototype alignment (PA) to align the potential domain shifts and guarantee minimal interference between classes as the number of classes increases. (3) Knowledge Update (KU): We propose a knowledge update method that enables continual knowledge enhancement without rehearsal.

Notably, the proposed method is also well-suited to non-cross-domain scenarios. To compare with previous works, we conduct experiments on two widely used benchmarks: mini-ImageNet [8] and CIFAR-100 [16]. For the cross-domain scenarios, we evaluate the method using the cross-domain dataset OfficeHome [17] with the newly introduced data setting. Additionally, to assess the model under more complex and diverse conditions, we use the Meta-Dataset [18] to simulate a larger-scale, real-world scenario. The main contributions are as follows:

- We investigate the significance of cross-domain knowledge in few-shot lifelong learning (FSLL) and enhance the model's ability by fusing cross-domain knowledge into the learning process.
- We propose a new learning framework that focuses on the unique challenge inherent in cross-domain FSLL. Besides, this paper is the first to investigate the impact of both debiased and non-debiased models in FSLL. Correspondingly, we propose an effective debiased base learning strategy to enhance the model's ability to learn and adapt in new sessions.
- The proposed learning framework has demonstrated effectiveness in cross-domain context, achieving an average prediction accuracy of 55.87%. Significantly, its applicability extends to non-cross-domain scenarios as well, where it achieves state-of-the-art results. This underscores the framework's adaptability across a wider range of applications.

2. Related work

2.1. Class-incremental lifelong learning

Class-incremental lifelong learning (CILL) [6] aims to learn new classes continually without forgetting old classes. Studies on CILL can be mainly divided into two categories: regularization-based methods [19,20] and replay-based methods [21,22]. Regularization-based methods aim to prevent forgetting by regularizing the weights [23] or predictions of a neural network [24]. Replay-based methods aim to prevent forgetting by storing and retrieving a small number of samples from previous classes for rehearsal [25,26]. Nevertheless, CILL usually requires a large number of samples for training, which limits its applicability in various real-world scenarios and practical use cases.

2.2. Few-shot lifelong learning

Few-shot lifelong learning (FSLL) [2] introduces a more realistic scenario compared to CILL, which requires learning new classes from a few labeled samples. Tao et al. [1] first defines the setting of FSLL and mitigates the forgetting issue by a neural gas network. Subsequent works have focused on adapting existing methods for CILL to address the challenges posed by FSLL [2,27,28]. For example, utilizing regularization techniques [29,30] or replaying past data [31,32]. However, within the context of FSLL, the data imbalance problem led to a severe decrease in performance. Moreover, CILL and FSLL both concentrate on continually learning new classes. They are manually constrained learning in each stage with mutually exclusive classes. Thus, the significance of cross-domain knowledge in lifelong learning has been overlooked.

2.3. Cross-domain lifelong learning

Currently, the cross-domain learning ability in lifelong learning has received limited attention. One related research focuses on incremental

domain adaptation [33,34]. Nevertheless, they fail to consider the learning of new classes [34], or they involve a single incremental step from the source domain to the target domain [33]. Another related research focuses on domain generalization [35], where the model is expected to learn from multiple domains simultaneously and generalize to an unseen domain.

A more related study to ours is CDFSCIL [14]. They propose to learn from cross-domain medical datasets. However, they impose a constraint where each domain is an independent task, overlooking the potential benefits that different domains can provide. In contrast, our work delves into the cross-domain knowledge for each class and we propose the knowledge acquisition and update. Another recent work by Yang [15] explores cross-domain task incremental learning, focusing on predicting within specific tasks. However, it struggles to quickly learn new tasks, as it requires a larger number of samples to train a new Siamese network for each new task. In contrast, we present a more general setting and the proposed method applies to few-shot learning scenarios.

2.4. Cross-domain few-shot learning

Cross-domain few-shot learning (CD-FSL) [36,37] is a closely related research topic to ours. CD-FSL focuses on learning a model from one or more domains and fast adaptation across different domains. Many efforts have been made in this area. For example, FWT [36] proposes a feature-wise transformation layer to simulate various distributions and thus improve the generalization ability to unseen domains. CHEF [38] builds an ensemble of learners using representation fusion to address the domain shifts in CD-FSL. Zhao et al. [39] suggest a prototypical feature alignment to tackle the shift between support and query samples. Meta-FDMixup [37] proposes to disentangle domain-irrelevant and domain-specific features to improve the model's ability to learn on few-shot samples.

Different from CD-FSL, cross-domain FSLL involves continual learning across domains and enhances the model's performance on all domains. Different from Zhao et al. [39], we propose a prototype alignment to minimize interference between classes as their number grows. Compared with Meta-FDMixup [37], we explore debiasing in the context of lifelong learning and propose a debiased base learning strategy to facilitate debiasing with only a base session available. The proposed strategy is also adaptable to other lifelong learning studies.

3. Method

3.1. Notation and preliminaries

We aim to learn new knowledge from a sequentially provided data stream with limited data while preserving old knowledge. New knowledge involves unseen classes and seen classes with a different domain. We have a sequence of S sessions $\{D^{(1)}, D^{(2)}, \dots, D^{(s)}, \dots, D^{(S)}\}$, where $D^{(s)} := \{(x_i^{(s)}, y_i^{(s)})\}_{i=1}^{|D^{(s)}|}$ represents the sth learning session with $|D^{(s)}|$ training samples. Each $x_i^{(s)}$ is an input sample belonging to class $y_i^{(s)} \in C^{(s)}$, where $C^{(s)}$ denotes the label space of session s. Specifically, the label spaces $C^{(s)}$ may contain known classes with different domain samples. The first session, $D^{(1)}$, serves as the base session, containing a relatively large number of training classes and samples. We follow the setting of previous works [40] to organize the data of base session into many episodes, where each episode is a c-way k-shot task. The goal of the base session is to learn from $D^{(1)}$ and distinguish between the classes in $C^{(1)}$. Subsequent new sessions, $D^{(s)}$ $(s > 1)$, arrive in a stream without rehearsal. These sessions with a size of $|D^{(s)}| = c \cdot k$, where $c = |C^{(s)}|$ represents the number of classes, and k represents the number of training samples per class (*i.e.* support data). In a given session s, we can only access $D^{(s)}$, and the data of previous sessions (from 1 to $s-1$) are not available. After learning on each session, the learned model is evaluated on an updated evaluation set $\tilde{C}^{(s)} := \cup_{i=1}^{s} C^{(i)}$ that covers all seen classes.

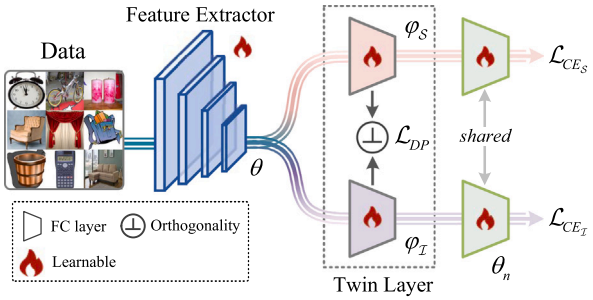

Fig. 2. The debiased base learning strategy is aiming to obtain a debiased base model, parameterized as $\theta_b = \{\theta, \varphi_I\}$. We opt for a twin layer over a twin model in practical implementation. φ_S is only used in the base training stage.

3.2. Debiased base learning

In FSLL, the model is expected to learn new knowledge. However, due to the limited availability of training samples on the base session, the model inevitably becomes biased toward domain-specific features [37]. To address this issue, we propose a Debiased Base Learning (DBL) strategy aimed at eliminating these biased features and improving the model's performance in acquiring new knowledge. Initially, the training objective of the base session is to minimize the cross-entropy loss $\mathcal{L}_{CE}$ on all episodes:

$$\mathbb{E}(f_\theta) = \frac{1}{N_e} \sum_{i=1..N_e} \mathcal{L}_{CE}^i(x, y; \theta), \tag{1}$$

where f_θ represents the trained model parameterized by θ and N_e is the number of episodes. Eq. (1) ensures an overall good performance of f_θ on the base session, which learns both domain-invariant and domain-specific features of input sample x, denote as $\mathcal{I}(x)$ and $\mathcal{S}(x)$, respectively. However, measuring $\mathcal{I}(x)$ and $\mathcal{S}(x)$ is non-trivial when only base session is available [41].

As the model inevitably learns from $\mathcal{S}(x)$, it inherently serves as an indicator of $\mathcal{S}(x)$. This insight leads us to propose the DBL strategy to develop a debiased base model. Fig. 2 illustrates the learning framework of the DBL strategy. Initially, we use a twin model $(f_{\theta_S}, f_{\theta_I})$ to realize the learning process, where f_{θ_S} serves as an indicator of $\mathcal{S}(x)$, and f_{θ_I} is responsible for learning $\mathcal{I}(x)$. In practice, instead of utilizing the entire model, we find that a fully connected layer can map localist features to a distributed representation. As a result, we exclusively consider a twin layer $(l_{\varphi_S}, l_{\varphi_I})$ and shared one f_θ, which proves to be adequate. The twin layer $(l_{\varphi_S}, l_{\varphi_I})$ is parameterized by φ_S and φ_I, respectively. We adopt a two-stage learning for DBL. In the first stage, we train the model on the base session and obtain layer l_{φ_S}. In the second stage, we encourage l_{φ_I} to capture $\mathcal{I}(x)$ with the guidance provided by l_{φ_S}. We achieve this purpose by minimizing the loss function $\mathcal{L}_{DP}$ to ensure orthogonality between l_{φ_S} and l_{φ_I}:

$$\mathcal{L}_{DP} = \frac{1}{d} \sum_{i=1}^{d} \| \varphi_{S_i} \, \varphi_{I_i}^T \|_F^2, \tag{2}$$

where d denotes the dimension of the output feature. $\| \cdot \|_F^2$ is the squared Frobenius norm. The overall optimization objective in base session can be denoted as:

$$\begin{aligned} \tilde{\mathbb{E}}(f_\theta, l_{\varphi_S}, l_{\varphi_I}, f_{\theta_n}) = \frac{1}{N_e} \sum_{i=1..N_e} \big(& \mathcal{L}_{CE_S}^i(x, y; \theta, \varphi_S, \theta_n) \\ & + \mathbb{I}_{[epoch \geqslant t]} \mathcal{L}_{CE_I}^i(x, y; \theta, \varphi_I, \theta_n) \\ & + \mathcal{L}_{DP}^i(\varphi_S, \varphi_I) \big), \end{aligned} \tag{3}$$

where θ_n is pre-trained in the base session and as the initialization for the subsequent sessions, we will introduce it in Section 3.3. t is the start epoch of the second stage. The $\mathcal{L}_{CE_S}$ and $\mathcal{L}_{CE_I}$ are cross-entropy losses. After completing the base learning, we obtain a feature encoder

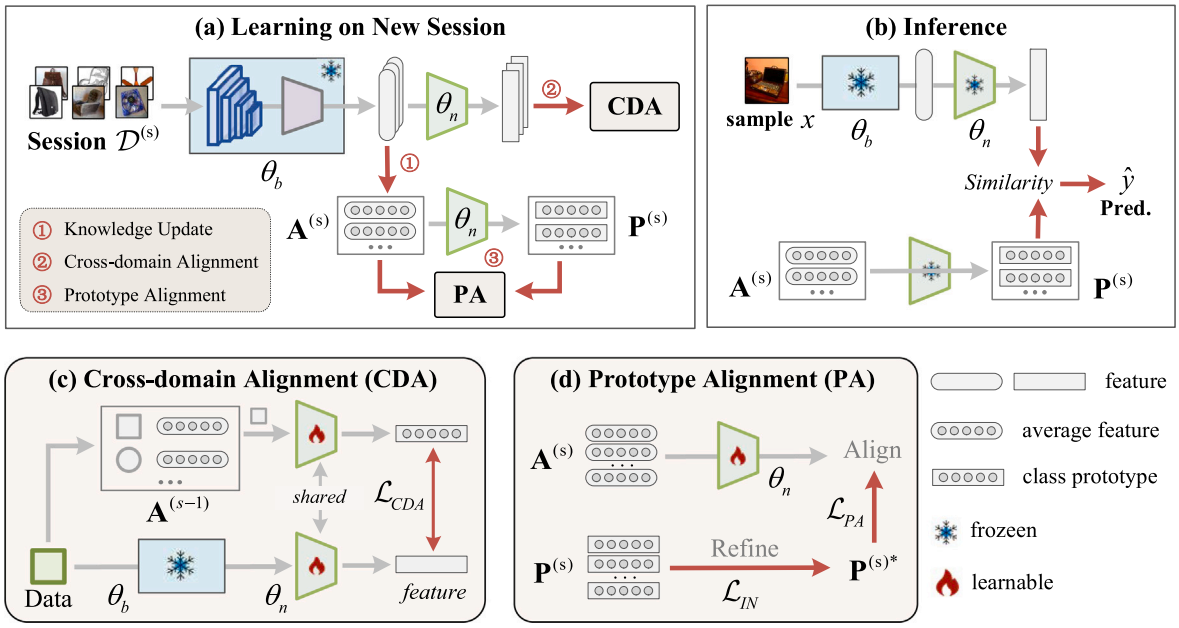

Fig. 3. Overview of the proposed framework. (a) The learning pipeline on the new session. θ_b is trained during the debiased base learning stage and stays frozen during new sessions. First, we update $\mathbf{A}^{(s)}$ through the knowledge update method to incorporate the average feature embedding from the current session. Second, we fine-tune θ_n through CDA to align the potential domain shifts. Third, we use PA to minimize interference between classes. (b) The inference pipeline. First, we extract the feature embedding of the input image through (θ_b, θ_n). Then, we calculate the cosine similarity between the feature embedding and the prototype of each class to obtain the final prediction result $\hat{y}$. (c) The proposed cross-domain alignment (CDA). For a new sample, we first obtain the average class embedding from memory $\mathbf{A}^{(s-1)}$. We then finetune f_{θ_n} to align domain shifts by optimizing the loss function $\mathcal{L}_{CDA}$. (d) The proposed prototype alignment (PA). We first refine the $\mathbf{P}^{(s)}$ to ensure a set of orthogonal prototypes $\mathbf{P}^{(s)*}$. We then align the prototypes derived from $f_{\theta_n}(\mathbf{A}^{(s)})$ with all refined prototypes using the loss function $\mathcal{L}_{PA}$.

$f_{\theta_b} = (f_\theta, l_{\varphi_I})$ endowed with domain-invariant knowledge. Note that l_{φ_S} is only used in the base training process. The necessity of the two-stage optimization scheme is analyzed in Section 4.4.

3.3. Knowledge acquisition

In cross-domain FSLL, the model is expected to continually learn new knowledge from cross-domain few-shot data while ensuring minimal interference between classes. To this end, we propose aligning potential domain shifts by cross-domain alignment and guaranteeing minimal interference between classes by prototype alignment. To achieve an efficient learning process, we utilize a feature memory: $\mathbf{A}^{(s)}$ to store the average feature embeddings of each class. Each element $\mathbf{a}^{(s)} \in \mathbf{A}^{(s)}$ represents the average class embedding in session s. $\mathbf{A}^{(s)}$ is updated through the proposed KU. Class prototype $\mathbf{P}^{(s)}$ is obtained from $f_{\theta_n}(\mathbf{A}^{(s)})$, and used to calculate the cosine similarity between the feature embedding and the prototype of each class to obtain the final prediction. An overview of the proposed methods are shown in Fig. 3, and the learning details are shown in Algorithm 1.

Cross-domain Alignment. Previous works fail to deal with samples from diverse domains, as they make the representation vulnerable to distribution shifts [42,43]. We introduce two key aspects to facilitate the learning from cross-domain knowledge and prevent catastrophic forgetting. First, we freeze the $f_{\theta_b} = (f_\theta, l_{\varphi_I})$ for retention of domain-invariant knowledge and to prevent overfitting to new sessions. Second, we integrate a learnable mapping with a bias vector to enable the model to learn new knowledge while dealing with the domain shifts. We propose the learnable mapping $\mathbf{M} : \mathbb{R}^d \to \mathbb{R}^n$, where $n > d$, to enhance the model's capability to learn new knowledge. Considering domain shifts, we incorporate a bias vector $\mathbf{b} \in \mathbb{R}^n$ into the learnable mapping, enabling the alignment of different shifts [44]. In practice, it enables us to conveniently implement the learnable mapping with a bias vector using a fully connected layer, denoted as f_{θ_n}, where $\theta_n = \{\mathbf{M}, \mathbf{b}\}$. In each session, if a new sample belongs to a seen class, we first obtain the average class embedding from memory $\mathbf{A}^{(s-1)}$. Then, we finetune f_{θ_n} to align potential domain shifts by optimizing the loss function $\mathcal{L}_{CDA}$. Fig. 3(c) illustrates the finetuning process of CDA.

$$\mathcal{L}_{CDA} = -\sum_{i=1}^{|\mathcal{D}^{(s)}|} \mathbb{I}_{[y_i \in \tilde{C}^{(s-1)}]} \mathrm{sim}(f_{\theta_n^{(s)}}(f_{\theta_b}(x_i^{(s)})), f_{\theta_n^{(s)}}(\mathbf{a}_{y_i}^{(s-1)})), \tag{4}$$

where $f_{\theta_n^{(s)}}$ is learnable mapping in session s. $\mathbf{a}_{y_i}^{(s-1)} \in \mathbf{A}^{(s-1)}$ is average feature embedding of class y_i in last session $s-1$. $\mathrm{sim}(\cdot,\cdot)$ is cosine similarity.

Prototype Alignment. To minimize interference between classes as the number of classes increases, we optimize f_{θ_n} to establish a suitable decision boundary for all seen classes within the current session. An intuitive solution is to increase the distance between all observed prototypes:

$$\mathcal{L}(\theta_n^{(s)}) = \sum_{\substack{i,j=1 \\ i \neq j}}^{|\tilde{C}^{(s)}|} \mathrm{sim}(f_{\theta_n^{(s)}}(\mathbf{a}_i^{(s)}), f_{\theta_n^{(s)}}(\mathbf{a}_j^{(s)})). \tag{5}$$

However, without the past samples available, Eq. (5) is difficult to optimize. Therefore, we first refine the $\mathbf{P}^{(s)} = f_{\theta_n^{(s)}}(\mathbf{A}^{(s)})$ to ensure a collection of orthogonal prototypes $\mathbf{P}^{(s)*}$, thereby reducing interference between classes:

$$\mathcal{L}_{IN}(\mathbf{P}^{(s)}) = \sum_{i=1}^{|\tilde{C}^{(s)}|} \mathrm{sim}(\mathbf{p}_i^{(s)}, \mathbf{P}^{(s)} \backslash \mathbf{p}_i^{(s)}) = \|\mathbf{P}^{(s)T}\mathbf{P}^{(s)} - \mathbf{I}\|_F^2, \tag{6}$$

where $\mathbf{P}^{(s)}$ is initial prototype set of session s. $\mathbf{P}^{(s)} \backslash \mathbf{p}_i^{(s)}$ denotes the set of all prototypes except ith prototype. $\mathbf{I}$ is a identity matrix. Then, we align the prototypes derived from $f_{\theta_n^{(s)}}$ with all refined prototypes $\mathbf{P}^{(s)*}$. Instead of forwarding samples to retrain $f_{\theta_n^{(s)}}$, it is effective by using $\mathbf{A}^{(s)}$, which eliminates the need for sample rehearsal and significantly reduces the computational cost. $\theta_n^{(s)}$ can be optimized via (with update rate β):

$$\mathcal{L}_{PA}(\mathbf{P}^{(s)*}, \theta_n^{(s)}) = -\sum_{i=1}^{|\tilde{C}^{(s)}|} \mathrm{sim}(\mathbf{p}_i^{(s)*}, f_{\theta_n^{(s)}}(\mathbf{a}_i^{(s)})), \tag{7}$$

$$\theta_n^{(s)*} = \theta_n^{(s)} - \beta \nabla_{\theta_n^{(s)}} \mathcal{L}_{PA}(\mathbf{P}^{(s)*}, \theta_n^{(s)}). \tag{8}$$

Algorithm 1 Learning algorithm for sessions s ($s > 1$). $\tilde{C}^{(s-1)}$ represents all classes from session 1 to $s-1$. Superscript (s) denotes the session index. r, k denote the number of iterations for CDA and PA, respectively.

Input: session $D^{(s)} = \{(x_i^{(s)}, y_i^{(s)})\}_{i=1}^{|D^{(s)}|}$ with label space $C^{(s)}$, model $(f_{\theta_b}, f_{\theta_n^{(s-1)}})$, feature memory $\mathbf{A}^{(s-1)}$, count vector $C^{(s-1)}$, hyper-parameter r, k.
Output: model $f_{\theta_n^{(s)}}$, feature memory $\mathbf{A}^{(s)}$, prototype set $\mathbf{P}^{(s)}$, count vector $C^{(s)}$.

for sample x in $D^{(s)}$ **do**
  Feature extraction: $\mathbf{x} = f_{\theta_b}(x)$
  Update number of samples: update $c^{(s)}$ ▷ Eq. (11)
end for
Calculate average feature: $\mathbf{a}^{(s)} = \{\mathbf{a}_1^{(s)}, \dots, \mathbf{a}_{|C^{(s)}|}^{(s)}\}$
Knowledge Update: update $\mathbf{A}^{(s)}$ ▷ Eq. (9)
if $C^{(s)} \cap \tilde{C}^{(s-1)} \neq \emptyset$ **then**
  for $i = 1, 2, \dots, r$ **do**
    Cross-domain Alignment: update f_{θ_n} ▷ Eq. (4)
  end for
end if
Refine prototype: update $\mathbf{P}^{(s)}$ ▷ Eq. (6)
for $i = 1, 2, \dots, k$ **do**
  Prototype Alignment: update f_{θ_n} ▷ Eq. (7)
end for
return updated $f_{\theta_n^{(s)}}$, $\mathbf{A}^{(s)}$, $\mathbf{P}^{(s)}$ (for inference), $C^{(s)}$

Fig. 3(d) illustrates the align process of PA. The final prototype set is determined by passing $\mathbf{A}^{(s)}$ through the updated f_{θ_n}, ensuring minimal interference in the current session.

3.4. Knowledge update

Previous works [45] calculated the class embedding using all samples within a single session. In realistic scenarios, however, it is not feasible to obtain all samples of a class simultaneously, and cross-domain knowledge unfolds continually. Naively averaging features across all sessions can result in shifts in the true embedding, thereby compromising the acquired knowledge. To this end, we propose a new method that enables continual knowledge updating without the need for having all data available or data rehearsal. Specifically, embeddings can be updated as follows:

$$\mathbf{A}^{(s)} = \frac{C^{(s-1)}}{C^{(s)}}\mathbf{A}^{(s-1)} + \frac{c^{(s)}}{C^{(s)}}\mathbf{a}^{(s)}, \tag{9}$$

where $C^{(s)} \in \mathbb{R}^{|\tilde{C}^{(s)}|}$ is a count vector that counts the number of samples belonging to each class from session 1 to s, $c^{(s)} \in \mathbb{R}^{|C^{(s)}|}$ count the number of samples of each class in current session s. $C^{(s)}$ and $c^{(s)}$ for class k are defined as:

$$C_k^{(s)} = \sum_{i=1}^{s}\sum_{j=1}^{|D^{(i)}|}\mathbb{I}[y_j^{(i)} = k], \tag{10}$$

$$c_k^{(s)} = \sum_{i=1}^{|D^{(s)}|}\mathbb{I}[y_i^{(s)} = k]. \tag{11}$$

Note that $C^{(s)}$ is a vector of real numbers and can be updated based on $C^{(s-1)}$ without accessing any information from past samples: $C^{(s)} = C^{(s-1)} + c^{(s)}$.

4. Experiments

To demonstrate the superiority of the proposed method, we conducted experiments in both cross-domain and non-cross-domain scenarios. For non-cross-domain scenario, we adhered to the widely-used benchmarks and conducted comparisons with state-of-the-art methods [1,6,25,26,30,40,46–49]. The benchmarks include CIFAR-100 [16] and mini-ImageNet [8]. For the cross-domain scenario, we evaluated the proposed method on OfficeHome [17] using a newly introduced split setting, which is more applicable in realistic applications. Moreover, we evaluated the proposed method using Meta-Dataset [18] to simulate a larger-scale, real-world scenario.

Table 1
Statistical metrics for OfficeHome, with four data domains: Art, Clipart, Product, and Real World.

Domain	Mean	Standard deviation	Kurtosis	Skewness
Art	122.65	82.65	−1.27	0.10
Clipart	146.70	105.71	−1.59	−0.31
Production	187.67	79.62	0.91	−1.16
Real world	146.56	82.66	−1.22	−0.21

4.1. Benchmarks

mini-ImageNet [8] is a subset of ImageNet [50], consisting of 100 classes, with each class comprising 500 training images and 100 testing images. All the images have a resolution of 84 × 84. **CIFAR-100** [16], on the other hand, comprised of 60,000 images with a resolution of 32 × 32, categorized into 100 distinct classes. To ensure comparability with FSLL works, we follow the same setting as described in [1]. Both datasets are divided into a base session and a series of novel sessions. For mini-ImageNet and CIFAR-100, 60 classes are selected as base classes. The remaining 40 classes are split into 8 incremental sessions, with each session containing 5 classes. Within each class, there are 5 training examples, i.e. 5-way 5-shot task.

OfficeHome [17] is a cross-domain object recognition dataset comprising 15,500 everyday objects distributed across 65 classes within 4 distinct domains: *Art*, *Clipart*, *Product*, and *Real World*. Art includes paintings, sketches, and artistic depictions. Clipart consists of clipart images, while Product includes images without a background. Real World encompasses regular images captured with a camera. We chose the Art, Clipart, and Product domains as the base session, each domain containing 40 classes. The remaining 25 classes within the three domains, along with the complete Real World domain (65 classes) are served as novel knowledge. We set 8 new sessions with 5-way 5-shot task. Within each session, 2 classes are chosen from the base classes with unseen domain (Real), while 3 classes are selected from the novel classes (comprising 4 domains, each with 25 classes). Fig. 4 shows the data setting and learning scenario. We divided the OfficeHome into training and testing sets with a ratio of 0.7/0.3. All images have a resolution of 224 × 224. We present the statistical metrics for 4 data domains in the OfficeHome dataset, as outlined in Table 1. Mean values offer insight into the central tendency of the data, representing the average value across the dataset and indicating the typical magnitude of the data points. Standard deviation measures the extent of variability or dispersion of data points from the mean, with a higher standard deviation suggesting greater variability in the dataset. Kurtosis is a measure of the shape of the probability distribution. Positive kurtosis indicates a relatively peaked distribution, while negative kurtosis suggests a flatter distribution. Skewness measures the asymmetry of the probability distribution. Positive skewness indicates a longer or fatter tail on the right, while negative skewness suggests a longer or fatter tail on the left.

Meta-Dataset [18] is a few-shot classification benchmark that consisting of 10 datasets: ILSVRC-2012 [51], Omniglot [52], FGVC-Aircraft (Aircraft) [53], CUB-200-2011 (Birds) [54], Describable Textures (DTD) [55], QuickDraw [56], FGVCx Fungi [57], VGG Flower [58], German Traffic Sign Recognition Benchmark (Traffic Signs) [59], and MSCOCO [60]. We evaluate FSLL performance in two settings: (1)

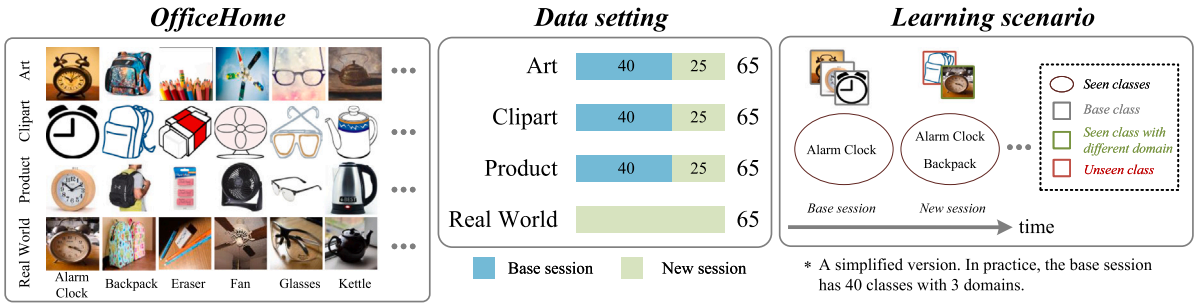

Fig. 4. Data setting and learning scenario: the base session is created with the Art, Clipart, and Product domains, each consisting of 40 selected classes. The remaining 25 classes, along with the entire Real World domain, represent novel knowledge.

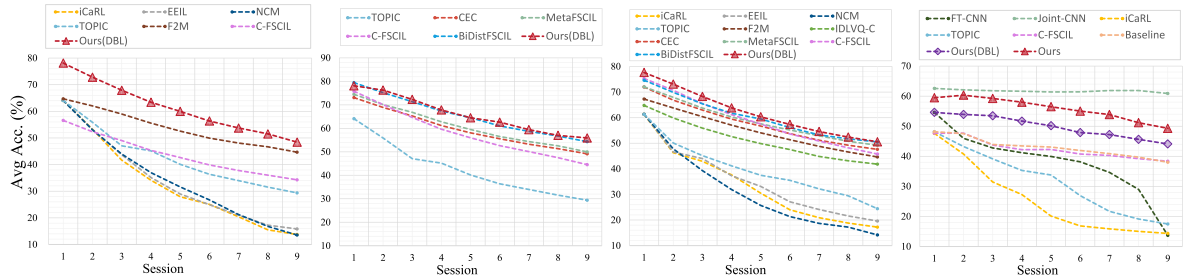

Fig. 5. Comparison results on CIFAR-100, mini-ImageNet, and OfficeHome. In each session, the models are evaluated based on the average accuracy (%) across all the seen classes $\tilde{C}^{(s)}$. "FT-CNN" represents fine-tuning with new sessions simply, while "Joint-CNN" indicates joint training involving all observed classes.

We use the first 8 datasets with their official training splits as the base session. Then, we randomly select 9 tasks from the test split of the Meta-Dataset to evaluate lifelong learning performance. (2) We use only the ImageNet subset as the base session and other datasets as novel sessions to evaluate cross-domain lifelong learning performance. To yield diverse and realistic tasks, we follow Meta-Dataset [18] to allow variable task ways and shots across the dataset, with each task originating from a distinct dataset. Further details on training are provided in the Training Details.

4.2. Experimental details

Architectures. For the comparison on mini-ImageNet and CIFAR-100, we adopt two commonly used embedding architectures, ResNet-18 and ResNet-20, which align with the recent works [1,40]. Specifically, the ResNet-18 is used for mini-ImageNet, while both architectures are used for CIFAR-100. For OfficeHome, we adopt ResNet-12 for all experiments. ResNet-12 has four basic blocks, each having three 3×3 convolutional layers and one residual operation. Moreover, each block contains a batch normalization layer and a 0.1 leaky ReLU. The output dimension for the four blocks has two options: [64, 128, 256, 512] and [64, 160, 320, 640]. We adopt the latter in the experiment for a fair comparison with the comparison works [48]. For Meta-Dataset, we use ResNet-18 to consistent with previous works [61].

Training Details. All models were trained using the SGD optimizer with a momentum of 0.9. We follow previous works [47] to perform pre-training on base classes from scratch. The initial learning rate is 0.1 and the decay factor is 0.1. The max training epochs is 200 and the learning rate decays at [60, 120]. Subsequently, the model is meta-trained for 100 epochs with an initial learning rate of 0.01 and a decay factor of 0.1. The learning rate decay at [30, 60]. Each epoch contains 1000 randomly sampled episodes. In the meta-training stage, we perform 60-way 5-shot task for mini-ImageNet and CIFAR-100, and 40-way 5-shot task for OfficeHome. For Meta-Dataset, each task has variable ways and shots to simulate a more realistic scenario. In the first setting, each novel session is randomly selected from the test split of the Meta-Dataset, which consists of 10 datasets. In the second setting, 9 datasets are reserved as novel datasets to evaluate cross-domain lifelong learning performance. Specifically, in each novel session, we add a new dataset based on the previous session, and the model is assessed on both the newly added dataset and all prior datasets. This evaluation process is repeated 100 times, and we report the average performance for each session. In the training stage, standard data augmentation is applied, including random crop, horizontal flip, and color jittering. In the validation stage, the model is evaluated on 600 episodes with randomly sampled classes and we calculate the average accuracy across all episodes.

Evaluation Protocol. We report top-1 accuracy (%) for each session and computed the overall average across all sessions. The performance of the ith session is evaluated based on the average accuracy on all the seen classes $\tilde{C}^{(s)}$. Additionally, we include the average accuracy across all sessions and the performance improvement against each comparison method.

Experimental Environment. The experiment was performed on Ubuntu 16.04 with 2 NVIDIA RTX 3090 GPU. The implementation is based on PyTorch 2.0 and Python 3.10. The training process for OfficeHome necessitates a GPU memory capacity of 30 GB.

4.3. Main results

Fig. 5 illustrates the visualization comparison results of the methods across the three main benchmarks. To align with recent works [1,40], we employ ResNet-18 and ResNet-20 for the comparison on CIFAR-100 and mini-ImageNet. For OfficeHome, we adopt ResNet-12 and report the average performance of top-1 accuracy in each session. For the comparison on CIFAR-100 and mini-ImageNet, we exclusively integrate DBL and PA, as the two benchmarks omit the cross-domain scenario from consideration. For the comparison on OfficeHome, "Baseline" denotes the model without the proposed training architecture (*i.e.* twin layer) and other proposed components. "Ours" denotes the proposed method under the same architecture. It is evident that merely fine-tuning with few-shot data from new sessions leads to a notable decline

Table 2
Comparison with state-of-the-art methods on mini-ImageNet. The backbone is ResNet-18. The best results are highlighted in **bold black**. The second-best results are highlighted by underline. "Impro." indicates the accuracy enhancement of our results compared to the comparison method.

Method	Sessions (mini-ImageNet)									Average Acc.	Impro. ↑
	1	2	3	4	5	6	7	8	9		
iCaRL [6]	61.31	46.32	42.94	37.63	30.49	24.00	20.89	18.80	17.21	33.29	28.73
EEIL [25]	61.31	46.58	44.00	37.29	33.14	27.12	24.10	21.57	19.58	34.97	27.05
NCM [26]	61.31	47.80	39.31	31.91	25.68	21.35	18.67	17.24	14.17	30.83	31.19
TOPIC [1]	61.31	50.09	45.17	41.16	37.48	35.52	32.19	29.46	24.42	39.64	22.37
F2M [30]	67.28	63.80	60.38	57.06	54.08	51.39	48.82	46.58	44.65	54.89	7.12
IDLVQ-C [46]	64.77	59.87	55.93	52.62	49.88	47.55	44.83	43.14	41.84	51.16	10.85
CEC [47]	72.00	66.83	62.97	59.43	56.70	53.73	51.19	49.24	47.63	57.75	4.27
MetaFSCIL [40]	72.04	67.94	63.77	60.29	57.58	55.16	52.90	50.79	49.19	58.85	3.16
CLOM [62]	73.08	68.09	64.16	60.41	57.41	54.29	51.54	49.37	48.00	58.48	3.53
Replay [45]	71.84	67.12	63.21	59.77	57.01	53.95	51.55	49.52	48.21	58.02	3.99
C-FSCIL [48]	75.32	70.67	65.51	61.18	57.52	53.63	50.85	48.21	45.79	58.74	3.27
BiDistFSCIL [49]	74.65	69.89	65.44	61.76	59.49	56.11	53.28	51.74	50.49	60.32	1.70
Ours (DBL+PA)	**77.67**	**73.11**	**68.29**	**63.82**	**60.32**	**57.35**	**54.58**	**52.41**	**50.57**	**62.01**	–

Table 3
Comparison with state-of-the-art methods on CIFAR-100, which was performed on ResNet-18 and ResNet-20.

Method	Sessions (CIFAR-100 w/ ResNet-18)									Average Acc.	Impro. ↑
	1	2	3	4	5	6	7	8	9		
iCaRL [6]	64.10	53.28	41.69	34.13	27.93	25.06	20.41	15.48	13.73	32.87	28.50
EEIL [25]	64.10	53.11	43.71	35.15	28.96	24.98	21.01	17.26	15.85	33.79	27.58
NCM [26]	64.10	53.05	43.96	36.97	31.61	26.73	21.23	16.78	13.54	34.22	27.15
TOPIC [1]	64.10	55.88	47.07	45.16	40.11	36.38	33.96	31.55	29.37	42.62	18.75
F2M [30]	64.71	62.05	59.01	55.58	52.55	49.96	48.08	46.67	44.67	53.70	7.67
C-FSCIL [48]	56.58	52.32	49.04	45.33	42.64	39.87	37.77	36.02	34.27	43.76	17.61
Ours (DBL+PA)	**78.12**	**72.82**	**67.91**	**63.41**	**59.98**	**56.38**	**53.74**	**51.52**	**48.44**	**61.37**	–
Method	Sessions (CIFAR-100 w/ ResNet-20)									Average Acc.	Impro. ↑
	1	2	3	4	5	6	7	8	9		
TOPIC [1]	64.10	55.88	47.07	45.16	40.11	36.38	33.96	31.55	29.37	42.62	23.38
CEC [47]	73.07	68.88	65.26	61.19	58.09	55.57	55.57	51.34	49.14	59.53	6.47
MetaFSCIL [40]	74.50	70.10	66.84	62.77	59.48	56.52	54.36	52.56	49.97	60.79	5.21
Replay [45]	74.40	70.20	66.54	62.51	59.71	56.58	54.52	52.39	50.14	60.78	5.23
C-FSCIL [48]	75.80	70.11	64.61	59.61	55.96	52.65	50.10	47.49	44.54	57.87	8.13
BiDistFSCIL [49]	**79.45**	75.20	71.34	67.40	**64.50**	61.05	58.73	56.73	54.31	65.41	0.59
Ours (DBL+PA)	78.12	**76.28**	**72.32**	**67.83**	64.49	**62.63**	**59.44**	**57.00**	**55.91**	**66.00**	–

in accuracy due to catastrophic forgetting ("FT-CNN"). The upper bound "Joint-CNN" denotes the model's maximum performance, as it is trained involving all observed classes. Notably, the comparison methods both have a performance decrease from session 1 to 3, while the proposed method can maintain stable performance and even improve the accuracy on seen classes. This is attributed to learning new knowledge contributes to an improved performance of the model on existing knowledge. We conducted tests with different session orders and found that the session order has an impact on this phenomenon but will not be substantial.

4.3.1. Comparison on non-cross-domain scenario

In the context of non-cross-domain scenario, the proposed method exclusively integrates Debiased Base Learning (DBL) and Prototype Alignment (PA), as non-cross-domain scenario omits the cross-domain learning from consideration. As outlined in Tables 2 and 3, the proposed method with DBL and PA demonstrates superior performance over all other methods across the two benchmarks with ResNet-18 and ResNet-20. In particular, the proposed method on ResNet-18 achieves an average accuracy of 62.01% and 61.37% on mini-ImageNet and CIFAR-100, respectively. The proposed method outperforms the second-best method by 1.70% and 7.67% on mini-ImageNet and CIFAR-100, respectively. The results demonstrate that a debiased base model can improve the performance of learning new knowledge and the proposed method is capable of learning new knowledge while preserving old knowledge. Note that the proposed method does not involve rehearsing old data [45,46,49].

4.3.2. Comparison on cross-domain scenario

For OfficeHome, we first perform meta-learning on the base session (40 classes), and then learn new knowledge in 8 new sessions. Different from FSLL, the new knowledge comprises both unseen classes and seen classes with different domains. As shown in Table 4, we first evaluate the performance of the proposed method without DBL, which achieves an average accuracy of 49.58% and brings an improvement of 8.11% compared to the baseline. The "Baseline" is the model without the proposed training architecture and other proposed components. Then, we incorporate all the proposed components, which achieve an average accuracy of 56.41% and bring improvement of 14.94%. Notably, the comparison method both have a performance decrease from session 1 to 3, while the proposed method can maintain stable performance and even improves the accuracy on seen classes. This demonstrates that the proposed method can learn new knowledge from different domains to improve the learned knowledge. Moreover, C-FSCIL has a performance drop rate of 9.35% from the base session to last session. In contrast, the proposed method without and with DBL has a performance drop rate of 7.36% and 8.82%, respectively. This demonstrates that the proposed method can alleviate catastrophic forgetting and improve overall performance by learning new knowledge.

4.3.3. Performance on Meta-Dataset

For the Meta-Dataset, we start by pretraining ResNet-18 on the base session. Next, we apply meta-learning on the base session and learn new knowledge across 9 novel sessions. The learning performance is detailed in Table 5. "Ours" denotes the proposed framework that

Table 4
Comparison results on OfficeHome. We reported the results of with and without Debiased Base Learning (DBL), respectively. Ours represents the combination of all the proposed components. Session 1 denotes the base session with 40 classes. "FT-CNN" represents fine-tuning on novel tasks simply, which leads to severe catastrophic forgetting. "Joint-CNN" represents joint training involving all observed classes, which indicates the upper bound of performance.

Method	Sessions (OfficeHome)									Average Acc.
	1	2	3	4	5	6	7	8	9	
FT-CNN	54.60	45.96	42.79	41.20	39.96	38.16	34.67	28.97	13.72	37.78
Joint-CNN	62.59	62.11	61.80	61.64	61.41	61.42	61.87	61.87	60.90	61.73
iCaRL [6]	47.63	40.77	31.45	27.27	20.12	16.85	15.88	15.02	14.35	25.48
TOPIC [1]	48.16	43.26	39.15	35.30	33.79	26.91	21.70	19.12	17.48	31.65
C-FSCIL [48]	47.70	47.70	43.75	42.19	42.27	40.74	40.29	39.20	38.35	42.47
Baseline	47.15	45.83	43.27	42.39	41.37	39.62	39.10	38.10	36.38	41.47
Ours w/o DBL	51.79	53.12	51.43	51.19	50.59	49.11	48.25	46.34	44.43	49.58
Ours	**59.47**	**60.33**	**60.37**	**58.10**	**56.52**	**55.28**	**54.50**	**52.45**	**50.65**	**56.41**

Table 5
Few-shot lifelong learning on Meta-Dataset tasks. In the first group, the base session is trained on the first 8 datasets of the Meta-Dataset (i.e., excluding Traffic Signs and MSCOCO), with each novel session selected from the test split of the Meta-Dataset. In the second group, the base session is trained on the ImageNet subset, and each novel session introduces a new dataset, building on the previous session.

Method	Sessions (Meta-Dataset)									Average Acc.
	1	2	3	4	5	6	7	8	9	
FSMB[a] [61]	54.87	52.81	52.31	51.39	51.24	50.39	50.37	49.01	48.98	51.26
C-FSCIL [48]	57.63	55.78	54.17	53.88	52.28	52.12	51.68	50.98	48.37	52.99
Ours w/o DBL	57.82	55.74	56.04	56.08	55.85	**55.76**	**54.57**	52.76	**51.29**	55.10
Ours	**58.74**	**57.69**	**57.31**	**57.30**	**56.54**	55.68	54.04	**53.49**	50.75	**55.73**
FSMB[a] [61]	44.59	44.27	42.95	42.81	42.58	42.30	41.16	40.44	37.90	42.11
C-FSCIL [48]	**51.94**	49.50	47.03	46.23	45.85	45.75	43.48	41.38	40.12	45.70
Ours w/o DBL	49.58	49.00	47.71	47.39	46.83	46.55	**46.51**	**45.65**	43.71	46.99
Ours	50.58	**50.38**	**49.44**	**48.15**	**48.05**	**47.15**	46.40	45.40	**44.70**	**47.81**

[a] Indicates reimplementation.

Table 6
Out-of-domain few-shot lifelong learning performance on Meta-Dataset, MINIST, and CIFAR-10/-100.

Method	Signs	MSCOCO	MNIST	CIFAR-10	CIFAR-100
FSMB* [61]	45.46	45.14	82.92	61.29	47.34
C-FSCIL [48]	46.53	54.40	87.82	69.29	56.55
Ours	**49.15**	**58.34**	**93.75**	**75.77**	**63.32**

incorporates all the proposed components. In the first group of the table, the base session is trained on the first 8 datasets with their official training splits, while the 9 novel sessions are selected from the test split of the Meta-Dataset. Note that the classes and samples in the test split are excluded from the training split. As shown in the table, our method achieves an average accuracy of 55.73%. In the second group of the table, the base session is trained on the ImageNet subset, and each novel session introduces a new dataset derived from the previous session. In other words, in each novel session, the model is evaluated on both the newly added dataset and all prior datasets, presenting a cross-domain challenge. As shown in the table, our method demonstrates superior cross-domain transferability, whereas comparison methods exhibit a more significant performance decline as sessions progress.

4.3.4. Out-of-domain performance

To evaluate the out-of-domain FSLL performance of the proposed method, we train the model on the official training splits of the first 8 datasets and evaluate on the last 2 datasets of Meta-Dataset (i.e., Traffic Signs and MSCOCO). Moreover, we follow [63] to evaluate three external benchmarks: MNIST, CIFAR-10, and CIFAR-100. The performance is shown in Table 6. We report the average accuracy across 9 novel tasks for each dataset. The proposed method shows a significant improvement over the comparison methods.

4.3.5. Class-wise performance analysis

Fig. 6 shows the class-wise performance analysis on OfficeHome. We present the performance in the final session, wherein the model is required to predict all previously encountered classes. The confusion matrix Fig. 6(a) highlights a noticeable concentration of values along the diagonal, indicating that the proposed method can mitigate forgetting. Furthermore, we present the cosine similarity Fig. 6(b) between the prototypes. Notably, the similarity within the same class surpasses that of different classes. This observation demonstrates the model can effectively learn classes while maintaining minimal interference between each class.

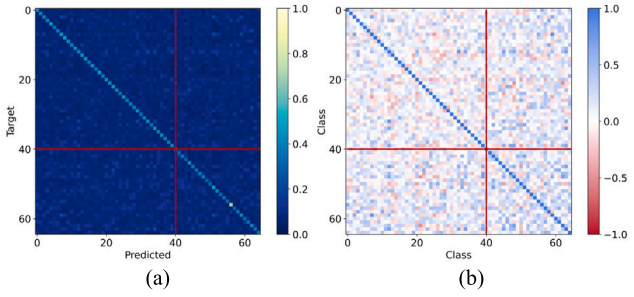

Fig. 6. Class-wise performance on OfficeHome. (a) The confusion matrix of prediction results with target labels. (b) Cosine similarity between the prototypes. The red cross splits the base session (40 classes) and the novel sessions.

4.4. Ablation analysis

Table 7 reports the ablation results on OfficeHome. The first row of the first and second groups represents the baseline model without and with the architecture (*i.e.,* twin layer) of the proposed method, respectively. Based on the two baseline models, the knowledge update (KU) method improves the average accuracy by 0.42% and 0.49%, respectively. The second and third groups represent the model without and with debiased base learning (DBL) strategy, respectively.

We can observe that the DBL improves the overall accuracy by 4.51%, which shows that learning new knowledge can benefit from

Table 7
Ablation study on OfficeHome. We evaluate the effectiveness of the Debiased Base Learning (DBL) strategy, Cross-domain Alignment (CDA), Prototype Alignment (PA), and Knowledge Update (KU).

DBL	CDA	KU	PA	Sessions									Average Acc.
				1	2	3	4	5	6	7	8	9	
				47.15	45.83	43.27	42.39	41.37	39.62	39.10	38.10	36.38	41.47
		✓		47.15	45.92	43.45	42.80	41.76	40.03	39.66	38.53	37.74	41.89
				48.16	47.57	43.92	43.46	43.09	41.93	40.89	39.70	38.04	42.97
	✓			48.16	47.57	44.92	44.37	44.02	43.19	42.41	40.59	38.89	43.79
		✓		48.16	47.92	44.18	43.67	43.75	42.34	41.74	40.48	38.93	43.46
			✓	51.79	52.99	50.95	50.08	49.14	48.18	47.73	46.16	44.06	49.01
	✓	✓		48.16	47.92	45.10	44.94	44.77	43.94	42.89	40.98	39.44	44.24
		✓	✓	51.79	53.12	51.26	50.41	49.57	48.10	47.73	46.34	44.67	49.22
	✓		✓	51.79	52.99	51.56	51.03	49.96	48.88	47.99	46.27	44.40	49.43
	✓	✓	✓	51.79	53.12	51.43	51.19	50.59	49.11	48.25	46.34	44.43	49.58
✓				55.15	54.12	49.09	48.23	47.54	45.31	44.49	42.65	40.73	47.48
✓	✓			55.15	54.12	50.09	49.18	47.89	46.24	45.31	43.36	42.02	48.15
✓		✓		55.15	54.25	49.22	48.40	47.97	45.94	45.35	43.54	41.58	47.93
✓			✓	59.47	60.11	58.94	57.61	56.48	54.58	53.91	51.81	49.35	55.81
✓	✓	✓		55.15	54.25	50.22	49.55	48.32	47.77	45.94	44.32	42.70	48.69
✓		✓	✓	59.47	60.33	59.11	57.89	**56.60**	54.65	53.79	51.78	49.39	55.89
✓	✓		✓	59.47	60.11	60.24	**58.14**	56.48	**55.43**	54.35	52.20	50.27	56.30
✓	✓	✓	✓	**59.47**	**60.33**	**60.37**	58.10	56.52	55.28	**54.50**	**52.45**	**50.65**	**56.41**

domain-invariant features. The effectiveness of PA is also noticeable, which improves the overall accuracy by 6.04%. During the learning process, the superiority of PA becomes progressively evident. It highlights the ability of PA to learn new knowledge while minimizing interference between classes. We can observe that the DBL exhibits a higher average accuracy throughout each session while showing a larger performance decline in the new sessions. This can be attributed to the fact that domain-invariant features can be transferred to the new task more effectively, which consequently presents the challenge of discriminating among the novel classes. This issue can be mitigated through the implementation of PA. With PA, the drop rate of the average accuracy in the new sessions is reduced from 14.42% to 10.12%. This indicates that the PA can alleviate the performance decline in new sessions. With all components integrates, the proposed method achieves a minimal performance decline of 8.82%. Note that the effectiveness of KU is more evident when relevant knowledge is involved. We will delve into this aspect further in the subsequent analysis.

4.4.1. Debiased base learning

Fig. 7 illustrates the effectiveness of DBL. We evaluate the base and novel accuracy in each session, respectively. The base accuracy is calculated on the predefined 40 base classes, and the novel accuracy is calculated on all seen novel classes in the current session. DBL improves the accuracy of base classes by 7.03%, and of novel classes by 9.39%. This demonstrates that a debiased base model can enhance the performance of learning new knowledge.

Fig. 8 shows the ablation study on the start epoch t of the second stage of DBL. Each epoch contains 1000 episodes which are randomly sampled from the training set. We can observe that there is a large improvement from $t = 0$ to $t = 1$, and the performance is stable when $t > 5$. Note that the choice of parameter t should take into account the episodes within each epoch and the characteristics of the base training set.

Twin model. We conducted experiments with varying degrees of training for the twin model. The average accuracy on all sessions is shown in Table 8. We chose the blocks from deep to shallow of the ResNet-12 as the twin model and evaluated the effects. We find that the performance of using the FC layer as the twin model is comparable to that of using the FC layer together with Blocks 4 and 3. Moreover, when using the whole network as the twin model, it requires more training epochs to achieve comparable performance. As a result, we adopt the FC layer as our twin model in all experiments for efficiency.

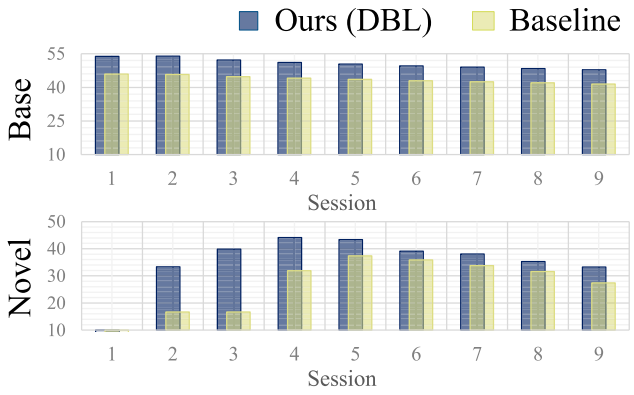

Fig. 7. Performance on base and novel classes. The base accuracy is evaluated on the predefined 40 base classes, and the novel accuracy is evaluated on all seen novel classes. The "Baseline" represents the model aligned with the architecture but without DBL.

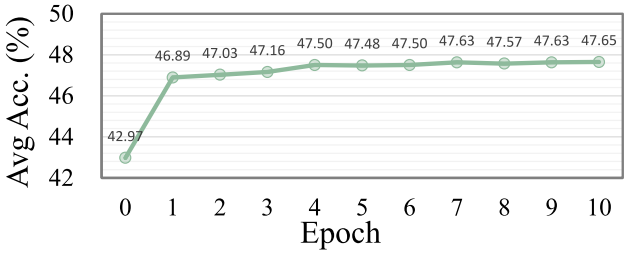

Fig. 8. The ablation study on the start epoch t of the second stage of DBL. We report the average accuracy (%) across all sessions.

Two-stage optimization. We use two-stage optimization for DBL. When optimizing twin layer $(l_{\varphi_S}, l_{\varphi_I})$ simultaneously, there is no assurance that l_{φ_I} will adaptively learn the domain-invariant features. Therefore, we first optimize l_{φ_S} and move l_{φ_I} away from l_{φ_S}. Then, we optimize l_{φ_I} to learn the domain-invariant features. We evaluate the one- and two-stage optimization, respectively. The results are shown in Fig. 9. DBL with two-stage training exhibits a higher average accuracy but shows a larger performance decline in the new sessions. This can be attributed to the fact that domain-invariant features are effective in transferring to new domains, but also presents the challenge of

Table 8
The ablation study of the twin model. Block 1–4 represents the basic block in the ResNet. The checkmark indicates that this module is part of the twin model.

ResNet-12				FC layer	Average Acc.
Block 1	Block 2	Block 3	Block 4		
				✓	47.48
			✓	✓	47.51
		✓	✓	✓	47.29
	✓	✓	✓	✓	46.97
✓	✓	✓	✓	✓	46.06

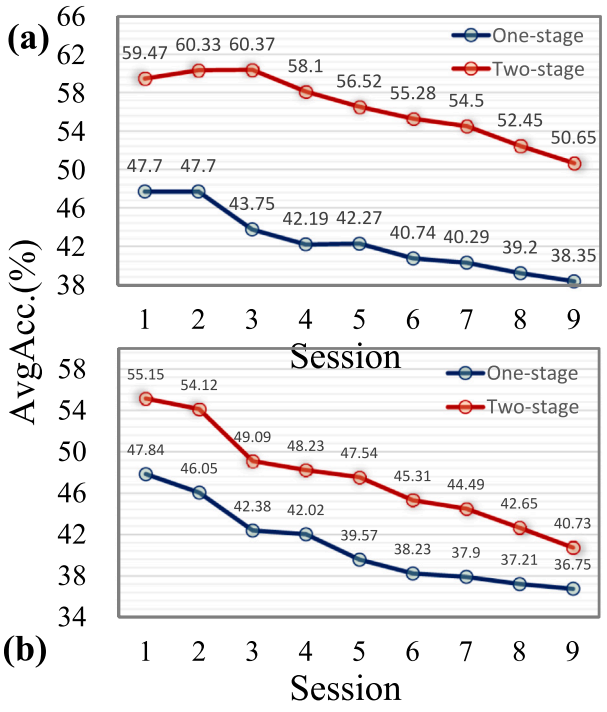

Fig. 9. Performance analysis of the one-stage and two-stage DBL strategies. We present the performance (a) with and (b) without PA.

discriminating among new classes. This can be mitigated through PA. Furthermore, we conduct the ablation study on the start epoch t of the second stage of DBL. We observed a large improvement from $t = 0$ to $t = 1$, and the performance is stable when $t > 5$.

4.4.2. Few-shot lifelong learning

The study of the iteration number r for cross-domain alignment (CDA) and k for prototype alignment (PA) is presented in Fig. 10. The evaluation on r is performed under the following settings: (*i*) the start epoch t of DBL is set at 5, (*ii*) the iteration number k of PA is set at 50, and (*iii*) in conjunction with KU. The optimizer is SGD with a learning rate of 0.1. We noticed an increase in average accuracy as the value of r increases from 1 to 8, while the rate of improvement diminishes when r is larger than 10. This phenomenon can be attributed to the model being likely to overfit the target domain as the iteration number becomes large. Therefore, we set $r = 8$ in our experiments. The evaluation on k is performed under the following settings: (*i*) the start epoch t of DBL is set at 5, (*ii*) the iteration number r of CDA is set at 8, and (*iii*) in conjunction with KU. We can observe that the model performs best at around $k = 50$. Fig. 11 illustrates the iteration sensitivity of the r and k. We can observe that the model achieves stable performance when k at the range of 40~50. When k is smaller (<25), the model needs a larger r to achieve comparable performance.

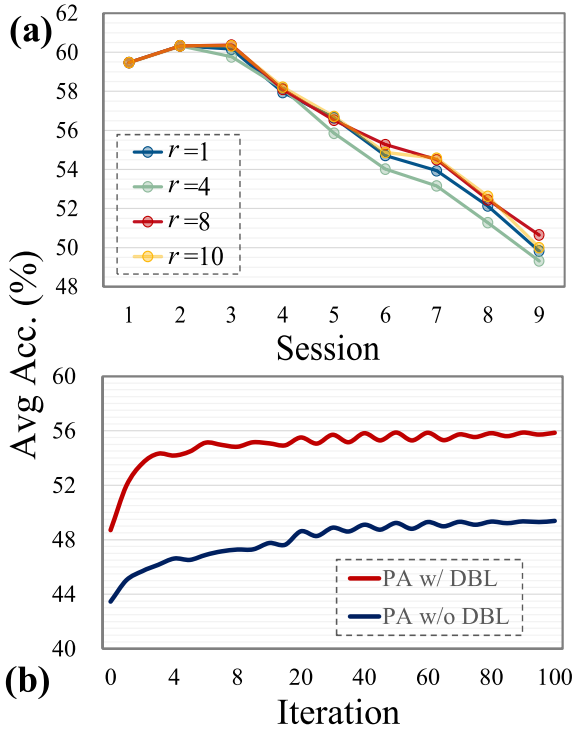

Fig. 10. Performance of CDA and PA in each session. (a) CDA with different iteration r. (b) PA with different iteration k.

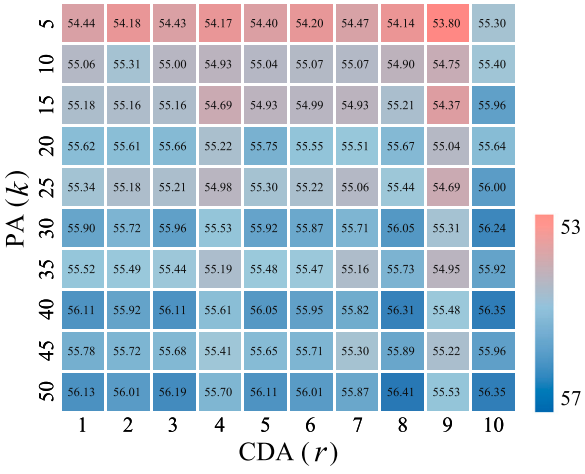

Fig. 11. Ablation study of the iteration number r in Cross-domain Alignment (CDA) and the iteration number k in Prototype Alignment (PA). We report the average accuracy (%) across all sessions.

In our experiment, we set these two hyperparameters to 8 and 50, respectively.

Fig. 12 illustrates the effectiveness of the knowledge update (KU). The increasing episode (e) represents the model encounters seen classes but in an unseen domain. From **(a)**, it is evident that the average accuracy improves with the increasing number of episodes. This demonstrates that KU can effectively update the acquired knowledge and fully leverage the limited data. Moreover, we compare the performance of KU under two baselines, i.e., with and without DBL. From **(b)**, we can observe that KU is effective in both baselines and the performance of

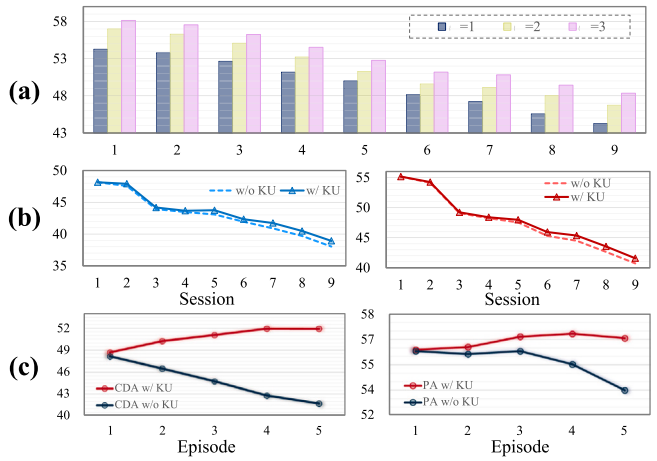

Fig. 12. (a) With increased episodes e, the performance is gradually increased. (b) With episode fix to 1, the impact of KU under two baselines, *i.e.*, with (red) and without (blue) DBL. (c) KU performs with CDA (top) and PA (bottom), respectively.

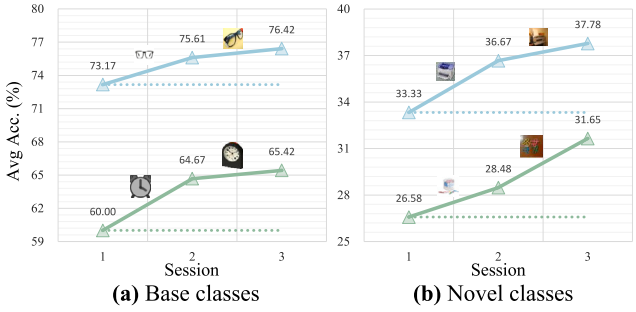

Fig. 13. The improvement made by knowledge update using various domain data from the same class. (a) and (b) represent the performance enhancement of the base (Alarm clock and Glasses) and novel (Printer and Paper clip) classes, respectively.

KU with DBL is consistently higher than that without DBL. Finally, we evaluate the performance of KU with CDA and PA. From **(c)**, we can observe that without KU, there is a noticeable decline in performance as the number of episode increase. This is mainly due to the feature shift, which is effectively mitigated by the implementation of KU.

To further assess the impact of cross-domain knowledge in FSLL, we evaluate the performance enhancement on base and novel classes, respectively. We perform the knowledge update by using different domain data from the same class. We show the performance improvement of the base and novel classes in Fig. 13. The observed improvements are noticeable in both base and novel classes, suggesting that cross-domain knowledge contributes to an overall performance boost for previously encountered classes. Specifically, we report two classes from the base classes (Alarm clock and Glasses) and two classes from the novel classes (Printer and Paper clip). The dashed line represents the performance of the base model. The solid line represents the performance of each class with the knowledge update. We can observe that the performance in both base and novel classes can be enhanced by learning knowledge from a different domain. Moreover, the novel classes have a lower start performance because the model has not learned the knowledge of the novel classes. However, the performance of the novel classes can be improved by learning the knowledge from the different domains.

4.5. Computation cost

Compared to the baseline network, Algorithm 1 incurs additional computational costs distributed among the three proposed components.

Table 9
The computation time of each components. The computation time of each component. We report the total time required for a complete session of learning.

Dataset	KU	CDA	PA	Session
mini-ImageNet	–	–	0.2 ms	2 s
CIFAR-100	–	–	0.2 ms	2 s
OfficeHome	0.1 ms	0.5 ms	0.2 ms	2 s
Meta-Dataset	0.8 ms	0.5 ms	0.2 ms	2 s

Specifically, during each novel session, the KU module requires $|C(s)|\times(3n+2)$ operations, resulting in a computational complexity of $O(n\cdot|C(s)|)$, where n is the dimension of the feature vector. The CDA module performs r updates, with computational cost arising from the forward inference of f_{θ_n} and the similarity computation. We denote the cost of forward inference as $O(f)$ and the similarity computation as $O(d)$. Therefore, the computational cost of CDA is $O(r\cdot|D^{(s)}|\cdot f\cdot d)$. Note that the inference in f_{θ_b} is part of the baseline network and is not included in the computation cost of CDA. The PA module requires k updates, where each update involves finetuning the f_{θ_n} for $|C(s)|$ times. The computational cost of $\mathcal{L}_{PA}$ is $O(|C(s)|\cdot f\cdot n)$, so the overall computational cost for updating f_{θ_n} is $(k\cdot|C(s)|\cdot f\cdot n)$. The computation time of each component is shown in Table 9, which reports the total time required for a complete session of learning. The experiments were conducted on an Intel Xeon Gold 6253 W CPU and an NVIDIA RTX 3090 GPU. Specifically, the Meta-Dataset varies in both the number of ways and shots for each task, so we report the average computational cost across repeated experiments.

5. Conclusions and discussions

In this paper, we investigate the effects of cross-domain knowledge in few-shot lifelong learning (FSLL) and correspondingly propose a new learning framework. The proposed framework equips the ability to continually learn in the cross-domain scenario while migrating forgetting and overfitting. Moreover, it also achieves state-of-the-art performance in non-cross-domain situations. Cross-domain FSLL has promising application value in real-world applications such as user authentication [64–67], which often face challenges due to diverse data sources, including variations across devices and environments [68,69]. The proposed framework offers three primary benefits for user authentication applications. First, it enables continual learning, allowing the system to adapt and improve itself over time with new data. Second, the framework incorporates cross-domain knowledge, enhancing the model's ability to generalize across various authentication scenarios. Third, it effectively handles few-shot data, ensuring robust performance even with limited training data.

Nonetheless, several potential challenges may arise when implemented in dynamic real-world scenarios. First, we pre-train the model in the base session to enhance knowledge transfer and its ability to prevent forgetting. However, the data from realistic downstream tasks exacerbates the risk of overfitting when continually updating the model, posing a challenge in maintaining the proposed model's generalizability for future tasks. Second, the proposed framework retains acquired knowledge in a memory buffer. In real-world scenarios, data streams often have greater complexity. Consequently, another challenge lies in updating, augmenting, and compressing the memory bank to align with the evolving learning scenarios dynamically. Third, the proposed cross-domain alignment mainly deals with the cross-domain issue within the image domain, which is short at learning across a larger domain gap, like the image-to-text or point cloud. For our future work, we aim to improve the robustness and practical relevance of our research by exploring additional real-world applications.

CRediT authorship contribution statement

Yaoyue Zheng: Writing – original draft, Software, Methodology, Investigation, Data curation, Conceptualization. **Xuetao Zhang:** Writing – review & editing, Supervision, Investigation, Funding acquisition. **Zhiqiang Tian:** Writing – review & editing, Supervision, Investigation, Funding acquisition. **Shaoyi Du:** Writing – review & editing, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by NSFC, China under Grant No. 62173269 and the Social Science Foundation of Shaanxi Province of China under Grant No. 2021K014.

Data availability

The data used in this study are publicly available.

References

[1] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, Y. Gong, Few-shot class-incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12183–12192, http://dx.doi.org/10.1109/CVPR42600.2020.01220.

[2] P. Mazumder, P. Singh, P. Rai, Few-shot lifelong learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 2337–2345, http://dx.doi.org/10.1609/aaai.v35i3.16334.

[3] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, N. Díaz-Rodríguez, Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges, Inf. Fusion 58 (2020) 52–68, http://dx.doi.org/10.1016/j.inffus.2019.12.004.

[4] P. Ruvolo, E. Eaton, ELLA: An efficient lifelong learning algorithm, in: International Conference on Machine Learning, PMLR, 2013, pp. 507–515.

[5] N.Y. Masse, G.D. Grant, D.J. Freedman, Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization, Proc. Natl. Acad. Sci. 115 (44) (2018) E10467–E10475, http://dx.doi.org/10.1073/pnas.1803839115.

[6] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C.H. Lampert, Icarl: Incremental classifier and representation learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2001–2010, http://dx.doi.org/10.1109/cvpr.2017.587.

[7] M. Masana, X. Liu, B. Twardowski, M. Menta, A.D. Bagdanov, J. Van De Weijer, Class-incremental learning: survey and performance evaluation on image classification, IEEE Trans. Pattern Anal. Mach. Intell. 45 (5) (2022) 5513–5533, http://dx.doi.org/10.1109/TPAMI.2022.3213473.

[8] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, Adv. Neural Inf. Process. Syst. 29 (2016) http://dx.doi.org/10.48550/arXiv.1606.04080.

[9] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, Adv. Neural Inf. Process. Syst. 30 (2017) http://dx.doi.org/10.48550/arXiv.1703.05175.

[10] D. Li, X. Yu, C. Xu, L. Petersson, H. Li, Transferring cross-domain knowledge for video sign language recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6205–6214, http://dx.doi.org/10.1109/CVPR42600.2020.00624.

[11] B. Li, P. Song, C. Zhao, Fusing consensus knowledge: A federated learning method for fault diagnosis via privacy-preserving reference under domain shift, Inf. Fusion (2024) 102290, http://dx.doi.org/10.1016/j.inffus.2024.102290.

[12] B. Zhang, X. Zhang, F. Huang, D. Miao, Cross-domain knowledge collaboration for blending-target domain adaptation, Inf. Process. Manage. 61 (4) (2024) 103730, http://dx.doi.org/10.1016/j.ipm.2024.103730.

[13] M. McCloskey, N.J. Cohen, Catastrophic interference in connectionist networks: The sequential learning problem, in: Psychology of Learning and Motivation, Vol. 24, Elsevier, 1989, pp. 109–165, http://dx.doi.org/10.1016/S0079-7421(08)60536-8.

[14] H. Yang, W. Huang, J. Liu, C. Li, S. Wang, Few-shot class-incremental learning for cross-domain disease classification, 2023, http://dx.doi.org/10.48550/arXiv.2304.05734, arXiv preprint arXiv:2304.05734.

[15] S. Yang, Z. Cai, Cross domain lifelong learning based on task similarity, IEEE Trans. Pattern Anal. Mach. Intell. (2023) http://dx.doi.org/10.1109/TPAMI.2023.3276991.

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

[17] H. Venkateswara, J. Eusebio, S. Chakraborty, S. Panchanathan, Deep hashing network for unsupervised domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5018–5027, http://dx.doi.org/10.1109/CVPR.2017.572.

[18] E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, U. Evci, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol, et al., Meta-dataset: A dataset of datasets for learning to learn from few examples, 2019, http://dx.doi.org/10.48550/arXiv.1903.03096, arXiv preprint arXiv:1903.03096.

[19] Z. Li, D. Hoiem, Learning without forgetting, IEEE Trans. Pattern Anal. Mach. Intell. 40 (12) (2017) 2935–2947, http://dx.doi.org/10.1109/TPAMI.2017.2773081.

[20] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A.A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al., Overcoming catastrophic forgetting in neural networks, Proc. Nat. Acad. Sci. 114 (13) (2017) 3521–3526, http://dx.doi.org/10.1073/pnas.1611835114.

[21] H. Shin, J.K. Lee, J. Kim, J. Kim, Continual learning with deep generative replay, Adv. Neural Inf. Process. Syst. 30 (2017) http://dx.doi.org/10.48550/arXiv.1705.08690.

[22] E. Belouadah, A. Popescu, Il2m: Class incremental learning with dual memory, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 583–592, http://dx.doi.org/10.1109/ICCV.2019.00067.

[23] F. Zenke, B. Poole, S. Ganguli, Continual learning through synaptic intelligence, in: International Conference on Machine Learning, PMLR, 2017, pp. 3987–3995, http://dx.doi.org/10.48550/arXiv.1703.04200.

[24] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, 2015, http://dx.doi.org/10.48550/arXiv.1503.02531, arXiv preprint arXiv:1503.02531.

[25] F.M. Castro, M.J. Marín-Jiménez, N. Guil, C. Schmid, K. Alahari, End-to-end incremental learning, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 233–248, http://dx.doi.org/10.1007/978-3-030-01258-8_15.

[26] S. Hou, X. Pan, C.C. Loy, Z. Wang, D. Lin, Learning a unified classifier incrementally via rebalancing, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 831–839, http://dx.doi.org/10.1109/CVPR.2019.00092.

[27] Y. Liu, Y. Su, A.-A. Liu, B. Schiele, Q. Sun, Mnemonics training: Multi-class incremental learning without forgetting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12245–12254, http://dx.doi.org/10.1109/CVPR42600.2020.01226.

[28] J. Bang, H. Kim, Y. Yoo, J.-W. Ha, J. Choi, Rainbow memory: Continual learning with a memory of diverse samples, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8218–8227, http://dx.doi.org/10.1109/CVPR46437.2021.00812.

[29] T. Lesort, A. Stoian, D. Filliat, Regularization shortcomings for continual learning, 2019, http://dx.doi.org/10.48550/arXiv.1912.03049, arXiv preprint arXiv:1912.03049.

[30] G. Shi, J. Chen, W. Zhang, L.-M. Zhan, X.-M. Wu, Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima, Adv. Neural Inf. Process. Syst. 34 (2021) 6747–6761, http://dx.doi.org/10.48550/arXiv.2111.01549.

[31] D. Lopez-Paz, M. Ranzato, Gradient episodic memory for continual learning, Adv. Neural Inf. Process. Syst. 30 (2017) http://dx.doi.org/10.48550/arXiv.1706.08840.

[32] A. Prabhu, P.H. Torr, P.K. Dokania, Gdumb: A simple approach that questions our progress in continual learning, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16, Springer, 2020, pp. 524–540, http://dx.doi.org/10.1007/978-3-030-58536-5_31.

[33] J.N. Kundu, R.M. Venkatesh, N. Venkat, A. Revanur, R.V. Babu, Class-incremental domain adaptation, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16, Springer, 2020, pp. 53–69, http://dx.doi.org/10.1007/978-3-030-58601-0_4.

[34] R. Volpi, D. Larlus, G. Rogez, Continual adaptation of visual representations via domain randomization and meta-learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4443–4453, http://dx.doi.org/10.1109/CVPR46437.2021.00442.

[35] C. Simon, M. Faraki, Y.-H. Tsai, X. Yu, S. Schulter, Y. Suh, M. Harandi, M. Chandraker, On generalizing beyond domains in cross-domain continual learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9265–9274, http://dx.doi.org/10.1109/CVPR52688.2022.00905.

[36] H.-Y. Tseng, H.-Y. Lee, J.-B. Huang, M.-H. Yang, Cross-domain few-shot classification via learned feature-wise transformation, 2020, http://dx.doi.org/10.48550/arXiv.2001.08735, arXiv preprint arXiv:2001.08735.

[37] Y. Fu, Y. Fu, Y.-G. Jiang, Meta-fdmixup: Cross-domain few-shot learning guided by labeled target data, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 5326–5334, http://dx.doi.org/10.1145/3474085.3475655.

[38] T. Adler, J. Brandstetter, M. Widrich, A. Mayr, D. Kreil, M. Kopp, G. Klambauer, S. Hochreiter, Cross-domain few-shot learning by representation fusion, 2020, http://dx.doi.org/10.48550/arXiv.2010.06498, arXiv preprint arXiv:2010.06498.

[39] Y. Zhao, T. Zhang, J. Li, Y. Tian, Dual adaptive representation alignment for cross-domain few-shot learning, IEEE Trans. Pattern Anal. Mach. Intell. (2023) http://dx.doi.org/10.1109/TPAMI.2023.3272697.

[40] Z. Chi, L. Gu, H. Liu, Y. Wang, Y. Yu, J. Tang, Metafscil: A meta-learning approach for few-shot class incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14166–14175, http://dx.doi.org/10.1109/CVPR52688.2022.01377.

[41] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: International Conference on Machine Learning, PMLR, 2015, pp. 1180–1189, http://dx.doi.org/10.48550/arXiv.1409.7495.

[42] M. Salzmann, C.H. Ek, R. Urtasun, T. Darrell, Factorized orthogonal latent spaces, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 701–708.

[43] C. Simon, P. Koniusz, M. Harandi, On learning the geodesic path for incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1591–1600, http://dx.doi.org/10.1109/CVPR46437.2021.00164.

[44] P.O. Pinheiro, Unsupervised domain adaptation with similarity learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8004–8013, http://dx.doi.org/10.1109/CVPR.2018.00835.

[45] H. Liu, L. Gu, Z. Chi, Y. Wang, Y. Yu, J. Chen, J. Tang, Few-shot class-incremental learning via entropy-regularized data-free replay, in: European Conference on Computer Vision, Springer, 2022, pp. 146–162, http://dx.doi.org/10.1007/978-3-031-20053-3_9.

[46] K. Chen, C.-G. Lee, Incremental few-shot learning via vector quantization in deep embedded space, in: International Conference on Learning Representations, 2020, https://openreview.net/forum?id=3SV-ZePhnZM.

[47] C. Zhang, N. Song, G. Lin, Y. Zheng, P. Pan, Y. Xu, Few-shot incremental learning with continually evolved classifiers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12455–12464, http://dx.doi.org/10.1109/CVPR46437.2021.01227.

[48] M. Hersche, G. Karunaratne, G. Cherubini, L. Benini, A. Sebastian, A. Rahimi, Constrained few-shot class-incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9057–9067, http://dx.doi.org/10.1109/CVPR52688.2022.00885.

[49] L. Zhao, J. Lu, Y. Xu, Z. Cheng, D. Guo, Y. Niu, X. Fang, Few-shot class-incremental learning via class-aware bilateral distillation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 11838–11847, http://dx.doi.org/10.1109/CVPR52729.2023.01139.

[50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255, http://dx.doi.org/10.1109/cvpr.2009.5206848.

[51] 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, Int. J. Comput. Vis. 115 (2015) 211–252, http://dx.doi.org/10.1007/s11263-015-0816-y.

[52] B.M. Lake, R. Salakhutdinov, J.B. Tenenbaum, Human-level concept learning through probabilistic program induction, Science 350 (6266) (2015) 1332–1338, http://dx.doi.org/10.1126/science.aab3050.

[53] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, A. Vedaldi, Fine-grained visual classification of aircraft, 2013, http://dx.doi.org/10.48550/arXiv.1306.5151, arXiv preprint arXiv:1306.5151.

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

[55] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3606–3613, http://dx.doi.org/10.48550/arXiv.1311.3618.

[56] J. Jongejan, H. Rowley, T. Kawashima, J. Kim, N. Fox-Gieg, The Quick, Draw!-Ai Experiment, Mount View, CA, 2016, p. 4, (Accessed 17 February 2018).

[57] B. Schroeder, Y. Cui, Fgvcx fungi classification challenge 2018, 2018, Available online: github.com/visipedia/fgvcx_fungi_comp (Accessed 14 July 2021).

[58] M.-E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, IEEE, 2008, pp. 722–729, http://dx.doi.org/10.1109/ICVGIP.2008.47.

[59] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, C. Igel, Detection of traffic signs in real-world images: The German traffic sign detection benchmark, in: The 2013 International Joint Conference on Neural Networks, IJCNN, Ieee, 2013, pp. 1–8, http://dx.doi.org/10.1109/IJCNN.2013.6706807.

[60] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755, http://dx.doi.org/10.1007/978-3-319-10602-1_48.

[61] Y. Chen, Z. Liu, H. Xu, T. Darrell, X. Wang, Meta-baseline: Exploring simple meta-learning for few-shot learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9062–9071, http://dx.doi.org/10.48550/arXiv.2003.04390.

[62] Y. Zou, S. Zhang, Y. Li, R. Li, Margin-based few-shot class-incremental learning with class-level overfitting mitigation, Adv. Neural Inf. Process. Syst. 35 (2022) 27267–27279, http://dx.doi.org/10.48550/arXiv.2210.04524.

[63] J. Requeima, J. Gordon, J. Bronskill, S. Nowozin, R.E. Turner, Fast and flexible multi-task classification using conditional neural adaptive processes, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. dÁlché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 7957–7968.

[64] C. Wu, K. He, J. Chen, Z. Zhao, R. Du, Toward robust detection of puppet attacks via characterizing fingertip-touch behaviors, IEEE Trans. Dependable Secure Comput. 19 (6) (2021) 4002–4018, http://dx.doi.org/10.1109/TDSC.2021.3116552.

[65] C.-W. Lien, S. Vhaduri, Challenges and opportunities of biometric user authentication in the age of iot: A survey, ACM Comput. Surv. 56 (1) (2023) 1–37, http://dx.doi.org/10.1145/3603705.

[66] H. Cao, D. Liu, H. Jiang, C. Cai, T. Zheng, J.C. Lui, J. Luo, HandKey: Knocking-triggered robust vibration signature for keyless unlocking, IEEE Trans. Mob. Comput. 23 (1) (2022) 520–534, http://dx.doi.org/10.1109/TMC.2022.3216868.

[67] T. Amesaka, H. Watanabe, M. Sugimoto, Y. Sugiura, B. Shizuki, User authentication method for hearables using sound leakage signals, in: Proceedings of the 2023 ACM International Symposium on Wearable Computers, 2023, pp. 119–123, http://dx.doi.org/10.1145/3594738.3611376.

[68] H. Cao, D. Liu, H. Jiang, J. Luo, MagSign: Harnessing dynamic magnetism for user authentication on IoT devices, IEEE Trans. Mob. Comput. 23 (1) (2022) 597–611, http://dx.doi.org/10.1109/TMC.2022.3216851.

[69] J. Al-Saraireh, M.R. AlJa'afreh, Keystroke and swipe biometrics fusion to enhance smartphones authentication, Comput. Secur. 125 (2023) 103022, http://dx.doi.org/10.1016/j.cose.2022.103022.