温州大學
WENZHOU UNIVERSITY

毕业设计 (论文)

外文资料翻译

题　　目：　基于卷积神经网络的手写数字及写字人识别

学　　院：计算机与人工智能学院　专　　业：　计算机科学与技术

班　　级：　22 级计算机 1 班　学　　号：　12350004

姓　　名：　王 小 明　指导教师：　王大明 (教授)

温州大学教务处制

外文资料来源及题目（注：含作者、书名、杂志名或外文数据库名等，英文文章或段落标题）

外文的题目为 Building a deep learning-based QA system from a CQA dataset，文章作者为 Jin Sol，Lian Xu，Jung Hanearl，Park Jinsoo，Suh Jihae，工作单位分别为 College of Business Administration, Seoul National University, Seoul, Republic of Korea;College of Business Administration, Seoul National University of Science Technology, Seoul, Republic of Korea。该文章被收在 Elsevier 期刊中，DOL 为 10.1016/J.DSS.2023.114038。

译成中文后题目

从 CQA 数据集中构建基于深度学习的问答系统

指导教师审阅意见：

签名：

年　　月　　日

译文：

1 摘要

人工机器阅读理解（MRC）数据集是训练的答案提取部分所必需的问答（QA）系统。然而，在现实世界中通常找不到具有问题-段落-答案对的高质量、结构良好的数据集。此外，更新或构建 MRC 数据集这是一件具有挑战性且代价高昂的事情。为了解决这些缺点，我们提出了一种使用大规模英语社区问答（CQA）数据集（即 Stack Exchange）由 3081834 个问答对组成。QA 系统采用了分类器-检索器-汇总器的结构设计。问题分类器和答案检索器部分基于来自变压器的双向编码器表示（BERT）谷歌的自然语言处理（NLP）模型，摘要部分介绍了一种基于深度学习的文本到文本转换转换器（T5）模型来总结长答案。我们实例化了提议的 QA 系统包含来自 CQA 数据集的 140 个主题（包括生物学、法律、政治等主题），并进行人工和自动评估。我们的系统给出了令人鼓舞的结果，考虑到它为测试集中的问题提供了高质量的答案，并满足了开发 QA 系统的要求 MRC 数据集。我们的研究结果显示了在没有受人造数据集的限制，这是开放领域或特定领域 QA 系统研究的重要一步。

2 相关工作

2.1 QA 系统

QA 系统可分为两大类：开放域和域特异性。开放领域 QA 系统可以回答来自不同领域的问题，不需要特定领域的词典。随着训练 MRC 数据集（如 SQuAD）的发布基于维基百科文章，CNN/每日邮报基于新闻文章，CBT 基于儿童书籍，越来越多的开放领域 QA 系统正在使用机器阅读方法进行回答提取。例如，Chen 等人提出了一个开放域 QA 系统 DrQA，使用由 SQuAD 数据集训练的 MRC 模型（多层 RNN 架构）来检测和提取答案从单个文档或一小部分文档集合；渡边等人提出了一种开放域 QA 系统，该系

统由用于检测答案的 MRC 模型（具有注意力和结构的 RNN）跨越检索到的文档或固定词汇表的文本。相比之下，特定领域的 QA 系统侧重于专业化领域（如就业、医学和农业问题）和使用提供更精确答案的特定语言资源。前面已经介绍了几个特定领域的 QA 系统工作，如医疗领域的 QA 系统 [3] 和烹饪领域的 QA 系统。由于特定领域的 QA 系统需要生成根据专家设计的词典提供专业解决方案在特定领域，在本研究中，我们只实现了开放域 QA 系统。CQA 网站是一种通过合作努力的网站，其中提问者和回答者聚在一起交流知识，满足他们的信息需求。寻求答案的提问者上传他们的问题到 CQA，任何具有相关知识的回答者都可以将答案发布到问题。这些社区使用游戏化机制，例如声誉积分和徽章，以激励用户提高参与度。回答者可以赢得声誉并建立通过在特定领域提供高质量的答案，使自己成为专家. 由 CQA 服务的答案是由人类生成的提问者期望 CQA 提供更好、更专业的答案，因为与传统的 QA 系统相比。CQA 上的一些答案服务，包括回答者的技术专长，尤其是他们的观点、推理解释和建议是很难通过一般的网络搜索获得。基于 CQA 数据集的优势，我们使用获得的问答对从 CQA 站点建立 QA 系统。然而，CQA 数据集仍然遭受低质量答案的困扰，例如：(1) 仅包含以下内容的答案没有任何其他解释的 URL 链接，(2) 不相关的答案问题或意见，(3) 具有非常简短描述的答案以及有限的理由和证据，以及 (4) 拼写错误的答案或一种非正式的写作风格。因此，基于 BERT 的语义采用相似算法过滤掉所有四种类型的低质量回答语料库，以便我们的系统能够提供更高质量和为用户提供相关答案。

2.1.1 深度学习技术

最近，深度学习和预训练模型已经证明在几个自然语言任务中取得了优异的成绩。特别地，微调预训练模型，如 OpenAI GPT（生成预训练）、GPT-2 和 BERT，已经获得了最先进的结果。GPT 是一个基于转换器的语言模型，它是设计目标如下：“给定之前的一组单词一个句子，预测下一个单词。”GPT-2 达到最先进水平文本生成的性能。BERT 是谷歌于 2018 年发布

的一种基于深度学习的 NLP 语言模型，旨在通过联合在所有层中调节左右上下文。相比使用以前的模型架构，其中只有单方面的上下文模型中包含知识，BERT 使用双向编码器可以从两个方向学习上下文信息。在对未标记的数据进行预训练后，BERT 采用微调方法相关的下游任务，如文本分类、MRC、情感分析和自然语言推理。许多研究提到 BERT 在几个 NLP 中取得了惊人的成绩任务，并优于大多数基于特征的表示方法，如 Word2Verc、Glove、CoVe 和 ELMo。QA 研究领域的重点不仅在于找到准确的答案，但在提供提问者要求的信息时更简洁。正如 Lloret 等人所指出的，文本摘要是一种 QA 系统的必要组成部分。T5 是来自谷歌的一个基于转换器的语言模型，它将每个语言问题转换为文本到文本格式。T5 是一个基于像 BERT 这样的变换器的预训练模型，不同之处在于它同时接收输入和以文本形式输出。因此，它专门从事文本总结、文本生成和机器翻译。谷歌的 Colossal CleanT5 任务中使用的爬行语料库（C4）数据集包含数百个千兆字节的英文文本。与 PEGASUS 模型相比，它的表现优于所有其他车型，如 BART 和 PEGASUS 对 BBC 新闻数据的文本摘要。

原文（可附 PDF 格式，可贴图片格式，可直接贴原文）

Information Fusion 103 (2024) 102020

Contents lists available at ScienceDirect

Information Fusion

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

Adaptive online continual multi-view learning

Yang Yu [a], Zhekai Du [b], Lichao Meng [b], Jingjing Li [b,*], Jiang Hu [a]

[a] Department of Orthopaedics, Sichuan Provincial People's Hospital; School of Medicine, University of Electronic Science and Technology of China, Chengdu, China
[b] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

ARTICLE INFO

Keywords:
Domain adaptation
Continual learning
Multi-view learning
Lifelong learning

ABSTRACT

Deep neural networks (DNNs) have gained great success in information fusion. However, recent studies report that DNNs are suffering from catastrophic forgetting, i.e., DNNs would forget the knowledge learned from previous tasks when training on the current task. To address this issue, continual learning is proposed to enhance long-term memories for DNNs. Since continual learning is very challenging, existing work simplifies the setting to simulate the sequentially online multi-task learning paradigm. Specifically, existing works commonly split one dataset into multiple disjoint categories to get multiple tasks that follow the same marginal distribution. We argue that this setting is too simple to approximate the real-world applications. In real-world scenarios, the data distributions of sequentially arrived tasks would change significantly from time to time, e.g., the lighting from day to night, and the background from spring to winter. Thus, the real-world applications are in a multi-view manner, yet existing methods ignore this challenge. To tame this, we propose Adaptive Online Continual Multi-view Learning (AOCML) to align distributions and reduce catastrophic forgetting as new tasks arrive. AOCML integrates experience replay and adversarial learning in an end-to-end framework, which stores samples in a memory buffer to replay previous tasks, while leveraging a discriminator to adaptively align distributions across views on-the-fly. In addition to common replay buffer, we also incorporate a soft label-based replay and an entropy-based reweighting to further prevent forgetting. Extensive experiments on four datasets verify that our method is able to significantly outperform previous CL methods and our method pushes CL one step forward towards practical multi-view orientation.

1. Introduction

Let us start our topic with the memory that you were learning numbers, e.g., from one to ten, as a kid. Generally, you cannot remember them all at once. You try to remember them progressively or sequentially, e.g., one, two for today and three, four for tomorrow. No one doubts that you would forget two when you learn three. However, recent studies on deep neural networks (DNNs) show that DNNs would forget the knowledge of previous tasks when it is trained on the current task, especially when the tasks are disjoint and learned sequentially [1–3]. This phenomenon is notoriously known as catastrophic forgetting [4], which is standing in the way of DNNs towards ideal artificial intelligence. To address this problem, continual learning [5–8], also known as lifelong learning, has been proposed recently. The goal of continual learning is to accumulate and improve knowledge in a sequence of tasks without forgetting.

Existing continual learning methods can be roughly grouped into three categories: memory-based methods [2,9,10], structure-based methods [11,12] and regularization-based methods [1,5]. Specifically, memory-based continual learning stores a few samples of previous tasks and replays them when training new tasks. These rehearsal samples can be real samples of previous tasks or synthesized samples generated by GANs and VAEs [10]. For instance, GEM [9] leverages an episodic memory, which stores a subset of the observed examples from previous tasks, to avoid forgetting. It is easy to find that memory-based methods focus on samples, the network capacity or architecture of the network is fixed. Structure-based methods, in contrast, learn dynamical structures to accommodate knowledge. Specifically, structure-based methods generally increase the network capacity to memory previous tasks [12]. At last, regularization-based methods put a constraint on the network parameters and tune them based on the possibility of forgetting. For instance, EWC [1] remembers old tasks by selectively slowing down learning on the weights important for those tasks.

Although existing continual learning methods have specific pros and cons, they relax or simplify the problem by splitting one dataset into several disjoint tasks as shown in Fig. 1. For instance, the digit dataset MNIST [13] is widely used in the community. Existing methods generally split it into five tasks, e.g., {0,1}, {2,3}, …,{8,9}, for evaluation [2,5,14]. In this paper, we argue that this setting is too simple to simulate

* Corresponding author.
E-mail address: lijin117@yeah.net (J. Li).

https://doi.org/10.1016/j.inffus.2023.102020
Received 4 May 2023; Received in revised form 20 July 2023; Accepted 13 September 2023
Available online 22 September 2023

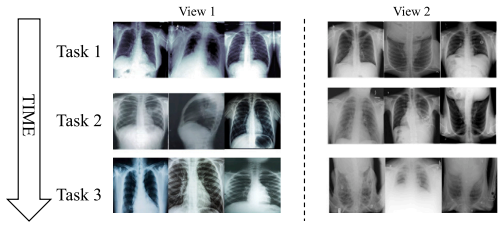


Fig. 1. An illustration of continual learning which consists of three tasks. In previous methods, different tasks are split from the same dataset which ignores the distributions gaps among different views. They assume that all the tasks are in a single-view setting. Our method, shown in the right part, challenges adaptive online multi-view learning where different sample distributions in different tasks can change dramatically, i.e., samples are from multiple views.

real-world continual learning scenarios. Specifically, a key challenge in learning from a never-ending stream of data is that the data distributions of the chronological tasks would change significantly from time to time. For instance, the varying weather conditions at different times, e.g., from foggy to sunny and then to rainy, can remarkably change the appearance of captured images. Besides, in medical image analysis, due to different scanners, scanning parameters, and subject groups, etc., there are often significant differences between different images. Data samples in these real-world continual learning tasks often follow different data distributions. In other words, real-world samples are from multiple views. However, samples split from the same dataset often have the same or very similar marginal distribution. It is well known that a machine learning algorithm would fail if the training set and testing set are not independent and identically distributed. Therefore, existing continual learning methods are vulnerable to multi-view continual learning demands. In the experiments reported in Section 4.4, we will show that the performance of previous state-of-the-art continual learning methods would drop significantly if samples from different views are used for testing, even if the tested instances have the same semantic information as previously learned samples.

In this paper, we investigate adaptive online continual learning which takes the distribution shifts among different tasks into consideration, i.e., via a multi-view learning manner [15,16]. The goal of our method is not only overcoming catastrophic forgetting but also aligning the feature representations of different views. Specifically, we leverage a very tiny memory buffer for each view to store a few samples from previous tasks since memory-based methods have been proven to be effective for the catastrophic forgetting issue. At the same time, we train an adversarial networks to mitigate the distribution gaps of various views. The adversarial networks consist of a generator and a discriminators. The generator learns feature representations for samples from different tasks. At the same time, the discriminator distinguishes whether a sample is from the replaying buffer or from the current task. The generator and the discriminator are adversarially trained by optimizing a two-player minimax game. It is worth noting that the adversarial networks have two merits. On one hand, since the samples in different replaying buffers may have different data distributions, confusing the discriminator is able to align the distributions among different views. On the other hand, since the replaying samples are from old tasks, confusing the discriminator encourages the generator to learn features which are indistinguishable between old tasks and new tasks, and therefore memorize old tasks. Besides, unlike previous methods that use either experience replay or adversarial learning alone, this unified framework enables online adaptation to continuously evolving multi-view tasks. We also introduce techniques including soft label-based replay and entropy-based re-weighting to focus on important samples in memory and further reduce forgetting. In summary, our contributions in this paper are three-fold: (1) We argue that existing continual learning settings are too simplified to reflect the real-world multi-view challenges. The distribution gaps among different views and tasks should be considered. (2) We propose a novel method named adaptive online continual multi-view learning (AOCML) to address the distribution shift issue, which unifies experience replay and adversarial learning in an integrated framework, coupled with a soft label-based replay and a sample reweighting schema. We propose a new experimental protocol which involves multiple datasets into evaluation to simulate the real-world scenarios. (3) We conduct extensive experiments on four datasets to verify that existing state-of-the-art continual learning methods are vulnerable to distribution gaps and our method can well handle the issue.

2. Related work

2.1. Continual learning

Continual learning [5,10,14,17,18] handles the catastrophic forgetting [1,3] issues in neural networks. Catastrophic forgetting refers to the problem that the neural networks may completely and abruptly forget previously learned information once new tasks are involved. The study on catastrophic forgetting has a long history since 1980s [4]. Recently, with the sweeping success of deep neural networks, catastrophic forgetting is attracting increasing attention in the community [2,14].

Several approaches have been proposed to avoid catastrophic forgetting under the name of continual learning. The simplest ones are like fine-tuning which fix several layers of the deep networks and tune latter layers for the new tasks [19]. The drawback of these methods is that they are hard to scale-up when facing a never-ending stream of data. As we briefed in the introduction, recent continual learning methods are mainly from three categories according to their working schemes. The memory-based methods [2,9,10,14] use an additional buffer to store previous samples. Samples from both the replaying buffer and the current task are used for training. An challenge in memory-based methods is that the size of replaying buffer is limited, it is hard to decide which samples should be stored. To tackle this challenge, Aljundi et al. [2] propose to store the samples which are most interfered. Rios et al. [20] bypass the storage of real data by training generative adversarial networks (GANs) to store data distributions of previous tasks. The structure-based methods [11,12] exploit modularity and connect the knowledge of a task to a specific modularity, such as several neurons or parameters. Then, they fix the old modularities and accumulate new knowledge by dynamically add new modularities. For instance, Rusu et al. [11] propose progressive neural networks which statically grows the architecture and fix previous learned modularities to achieve a very low forgetting rate. The regularization-based methods [1,5] deploy a metric to evaluate the importance of the parameters in a neural network. A penalty or regularization would be applied if the important parameters are changed when training new tasks. Our method falls into the category of leveraging a replaying buffer to overcome catastrophic forgetting. We find that memory-based methods are easy to implement and able to achieve good performance. Noting that Kurle et al. [21] strive to achieve continual learning for non-stationary data through a memory-based online variational Bayes. They primarily focus on addressing the distribution shift along the sequential training data. In contrast, our work confronts the distribution shift that occurs between the training and test data.

At last, it is worth noting that continual learning is related to multi-task learning [22]. The difference is that the training samples in all tasks are available for multi-task learning, while samples in different tasks arrive in sequence for continual learning. We cannot access the data of task 2 when we learn task 1. Therefore, multi-task learning can be seen as an upper bound of continual learning. In continual learning, we fight with catastrophic forgetting. In multi-task learning, we exploit commonalities and differences across tasks (see Fig. 2).

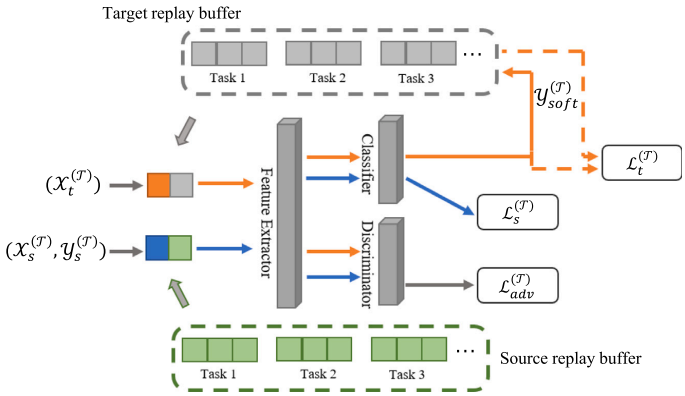


Fig. 2. Illustration of AOCML when learning task $\mathcal{T}$. AOCML consists of a feature extractor F, a classifier C and a domain discriminator D. The replay buffer stores the labeled source samples $(\mathcal{X}_s^{(\mathcal{T})}, \mathcal{Y}_s^{(\mathcal{T})})$ (blue) and the target samples (orange) with the soft label $(\mathcal{X}_t^{(\mathcal{T})}, \mathcal{Y}_{soft}^{(\mathcal{T})})$. We feed both the replayed samples and incoming samples into the model. The green and gray squares denotes source replayed samples and target replayed samples, respectively. The output contains (a) the cross-entropy loss on source samples (blue solid lines), (b) adversarial loss for adversarial learning (gray solid lines), and (c) MSE loss on replayed target samples (orange dashed lines).

2.2. Adaptive learning

There is a common assumption behind machine learning algorithms that the training set and the testing set should follow a same data distribution, e.g., marginal distribution and conditional distribution. However, this assumption cannot be guaranteed in many real-world applications. Thus, how to apply a model trained on a source domain to a target domain which has different data distribution with the source domain is an important topic in the community. Methods which addresses the distribution gaps between different domains are known as transfer learning or domain adaptation [23–26]. Adaptive learning has been successfully used in many fields such as image classification [23], object detection [27] and semantic segmentation [24]. The main idea of adaptive learning is to deploy a distance metric [23] or an adversarial loss [28] to align the distributions of the source domain and the target domain. In this paper, we leverage the idea of adaptive learning to handle the domain shift in continual learning.

2.3. Multi-view learning

In visual recognition tasks, an object can be represented by multiple views, such as different angles, different modalities and different devices. Although different views share the same semantic information, the data distributions varies from view to view. Existing multi-view learning methods [16,16,29] try to learn the view-invariant information to simultaneously handle the data from multiple views. In this paper, we focus on continual learning [5] where samples in different tasks are from multiple views. We leverage adversarial learning to mitigate the data distribution gaps among various views.

3. Proposed method

3.1. Problem setup

Continual learning attempts to challenge the problem of catastrophic forgetting introduced by continuous arrivals of training data in conventional machine learning. Many previous methods in continual learning area follow the same protocol [2,9,30,31]. Different from them, we also consider the distribution discrepancy within each task.

In this paper, we have labeled samples from a source distribution D_s and unlabeled samples from a target distribution D_t, which share the same feature space and output space. Then we split the source and the target data into sequences of disjoint tasks delineated by clear boundaries. Specifically, let $D_s^{(n)}$ and $D_t^{(n)}$ be the distributions of task n from the source domain and the target domain, respectively. $D_s = \{D_s^{(1)}, \dots, D_s^{(n)}\}$ and $D_t = \{D_t^{(1)}, \dots, D_t^{(n)}\}$ are the sets of marginal distributions. $(\mathcal{X}_s^{(n)}, \mathcal{Y}_s^{(n)})$ and $(\mathcal{X}_t^{(n)}, \mathcal{Y}_t^{(n)})$ are from $D_s^{(n)}$ and $D_t^{(n)}$ respectively, where $\mathcal{X}$ is the set of input data and $\mathcal{Y}$ the set of corresponding ground-truth labels.

During the learning process, the model receives a batch of new data $(\mathcal{X}_s^{(n)}, \mathcal{Y}_s^{(n)})$ and $(\mathcal{X}_t^{(n)})$ without the ground-truth labels $\mathcal{Y}_t^{(n)}$. For the current task n, our goal is to learn a function $f : x \to y$, which is comprised of a feature extractor F and a classifier C, to minimize the classification loss on incoming source data $(\mathcal{X}_s^{(n)}, \mathcal{Y}_s^{(n)})$, as well as reduce the distribution discrepancy between the labeled source data and the unlabeled target data. Besides, as the conventional continual learning supposes, we should also significantly reduce the performance degradation of previous tasks. Mathematically, we aim to minimize the empirical risk of all seen source data and target data:

$$\begin{aligned}&\sum_{n=1}^{\mathcal{T}} \mathbb{E}_{(\mathcal{X}_s^{(n)}, \mathcal{Y}_s^{(n)}) \sim D_s^{(n)}} [\ell(C(F(\mathcal{X}_s^{(n)}; \theta_F); \theta_C), \mathcal{Y}_s^{(n)})] \\ &+ \sum_{n=1}^{\mathcal{T}} \mathbb{E}_{(\mathcal{X}_t^{(n)}, \mathcal{Y}_t^{(n)}) \sim D_t^{(n)}} [\ell(C(F(\mathcal{X}_t^{(n)}; \theta_F); \theta_C), \mathcal{Y}_t^{(n)})],\end{aligned} \tag{1}$$

where ℓ is a loss function. $\mathcal{T}$ is the index of the current task. θ_F and θ_C represent the parameters of the feature extractor and the classifier, respectively.

3.2. Experience replay with soft labels

In continual learning, the data from previous tasks is no longer available when learning the current task. Therefore, a shared classifier may adjust parameters in a way that increases performance on the current task, while hurts performance on previously learned tasks, which is also known as catastrophic forgetting. To alleviate this issue, a straightforward idea is to maintain a replay memory by storing a limited number of previous data that can be replayed later. In this paper, our method falls into this paradigm, we replace the loss ℓ by the cross-entropy loss ℓ_{ce}. By jointly training the samples from the current task $\mathcal{T}$ and the replay buffer $\mathcal{R}$, we have the following supervised loss,

$$\begin{aligned}\mathcal{L}_s^{(\mathcal{T})} = &\frac{1}{N_{\mathcal{T}}} \ell_{ce}(C(F(\mathcal{X}_s^{(\mathcal{T})}; \theta_F), \mathcal{Y}_s^{(\mathcal{T})}; \theta_C)) \\ &+ \frac{1}{N_{\mathcal{R}}} \ell_{ce}(C(F(\mathcal{X}^{(\mathcal{R}_s)}; \theta_F), \mathcal{Y}^{(\mathcal{R}_s)}; \theta_C)),\end{aligned} \tag{2}$$

Algorithm 1 Learning Process for current task $\mathcal{T}$

Input: Learning rate α; Batch size $\mathcal{B}$; Buffer batch size $\mathcal{B}_b$; Epoch $\mathcal{E}$; Source memory size per task C_s; Target memory size per task C_t;
Output: $\Theta^* = \{\theta_F^*, \theta_C^*, \theta_G^*\}$.
Initialize:
Source memory buffer $\mathcal{M}_s$ with size C_s;
Target (soft) memory buffer $\mathcal{M}_t$ with size C_t;
Parameters Θ;
Optimization:

```
for E iterations do
    (X^(R_s), Y^(R_s)) ← Sample B_b source replay samples from previous tasks
    (X^(R_t), Y_soft^(R_t)) ← Sample B_b target replay samples from previous tasks
    (X_s^(T), Y_s^(T)) ← Sample B source samples from the distribution of the current task t
    (X_t^(T)) ← Sample B target samples from the distribution of the current task t
    (X_s, Y_s) ← (X_s^(T), Y_s^(T)) ∪ (X^(R_s), Y^(R_s))
    (X_t) ← (X_t^(T)) ∪ (X^(R_t))
    Calculate L_s^(T) with (x_s, y_s) according to Eq. (2).
    Calculate L_t^(T) with (X^(R_t), Y_soft^(R_t)) according to Eq. (3).
    Calculate L_adv^(T) with x_s ∪ x_t according to Eq. (5).
    Perform parameter updating for θ_F and θ_C according to Eq. (6)
    Perform parameter updating for θ_D according to Eq. (7)
end for
```

where $N_{\mathcal{T}}$ and $N_{\mathcal{R}}$ represent the batch size of new incoming data and the number of replayed data, $(\mathcal{X}^{(\mathcal{R}_s)}, \mathcal{Y}^{(\mathcal{R}_s)})$ is a set of samples drown from the source replay buffer.

In addition to general replay buffer, in this paper, we also consider the unlabeled data in each task. For the same purpose, we maintain a replay buffer $\mathcal{R}_t$ for the unlabeled samples from previous tasks. Meanwhile, we perform a adaptation between the labeled source data and the unlabeled target data, which will be elaborated in Section 3.3. After the adaptation, the classifier is expected to perform a considerable prediction for the target data of current task $\mathcal{T}$. In order to enhance the performance on replayed target samples, we further introduce the soft labels, which refers to the logits output by the classifier, denoted as $\mathcal{Y}_{soft}^{(\mathcal{T})} = C(F(\mathcal{X}_t^{(\mathcal{T})}))$. We store the soft labels alongside with the corresponding target samples $\mathcal{X}_t^{(\mathcal{T})}$ in $\mathcal{R}_t$. Later, we replay the target samples to perform a training as well as these from the source memory. As a result, we have the following loss:

$$\mathcal{L}_t^{(\mathcal{T})} = \frac{1}{N_{\mathcal{R}}} \ell_{mse}(C(F(\mathcal{X}^{(\mathcal{R}_t)}; \theta_F), \mathcal{Y}_{soft}^{(\mathcal{R}_t)}; \theta_C)), \tag{3}$$

where ℓ_{mse} denotes the mean squared error loss, $(\mathcal{X}^{(\mathcal{R}_t)}, \mathcal{Y}_{soft}^{(\mathcal{R}_t)})$ is a set of samples drown from the target replay buffer. For these difficult samples, the equal importance may cause a degradation of performance. To alleviate this issue, we use the entropy criterion $H(g) = -\sum_{c=1}^{C} g_c \log g_c$, where C is the number of classes and g_c is the probability of predicting an example to class c, to quantify the uncertainty of the prediction results of these samples, then impose an weighted importance on different target samples by applying $w(H(g)) = 1 + e^{-H(g)}$ to each sample. Finally, the overall objective function of our supervised learning is:

$$\min_{\theta_F, \theta_C} (\mathcal{L}_s^{(\mathcal{T})} + \mathcal{L}_t^{(\mathcal{T})}). \tag{4}$$

3.3. Adversarial learning with replayed samples

In adaptive conditional learning, our goal is to learn a model which can predict accurately on both labeled and unlabeled data, in order to deal with the unlabeled target data which has a different distribution from the labeled data, we leverage the adversarial learning, which shares the spirit of GANs and is widely used to reduce the discrepancy across domains. We employ a discriminator network D parameterized by θ_D to distinguish the target data from the source data. Meanwhile, a feature extractor F is trained to fool the discriminator. Through this adversarial learning, the discrepancy between the source domain and the target domain can be minimized. Different from general adversarial learning, the distributions of the source domain and the target domain are both agnostic due to the loss of samples from previous tasks. As an alternative, we estimate the up-to-date empirical distributions by the incoming data and the replayed data. Formally, we give the following objective of the above minimax game:

$$\min_F \max_D \mathcal{L}_{adv}^{(\mathcal{T})} = \mathbb{E}\left[\log D\left(F\left(X_s\right)\right)\right] + \mathbb{E}\left[\log\left(1 - D\left(F\left(X_t\right)\right)\right)\right], \tag{5}$$

where X_s and X_t represent the samples drown from the estimated distributions $(\mathcal{X}_s^{(\mathcal{T})} \cup \mathcal{X}^{(\mathcal{R}_s)})$ and $(\mathcal{X}_t^{(\mathcal{T})} \cup \mathcal{X}^{(\mathcal{R}_t)})$, respectively. By optimizing Eq. (5), the distribution discrepancy between two task sequences D_s and D_t are minimized. As a result, the feature extractor F tends to extract domain-invariant features, so that a classifier trained on the labeled source data can also perform well on the unlabeled target data.

3.4. Overall objective function

On one hand, our AOCML augments the general replay buffer with the unlabeled target data, then performs the supervision on the labeled data and the unlabeled data with soft labels simultaneously. This prevents the catastrophic forgetting of the previous tasks and improves the accuracy across the tasks. On the other hand, we conduct adversarial learning to align the distributions of the labeled source data and the unlabeled target data, which improves the accuracy within each task. The overall objective of the feature extractor F and the classifier C is given by

$$\min_{\theta_F, \theta_C} (\mathcal{L}_s^{(\mathcal{T})} + \mathcal{L}_t^{(\mathcal{T})} - \mathcal{L}_{adv}^{\mathcal{T}}). \tag{6}$$

Then, the discriminator D is trained to distinguish the source samples from the target samples, the objective is formulated as follows,

$$\min_{\theta_D} (\mathcal{L}_{adv}^{\mathcal{T}}). \tag{7}$$

The above two steps are conducted alternatively until convergence. The overall learning process of task $\mathcal{T}$ can refer to Algorithm 1.

4. Experiments

4.1. Datasets

MNIST Split is a multi-task learning dataset which splits MNIST [13] into 5 tasks with no-overlapping categories in each subset, i.e., 2 categories in each task. We use all samples for all tasks. The resolution are interpolated to size of 32 × 32 and the channel is expanded to 3 to be consistent with SVHN dataset.

SVHN Split splits street view house number (SVHN) [32] dataset into 5 disjoint tasks. SVHN consists of the original, variable-resolution, color house-number images obtained from house numbers in Google Street View. In our experiments, we have totally 69,590 images for training, 3666 for validation and 26,031 for testing.

USPS Split splits USPS handwritten digits dataset [33] into 5 disjoint tasks. USPS images were scanned from mail in a working post. The resolutions are interpolated to size of 32 × 32 and the channel is expanded to 3 to be consistent with SVHN dataset.

Office-31 Split splits the Office-31 dataset [34], where each image represents a common office object like calculator and monitor. Office-31 contains 31 classes with 4652 images from three domains: *Amazon*(**A**), *Webcam*(**W**) and *DSLR*(**D**). For each domain, we split 31

Table 1
Evaluation results of source (in first 2 rows) and target on Office-31 Split. Larger is better for average accuracy, lower is better for average forgetting. "Src", "Tgt" are short for source and target, respectively. Oracle means the experiments using the domain adversarial learning without continual learning setting, i.e., we can access all the source and the target data without suffering catastrophic forgetting.

		A→D		A→W		D→A		D→W		W→A		W→D	
		Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting
Src	Random replay	86.6 ± 0.5	8.4 ± 0.4	86.3 ± 1.3	9.2 ± 1.1	96.6 ± 1.0	1.9 ± 0.9	96.2 ± 1.0	2.7 ± 1.4	96.1 ± 0.4	2.5 ± 0.4	96.5 ± 0.5	2.2 ± 0.6
	AOCML (ours)	74.8 ± 1.4	11.3 ± 3.3	79.3 ± 1.2	9.6 ± 1.0	96.1 ± 0.9	2.3 ± 0.9	95.6 ± 1.3	3.2 ± 1.3	94.7 ± 0.6	2.9 ± 1.2	94.0 ± 0.5	4.0 ± 0.8
Tgt	Random replay	51.9 ± 1.9	20.8 ± 1.9	43.9 ± 1.7	26.5 ± 2.9	39.0 ± 1.6	22.2 ± 2.7	75.5 ± 1.1	14.4 ± 1.7	42.2 ± 1.5	19.5 ± 1.5	90.2 ± 0.9	5.4 ± 0.9
	AOCML (ours)	66.3 ± 2.8	11.3 ± 3.3	69.6 ± 1.6	9.2 ± 1.9	52.0 ± 1.3	12.7 ± 1.1	86.5 ± 2.1	8.2 ± 1.9	55.3 ± 0.9	8.7 ± 0.6	93.2 ± 1.2	3.5 ± 1.2
	Oracle	75.1 ± 0.2	–	77.9 ± 0.3	–	54.5 ± 0.3	–	96.9 ± 0.2	–	57.5 ± 0.4	–	100.0 ± 0.0	–

Table 2
Evaluation results on Digits-x Split. Larger is better for average accuracy, lower is better for average forgetting.

	Source						Target					
	S→M		U→M		M→U		S→M		U→M		M→U	
	Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting
Random replay	62.6 ± 4.1	16.8 ± 2.8	84.2 ± 4.0	6.5 ± 4.5	91.9 ± 3.6	2.7 ± 1.2	36.1 ± 3.5	30.2 ± 3.6	58.7 ± 3.8	12.3 ± 5.8	68.5 ± 5.4	12.0 ± 4.4
GEM [9]	37.7 ± 5.2	73.0 ± 6.3	71.8 ± 4.0	15.7 ± 3.0	92.4 ± 0.2	12.4 ± 1.2	11.1 ± 1.9	55.5 ± 6.3	25.6 ± 1.5	41.0 ± 1.0	23.3 ± 1.1	9.3 ± 3.2
DERPP [35]	63.8 ± 10.4	39.9 ± 13.1	88.4 ± 0.0	4.0 ± 0.7	98.2 ± 0.6	2.7 ± 1.2	9.9 ± 0.7	56.3 ± 6.5	26.6 ± 1.7	26.1 ± 0.7	39.3 ± 11.2	26.3 ± 7.4
RA+MIR [2]	61.6 ± 4.2	27.0 ± 2.3	77.3 ± 7.0	17.2 ± 5.3	82.2 ± 4.2	13.1 ± 3.6	47.4 ± 8.1	35.6 ± 6.4	48.9 ± 8.2	31.1 ± 2.5	56.1 ± 7.0	33.7 ± 6.1
CLCV [8]	63.1 ± 1.2	22.3 ± 2.1	79.5 ± 4.2	15.1 ± 3.6	85.7 ± 2.9	12.5 ± 4.1	52.6 ± 3.2	31.9 ± 4.7	55.6 ± 6.0	27.9 ± 3.4	59.9 ± 3.9	30.6 ± 5.3
AOCML (ours)	60.4 ± 0.4	33.1 ± 1.0	92.9 ± 0.7	4.2 ± 1.0	93.7 ± 1.3	3.6 ± 0.9	88.5 ± 2.2	6.2 ± 1.2	88.3 ± 3.4	2.6 ± 0.3	86.6 ± 1.6	6.8 ± 1.9

(a) MNIST (b) USPS (c) SVHN (d) Office-31

Fig. 3. Samples of the datasets we used.

classes into 5 tasks with 7, 7, 7, 7, 3 classes respectively. We use the full dataset for training and testing.

For a clearer understanding, we report several samples of each dataset in Fig. 3. Since the experimental settings for MINIST, SVHN and USPS are same, we use digits-x to refer the three.

4.2. Implementation details

The complete model of our AOCML consists of three parts: a feature extractor F, a classifier C and a domain discriminator D. For the implementation of the replaying buffer, the total number of slots in a replay buffer is $mem\_size \times nclasses$. If the buffer is full, it will randomly select positions and replace samples in these positions. Then, we randomly choose samples from previous tasks to replay. It is possible that a sample may stored or replayed multiple times.

For digits-x tasks, we use ResNet-18 which consists of four basic blocks as the feature extractor. The classifier is a single fully-connected layer. The discriminator network is a three-layer MLP with gradients reversal layer. All the networks are optimized by SGD with a fixed learning rate 0.1. We have 5 tasks with label sets {0, 1}, {2, 3}, {4, 5}, {6, 7} and {8, 9}, and the continual learning starts from task {0, 1} and end with task {8, 9}. The experiments are implemented with PyTorch-1.7 on the NVIDIA 2080Ti with CUDA 11.

For the Office-31 task, we use the pre-trained ResNet-50 which consists of four bottleneck blocks as the feature extractor. The classifier and the discriminator are the same as which in digits-x tasks. F, C and D are optimized by SGD with learning rate 0.01, 0.01, 0.1, respectively.

Due to the randomness of the experiments, each reported result in this paper is an average of 5 runs. The source memory size and the target memory size are both set to 50 by default. The batch size is set to 10 by default for the incoming samples and the replayed samples.

4.3. Metrics and baselines

Following previous work [2], we use two metrics in this paper. The first is average accuracy and the second is average forgetting. Specifically, both of them are calculated on all the tasks and then averaged. Let $a_{i,j}$ be the accuracy of the test task j after training the task i, the average accuracy after training the task t is $\frac{1}{t}\sum_{j=1}^{t} a_{t,j}$. Let $f_{i,j}$ be the forgetting on the test task j after training the task i, which is computed as:

$$f_{i,j} = \max_{l\in\{1,\ldots,i-1\}} a_{l,j} - a_{i,j}. \tag{8}$$

Then, the average forgetting at task t is given by

$$F_t = \frac{1}{t-1}\sum_{j=1}^{t-1} f_{t,j}. \tag{9}$$

For example, suppose the accuracy of task 1 is 0.9 when we training on task 1 and dropped to 0.6 when training task 2, then the forgetting would be 0.3. Note that lower accuracy does not always lead to larger forgetting.

For Digits-x datasets, we compare our method with state-of-the-art methods: GEM [9], DERPP [35] random replay and CLCV [8]. We also adopt MIR [2] with our adversarial learning as a baseline, reported as RA+MIR. For Office-31 datasets, oracle means the experiments using the domain adversarial learning without continual learning setting, that means we can access all the source and the target data without suffering catastrophic forgetting. Other results are the best results we can achieve by running experiments on mammoth [35], which is a general continual learning framework in PyTorch.

4.4. Results

The results on Office-31 Split are reported in Table 1, which are obtained with pre-trained ResNet-50. We can see that our method has made improvement than original random replay method on the target domain with an extra gain of 27%. Some of them are closed to the results of the oracle without continual learning setting. The average forgetting on the target domain is dropped significantly as well, up to 17%. Overall, the improvements are more obvious for harder adaptations such as W→A.

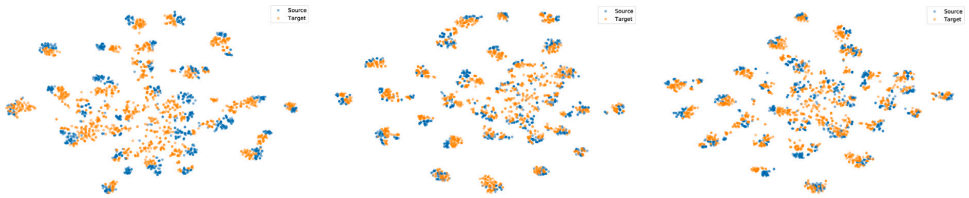
Table 2 reports the results on SVHN→MNIST Split, USPS→MNIST Split and MNIST→USPS Split experiments. Also, it is easy to observe the

Table 3
Comparison on S→M Split with different **target** memory size. Larger is better for average accuracy, lower is better for average forgetting.

		S→M								A→W					
		10		50		100		600		10		50		100	
		Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting
Src	Test	60.5 ± 3.0	33.0 ± 2.7	60.4 ± 0.4	33.1 ± 1.0	62.2 ± 2.1	30.1 ± 2.0	64.0 ± 3.1	27.6 ± 3.4	76.6 ± 0.9	10.0 ± 0.8	74.8 ± 1.4	11.1 ± 1.5	76.3 ± 0.9	9.9 ± 0.9
	Valid	61.6 ± 4.0	32.7 ± 3.4	59.2 ± 0.6	35.0 ± 1.1	62.6 ± 2.7	29.7 ± 2.3	63.8 ± 3.9	27.6 ± 4.3						
Tgt	Test	77.5 ± 4.6	16.5 ± 6.2	88.5 ± 2.2	6.2 ± 1.2	91.6 ± 2.0	4.3 ± 0.7	90.4 ± 2.1	2.9 ± 0.4	64.8 ± 1.0	13.5 ± 0.6	67.3 ± 1.2	12.1 ± 0.8	67.3 ± 2.0	12.6 ± 1.7
	Valid	75.8 ± 4.6	17.5 ± 6.2	86.3 ± 2.3	8.0 ± 1.2	90.7 ± 2.0	4.2 ± 0.6	89.0 ± 2.3	3.1 ± 0.4						

Table 4
Comparison on S→M Split with different **source** memory size. Larger is better for average accuracy, lower is better for average forgetting.

		S→M								A→W					
		10		50		100		600		10		50		100	
		Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting
Src	Test	41.6 ± 2.1	51.7 ± 2.3	60.4 ± 0.4	33.1 ± 1.0	67.3 ± 2.0	25.2 ± 3.4	81.5 ± 2.5	10.2 ± 3.5	73.1 ± 1.1	13.6 ± 0.6	74.8 ± 1.4	11.1 ± 1.5	76.4 ± 0.6	9.9 ± 0.5
	Valid	39.7 ± 1.7	53.3 ± 1.7	59.2 ± 0.6	35.0 ± 1.1	69.0 ± 1.9	23.8 ± 3.1	82.0 ± 2.2	10.3 ± 3.2						
Tgt	Test	87.9 ± 2.3	8.2 ± 1.7	88.5 ± 2.2	6.2 ± 1.2	87.9 ± 2.7	6.2 ± 1.1	90.1 ± 1.6	4.0 ± 1.3	67.3 ± 1.5	13.7 ± 1.8	67.3 ± 1.2	12.1 ± 0.8	66.8 ± 1.0	11.9 ± 1.4
	Valid	86.8 ± 2.2	8.6 ± 1.4	86.3 ± 2.3	8.0 ± 1.2	86.2 ± 3.0	6.8 ± 1.3	89.0 ± 1.4	4.3 ± 1.3						



(a) Random Replay (b) w/o SoftLabels (c) Ours

Fig. 4. t-SNE Visualization. We take D (blue)→W (orange) as an example.

Table 5
Ablation study on SVHN → MNIST Split and Amazon → Webcam Split. The w/ and w/o are short for with and without, respectively.

		S→M		A→W	
		Accuracy	Forgetting	Accuracy	Forgetting
Src	w/o Both	62.6 ± 4.1	16.8 ± 2.8	86.2 ± 1.3	9.5 ± 1.3
	w/o SoftLabels	60.8 ± 1.2	33.1 ± 1.8	77.6 ± 0.9	12.2 ± 0.6
	w/o Reweight	58.6 ± 3.1	35.9 ± 2.8	77.5 ± 0.6	11.9 ± 0.2
	w/ Both	60.4 ± 0.4	33.1 ± 1.0	78.8 ± 0.9	9.8 ± 0.8
Tgt	w/o Both	36.1 ± 3.5	30.2 ± 3.6	45.1 ± 1.9	25.2 ± 2.8
	w/o SoftLabels	75.6 ± 2.1	18.1 ± 1.0	65.5 ± 1.3	15.5 ± 0.4
	w/o Reweight	86.4 ± 1.2	9.6 ± 2.7	65.6 ± 1.9	12.8 ± 1.1
	w/ Both	88.5 ± 2.2	6.2 ± 1.2	69.1 ± 1.3	9.8 ± 1.3

improvement on the target for these datasets against previous methods, while keeping a small risk on the source data. The conclusion is consistent with the results on Office-31 dataset. Notably, for SVHN→MNIST Split, our method achieves 88.5% on average accuracy and 6.2% on average forgetting, which is quite impressive.

4.5. Model analysis

Effect of the target memory size. The target memory is an important component in our method. The memory size controls the number of samples stored in the target memory. Obviously, a larger target memory size provides a better approximation of the ground-truth data distribution, which would benefit the adaptation. We report the results of our method with different target memory sizes in Table 3 with the source memory size fixed to 50. The results show that our method can tolerate a very limited target memory size (like 10), because the performance is much close to best, compared with the source under the limited source memory size. Besides, the accuracy grows rapidly with a larger target memory size, which verifies the effect of the target memory size. However, when the target memory size is larger than 100, it cannot continually produce better results on S→M, and 50 on A→W. This indicates that our method do not require as much storage resources for the target data as the source data.

Effect of the source memory size. The source memory size controls the number of samples stored in source memory. Larger source memory size often leads to better performance on the source domain according to previous studies since it provides a better approximation on the source distribution. A small source memory size is enough to approximate the distribution needed by aligning with target distribution, a larger source memory size may not contribute effectively to the performance on the target domain. Intuitively, the first impression matters for training target. To verify the effect of source memory size, we report the results of our method with different source memory size in Table 4, with target memory size set to 50. It can be seen that a larger source memory size indeed improves the performance for the source domain, but has no obvious effect on the target domain.

Ablation Study. The proposed method contains two modules for adaptive continual learning on the target domain. To verify the effects of each module, we report the results of ablation study in Table 5. **w/o Both** is identical to experiments with random replay. **w/o SoftLabels** only uses adversarial learning without the soft labels. **w/o Re-weighting** uses both adversarial learning and original soft labels without re-weighting. **w/ Both** uses both modules, which is our default setting in aforementioned experiments. It can be seen that the adversarial learning plays an important role in the overall performance on the target, but also with some cost on the performance of the source domain. Other modules also contribute the improvement on the target domain. It is worth noting that entropy-based re-weighting is able to decrease the accuracy degradation on the source domain.

Learning process across tasks. We report the change of loss in Fig. 5 and the test accuracy across the training process in Table 6, the random replay is used for comparing. The results in Fig. 5 show that the loss reduces during the learning process in each task, the peaks represent the beginning of a new training task. The baseline accuracies of previous target tasks degrade but shows no meaningful difference on

Table 6
Accuracy (%) of test set during the training process. The numbers in columns stands for the training task, while the numbers in rows represent the test task. We take D→W on Office-31 as an example.

	Source															Target													
	AOCML (ours)					w/o SoftLabels					Random replay					AOCML (ours)					w/o SoftLabels					Random replay				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	100	99	99	100	99	100	99	97	99	98	100	100	100	99	98	98	98	97	96	95	98	97	95	93	94	99	96	90	88	73
2		96	95	93	97		99	98	95	95		95	97	95	98		98	90	94	93		96	94	97	92		88	90	77	80
3			98	92	96			99	97	95			98	95	92			96	92	85			96	88	83			91	74	75
4				97	91				97	92				99	91				87	69				89	66				79	58
5					100					93					97					91					97					88

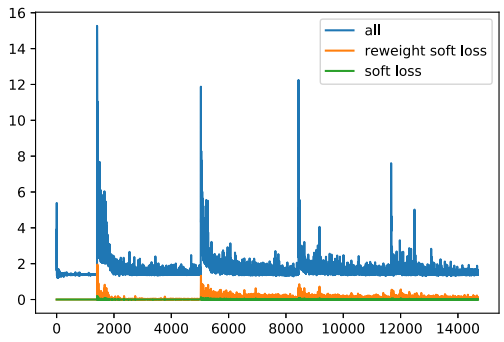


Fig. 5. Loss curve during learning process. Curve in blue, green and orange represent the total loss, the original soft label loss and the re-weighted soft label loss, respectively. We take S→M as an example.

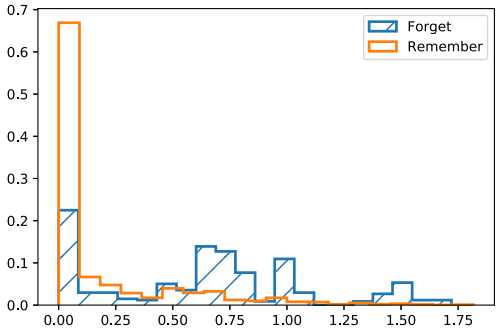


Fig. 6. Entropy distribution for forgotten (Blue) and remembered (Orange) sample when training task 5. We take S→M as an example.

source tasks. On the contrary, our method can achieve a better accuracy at the beginning with adversarial learning and reduce the forgetting by soft labels.

Visualization. In Fig. 4, we use t-SNE [36] to visualize the feature representations of task D→W on Office-31 dataset. t-SNE maps high dimension data into a low-dimensional space by learning the distance between data points. Here, we use the features outputted by the last convolutional layer of the feature extractor network. Each visually recognizable cluster stands for a category. The aligned result with adversarial learning in Fig. 4(b) is clearly better than the random replay in Fig. 4(a), as it has effects on reducing the distribution shift, which matches the analysis above. The result with soft labels in Fig. 4(c) shows no significant difference with Fig. 4(b). The results verify that soft labels contribute to reducing forgetting since they do not affect the distribution (the overall performance can be improved by aligning distributions or reducing forgetting).

Effect of the entropy-based re-weighting. In Fig. 6, we report the entropy of a set of samples during the training process. Samples that have correct predictions at first and then have wrong predictions after learning other tasks are flagged as forgotten, and these have consistent correct predictions are flagged as remember. The distribution of entropy for forgotten samples shows a clearly difference from those who are remembered, i.e., the former has larger entropy than the latter. It indicates that the predictions of forgotten samples is uncertainty. Therefore entropy is a safe metric to evaluate the forgotten samples.

5. Conclusion

In this paper, we investigate an adaptive continual multi-view learning setting where the data distribution would change from training to testing. To tackle this problem, we propose a novel method named adaptive online continual multi-view learning (AOCML) to address the issue of distribution discrepancy. AOCML leverages the idea of adversarial learning and replay memory, which not only aligns the distribution between different tasks, but also reduces the forgetting on aligned data. In addition, soft label-based replay and entropy-based re-weighting are employed to focus on important samples in memory and further reduce forgetting. Extensive experiments on various datasets verify the effectiveness of our proposed approach.

CRediT authorship contribution statement

Yang Yu: Methodology, Writing – reviewing & editing. **Zhekai Du:** Data curation, Validation. **Lichao Meng:** Conceptualization, Methodology, Writing –reviewing & editing. **Jingjing Li:** Supervision.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by Sichuan Science and Technology Program under Grant 2023NSFSC0483, in part by Medico-Engineering Cooperation Funds from University of Electronic Science and Technology of China (No. ZYGX2021YGCX016).

References

[1] 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. Natl. Acad. Sci. 114 (13) (2017) 3521–3526.

[2] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, L. Page-Caccia, Online continual learning with maximal interfered retrieval, in: Advances in Neural Information Processing Systems, 2019, pp. 11849–11860.

[3] R. Kemker, M. McClure, A. Abitino, T. Hayes, C. Kanan, Measuring catastrophic forgetting in neural networks, arXiv preprint arXiv:1708.02072.

[4] 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.

[5] J. von Oswald, C. Henning, J. Sacramento, B.F. Grewe, Continual learning with hypernetworks, arXiv preprint arXiv:1906.00695.

[6] M.F. Criado, F.E. Casado, R. Iglesias, C.V. Regueiro, S. Barro, Non-iid data and continual learning processes in federated learning: A long road ahead, Inf. Fusion 88 (2022) 263–280.

[7] Z. Le, J. Huang, H. Xu, F. Fan, Y. Ma, X. Mei, J. Ma, Uifgan: An unsupervised continual-learning generative adversarial network for unified image fusion, Inf. Fusion 88 (2022) 305–318.

[8] J. He, F. Zhu, Online continual learning via candidates voting, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 3154–3163.

[9] D. Lopez-Paz, M. Ranzato, Gradient episodic memory for continual learning, in: Advances in Neural Information Processing Systems, 2017, pp. 6467–6476.

[10] H. Shin, J.K. Lee, J. Kim, J. Kim, Continual learning with deep generative replay, in: Advances in Neural Information Processing Systems, 2017, pp. 2990–2999.

[11] A.A. Rusu, N.C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, R. Hadsell, Progressive neural networks, arXiv preprint arXiv:1606.04671.

[12] J. Yoon, E. Yang, J. Lee, S.J. Hwang, Lifelong learning with dynamically expandable networks, arXiv preprint arXiv:1708.01547.

[13] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[14] S. Ebrahimi, F. Meier, R. Calandra, T. Darrell, M. Rohrbach, Adversarial continual learning, arXiv preprint arXiv:2003.09553.

[15] C. Tang, X. Zheng, W. Zhang, X. Liu, Z. Xinzhong, Z. En, Unsupervised feature selection via multiple graph fusion and feature weight learning, Inform. Sci. 66 (152101) (2023) 1–152101.

[16] C. Tang, Z. Li, J. Wang, X. Liu, W. Zhang, E. Zhu, Unified one-step multi-view spectral clustering, IEEE Trans. Knowl. Data Eng..

[17] G.I. Parisi, R. Kemker, J.L. Part, C. Kanan, S. Wermter, Continual lifelong learning with neural networks: A review, Neural Netw. 113 (2019) 54–71.

[18] M. Farajtabar, N. Azizan, A. Mott, A. Li, Orthogonal gradient descent for continual learning, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 3762–3773.

[19] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1717–1724.

[20] A. Rios, L. Itti, Closed-loop memory gan for continual learning, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 2018, pp. 3332–3338.

[21] R. Kurle, B. Cseke, A. Klushyn, P. Van Der Smagt, S. Günnemann, Continual learning with bayesian neural networks for non-stationary data, in: International Conference on Learning Representations, 2019.

[22] S. Ruder, An overview of multi-task learning in deep neural networks, arXiv preprint arXiv:1706.05098.

[23] J. Li, E. Chen, Z. Ding, L. Zhu, K. Lu, H.T. Shen, Maximum density divergence for domain adaptation, IEEE Trans. Pattern Anal. Mach. Intell. (2021) 3918–3930.

[24] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, T. Darrell, Cycada: Cycle-consistent adversarial domain adaptation, in: International Conference on Machine Learning, PMLR, 2018, pp. 1989–1998.

[25] J. Li, Z. Du, L. Zhu, Z. Ding, K. Lu, H.T. Shen, Divergence-agnostic unsupervised domain adaptation by adversarial attacks, IEEE Trans. Pattern Anal. Mach. Intell. (2022) 8196–8211.

[26] J. Li, M. Jing, H. Su, K. Lu, L. Zhu, H.T. Shen, Faster domain adaptation networks, IEEE Trans. Knowl. Data Eng. (2022) 5770–5783.

[27] K. Saito, Y. Ushiku, T. Harada, K. Saenko, Strong-weak distribution alignment for adaptive object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6956–6965.

[28] A. Ma, J. Li, K. Lu, L. Zhu, H.T. Shen, Adversarial entropy optimization for unsupervised domain adaptation, IEEE Trans. Neural Netw. Learn. Syst. (2022) 6263–6274.

[29] J. Li, Y. Wu, J. Zhao, K. Lu, Low-rank discriminant embedding for multiview learning, IEEE Trans. Cybern. 47 (11) (2016) 3516–3529.

[30] 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.

[31] A. Chaudhry, M. Ranzato, M. Rohrbach, M. Elhoseiny, Efficient lifelong learning with a-gem, arXiv preprint arXiv:1812.00420.

[32] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A.Y. Ng, Reading digits in natural images with unsupervised feature learning.

[33] J.J. Hull, A database for handwritten text recognition research, IEEE Trans. Pattern Anal. Mach. Intell. 16 (5) (1994) 550–554, http://dx.doi.org/10.1109/34.291440.

[34] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in: Proceedings of the 11th European Conference on Computer Vision: Part IV, ECCV'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 213–226.

[35] P. Buzzega, M. Boschini, A. Porrello, D. Abati, S. Calderara, Dark experience for general continual learning: a strong, simple baseline, in: Advances in Neural Information Processing Systems (NIPS), 2020.

[36] L. van der Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (86) (2008) 2579–2605, URL http://jmlr.org/papers/v9/vandermaaten08a.html.