Information Sciences 676 (2024) 120813

Contents lists available at ScienceDirect

Information Sciences

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

Cross-domain continual learning via CLAMP

Weiwei Weng [a,*], Mahardhika Pratama [b,*], Jie Zhang [a], Chen Chen [a], Edward Yapp Kien Yie [c], Ramasamy Savitha [d]

[a] SCSE, Nanyang Technological University, Nanyang Avenue, Singapore, Singapore
[b] STEM, University of South Australia, Mawson Lakes Boulevard, Adelaide, Australia
[c] SIMTech, A*Star, Fusionopolis Way, Singapore, Singapore
[d] I2R, A*Star, Fusionopolis Way, Singapore, Singapore

ARTICLE INFO

Dataset link: https://github.com/wengweng001/CLAMP_torch.git

Keywords:
Continual learning
Domain adaptation
Lifelong learning
Incremental learning
Cross domain adaptation
Unsupervised domain adaptation

ABSTRACT

Artificial neural networks, celebrated for their human-like cognitive learning abilities, often encounter the well-known catastrophic forgetting (CF) problem, where the neural networks lose the proficiency in previously acquired knowledge. Despite numerous efforts to mitigate CF, it remains the significant challenge particularly in complex changing environments. This challenge is even more pronounced in cross-domain adaptation following the continual learning (CL) setting, which is a more challenging and realistic scenario that is under-explored. To this end, this article proposes a cross-domain CL approach making possible to deploy a single model in such environments without additional labelling costs. Our approach, namely continual learning approach for many processes (CLAMP), integrates a class-aware adversarial domain adaptation strategy to align a source domain and a target domain. An assessor-guided learning process is put forward to navigate the learning process of a base model assigning a set of weights to every sample controlling the influence of every sample and the interactions of each loss function in such a way to balance the stability and plasticity dilemma thus preventing the CF problem. The first assessor focuses on the negative transfer problem rejecting irrelevant samples of the source domain while the second assessor prevents noisy pseudo labels of the target domain. Both assessors are trained in the meta-learning approach using random transformation techniques and similar samples of the source domain. Theoretical analysis and extensive numerical validations demonstrate that CLAMP significantly outperforms established baseline algorithms across all experiments by at least 10% margin.

1. Introduction

The underlying goal of continual learning (CL) is to design a learning algorithm handing never-ending environments efficiently. Unlike conventional learning algorithms dealing with a single task, the CL model is supposed to cope with a sequence of different tasks $T_1, T_2, T_3, ..., T_K$ where K denotes the number of tasks which might be unknown prior to process runs. Note that the continual learning problem distinguishes itself from the multi-task learning problem because the learning tasks are received sequentially and not available at once. Each task possesses non-stationary characteristics, i.e., different data distributions, different class labels, or

* Corresponding authors.
E-mail addresses: weiwei002@e.ntu.edu.sg (W. Weng), dhika.pratama@unisa.edu.au (M. Pratama).

https://doi.org/10.1016/j.ins.2024.120813
Received 24 January 2024; Received in revised form 27 May 2024; Accepted 28 May 2024

Available online 31 May 2024

combinations between the two types. The key challenge of continual learning lies in the fast adaptation to new environments while preventing the catastrophic forgetting (CF) problem with the absence of data samples of old tasks T_{k-1}, i.e., previously relevant parameters are overwritten when learning new tasks.

Numerous works have been devoted in the past few years where the main focus is to combat the issue of CF paving a way for knowledge accumulation across streaming tasks. The regularization-based approaches Kirpatrick et al. [1] addresses the catastrophic forgetting problem where the regularization term protects important parameters of old tasks from changing. Although this approach is computationally efficient and easy to implement, it does not scale well for large-scale problems. Mao et al. [2] aims to address this problem by considering mutual and shareable information across tasks. Nonetheless, the regularization-based approach is over-dependent on the task IDs for inference process. The structure-based approach adds new network components to embrace new tasks while isolating previous network parameters Rusu et al. [3]. Some approaches propose network evolution criteria for growing/pruning whereas other works put forward search-based approaches to find near-optimal network structures to accommodate new tasks Pratama et al. [4]. The first approach is fast to compute but does not assure optimal solutions whereas the second approach is computationally prohibitive. As with the regularization-based approach, the structure-based approach relies on the task IDs imposing extra domain knowledge. The memory-based approach Rebuffi et al. [5] is applicable without the task IDs where the experience replay mechanism is carried out using old samples stored in the memory. However, it requires hundreds of samples per class to be stored in the memory. This mechanism is tackled in Shin et al. [6] using the pseudo-rehearsal approach via generative models such as generative adversarial network (GAN) or variational auto-encoder (VAE). The pseudo-rehearsal mechanism is inferior compared to the conventional approaches because synthetic samples of generative models have lower qualities than original samples.

The continual learning problem still requires in-depth study because the vast majority of existing works are limited to only a single domain and the cross-domain CL problem remains a relatively uncharted territory. This problem differs from multistream classification problems in Chandra et al. [7] because they still consider a single task. To the best of our knowledge, only Lao et al. [8] has explored the cross-domain continual learning problem. Our approach differs from this work where a meta-learning strategy is proposed to induce two assessors guiding the continual learning processes. In addition, the class-aware adversarial domain adaptation is put forward for domain alignments to address the domain shift problems.

Continual learning approach for many processes (CLAMP) is proposed as a solution of the cross-domain continual learning problems where there exist a fully labelled source domain and an unlabelled target domain. The underlying goal is to craft a predictive model of the target domain without any labels using discriminative information of the source domain. The source domain and the target domain are different but related where they share the same labelling function. The key difference with the multistream classification problem in Chandra et al. [7] lies in the cross-domain nature handling not only a single time-varying task but also a sequence of different tasks across different domains. Our problem also differs from Lin et al. [9] because the source domain is incremental in nature and does not carry complete class information. This problem extends the conventional continual learning problem where a continual learner not only handles a non-stationary process without the CF problem but also addresses a domain shift between source domain and target domain.

CLAMP is designed to cope with the cross-domain CL problems where a class-aware adversarial domain adaptation approach Ganin et al. and Li et al. [10,11] is integrated to produce a domain-invariant network. The domain alignment strategy makes use of a min-max game between a feature extractor and a domain classifier. The feature extractor is trained in an adversarial manner to generate indistinguishable features of source domain and target domain while the domain classifier predicts sample's origins. Our approach goes one step ahead of the vanilla adversarial domain adaptation performing the domain alignment only and ignoring the class alignment of the target domain and the source domain. The class alignment is achieved by performing a self-training of the target domain where a pseudo-labelling process is committed to highly confident samples of the target domain. As with Li et al. and Zheng et al. [11,12], the meta-learning-based regularization approach is applied to combat the noisy pseudo label problem. The key difference of our approach lies in dual meta-training loops also weighting every sample of the source domain to overcome the negative transfer issue in addition to the target domain to address the noisy pseudo-label problem. Our approach also relies on sequence-aware assessors producing a set of weights for every sample rather than a single weight. This approach not only controls the sample's influences addressing the issues of negative transfers and noisy pseudo labels but also the interactions of the three loss functions to achieve a proper tradeoff between the stability and the plasticity, a key issue of continual learning.

This paper at least conveys five contributions:

- we propose a highly challenging but rarely discussed problem, namely the cross-domain continual learning problem. This problem features two different but related continual learning problems drawn from two different domains while sharing the same label space such that the CF problem and the domain shift problem must be tackled simultaneously. Source domain is fully labelled but the target domain suffers from the absence of any labels. The goal is to craft a continual learning model that performs well for both domains benefiting from label information of the source domain to the target domain. Such problem differs from Chandra et al. [7] because both source and target domains characterize the continual learning problem and Lin et al. [9] because the source domain also features the continual learning process. Our problem is similar to those of De Carvalho [13] and Lao et al. [8] but we offer different approaches here.
- CLAMP is proposed to resolve the cross-domain continual learning problem. This approach is inspired by the concept of meta weighted adversarial domain adaptation Li et al. [11] where we expand this approach for the continual learning problem involving a multi-objective loss function. Our approach not only includes a meta-training loop in the target domain to reject noisy pseudo label but also in the source domain to avoid the issue of negative transfer. That is, a source-domain sample can be

down-weighted if it poses a risk of negative transfer. Pseudo labels can be rejected by assigning low weights to minimize their effects;
- The concept of assessor-guided learning is put forward to improve the learning process where dual assessors are put forward to address the issue of negative transfer and noisy pseudo label as well as to attain proper trade off between the plasticity and the stability. This strategy innovates Zheng et al. and Masum et al. [12,14] where the assessor is not yet applied in such configurations and limited to a static setting. Note that Masum et al. [14] applies the concept of assessor-guided learning to the ordinary continual learning problems while Zheng et al. [12] develops such concept only to a single-task metric learning problem;
- The theoretical analysis analyzing the generalization bound of the target domain is provided;
- the source codes of CLAMP are made publicly available in https://github.com/wengweng001/CLAMP_torch.git for reproducibility and convenient further study.

The remainder of this paper is structured as follows: Section 2 discusses the related works; Section 3 outlines the problem formulation; Section 4 explains the methodology of our paper; Section 5 is devoted to the theoretical analysis; Section 6 describes the experiments, discussions and limitations; Some concluding remarks are drawn in Section 7.

2. Related works

This section discusses the related works which comprise three sub-sections: continual learning, meta-learning and multi-stream mining.

2.1. Continual learning

Regularization-based approach [1,2] addresses the catastrophic forgetting problem in the continual learning domain with the application of additional regularization term protecting important parameters of old tasks from changing. Nevertheless, the regularization-based approach requires the task IDs and the task boundaries to be known hindering its feasibility in practise. **Structure-based approach** [3,4,15,16] expands network structures to deal with new tasks leaving old network parameters unchanged. As with regularization-based approach, this approach is sensitive to the task IDs. **Memory-based Approach** [17,18] takes different strategies where it stores a subset of old samples in the episodic memory. These samples are interleaved with new samples when learning a new task to reduce the CF problem. Wang et al. [19] recently offers a rehearsal-free approach using the concept of prompts. That is, learnable prompts are inserted to every task while fixing the backbone ViT network. These works have been successful without the use of any memories in the challenging class-incremental learning problems. All these works still consider a single domain CL problem where we aim to extend here into cross-domain CL problems where the goal is to generalize over an unlabelled target domain from a fully labelled source domain. To the best of our knowledge, this problem is only addressed in Lao et al. [8] using the pseudo-rehearsal mechanism to address the catastrophic forgetting problem. Our approach differs from this work where the assessor-guided learning strategy and the class-aware adversarial domain adaptation approach are put forward. This problem also goes one step ahead of the multistream mining topic Chandra et al. [7] yet considering a single non-stationary task and of the class-incremental unsupervised domain adaptation Lin et al. [9] possessing all classes in the source domain. Recent work by De Carvalho et al. [13] addresses the same problem, the cross-domain continual learning problem. However, our approach is significantly different from this work where the assessor-guided learning technique containing the dual meta-training loop is proposed. Our numerical study also finds that our approach, CLAMP, outperforms this work.

2.2. Meta-learning

Meta-learning Finn et al. and Schmidhuber et al. [20,21] constitutes a learning-to-learn technique where the goal is to improve the learning performance of another learning algorithm. Such approach has been explored for CL [22,23,18] where Javed et al. [22] introduces two networks, representation network and prediction network, trained in the meta-learning manner and Gupta et al. [23] extends the concept of model agnostic meta-learning (MAML) to continual learning problems and puts forward the concept of learnable per variable learning rates while Pham et al. [24] and Dam et al. [18] adopt this concept for online task-incremental learning problems. We go one step ahead of these works where the meta-learning strategy is applied for cross-domain continual learning problem. That is, it aims to generalize well for a sequence of unlabelled target learning tasks provided with a sequence of labelled source learning tasks. The notion of assessor-guided learning is put forward in Zheng et al. [12] for a single-task metric learning. We innovate over this concept where it is applied in the continual multi-task environments. In addition, it not only performs a soft sample selection mechanism but also undertakes selections of learning strategies, i.e., which losses to be favoured in the multi-objective optimization problem. In Masum et al. [14], the notion of assessor-guided learning is adopted to cope with the continual learning problems. Our work distinguishes itself where such idea is extended for the cross-domain continual learning problems. Such problem requires the catastrophic forgetting problem and the domain shift problem to be overcome simultaneously.

2.3. Multistream mining

Multistream mining is seen as an extension of the unsupervised domain adaptation problem where each domain features a streaming process rather than a fixed process Chandra et al. [25]. As with the unsupervised domain adaptation problem, it aims to

develop a stream-invariant model to handle an unlabelled target stream by transferring relevant knowledge of a source stream in which true class labels are available. The underlying challenge lies in the presence of asynchronous drift because the source stream and the target stream independently run. A pioneering study in this area is proposed in Chandra et al. [25] where it integrates the kernel mean matching (KMM) as a domain adaptation method and a drift detection approach to detect concept drifts in each stream. FUSION is proposed in Haque et al. [26] using KLIEP as the domain adaptation method and a density ratio to signal the asynchronous drifts to reduce computational burden of MSC. A deep learning solution is offered in Pratama et al. [27], where the multistream classification problem is handled using the encoder-decoder structure having shared parameters as well as the KL divergence approach.

The multistream classification problem is expanded to consider the presence of multi-source stream in Du et al. [28]. This work is extended in Du et al. [29]. Recently, the same problem is considered in Xie et al. [30] where the CMD-based regularization approach is proposed. The aforementioned works rely on a single domain assumption where the source stream and the target stream are drawn from the same feature space but different distributions. The problem of cross-domain multistream classification is attacked in Carvalho et al. [31] where each stream features a unique feature space but shares the same target attributes. The adversarial domain adaptation technique is utilized in Carvalho et al. [31]. Weng et al. [32] considers the problem of label scarcity where no labels are provided in the source stream and target stream for model updates. This paper distinguishes itself from these works because the source domain and the target domain present a continual learning problem, i.e., a model is exposed with a sequence of different tasks in both the source domain and the target domain. In other words, we go beyond a single non-stationary task where a model not only handles the domain shift problem and the asynchronous drift problem but also addresses the task shift problem and the catastrophic forgetting problem. To the best of our knowledge, this problem is only addressed in Lao et al. [8] using the pseudo-rehearsal mechanism to address the catastrophic forgetting problem. Our approach differs from this work where the assessor-guided learning strategy and the class-aware adversarial domain adaptation approach are put forward.

3. Problem formulation

The cross-domain continual learning problem is defined here as a learning problem to handle two different but related continual learning problems: $\mathcal{T}_1^S, \mathcal{T}_2^S, ..., \mathcal{T}_K^S$ a source domain and $\mathcal{T}_1^T, \mathcal{T}_2^T, ..., \mathcal{T}_K^T$ a target domain $k \in \{1, ..., K\}$ where K stands for the number of tasks. Labelled samples are only provided to the source domain $\mathcal{T}_k^S = \{(x_i^S, y_i^S)\}_{i=1}^{N_k^S}$ leaving the target domain completely unlabelled $\mathcal{T}_k^T = \{(x_i^T)\}_{i=1}^{N_k^T}$. The source domain and the target domain are different but related because they possess different feature space $x_k^S \in \mathcal{X}_k^S, x_k^T \in \mathcal{X}_k^T, \mathcal{X}_k^S \neq \mathcal{X}_k^T$ but share the same target variables $y_k^S, y_k^T \in \mathcal{Y}_k, y_i = [l_1, l_2, ..., l_m]$ where $\mathcal{X}_k^S \times \mathcal{Y}_k \in \mathcal{D}_k^S$ and $\mathcal{X}_k^T \times \mathcal{Y}_k \in \mathcal{D}_k^T$. m is the number of target classes. The source domain and the target domain run independently and follow different distributions $P(x_S) \neq P(x_T)$ such that $\mathcal{D}_k^S \neq \mathcal{D}_k^T$. N_k^S, N_k^T denote the size of the $k-th$ task of the source domain and the target domain respectively where $N_k^S \neq N_k^T$.

Every task of the source domain and the target domain features non-stationary characteristics: domain-incremental learning, task-incremental learning and class-incremental learning. The domain-incremental learning problem presents changing data distributions or concept drifts of each task $P(x,y)_k^S \neq P(x,y)_{k+1}^S, P(x,y)_k^T \neq P(x,y)_{k+1}^T$. The task-incremental learning problem and the class-incremental learning problem feature different label sets of each task. Suppose that $L_k^{S,T}$ denotes a label set of the $k-th$ task of the source domain and the target domain, i.e., both domains carry the same label set, $L_k^{S,T} \cap L_{k'}^{S,T} = \emptyset, k, k' \in \{1, ..., K\}$, i.e., every task carries unique label sets. The key difference of the task-incremental learning problem and the class-incremental learning problem lies in the presence of a task IDs for the task-incremental learning problem making possible to deploy different classifiers with a shared feature extractor. Since the class-incremental learning problem is more challenging than the task-incremental learning problem, we only focus on the class-incremental learning here. A model is only offered by data samples of the current task $(x^S, y^S)_k \backsim \mathcal{D}_k^S$ and $(x^T)_k \backsim \mathcal{D}_k^T$ whereas old samples are discarded for the sake of memory and computational efficiencies. This strategy leads to the catastrophic forgetting problem. This problem is visualized in Fig. 1.

The goal of this problem is to develop a predictive model $g_\phi(f_\theta(.))$ where $g_\phi(.)$ is a classifier parameterized by ϕ and $f_\theta(.)$ is a feature extractor parameterized by θ to perform well on all tasks of the unlabelled target domain by transferring knowledge of the labelled source domain. This is achieved by learning the source domain and the difference of the source domain and target domain $\frac{1}{K}\sum_{k=1}^{K} \mathcal{L}_k, \mathcal{L}_k \triangleq \mathbb{E}_{(x,y)_k \backsim \mathcal{D}_k^S}[l(g_\phi(f_\theta(x)), y)] + d(\mathcal{D}_k^S, \mathcal{D}_k^T)$.

4. Learning procedure of CLAMP

CLAMP relies on the class-aware adversarial training strategy [10,11] comprising the feature extractor $f_\theta(.)$, the classifier $g_\phi(.)$ and the domain classifier $\xi_\psi(.)$. The feature extractor $f_\theta(.)$ is trained in an adversarial manner to fool the domain classifier $\xi_\psi(.)$ classifying whether a data sample belongs from the target domain or the source domain. A domain-invariant network is produced if the feature extractor generates indistinguishable samples. In other words, the feature extractor and the domain classifier play a minimax game with a gradient reversal strategy to update the feature extractor converting the minimization problem into the maximization problem. The classifier $g_\phi(.)$ delivers the final prediction of the network. The continual learning strategy across the source domain and the target domain is formulated as a meta-weighted combination of the three loss functions, the cross entropy loss function, the dark experience replay loss function Buzzega et al. [33] and the knowledge distillation loss function Rebuffi et

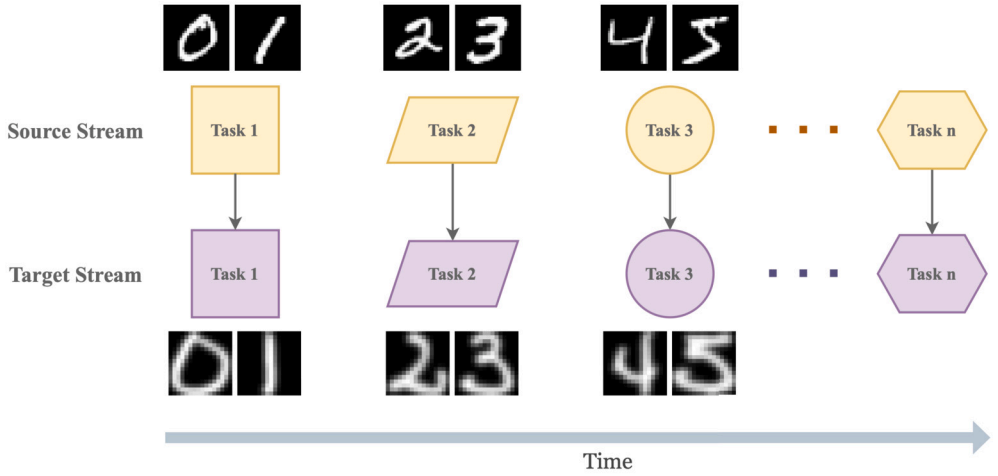


Fig. 1. Cross-Domain Continual Learning Problems.

al. [5]. That is, two assessors trained in the meta-learning manner are deployed to guide the learning process of the base learner, i.e., they control the sample influences making sure only positive samples to obtain high weights as well as the interaction of the three loss functions determining the best combination in learning a sample - proper trade off between the plasticity and the stability. This approach makes use of an episodic memory $\mathcal{M}$ based on reservoir sampling where two episodic memories, namely the source memory $\mathcal{M}_S$ and the target memory $\mathcal{M}_T$, are used here. Fig. 2 shows the network structure of CLAMP.

4.1. Training strategy of source domain

Referring to Ben David et al. [34], the generalization power of the target domain is upper bounded by the empirical error of the source domain. This requires a model to be trained in the source domain which can be done simply by minimizing the cross entropy loss function because of the presence of true class labels. Nonetheless, this process risks on the negative transfer problem. Our approach is inspired by the assessor-guided training process Zheng et al. and Masum et al. [12,14] performing the soft-weighting mechanism for every sample of the source domain. We adapt this approach to the cross-domain continual learning problem where the assessor generates a set of weights corresponding to the three loss functions, the cross entropy loss function, the DER loss function Buzzega et al. [33] and the distillation loss function Rebuffi et al. [5] thereby leading to a suitable weighted linear combination in learning a data sample. That is, a sequence-aware assessor produces a set of weights $\{\alpha_S, \beta_S, \gamma_S\} = \kappa^S_{\zeta_S}(.) \in [0,1]$ where ζ_S is the parameters of the assessor. $\alpha_S, \beta_S, \gamma_S$ respectively stand for the cross entropy weight, the DER weight and the distillation weight. A high weight, close to one, is provided to good samples whereas a low weight is produced to poor samples minimizing its effects. The loss function of the source domain is set as follows:

$$\mathcal{L}_S = \underbrace{\mathbb{E}_{(x,y)\backsim \mathcal{D}^S_k \cup \mathcal{M}^S_{k-1}}[\alpha_S l(s_W(o), y)]}_{L_{CE}} + \underbrace{\mathbb{E}_{(x,y)\backsim \mathcal{M}^S_{k-1}} \beta_S[||o-h|| + l(s_W(o), y)]}_{L_{DER}} + \underbrace{\mathbb{E}_{(x,y)\backsim \mathcal{M}^S_{k-1}}[\gamma_S l(o,h)]}_{L_{distill}} \tag{1}$$

where $o = g_\phi(f_\theta(x))_k, h = g_\phi(f_\theta(x))_{k-1}$ denote the current output logit, i.e. the pre-softmax response, and the previous output logit before learning the current task. $l(.)$ stands for the cross entropy loss function while $s_W(.)$ labels the softmax layer parameterized by W. The assessor $\kappa^S_{\zeta_S}(.)$ is implemented as an LSTM with a convolutional feature extractor taking into account a sequence of past weights in generating a current weight. A fully connected layer with a sigmoid activation function is inserted at a last layer assuring a bounded output in between 0 and 1. It is seen that the soft-weighting mechanism is carried out here rather than the binary hard-weighting approach to induce a smooth learning process. The cross entropy loss function focuses on the current and past concepts, the plasticity to new knowledge, whereas the DER loss function and the distillation loss function mitigates the catastrophic forgetting effect, the stability of old concepts. The meta-weighting approach to the three loss functions is applied here to achieve proper tradeoff between the plasticity and the stability.

The cross entropy (CE) loss function learns both current samples of the current task and past samples of the past tasks and is weighted by a meta-weight generated by the assessor. That is, it learns the current concept while retaining the previously learned knowledge to prevent the CF problem. The use of CE loss function alone does not suffice to avoid the CF problem because the number of old tasks is much more than the current task and restricted to a tiny memory storing past samples. To this end, the DER loss function by Buzzega et al. [33] is integrated and considers both the soft label and the hard label. The hard label is incorporated to prevent the output bias problem. As with the CE loss function, the DER loss function is meta-weighted by the assessor controlling its contributions to the training process. Since the DER loss function utilizes the L2 distance function to compare the current and previous output logits, the distillation loss function Rebuffi et al. [5] is integrated to further combat the CF problem and uses the cross entropy loss function to match the current and previous output logits. The influence of the distillation loss function is governed

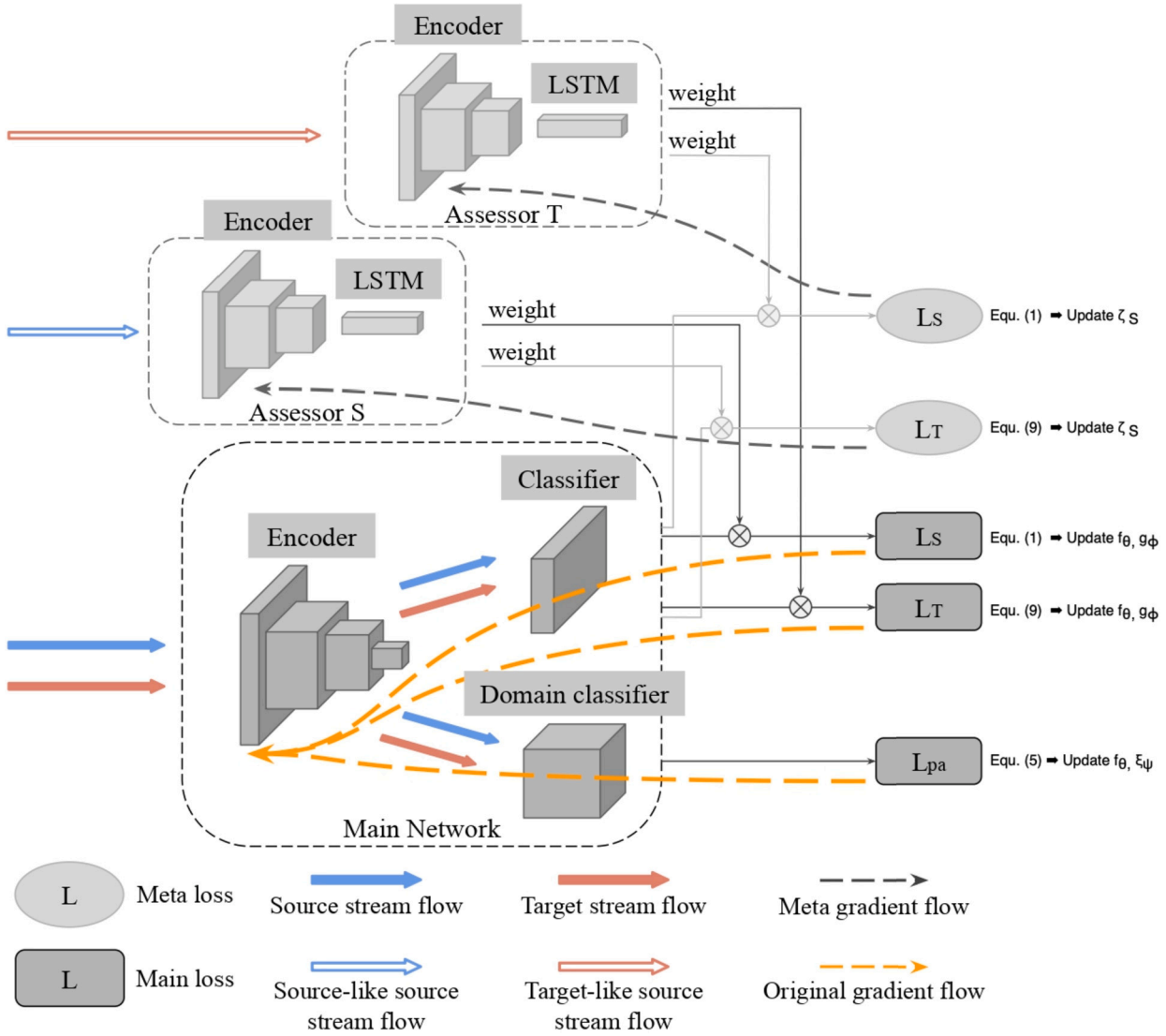


Fig. 2. CLAMP architecture includes the main module (encoder f_θ, classifier g_ϕ and domain classifier ξ_ψ) and two assessors (ζ_S and ζ_T). In cross-domain continual learning scenario, the adversarial learning process of the base learner, encoder and classifier is adjusted by two assessors trained in the meta-learning method. Domain classifier of main network aims to solve unsupervised domain adaptation.

by the assessor. That is, the assessor controls how much a model should learn from the current and past concepts or balances the stability and the plasticity, i.e., the CE loss function learns both the current and past concepts while the DER loss function and the distillation loss function focus on the past concept in the memory.

The meta-learning approach is adopted to train both the base learner and the assessor. Two data partitions, namely a training set $\mathcal{T}^k_{train}$ and a validation set $\mathcal{T}^k_{val}$, are created for this purpose. The training set is nothing but data samples of the current task and data samples of the episodic memory $\mathcal{T}^k_{train} = \mathcal{T}^S_k \cup \mathcal{M}^S_{k-1}$ whereas the validation set applies the random transformation approach to the training set Masum et al. [14]. Specifically, three random transformations, namely imageinvert, Gaussian noise perturbation, RGB-rand perturbation, are applied to the original images of the training set $\mathcal{T}^k_{val} = \{T(x_i), y_i\}^{N_{train}}_{i=1} \backsim \mathcal{T}^k_{train}$ where $T(.)$ is an image transformation operation. The training process of the base learner and the assessor is formulated as a bi-level optimization problem where the base learner is updated using the training set in the outer loop while the assessor utilizes the validation set in the inner loop.

$$\begin{aligned} \{\theta^*, \phi^*\} &= \arg\min_{\theta,\phi} \mathbb{E}_{(x,y)\backsim\mathcal{T}^k_{train}} \mathcal{L}_S \\ s.t \quad &\min_{\zeta_S} \mathbb{E}_{(x,y)\backsim\mathcal{T}^k_{val}} \mathcal{L}_S \end{aligned} \tag{2}$$

where $\{\theta^*, \phi^*\}$ are optimal parameters of the base learner in respect to the current assessor ζ_S. Because of the absence of ground truth, the meta-training strategy is carried out where the assessor is fine-tuned to minimize the validation loss of the base learner.

That is, the base learner is evaluated using the validation set $\mathcal{T}_{val}^{k}$ returning the validation loss. This implies the use of a meta-objective in adapting the base learner. The validation loss is differentiable in respect to the assessor because it is induced by the cross entropy weight α, the DER weight β and the distillation weight γ. The assessor is updated first here because it determines the optimal base network:

$$\zeta_S' = \zeta_S - \beta \sum_{(x,y)\in\mathcal{T}_{val}^{k}} \nabla_{\zeta_S} \mathcal{L}_S \tag{3}$$

where β is the learning rate of the inner loop. The next step is to train the base network making use of the three meta-weights of the assessor as follows:

$$\{\phi,\theta\} = \{\phi,\theta\} - \mu \sum_{(x,y)\in\mathcal{T}_{train}^{k}} \nabla_{\{\phi,\theta\}} \mathcal{L}_S \tag{4}$$

where μ is the learning rate of the outer loop. This strategy implies a joint update of the base network and the assessor where they work in the collaborative manner, i.e., the adjustment of the base learner involves the tuning process of the assessor. It also simulates the training and testing phases of the base learner where the training process is done using the training set with the help of the assessor. Its generalization power is evaluated using the validation set where the validation loss controls the training process of the assessor. It leads to an improved performance of the base learner because the base learner is adjusted using the three meta-weights generated by an updated assessor $\{\alpha_S,\beta_S,\gamma_S\} = \kappa_{\zeta_S'}^{S}(x)$.

4.2. Unsupervised domain adaptation strategy

A domain adaptation step is required here to achieve a domain-invariant network because the source domain and the target domain are different but related. That is, they follow different distributions $P(x_s) \neq P(x_t)$ and are drawn from distinct feature spaces $\mathcal{X}_S \neq \mathcal{X}_T$ but are related because of the same label spaces. This process has to be carried out in an unsupervised way due to the absence of labelled samples of the target domain. Our approach makes use of the class-aware adversarial domain adaptation strategy Ganin et al. and Li et al. [10,11] deploying the domain classifier $\hat{d}_i = \xi_\psi(f_\theta(x_i))$ formed as a single hidden layer network to classify the origin of features generated by the feature extractor $f_\theta(.)$, i.e., the domain classifier is tasked to perform a binary classification problem whether a data sample belongs to the source domain or the target domain. The domain adaptation loss is formulated as the domain classifier loss:

$$\begin{aligned}\mathcal{L}_{pa} &= \mathbb{E}_{(x,d_n)\backsim D_k^S \cup \mathcal{M}_{k-1}^S}[\log(\xi_\psi(f_\theta(x)))] + \mathbb{E}_{(x,d_n)\backsim D_k^T \cup \mathcal{M}_{k-1}^T}[\log(1-\xi_\psi(f_\theta(x)))] \\ &= \frac{1}{N_s}\sum_{i=1}^{N_S} \mathcal{L}_d(d_i,\hat{d}_i) + \frac{1}{N_T}\sum_{i=1}^{N_t} \mathcal{L}_d(d_i,\hat{d}_i) \\ \mathcal{L}_d(d_i,\hat{d}_i) &= -[d_i \log(\hat{d}_i) + (1-d_i)\log(1-\hat{d}_i)]\end{aligned} \tag{5}$$

where $d_n \in \{0,1\}$ is the origin of data samples, i.e., 1 for the source domain and 0 for the target domain. In a nutshell, the loss function $\mathcal{L}$ is optimized by searching for a saddle point solution ϕ,ψ,θ of the minimax problem.

$$\begin{aligned}(\hat{\phi},\hat{\theta}) &= \arg\min_{\phi,\theta} \mathcal{L}(\phi,\theta,\hat{\psi}) \\ \hat{\psi} &= \arg\max_{\psi} \mathcal{L}(\hat{\phi},\hat{\theta},\psi)\end{aligned} \tag{6}$$

The gradient reversal layer is applied when adjusting the feature extractor thus converting the minimization problem into the maximization problem. This strategy minimizes the divergence of the source domain and the target domain because the feature extractor is trained to produce indistinguishable feature representations, i.e., maximizing similarities across the two domain thereby fooling the domain classifier. The gradient reversal layer has no parameters and simply reverses the update directions, i.e., multiplying the gradient with -1. We apply the stochastic gradient descent method for model updates.

$$\begin{aligned}\theta &= \theta + \lambda \frac{\partial \mathcal{L}_d^i}{\partial \theta} \\ \phi &= \phi - \lambda \frac{\partial \mathcal{L}_d^i}{\partial \phi} \\ \psi &= \psi - \lambda \frac{\partial \mathcal{L}_d^i}{\partial \psi}\end{aligned} \tag{7}$$

where the positive sign of the feature extractor update rule is as a result of the gradient reversal layer. Although this approach is effective in closing the gap of the source domain and the target domain, it only executes the global alignment approach excluding the class-wise discriminability on representations. CLAMP also involves the training process of the target domain based on the pseudo labelling approach to enhance the generalization performance.

4.3. Training strategy of target domain

The vanilla adversarial domain adaptation approach only performs a global domain alignment strategy ignoring the class alignment of the two domains Li et al. [11]. This problem is addressed here using the pseudo labelling strategy of the target domain. Pseudo labelled samples of the target domain are learned in the discriminative fashion using the same loss as the source domain (1). In other words, the self-learning mechanism is carried out for the target domain where a pseudo-label is derived from a predictive output of a model. The pseudo-labelling approach is undertaken:

$$\hat{y}^T = \arg\max_{o=1,\ldots,m} P(y|x), P(y|x) = s_W(g_\phi(f_\theta(x^T))) \tag{8}$$

The pseudo labelling strategy returns the training set of the target domain $\mathcal{T}_{ps}^k = \{(x_i^T, \hat{y}_i^T)\}_{i=1}^{N_k^{ps}}$ where N_k^{ps} denotes the number of pseudo-labelled samples of the target domain of the $k-th$ task. Note that the pseudo-labelling mechanism is also performed to those of the target episodic memory $\mathcal{M}_{k-1}^T$. Nevertheless, this mechanism risks on noisy pseudo labels undermining model's generalization performance. The assessor-guided learning mechanism using the sequence-aware assessor is implemented here to reject noisy pseudo-labels. We adopt similar strategy as Li et al. [11] but the key difference lies in the use of the sequence-aware target assessor $\kappa_{\zeta_T}^T(.) \in [0,1]$. As with the source domain, the target assessor delivers a set of weights $\alpha_T, \beta_T, \gamma_T$ rather than a single weight as per Li et al. [11] to achieve a proper tradeoff between plasticity and stability in the continual learning setting. The loss function of the target domain is formulated:

$$\mathcal{L}_T = \underbrace{\mathbb{E}_{(x,y)\backsim \mathcal{D}_k^T \cup \mathcal{M}_{k-1}^T}[\alpha_T l(s_W(o), y)]}_{L_{CE}} + \underbrace{\mathbb{E}_{(x,y)\backsim \mathcal{M}_{k-1}^T} \beta_T[||o-h|| + l(s_W(o), y)]}_{L_{DER}} + \underbrace{\mathbb{E}_{(x,y)\backsim \mathcal{M}_{k-1}^T}[\gamma_T l(o,h)]}_{L_{distill}} \tag{9}$$

As with the source domain, the assessor of the target domain performs the soft-weighting mechanism rejecting noisy pseudo-labels by allocating small weights. The assessor of the target domain is trained with a meta-objective using the validation loss of the base network.

The validation set $\mathcal{T}_{vl}^k = \{(x_i^S, y_i^S)\}_{i=1}^{N_k^{val}}$ is constructed from highly similar samples of the source domain Li et al. [11]. Such samples are detectable from the output of the domain classifier $\xi_\psi(f_\theta(x))$ where highly similar samples must be close to 0.5. That is, we measure the absolute differences of the domain classifier and 0.5 as $|\xi_\psi(f_\theta(x)) - 0.5|$. The differences are sorted in the descending order and the top $p = 2$ samples are taken for each class making sure a balanced class proportion $N_k^{val} = p * m$. As with the training set, the validation set includes data samples of the source episodic memory $\mathcal{M}_{k-1}^S$. The meta training process is carried out similarly as the training strategy of the source domain where the bi-level optimization strategy is formulated:

$$\begin{aligned} \{\theta^*, \phi^*\} &= \arg\min_{\theta,\phi} \mathbb{E}_{(x,y)\backsim \mathcal{T}_{vl}^k} \mathcal{L}_T \\ s.t \quad & \min_{\zeta_T} \mathbb{E}_{(x,y)\backsim \mathcal{T}_{ps}^k} \mathcal{L}_T \end{aligned} \tag{10}$$

where $\{\theta^*, \phi^*\}$ are the optimal parameters of the base learner with respect to the current assessor of the target domain $\kappa_{\zeta_T}^T(.)$. The inner loop focuses on the training strategy of the assessor using the training set while the outer loop is directed to the training process of the base learner using the validation set. Both the inner and outer loops are solved with the SGD approach since they only call for the first order optimization process. The base learner is evaluated with the validation set resulting in the validation loss utilized to update the target assessor.

$$\zeta_T' = \zeta_T - \beta \sum_{(x,y)\in \mathcal{T}_{vl}^k} \nabla_{\zeta_T} \mathcal{L}_T \tag{11}$$

where β is the learning rate of the inner loop. The updated target assessor produces the three weights $\{\alpha_T, \beta_T, \gamma_T\} = \kappa_{\zeta_T'}^T(x)$ navigating the learning process of the base learner.

$$\{\phi, \theta\} = \{\phi, \theta\} - \mu \sum_{(x,y)\in \mathcal{T}_{ps}^k} \nabla_{\{\phi,\theta\}} \mathcal{L}_T \tag{12}$$

where μ is the learning rate of the outer loop. A joint update of the base learner and the assessor is presented here where every outer loop contains the inner loop. Both assessor and base learner work in collaborative manner.

4.4. Loss function

The overall loss function of CLAMP for the $k-th$ task is written:

$$\mathcal{L}_{all} = \mathcal{L}_S - \mathcal{L}_{pa} + \mathcal{L}_T \tag{13}$$

where a negative sign is inserted to realize the gradient reversal strategy applied to the feature extractor $f_\theta(.)$. The alternate optimization approach is implemented here where at first CLAMP is trained to minimize $\mathcal{L}_S$ affecting the classifier and the feature

extractor $g_\phi(f_\theta(.))$. It is done in the meta-learning manner with the inner loop to update the source assessor $\kappa_{\zeta_S}(.)$ and the outer loop to tune the base network $g_\phi(f_\theta(.))$. The domain adaptation step is carried out afterward where the domain classifier $\xi_\psi(.)$ and the feature extractor $f_\theta(.)$ are updated in respect to $\mathcal{L}_{pa}$ where the gradient reversal layer is applied to the feature extractor $-\frac{\partial \mathcal{L}_k^{ps}}{\partial \theta}$ whereas a normal gradient update is taken for the domain classifier $\frac{\partial \mathcal{L}_k^{ps}}{\partial \psi}$. The next stage addresses the target domain minimizing $\mathcal{L}_T$. As with the source domain, it is carried out with the meta learning approach where the target assessor $\kappa_{\zeta_T}(.)$, inner loop, and the base network $g_\phi(f_\theta(.))$, outer loop, work collaboratively.

5. Theoretical analysis

Theorem 1 *(Ben David et al. and Lao et al. [34,8]). Under a non-continual learning framework, the generalization error of the target domain $\mathcal{D}^T$ is upper bounded by the empirical error of a model on the source domain $\mathcal{D}^S$ plus discrepancy between two domains:*

$$\epsilon_T \leq \epsilon_S + d(\mathcal{D}^S, \mathcal{D}^T) + g* \tag{14}$$

where $d(\mathcal{D}^S, \mathcal{D}^T)$ is the discrepancy of the two domains which can be modelled by the $\mathcal{H}$ divergence while $g*$ is the error of an optimal classifier for both source and target domains. This is expandable for the continual learning setting where sequences of different tasks are observed from the unlabelled target domain and the labelled source domain $\mathcal{T}_k^S, \mathcal{T}_k^T, k \in [1, ..., K]$. Let $\lambda_k = d(\mathcal{D}_k^S, \mathcal{D}_k^T)$ be the discrepancy of the source domain and the target domain in the embedding space $f_\theta(x^S)$ and $f_\theta(x^T)$. The error of the $k-th$ task of the target domain is formalized:

$$\epsilon_{T_k} \leq \epsilon_{S_k} + \lambda_k + g* \tag{15}$$

where $g* = \arg\min_{\phi,\theta}(\epsilon_{T_k} + \epsilon_{S_k})$ is the error of an optimal classifier for both source domain and target domain.

Theorem 2 *(Lao et al. [8]). The total error of the target domain after executing K tasks is bounded as follows:*

$$\epsilon_T \leq \sum_{k=1}^{K}(\epsilon_{S_k} + \lambda_k) + \sum_{k=1}^{K-1}(\epsilon_{\mathcal{M}_k^S} + \epsilon_{\mathcal{M}_k^T}) + g* \tag{16}$$

where $\epsilon_{\mathcal{M}_k}^S, \epsilon_{\mathcal{M}_k}^T$ are the compression errors due to the episodic memory for experience replay.

Proof. We use $\hat{\epsilon}_k^T$ to estimate ϵ_k^T because the episodic memory is used to generate previous experiences $k \in [1, K-1]$. The total error of the target domain is thus formalised as follows:

$$\begin{aligned}\epsilon_T &= \epsilon_{T_k} + \sum_{k=1}^{K-1}(\hat{\epsilon}_{T_k} + \epsilon_{\mathcal{M}_k^T}) \\ &= \epsilon_{S_k} + \lambda_k + \sum_{k=1}^{K-1}(\hat{\epsilon}_{S_k} + \lambda_k + \epsilon_{\mathcal{M}_k^T}) + g*\end{aligned} \tag{17}$$

As with the target domain, the episodic memory $\mathcal{M}_k^S$ is applied in the source domain to perform the experience replay mechanism. This issue results in the compression error. Hence, the error of the source domain at the $k-th$ task is defined as follows:

$$\hat{\epsilon}_{S_k} = \epsilon_{S_k} + \epsilon_{\mathcal{M}_k^S} \tag{18}$$

The total error of the target domain is defined as follows:

$$\begin{aligned}\epsilon_T &\leq \epsilon_{S_k} + \lambda_k + \sum_{k=1}^{K-1}(\epsilon_{S_k} + \lambda_k + \epsilon_{\mathcal{M}_k^T} + \epsilon_{\mathcal{M}_k^S}) + g* \\ &= \sum_{k=1}^{K}\epsilon_{S_k} + \lambda_k + \sum_{k=1}^{K-1}(\epsilon_{\mathcal{M}_k^T} + \epsilon_{\mathcal{M}_k^S}) + g*\end{aligned} \tag{19}$$

The error bound of the target domain implies the importance of CLAMP's learning modules. The training strategy of the source domain $\mathcal{L}_S$ contributes to reduction of ϵ_{S_k} and $\epsilon_{M_k^S}$ while the discrepancy between the source domain and the target domain λ_k is minimized by the unsupervised domain adaptation strategy $\mathcal{L}_{pa}$. Since the training process of the target domain models the target domain directly, it has an impact on ϵ_T, λ_k, and $\epsilon_{M_k^T}$. It should be carefully committed due to the risk of noisy pseudo labels.

6. Experiments

This section presents our numerical validations of CLAMP including benchmark problems, comparisons against prior arts, ablation study, memory analysis, sensitivity analysis and the t-sne analysis.

6.1. Setup

Datasets: we evaluate our method, CLAMP, on four problems to simulate up to 21 cross-domain CL problems. 1) Digit Recognition is constructed by two domains, MNIST(MN)↔USPS(US). 2) Office-31 Saenko et al. [35] has 4652 coloured images from three domains: Amazon (A), DSLR (D) and Webcam (W). Each domain consists of 31 classes. 3) Office-Home Ventakeswara et al. [36] contains four distinct domains: Artistic images (Ar), Clip art (Cl), Product images (Pr), and Real-World images (Rw) where each domain has 65 classes and 12 problems are set. 4) VisDA 2017 Peng et al. [37] consists of a source dataset of synthetic 2D rendering of 3D models generated from different angles and lightning conditions while a target dataset contains photo-realistic images.
Task Division: We detail the task division for each dataset. 1) for the digit recognition case (MNIST and USPS), we divide the 10 classes in each domain into 5 tasks, where each task encompasses 2 classes. 2) for Office-31, we sort the classes in alphabetically ascending order and divide each domain into 5 tasks. The first four tasks include 6 classes each, and the last task includes 7 classes. 3) for Office-Home, we split the 65 classes into 13 subsets of 5 classes per task. 4) for VisDA problem, there exist 12 classes split into 4 tasks of 3 classes each.
Benchmark Algorithms: CLAMP is compared against ten prior arts: EWC Kirkpatrick et al. [1], LwF Li et al. [38], SI Zenke et al. [39], MAS Aljundi et al. [40], RWalk Chaudhry et al. [41], iCARL Rebuffi et al. [5], IL2M Belouadah et al. [42], EEIL Castro et al. [43], DANN Ganin et al. [10], HAL Chaudhry et al. [44], AGLA Masum et al. [14] and CDCL De Carvalho et al. [13]. No comparisons are done against Lao et al. [8] due to the absence of its public source codes. *The source codes for CLAMP, including benchmarks and task split details, are publicly available at https://github.com/wengweng001/CLAMP_torch.git*.
Evaluation Metric: the performance of consolidated algorithms is evaluated using the average classification accuracy of unlabelled target domain assessing their performances for all already seen tasks, thus reflecting the existence of CF problems Chaudhry et al. [44]. The target domain is chosen to measure their performances against the domain shift problem whether or not the domain adaptation mechanism succeeds. The average accuracy $\mathcal{A} \in [0, 1]$ is mathematically expressed as follows:

$$\mathcal{A}_{\uparrow} = \frac{1}{k} \sum_{j=1}^{k} a_{k,B_{k,j}} \tag{20}$$

where $a_{k,B_{k,j}} \in [0, 1]$ denotes the accuracy evaluated on the test set of task j after the model has been trained with the task k. In a nutshell, the average accuracy $\mathcal{A}_T$ evaluates the accuracy of all tasks after learning the last task.
Implementation Details: We report the average accuracy over five different runs under the same computational resources to ensure fair comparisons; however, for VisDA, the average accuracy is reported from three runs. The hyper-parameters of all algorithms are selected using the grid search approach and reported in the appendix. LeNet is used as a backbone network for the digit recognition problem, while for Office-31 and VisDA, we use a ResNet-34 pretrained on ImageNet, and for the Office-Home, we apply a pretrained ResNet-50. We train our model using SGD optimizer. Batch size is 128 for digit recognition and 32 for Office-31, Office-Home, and VisDA.

6.2. Numerical results

Table 1 & 2 report numerical results of consolidated algorithms in the digit recognition problem, the VisDA problem and the office-31 problem separately. CLAMP's superiority is evident in the digit recognition problem, where it beats other algorithms with large margins, at least 41%. This finding demonstrates that the problem of cross-domain continual learning is extremely challenging and unsolvable with classic single-domain CL algorithms. Note that the digit recognition problem is considered as a toys CL problem under a single domain configuration. This fact also substantiates the importance of domain alignment step to dampen the gap across two domains while protecting against the CF problem as exemplified by CLAMP. CLAMP's performance, surpassing even joint training, which was jointly trained on all tasks of the source process, attests to its efficacy in domain alignment. On the other hand, CLAMP demonstrates its advantage in the VisDA problem outperforming all other methods with significant margin except the joint training serving as the upper bound. The case of office 31 presents similar finding where CLAMP beats other methods with significant margins.

A similar pattern is observed in the Office-Home problem, where CLAMP consistently outperforms the performance of other algorithms by at least 10% for each experiment and approximately 22% across different tasks on average. Moreover, in the more challenging Office-Home dataset, CLAMP is also superior to other algorithms by a minimum of 14% across 13 tasks, as exhibited in Table 3. Analyzing the Office-31 and Office-Home experiments, we hypothesize that significant domain discrepancies in two specific pairings of Office-31, A↔D and A↔W, and across most tasks in Office-Home, given the overall poor performance of other methods. In such challenging conditions, CLAMP demonstrates its robustness by achieving a notable margin of superiority. The problem of cross-domain continual learning is difficult to tackle because of the presence of large domain discrepancies and the CF issue. This issue cannot be sufficiently overcome with single-process continual learning algorithms or traditional domain adaptation algorithms alone. Fig. 3 depicts the difference across various methods for four benchmarks: digit recognition, Office-31, Office-Home and VisDA. CLAMP surpasses competing methods in all evaluation scenarios with notable gaps.

Table 1
Average query accuracy (%) of all learned tasks on digit recognition, MNIST(MN) & USPS(US), and VisDA.

	MN→US	US→MN	VisDA
Source Only	17.80 ± 0.2	12.55 ± 0.9	12.85 ± 1.2
Joint Training	77.81 ± 1.9	49.81 ± 0.7	44.76 ± 0.9
EWC	17.88 ± 0.3	13.30 ± 1.8	12.65 ± 3.5
LwF	17.74 ± 0.4	12.94 ± 1.5	13.10 ± 1.2
SI	17.80 ± 0.2	12.54 ± 1.0	13.40 ± 2.1
MAS	17.92 ± 0.3	13.18 ± 2.0	13.45 ± 1.0
RWalk	17.92 ± 0.3	13.39 ± 1.7	13.91 ± 1.7
iCaRL	42.11 ± 2.5	24.04 ± 0.9	15.67 ± 1.7
IL2M	43.67 ± 3.3	23.16 ± 1.9	19.58 ± 1.4
EEIL	38.75 ± 3.0	22.46 ± 1.4	15.99 ± 1.8
HAL	72.31 ± 1.4	47.55 ± 2.3	16.87 ± 1.6
AGLA	62.75 ± 4.2	47.98 ± 1.1	25.8 ± 2.7
CDCL	66.73	52.50	10.16
DANN	17.89 ± 0.9	16.90 ± 3.1	16.16 ± 0.1
CLAMP	**84.98 ± 1.3**	**89.63 ± 1.0**	**30.71 ± 1.8**

The results for CDCL are taken from the original paper.

Table 2
Average query accuracy (%) of all learned tasks on Office-31.

Source Target	A D	A W	D A	D W	W A	W D	*Office-31 Avg.*
Source Only	14.96 ± 3.0	14.90 ± 1.6	8.44 ± 2.5	20.07 ± 2.8	14.91 ± 3.0	24.29 ± 3.0	*16.26 ± 5.4*
Joint Training	68.59 ± 2.6	59.27 ± 1.3	42.67 ± 2.8	80.88 ± 2.9	49.24 ± 2.0	90.43 ± 2.3	*65.18 ± 18.4*
EWC	15.05 ± 2.8	15.09 ± 1.5	8.67 ± 2.7	20.25 ± 2.3	14.64 ± 3.4	25.65 ± 3.0	*16.56 ± 5.8*
LwF	19.85 ± 2.1	18.30 ± 1.9	7.67 ± 2.4	21.18 ± 4.8	16.23 ± 2.3	27.20 ± 2.4	*18.41 ± 6.4*
SI	15.59 ± 2.3	14.67 ± 1.9	8.38 ± 2.7	19.86 ± 2.9	14.91 ± 3.0	24.29 ± 2.3	*16.28 ± 5.4*
MAS	15.95 ± 3.9	14.90 ± 1.5	8.67 ± 2.7	20.25 ± 2.3	14.52 ± 3.4	25.65 ± 3.0	*16.66 ± 5.8*
RWalk	15.95 ± 3.0	15.09 ± 1.5	8.61 ± 2.7	20.25 ± 2.3	14.64 ± 3.4	25.28 ± 3.0	*16.64 ± 5.6*
iCaRL	47.68 ± 3.8	37.87 ± 1.8	38.77 ± 5.2	79.84 ± 3.6	42.34 ± 1.8	88.60 ± 1.8	*55.85 ± 22.4*
IL2M	48.12 ± 2.0	43.18 ± 3.3	39.95 ± 5.4	79.61 ± 2.2	42.84 ± 1.7	86.46 ± 1.6	*56.69 ± 20.7*
EEIL	45.51 ± 2.3	41.02 ± 3.9	40.23 ± 5.0	79.21 ± 2.4	42.76 ± 1.9	86.72 ± 1.9	*55.91 ± 21.2*
HAL	33.51 ± 1.4	32.55 ± 3.9	25.62 ± 1.7	62.81 ± 2.8	29.42 ± 3.3	53.52 ± 2.4	*39.57 ± 2.58*
AGLA	55.83 ± 5.98	49.81 ± 3.77	29.61 ± 4.08	62.50 ± 6.02	31.26 ± 1.20	56.85 ± 5.59	47.64 ± 13.93
CDCL	9.68	10.98	7.02	27.97	6.73	28.98	*15.23*
DANN	9.03 ± 1.2	17.77 ± 2.8	2.07 ± 1.1	4.73 ± 3.9	17.09 ± 2.0	17.88 ± 8.2	*11.43 ± 7.1*
CLAMP	**84.29 ± 2.9**	**80.39 ± 2.0**	**61.37 ± 1.3**	**94.85 ± 0.3**	**59.08 ± 4.2**	**98.47 ± 2.2**	***79.74 ± 16.5***

The results for CDCL are taken from the original paper.

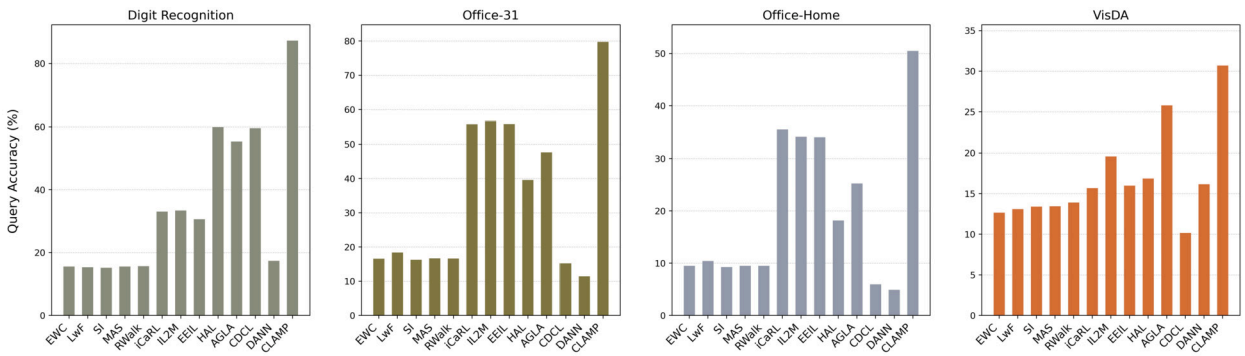


Fig. 3. Average query accuracy comparison across different baselines for four benchmarks: digit recognition, Office-31, Office-Home and VisDA.

6.3. Ablation studies

Ablation study is performed to understand the impact of each learning component where numerical results are reported in Table 4. It is perceived that the naive training on the source domain leads to the poorest results since the catastrophic forgetting problem and the domain discrepancies between the source and target domains occur. An additional adversarial domain adaptation step as shown in the PA does not help to improve numerical results. Baseline 1 integrating the pseudo labelling strategy of the target domain and the adversarial domain adaptation method improves numerical results of the naive approach. The deployment of assessors to produce meta-weights as shown in Baseline 2 also boosts numerical results compared to Baseline 1. The use of episodic memory addresses the catastrophic forgetting problem as shown in Baseline 3 but still does not deliver good performances because of the absence of

Table 3
Average query accuracy (%) of all learned tasks on Office-Home.

Method	Ar→Cl	Ar→Pr	Ar→Ew	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	*Avg.*
Source Only	7.75 ± 1.7	11.14 ± 1.1	15.47 ± 1.3	5.11 ± 2.6	7.89 ± 0.8	7.32 ± 0.7	7.09 ± 1.5	6.62 ± 1.1	10.38 ± 0.3	13.70 ± 0.9	8.28 ± 1.7	13.89 ± 0.1	*9.55 ± 3.3*
Joint Training	37.17 ± 1.3	57.04 ± 1.1	67.62 ± 1.9	40.96 ± 2.2	48.90 ± 1.5	54.86 ± 1.0	43.77 ± 1.6	34.25 ± 2.5	64.62 ± 1.2	60.06 ± 1.7	40.72 ± 1.2	72.53 ± 1.1	*51.88 ± 12.7*
EWC	7.34 ± 1.6	11.68 ± 1.5	15.38 ± 1.4	5.06 ± 1.6	7.70 ± 0.6	7.51 ± 0.7	7.68 ± 1.4	6.42 ± 0.7	10.35 ± 0.6	13.78 ± 1.5	7.75 ± 1.8	13.43 ± 0.2	*9.51 ±3.3*
LwF	8.11 ± 2.1	11.91 ± 1.4	16.20 ± 1.3	5.09 ± 2.6	8.68 ± 0.8	9.40 ± 1.1	8.30 ± 1.6	7.35 ± 1.2	12.65 ± 1.7	13.56 ± 1.3	8.51 ± 1.7	15.02 ± 0.7	*10.40 ± 3.4*
SI	7.44 ± 1.7	11.15 ± 1.0	15.50 ± 1.4	4.28 ± 1.9	7.59 ± 0.8	7.05 ± 1.0	7.32 ± 1.2	6.35 ± 1.3	9.92 ± 1.0	12.85 ± 1.4	7.87 ± 1.8	13.73 ± 0.4	*9.25 ± 3.4*
MAS	7.38 ± 2.1	11.59 ± 1.0	15.51 ± 2.0	4.69 ± 1.8	7.42 ± 1.1	7.12 ± 0.7	7.81 ± 1.9	6.44 ± 1.0	10.87 ± 0.7	14.09 ± 2.1	7.46 ± 1.7	13.37 ± 0.4	*9.48 ± 3.5*
RWalk	7.18 ± 1.8	11.53 ± 1.4	15.25 ± 1.5	5.15 ± 2.2	7.67 ± 0.6	7.45 ± 0.4	7.53 ± 1.7	6.54 ± 1.0	10.91 ± 0.6	13.56 ± 1.8	7.75 ± 1.6	13.53 ± 0.4	*9.5 ±3.3*
iCaRL	28.03 ± 1.0	44.93 ± 1.3	53.91 ± 2.1	19.81 ± 1.0	30.43 ± 2.0	33.11 ± 1.3	30.28 ± 2.1	23.39 ± 1.5	46.09 ± 2.1	38.72 ± 1.4	25.93 ± 1.4	52.40 ± 0.9	*35.59 ± 11.4*
IL2M	26.60 ± 1.0	43.82 ± 1.8	52.72 ± 1.1	18.03 ± 0.8	28.14 ± 1.2	30.85 ± 1.5	28.17 ± 2.5	20.24 ± 0.8	46.12 ± 1.5	37.52 ± 1.5	25.85 ± 1.8	52.08 ± 1.0	*34.18 ±12.0*
EEIL	26.70 ± 1.2	44.10 ± 1.9	52.80 ± 1.2	17.21 ± 1.0	28.64 ± 0.8	31.37 ± 1.2	28.35 ± 1.6	20.41 ± 0.9	45.52 ± 1.1	36.26 ± 1.6	25.83 ± 1.4	51.74 ± 0.9	*34.08 ± 11.9*
HAL	10.90 ± 1.4	20.34 ± 1.4	29.51 ± 1.3	10.17 ± 1.7	13.04 ± 1.9	15.72 ± 1.9	14.77 ± 2.3	12.38 ± 1.6	26.32 ± 1.4	20.50 ± 2.4	13.34 ± 1.3	31.13 ± 1.0	*18.18 ± 1.6*
AGLA	18.58 ± 1.5	29.89 ± 2.1	34.20 ± 4.0	17.84 ± 1.5	23.39 ± 1.3	28.57 ± 2.9	20.02 ± 1.0	19.19 ± 2.3	32.89 ± 1.9	28.67 ± 5.4	20.61 ± 3.0	41.17 ± 2.6	*25.25 ± 7.5*
CDCL	4.15	4.77	6.01	5.56	8.32	6.35	3.92	5.02	6.29	6.45	6.26	8.35	*5.95*
DANN	4.03 ± 0.8	4.40 ± 1.1	5.82 ± 0.6	2.62 ± 3.0	5.00 ± 1.7	5.01 ± 1.8	4.71 ± 1.2	3.48 ± 1.7	5.82 ± 1.0	6.06 ± 1.4	4.21 ± 0.6	7.59 ± 0.7	*4.90 ± 1.3*
CLAMP	**42.57 ± 0.9**	**62.20 ± 1.0**	**63.63 ± 1.7**	**38.50 ± 2.0**	**49.47 ± 1.5**	**49.92 ± 0.9**	**40.63 ± 3.1**	**40.45 ± 1.0**	**55.29 ± 1.9**	**54.88 ± 1.8**	**45.21 ± 1.3**	**63.18 ± 1.8**	***50.49 ± 9.3***

The results for CDCL are taken from the original paper.

Table 4
Ablation studies of different learning modules for CLAMP on MNIST(MN)↔USPS(US) and Office-31.

Method	PA	PL	Meta	R_1	R_2	MU → US	US → MN	A → D	A → W	D → A	D → W	W → A	W→D	*Office-31 avg.*
Naive	✗	✗	✗	✗	✗	17.49 ± 2.2	17.15 ± 1.2	13.60 ± 1.1	15.28 ± 3.6	12.36 ± 2.0	29.90 ± 1.8	16.63 ± 1.3	33.97 ± 1.7	*20.29 ± 9.2*
PA	✓	✗	✗	✗	✗	18.90 ± 0.4	16.45 ± 6.4	13.60 ± 3.1	13.95 ± 1.5	14.04 ± 1.7	28.72 ± 1.6	14.15 ± 1.6	35.82 ± 1.4	*20.05 ± 9.7*
Baseline 1	✓	✓	✗	✗	✗	19.29 ± 0.6	19.47 ± 0.7	14.33 ± 2.5	15.05 ± 2.8	15.49 ± 2.1	35.91 ± 1.6	14.53 ± 2.2	35.72 ± 2.3	*21.84 ± 10.8*
Baseline 2	✓	✓	✓	✗	✗	19.21 ± 0.6	19.46 ± 0.7	12.10 ± 2.0	16.81 ± 3.5	16.01 ± 3.1	34.64 ± 2.0	14.92 ± 1.9	40.16 ± 2.7	*22.44 ± 11.8*
Baseline 3	✓	✗	✗	✓	✗	53.27 ± 5.4	65.43 ± 2.3	66.02 ± 1.6	55.82 ± 2.8	49.52 ± 2.6	92.20 ± 1.6	51.11 ± 1.4	95.35 ± 1.8	*68.34 ± 20.1*
Baseline 4	✓	✓	✗	✓	✗	58.31 ± 4.3	58.31 ± 4.3	74.29 ± 3.6	67.53 ± 2.1	55.23 ± 2.5	92.63 ± 1.7	55.56 ± 1.7	96.97 ± 1.2	*73.70 ± 17.9*
Baseline 5	✓	✓	✗	✓	✓	79.50 ± 2.0	89.61 ± 0.8	79.90 ± 2.1	77.54 ± 2.8	62.91 ± 1.5	94.13 ± 0.9	62.58 ± 3.8	99.20 ± 1.3	*79.38 ±15.3*
CLAMP	✓	✓	✓	✓	✓	**84.98 ± 1.3**	**89.63 ± 1.0**	**84.29 ± 2.9**	**80.39 ± 2.0**	**61.37 ± 1.3**	**94.85 ± 0.3**	**59.08 ± 4.2**	**98.47 ± 2.2**	***79.74 ± 16.5***

PA: process adaptation, PL: pseudo label, Meta: assessor, R_1: previous source tasks memory replay, R_2: previous target tasks memory replay.

Table 5
Accuracy (%) of CLAMP with different exemplar sizes per class on Office-31.

Exemplar per class	1	3	5	**10(ours)**
A→D	62.97	79.64	81.3	**84.29**
A→W	61.85	72.73	75.29	**80.39**
D→A	46.53	58.85	60.6	**61.37**
D→W	88.74	92.19	93.74	**94.85**
W→A	45.41	57.42	60.78	**59.08**
W→D	92.96	97.12	98.21	**98.47**
Avg.	*66.41*	*76.33*	*78.32*	***79.74***

Table 6
Results of using different inner-iteration steps to update assessor network parameters on MNIST(MN)↔USPS(US).

MN→US	Steps	1(ours)	2	3	4	5
	Accuracy (%)	84.98 ± 1.3	84.88 ± 2.1	84.26 ± 2.1	83.83 ± 1.3	82.28 ± 2.3
	Training Time (sec.)	2267 ± 413.8	3802 ± 110.3	4920 ± 81.2	6132 ± 89.8	7404 ± 161.4
US→MN	Steps	1(ours)	2	3	4	5
	Accuracy (%)	89.63 ± 1.0	89.40 ± 1.0	88.84 ± 1.9	89.49 ± 0.9	89.57 ± 0.7
	Training Time (sec.)	597 ± 2.4	1030 ± 23.3	1153 ± 15.4	1270 ± 25.9	1381 ± 20.5

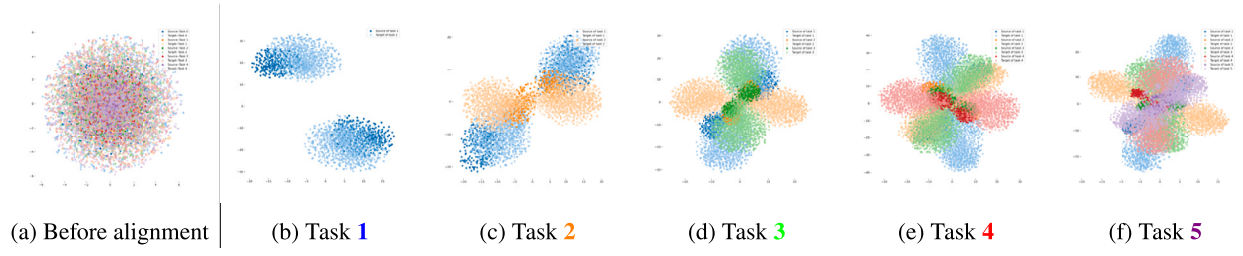


Fig. 4. t-SNE plots of the USPS (in light colours) → MNIST (in dark colours) case.

domain adaptations. Baseline 4 improves numerical results of baseline 3 but suffers from the catastrophic forgetting problem of the target domain because of the absence of episodic memory of the target domain. The vital role of assessors is confirmed in Baseline 5 where their absences deteriorate numerical results. These findings substantiate positive impacts of CLAMP's learning modules.

6.4. Memory analysis

Table 5 tabulates numerical results of CLAMP under different memory sizes per class for both target domain and source domain. It is perceived that the performance of CLAMP degrades when decreasing the number of exemplar per class in the memory. This situation occurs because reduction of memory sizes leads to the extreme class imbalanced problem between the current task and the episodic memory leading to the catastrophic forgetting problem. However, the performance gaps gradually decrease with the increase of the memory size. Note that CLAMP utilizes 10 exemplars per class in the main numerical results showing modest memory usages.

6.5. Effect of inner loop steps in assessor network update

To validate the approximation of the assessor's single-step update, we investigate the influence of different step sizes on the final average accuracy and corresponding training time. The results are summarized in Table 6, where steps indicate how many times the inner iteration updates the assessor network parameters before determining the three meta-weights. It turns out that larger inner iteration steps do not significantly improve performance and impose more processing time, which is similar to Li et al. [11]. The increase of inner-iteration of the assessor network update results in the unstable impact on the optimization of the main network parameters. Based on the results, the single meta update step approaches satisfactory performance and incurs much less execution time. Thus, we apply the single-step update in the inner loop.

6.6. T-SNE analysis

The t-sne analysis is performed to evaluate the embedding qualities of CLAMP using the USPS→MNIST case. Fig. 4 visualizes the t-SNE plot across each task. Before the training process begins, there do not exist any clear patterns indicating confused predictions. The advantage of CLAMP is clearly seen after the training process where the source domain and target domain are successfully aligned and show remarkable cluster structures showing confident predictions. In addition, different classes are projected differently in the feature space signifying the robustness against catastrophic forgetting problem.

Table 7
Average query accuracy comparison of learning rate impact on base learner (l_b) and assessors (l_a) performance in MNIST→USPS experiment, highlighting the optimal method results in bold.

l_a	0.1	0.01	0.001	0.0001
$l_b = 0.001$	38.13 ± 2.9	78.05 ± 8.0	82.34 ± 2.5	**84.98 ± 1.3**
l_b	0.1	0.01	0.001	0.0001
$l_a = 0.0001$	63.32 ± 5.8	79.88 ± 1.9	**84.98 ± 1.3**	53.98 ± 4.4

Table 8
Sensitivity analysis with varying threshold (γ) for MNIST↔USPS.

γ	MNIST→USPS	USPS→MNIST
0.7	83.24 ± 1.57	88.80 ± 0.94
0.75	83.58 ± 0.97	89.65 ± 0.91
0.8	84.56 ± 1.21	89.31 ± 1.00
0.85(ours)	84.98 ± 1.3	89.63 ± 1.0
0.9	84.07 ± 1.26	89.29 ± 1.15

6.7. Sensitivity analysis

The influence of learning rate for base learner (l_b) and assessors (l_a) on the performance of CLAMP is elaborated in this section. Table 7 shows learning rate analysis results on the digit recognition experiment. A careful examination of Table 7 reveals significant insights into the optimal learning rate settings for both base learners and assessors in the context of MNIST→USPS. It is evident that the base learner's performance improves with a decrease in the learning rate, with the most notable improvement observed when l_b is set to 0.001. The accuracy peaks at 84.98% (± 1.3), indicating a higher stability and efficiency at this learning rate. To strive a balance between performance and training time, we select 0.0001 as the assessor's learning rate. Conversely, for assessors, the learning rate of 0.0001 with l_b fixed at 0.001 demonstrates the best performance, achieving an accuracy of 84.98% (± 1.3). This suggests that a lower learning rate for assessors is conducive to achieving optimal performance. Additionally, the results indicate a clear trend of diminishing returns at extremely low or high learning rates, as seen with the other combinations of l_a and l_b, which yield lesser accuracy and higher standard deviations. The highlighted results in bold in the table further emphasize the optimal learning rate combinations for maximizing accuracy in digit recognition tasks within the CLAMP framework. This analysis aligns well with the fact that CLAMP forms a bi-level optimization approach between the assessor in the inner loop and the base learner in the outer loop. That is, the assessor should be updated in the slower pace than that the base learner to assure model's convergences. Table 8 reports our numerical results when varying the pseudo label thresholds. It is seen that a too low threshold leads to performance degradation by about 1% because it leads to too many noisy pseudo labels to be involved in the training process. On the other hands, a too high threshold also results in performance deterioration because too few pseudo labels are induced. Generally speaking, CLAMP is not too sensitive to the pseudo label threshold because of the presence of a meta-training loop in the target domain contributing to reject noisy pseudo labels. The pseudo label threshold is fixed at 0.85 in all our experiments.

6.8. Analysis of model generalization

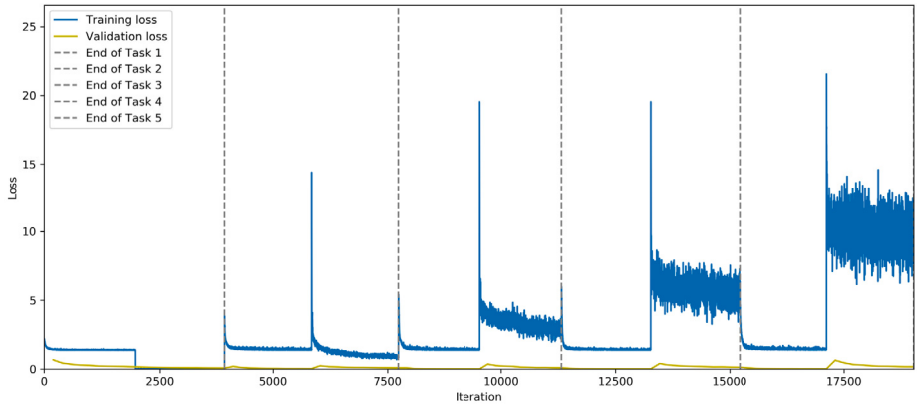
This section is meant to analyze the generalization performance of CLAMP where the trace of training losses and validation losses are portrayed in Fig. 5 for the digit recognition problem. It is evident that CLAMP does not exhibit any over-fitting signs where the validation losses are well below the training losses during the cross-domain continual learning process. That is, CLAMP generalizes well to unseen samples of the validation set.

6.9. Analysis of model's uncertainties

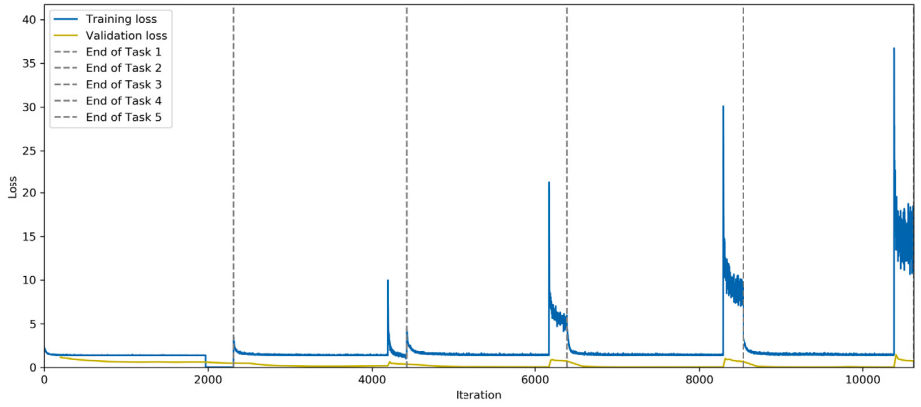
Using the entropy formulation $H(X) = -\sum_{i=1}^{n} P(x_i) \log_b P(x_i)$, we analyze the model's performance in relation to entropy. Fig. 6 demonstrates that for digit recognition tasks on MNIST and USPS datasets, CLAMP achieves the highest average query accuracy among all compared methods. This plot shows the correlation between model uncertainty and entropy, in which lower entropy indicates greater certainty in model predictions. Specifically, CLAMP achieves an entropy of 0.8445 for MNIST→USPS and 0.8745 for USPS→MNIST. While these values are not the lowest, they are comparatively low, denoting a respectable level of certainty in its predictions. Note that HAL displays the lowest entropy, suggesting it is more certain in its predictions than CLAMP for the MNIST→USPS task. The figure clearly shows that CLAMP maintains high accuracy while achieving relatively lower entropy compared to other methods. This visually confirms that CLAMP is generally more accurate and certain in its predictions than most other approaches.

6.10. Statistical test

Table 9, Table 10 and Table 11 present the results of T-tests comparing CLAMP with benchmark algorithms on all experiments. The T-values in the table provide statistical evidence of significant differences in performance between CLAMP and the other algorithms,



(a) MNIST→USPS



(b) USPS→MNIST

Fig. 5. CLAMP loss trajectories in digits recognition: (a) MNIST→USPS, (b) USPS→MNIST. It illustrates the trajectories of training loss and validation loss over the entire training process in the cross-domain continual learning scenario.

Table 9
T-test results (significance level: 0.05) comparing CLAMP with benchmark algorithms on digit recognition and VisDA.

Source Target	Split MNIST Split USPS	Split USPS Split MNIST	VisDA
SourceOnly	2288.18	5729.31	230.28
Joint Training	139.53	3262.18	-194.73
EWC	2253.13	3706.91	127.99
LwF	2214.7	4254	227.05
SI	2288.18	5451.09	174.56
MAS	2251.79	3418.95	233.8
RWalk	2251.79	3865.52	189.26
iCaRL	681.58	4875.27	169.44
IL2M	521.78	3095.82	136.14
EEIL	633.44	3904.18	161.29
HAL	297.1	1677.84	160.29
DANN	1900.91	2232.83	225.12

with notably high t-values. It is important to note that the p-values, although they are not presented in the table, are extremely small, all well below 0.0001, indicating highly significant differences. Overall, the T-values provide strong statistical evidence that CLAMP performs significantly better than the benchmark algorithms in most cross-domain continual learning tasks. This superiority is evident in various cases, except when compared with Joint Training on the VisDA dataset and in six experiments of Office-Home, as indicated by red markings. This exception may be attributed to the elevated discrepancies between source and target domains

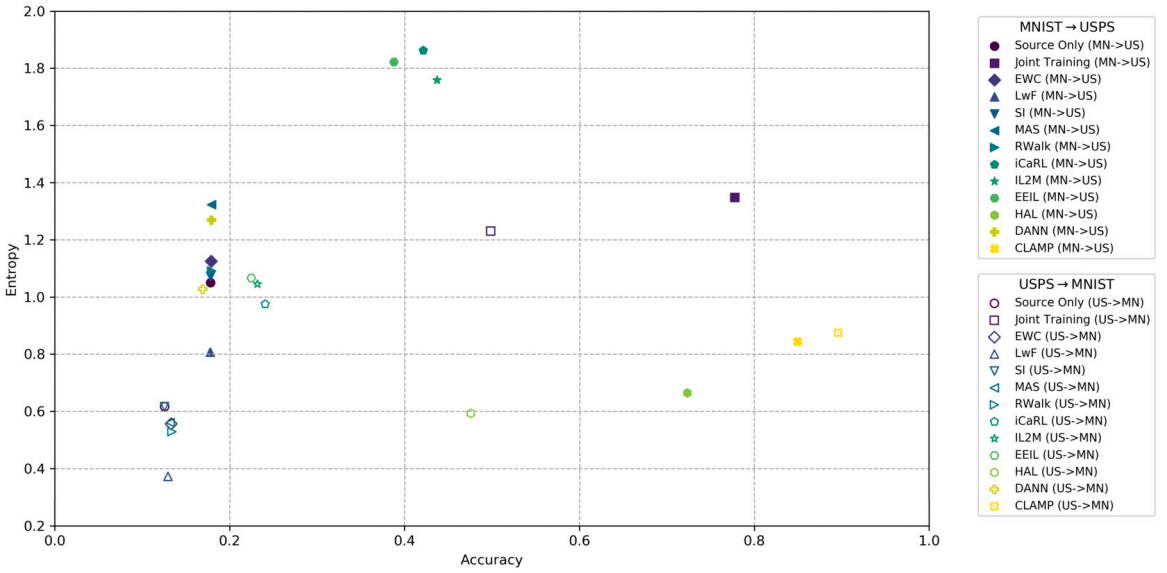


Fig. 6. Comparative Analysis of Model Performance on digit recognition tasks: Accuracy vs. Entropy (predictive uncertainty).

Table 10
T-test results (significance level: 0.05) comparing CLAMP with benchmark algorithms on the Office-31.

	A→D	A→W	D→A	D→W	W→A	W→D
Source Only	370.8	720.95	996.98	748.74	454.21	444.97
Joint Training	26.46	249.64	321.51	135.1	112.27	56.37
EWC	118.65	736.47	933.4	906.84	436.49	436.81
LwF	90.99	634.62	1044.21	431.9	474.94	488.5
SI	115.26	671.72	938.53	725.23	454.21	520.11
MAS	110.62	738.62	933.4	906.84	437.67	436.81
RWalk	111.91	736.47	934.46	906.84	436.49	439.03
iCaRL	37.97	445.56	223.79	117.15	194.44	77.49
IL2M	37.71	271.89	204.68	193.53	190.23	98.52
EEIL	39.86	253.27	217.18	182.32	187.9	90.2
HAL	54.15	307.76	886.62	320.8	294.72	308.1
DANN	199.39	513.12	1848.2	649.62	479.08	211.83

Table 11
T-test results (significance level: 0.05) comparing CLAMP with benchmark algorithms on the Office-Home.

Source Target	Art Clipart	Art Product	Art Real World	Clipart Art	Clipart Product	Clipart Real World	Product Art	Product Clipart	Product Real World	Real World Art	Real World Clipart	Real World Product
Source Only	386.55	744.62	485.27	165.7	530.25	805.68	158.54	485.95	503.46	333.11	368.49	592.74
Joint Training	72.93	75.25	-33.75	-17.67	5.83	-79.18	-14.65	49.17	-89.53	-34.06	54.19	-96.09
EWC	409.81	607.53	472.45	278.8	560.52	802.09	157.69	595.32	486.37	285.55	360.27	595.53
LwF	322.08	633.7	477.91	217.5	520.18	614.78	150.86	452.5	360.65	302.94	366.2	540.61
SI	390	782.58	471.27	264.89	534.08	687.13	163.12	443.98	455.66	300.04	359.11	581.4
MAS	328.9	775.84	395.31	268.32	490.09	809.47	146.94	513.54	473.06	240.07	376.67	585.63
RWalk	375.52	638.49	460.16	239.53	560.93	929.87	152.4	512.03	480.31	264.24	388.02	583.75
iCaRL	230.78	228.28	77.58	178.49	165.11	229.26	45	202.08	70.05	115.36	215.5	116.13
IL2M	253.48	193.51	116.19	202.93	240.73	235.08	50.93	337	81.69	120.61	186.19	116.87
EEIL	225.93	182.76	112.23	203.32	265.64	266.67	57.3	318.08	95.96	125.86	216.62	123.24
HAL	406.34	527.48	343.8	230.47	326.26	350.79	109.06	317.69	264.7	186.56	370.17	337.44
DANN	683.46	842.91	691.49	212.5	425.24	481.22	175.9	400.27	496.84	348.51	611.49	624.01

in these specific experiments. Note that the joint training presents the case of upper bound for continual learning. These findings underscore the effectiveness of CLAMP in addressing cross-domain continual learning problem.

6.11. Experiments with DomainNet

Additional experiments are performed using a highly challenging problem for domain adaptation, DomainNet [45]. The DomainNet problem presents 345 classes and 6 domains where the continual learning problem is constructed by splitting 345 classes into 15 tasks with 23 classes each. CLAMP is compared with CDCL [13] representing prior arts in the cross-domain continual learning. 2

Table 12
Average query accuracy (%) of all learned tasks on DomainNet (clipart and sketch).

Method	clipart → sketch	sketch → clipart
CDCL	3.49	2.17
CLAMP	**12.87**	**13.67**

The results for CDCL are taken from the original paper.

cross domain experiments, namely $sketch \leftrightarrow clipart$, are executed. Table 12 reports our numerical results. It is perceived that CLAMP outperforms CDCL with over 10% margins across 2 cross-domain cases, $sketch \leftrightarrow clipart$. Note that the DomainNet problem is highly challenging because of the large number of classes and significant discrepancies across the two domains, i.e., CDCL only obtains $2-3\%$ accuracy. These results further confirm the robustness of our finding in respect to the advantage of CLAMP for cross-domain continual learning problems.

6.12. Complexity analysis

In this part, we analyze the complexity of CLAMP. Let N_k^S and N_k^T denote the number of samples for the k-th task in the source and target domains, respectively, where $k \in \{1, \ldots, K\}$. Let M_{k-1}^S and M_{k-1}^T represent the episodic memory for the source and target domains of the previous task, starting from $k = 2$. E is the number of epochs, and E_{inner} and E_{outer} are the number of inner and outer epochs in the meta learning process, respectively. At k-th task, the complexity of unsupervised process adaption (Stage 1 in Algorithm 1) is $O\left(\min(N_k^S + M_{k-1}^S, N_k^T + M_{k-1}^T)\right)$, the complexity of Stage 2 is $O\left(E_{outer} \times \left(N_{trans} \times (N_k^S + M_{k-1}^S) + E_{inner} \times N_{trans} \times (N_k^S + M_{k-1}^S)\right)\right)$ and that of Stage 3 is $O(E_{outer} \times (N_{trans} \times (N_k^T + M_{k-1}^T) + E_{inner} \times N_{trans} \times (N_k^T + M_{k-1}^T)))$. N_{trans} represents the number of transformations, which is three in CLAMP. Given E_{inner} and E_{outer} are both 1 (Appendix D) and assuming $min(N_k^S + M_{k-1}^S, N_k^T + M_{k-1}^T)$ is represented as $N_k + M_k$. Thus, the complexity of each stage simplifies to $O\left((N_k + M_k)\right)$ for Stage 1, $O(6(N_k^S + M_{k-1}^S))$ for Stage 2 and $O(6(N_k^T + M_{k-1}^T))$ for Stage 3. The overall complexity of CLAMP can be expressed as:

$$O_{\text{overall}} = E \times \sum_{k=1}^{K} (O(\text{Stage 1}) + O(\text{Stage 2}) + O(\text{Stage 3})) \tag{21}$$

$$\begin{aligned} &= E \times \sum_{k=1}^{K} \Big(O\left(\min(N_k^S + M_{k-1}^S, N_k^T + M_{k-1}^T)\right) \\ &\quad + O\left(E_{outer} \times \left(3 \times (N_k^S + M_{k-1}^S) + E_{inner} \times 3 \times (N_k^S + M_{k-1}^S)\right)\right) \\ &\quad + O\left(E_{outer} \times \left(3 \times (N_k^T + M_{k-1}^T) + E_{inner} \times 3 \times (N_k^T + M_{k-1}^T)\right)\right)\Big) \end{aligned} \tag{22}$$

$$= E \times \sum_{k=1}^{K} \left(O\left((N_k + M_k)\right) + O(6(N_k^S + M_{k-1}^S)) + O(6(N_k^T + M_{k-1}^T))\right) \tag{23}$$

$$\leq \sum_{k=1}^{K} O(7E \times (N_k + M_k)) \tag{24}$$

6.13. Discussion

Five aspects are observed from our experiments:

- The cross-domain continual learning problem is challenging where it suffers from both the CF issue and the domain shift issue. In addition, because both the source domain and the target domain characterize sequences of tasks, the CF problem occurs in both domain.
- The performance of other algorithms is poor because they only handle the CF problem in the source domain ignoring the CF problem in the target domain as well as the issue of domain shift due to the absence of domain alignment procedures.
- CLAMP performs satisfactorily compared to other methods because it addresses the CF problem in both domain as well as the domain shift problem. The presence of adversarial domain adaptation not only dampens the gap between the source domain and the target domain, i.e., the domain alignment but also attains the class alignment because of the pseudo-labelling mechanism. In addition, the dual assessors contribute positively to balance the stability and the plasticity with generations of meta-weights steering the roles of loss functions.
- The learning rates play vital roles in the meta-training configuration. We find that the inner loop should be set lower than the outer loop to assure model's convergence.
- Joint training usually serving as an upper bound in the continual learning area does not perform well in the cross-domain continual learning problems because it does not have any strategies to overcome the issue of domain shift.

- We are certain that the problem of shortcut learning where a model only focuses on simple features does not exist in our method because these models are given with sufficient samples in the source domain and the target domain.

6.14. Limitation

Although CLAMP delivers commendable performances in the cross-domain continual learning problems compared to the prior arts, several issues remain open:

- CLAMP still utilize the episodic memories storing old data samples for the source domain and the target domain. Notwithstanding that the episodic memory is common in the continual learning, it imposes additional memory expenditures and the size of episodic memory grows as the number of tasks.
- CLAMP imposes high complexity because two assessor networks need to be trained simultaneously aside from the main networks.
- Our setting assumes that the source domain and the target domain share the same label space which may not hold in several practical applications. This setting can be extended to the partial domain adaption, the open-set domain adaptation or the universal domain adaptation.
- CLAMP requires an access to the source domain samples, which might not be available due to the issue of privacy. In other words, our setting can be extended to the source-free case where no source domain samples are offered while performing domain adaptations.
- CLAMP is not applicable yet for the control system context. Recently, three exist some interesting works in this area. Song et al. [46] proposes an event-triggered state estimation for reaction–diffusion neural networks (RDNNs) subject to Denial-of-Service (DoS) attacks. Zhuang et al. [47] puts forward an optimal iterative learning control (ILC) algorithm for linear time-invariant multiple-input–multiple output (MIMO) systems with nonuniform trial lengths under input constraints. Wang et al. [48] proposes a Q-learning based fault estimation (FE) and fault tolerant control (FTC) scheme under ILC framework.

7. Conclusion

This paper presents the cross-domain continual learning problem calling for a model to overcome the issues of domain shift and catastrophic forgetting simultaneously. To this end, continual learning approach for many processes (CLAMP) is proposed where it is built upon the concept of dual assessors meta-weighting the multi-objective function such that the trade-off between the stability and the plasticity is achieved and the concept of class-aware adversarial domain adaptation dampening the gaps between the source domain and the target domain. Our experimental results confirm the advantage of CLAMP beating prior arts with significant margins. Moreover, our ablation study demonstrates the positive contributions of each learning module while our analysis of generalization show no over-fitting signals of CLAMP and CLAMP utilizes low memory budgets. Our future study is devoted to reduce the complexity of CLAMP where three networks have to be trained at once.

CRediT authorship contribution statement

Weiwei Weng: Writing – review & editing, Visualization, Validation, Software, Data curation. **Mahardhika Pratama:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Jie Zhang:** Supervision, Project administration, Funding acquisition. **Chen Chen:** Visualization, Validation, Software, Data curation. **Edward Yapp Kien Yie:** Supervision, Project administration, Funding acquisition. **Ramasamy Savitha:** Writing – review & editing, 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

https://github.com/wengweng001/CLAMP_torch.git.

Appendix A. Pseudo code

We provide the details of CLAMP in Algorithm 1.

Appendix B. Experimental task division

In this section, we provide details of the division for each subset in all benchmark datasets, namely, MNIST LeCun et al. [49], USPS Hull et al. [50], Office-31 Saenko et al. [35] and Office-Home Venkateswara et al. [36]. The division details are presented in Table 13. For the digit recognition case, which includes MNIST and USPS, the split step is 2. For Office-31, we sort the classes in

Algorithm 1: CLAMP.

Input: current task k, labelled source data $\mathcal{T}_k^S = \{x_i^S, y_i^S\}_{i=1}^{N_k^S}$, unlabelled target data $\mathcal{T}_k^T = \{x_i^T\}_{i=1}^{N_k^T}$, iteration numbers n_{inner}, n_{outer}, training epoch E, pseudo label selection threshold γ for target data, learning rate η and μ

Output: feature extractor f_θ, the classifier g_ϕ and the domain classifier ξ_ψ, source assessor network $\kappa_{\zeta_S}^S$, target assessor network $\kappa_{\zeta_T}^T$

Init: $M_S \leftarrow \emptyset$, $M_T \leftarrow \emptyset$ when k = 1 ; // Initialize episodic memories

$\mathcal{T}_{train_S}^k \leftarrow \mathcal{T}_k^S \cup M_S$;

for $e = 1, ..., E$ **do**

  Stage 1 Unsupervised Process Adaptation ; // $1 \le e \le E_{pa}$

  Update f_θ, ξ_ψ by $\mathcal{L}_{pa_{(x,y)\in\mathcal{T}_{train_S}^k \& (\mathcal{T}_k^T \cup M_T)}}$;

  Stage 2 Source Domain ; // $e > E_{pa}$

  for $iter_o = 1, ..., n_{outer}$ **do**

    Form validation set $\mathcal{T}_{val_S}^k$ by applying three random transformations to $\mathcal{T}_{train_S}^k$;

    for $iter_i = 1, ..., n_{inner}$ **do**

      Update $\kappa_{\zeta_S}^S$ based on $\zeta_S \leftarrow \zeta_S - \mu \nabla_{\zeta_S} \mathcal{L}_{S(x,y)\in\mathcal{T}_{val_S}^k}$; // μ: inner iteration learning rate

    end

    Update f_θ, g_ϕ by $\mathcal{L}_{CE(x,y)\in\mathcal{T}_{train_S}^k}$, $\mathcal{L}_{DER(x,y)\in M_S}$, $\mathcal{L}_{distill(x,y)\in M_S}$

  end

  Stage 3 Target Domain ; // $e > E_{pa}$

  Select pseudo-labelled target samples set $\mathcal{T}_k^{ps} = \{x^T, \widetilde{y}\}$;

  $\mathcal{T}_{train_{ps}}^k \leftarrow \mathcal{T}_k^{ps} \cup M_T$;

  for $iter_o = 1, ..., n_{outer}$ **do**

    Select validation set $\mathcal{T}_{val_T}^k$ based on process similar samples from $\mathcal{T}_{train_S}^k$;

    for $iter_i = 1, ..., n_{inner}$ **do**

      Update $\kappa_{\zeta_T}^T$ based on $\zeta_T \leftarrow \zeta_T - \mu \nabla_{\zeta_T} \mathcal{L}_{T(x,y)\in\mathcal{T}_{val_T}^k}$

    end

    Update f_θ, g_ϕ by $\mathcal{L}_{CE(x,y)\in\mathcal{T}_{train_{ps}}^k}$, $\mathcal{L}_{DER(x,y)\in M_T}$, $\mathcal{L}_{distill(x,y)\in M_T}$

  end

end

$M_S \leftarrow M_S \cup Sample(\mathcal{T}_k^S)$, $M_T \leftarrow M_T \cup Sample(\mathcal{T}_k^{ps})$; // Update episodic memory

Table 13
Class names in each task on all benchmarks.

Dataset	Task	Class Index	Class Name
MNIST & USPS	1	[0,1]	zero, one
	2	[2,3]	two, three
	3	[4,5]	four, five
	4	[6,7]	six, seven
	5	[8,9]	eight, nine
Office-31	1	[0,1,2,3,4,5]	back pack, bike, bike helmet, bookcase, bottle, calculator
	2	[6,7,8,9,10,11]	desk chair, desk lamp, desktop computer, file cabinet, headphones, keyboard
	3	[12,13,14,15,16,17]	laptop computer, letter tray, mobile phone, monitor, mouse, mug
	4	[18,19,20,21,22,23]	paper notebook, pen, phone, printer, projector, punchers
	5	[24,25,26,27,28,29,30]	ring binder, ruler, scissors, speaker, stapler, tape dispenser, trash can
Office-Home	1	[0,1,2,3,4]	Alarm Clock, Backpack, Batteries, Bed, Bike
	2	[5,6,7,8,9]	Bottle, Bucket, Calculator, Calendar, Candles
	3	[10,11,12,13,14]	Chair, Clipboards, Computer, Couch, Curtains
	4	[15,16,17,18,19]	Desk Lamp, Drill, Eraser, Exit Sign, Fan
	5	[20,21,22,23,24]	File Cabinet, Flipflops, Flowers, Folder, Fork
	6	[25,26,27,28,29]	Glasses, Hammer, Helmet, Kettle, Keyboard
	7	[30,31,32,33,34]	Knives, Lamp Shade, Laptop, Marker, Monitor
	8	[35,36,37,38,39]	Mop, Mouse, Mug, Notebook, Oven
	9	[40,41,42,43,44]	Pan, Paper Clip, Pen, Pencil, Postit Notes
	10	[45,46,47,48,49]	Printer, Push Pin, Radio, Refrigerator, Ruler
	11	[50,51,52,53,54]	Scissors, Screwdriver, Shelf, Sink, Sneakers
	12	[55,56,57,58,59]	Soda, Speaker, Spoon, TV, Table
	13	[60,61,62,63,64]	Telephone, ToothBrush, Toys, Trash Can, Webcam
VisDA	1	[0,1,2]	aeroplane, bicycle, bus
	2	[3,4,5]	car, horse, knife
	3	[6,7,8]	motorcycle, person, plant
	4	[9,10,11]	skateboard, train, truck

Table 14
Shared hyper-parameters for compared methods.

	MNIST ↔ USPS	Office-31	Office-Home	VisDA
Learning rate	{0.0001, **0.001**, 0.01}	{0.00001, *0.0001(others)*, **0.001(CLAMP)**, 0.01, 0.1}	{0.00001, *0.0001(others)*, **0.001(CLAMP)**, 0.01}	0.001
Memory size	50	10	5	10
Epoch	20	{10, **20**, 50}	{10, **20**}	10

alphabetic order and divide each domain into 5 tasks. The first four tasks include 6 classes each, and the last task includes 7 classes. Office-Home consists of 65 classes, and we divide them into 13 subsets of 5 classes per task. For VisDA, it is divided into 4 tasks of 12 classes in total.

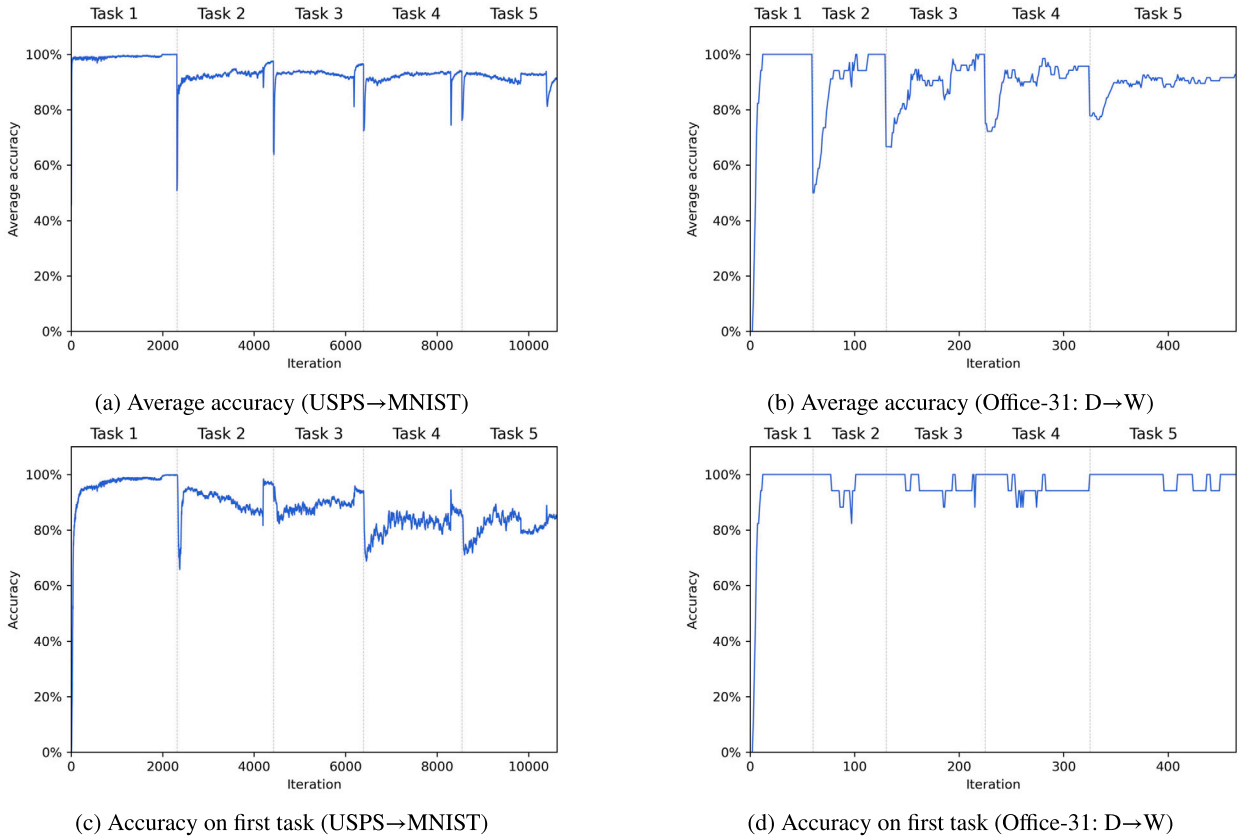
Appendix C. Model architecture

In CLAMP, we use LeNet with one additional linear layer of 100 units for MNIST↔USPS, ResNet-34 with one additional linear layer of 512 units for Office-31 and VisDA, and ResNet-50 with one additional linear layer of 1000 units for Office-Home as the backbone network of the main network. For the assessor network, we use a multi-layer perceptron with 2 hidden-layers of 256 nodes, followed by a two-layer LSTM with 64 units, and then connect to 2 linear-layers with 64 and 3 nodes each for MNIST↔USPS. As for Office-31, Office-Home and VisDA, we adopt ResNet-18 as part of the assessor network, followed by a two-layer LSTM with 128 units, and then 2 linear-layers with 128 and 3 nodes each. All ResNet models used in this work are pre-trained on ImageNet.

Appendix D. Hyper-parameter selection

This section outlines the hyper-parameters choices for the various experiments conducted in this work. To enhance clarity, we use the same hyper-parameters descriptions and notation as in the corresponding original papers, and similarly to our proposed method. Below, we provide the hyper-parameters values or grid, and mark the best results we adopt in bold. Table 14 presents the shared hyper-parameters across different methods that are not specifically mentioned below. We use the SGD optimizer with weight decay and momentum set to 0.9 and 0.0005, respectively, for all methods discussed in this work. For CLAMP, we select pseudo target labels that meet $\hat{y}^T \geq 0.85$ to train the target domain.

- CLAMP
  - learning rate: { 0.01, **0.001**, 0.002, 0.0001 } (MNIST↔ USPS), { 0.00001, **0.0001**, 0.0002, 0.001, 0.002, 0.01, 0.1} (Office-31), { 0.01, 0.001, **0.0001**, 0.00001 } (Office-Home), **0.001** (VisDA)
  - epoch: {5, 10, 15, **20**, 25, 50} (MNIST↔ USPS), {2, 3, 5, **10**, 11, 13, 15, 20, 25, 30, 40, 50} (Office-31), { 5, **10**, 15, 20, 25 } (Office-Home)
  - process adaptation epoch: { 3, 5, **10** } (MNIST↔ USPS), { 1, 3, **5**, 7, 8, 10, 15, 20, 25 } (Office-31), { 1, 3, **5**, 10, 15 } (Office-Home)
  - memory: { 5, 10, 25, **50**, 100, 200, 250 } (MNIST↔ USPS), { 1, 3, 5, **10**, 15 } (Office-31), { 1, 3, **5**, 10, 15 }
  - pseudo label selection threshold γ: 0.85
- EWC
  - λ: 5000
- LwF
  - T: 2.0
  - regularization R: 1.0
- SI
  - strength parameter c: 0.1
  - damping parameter ξ: 0.1
- MAS
  - regularization λ: 1.0
- RWalk
  - regularization λ: 1.0
- EEIL
  - epochs for balanced fine-tuning: 30
  - distillation parameter T: 2.0
- HAL
  - regularization λ: 0.1
  - decay rate β: 0.5
  - mean embedding strength γ: 0.1
- AGLA
  - Inner epoch: 1
  - Outer epoch: 1



(a) Average accuracy (USPS→MNIST)

(b) Average accuracy (Office-31: D→W)

(c) Accuracy on first task (USPS→MNIST)

(d) Accuracy on first task (Office-31: D→W)

Fig. 7. (a) Average query accuracy for all learned tasks throughout the training phase in USPS→MNIST; (b) Average query accuracy for all learned tasks throughout the training in Office-31: D→W; (c) Query accuracy on the first task during the training phase in USPS→MNIST; (d) Query accuracy on the first task during the training phase on Office-31: D→W.

Table 15
Accuracy (%) of CLAMP with different exemplar sizes per class on MNIST↔USPS.

Exemplar per class	MNIST → USPS	USPS → MNIST
5	71.93 ± 4.2	84.40 ± 0.9
25	81.69 ± 1.1	89.35 ± 1.1
50 (ours)	**84.98 ± 1.3**	**89.63 ± 1.0**
100	84.82 ± 1.5	89.09 ± 0.6
250	85.02 ± 4.0	88.33 ± 3.0

- DANN
  - adaptation parameter λ: 1.0

Appendix E. Additional experimental results

In this section, we evaluate the ability of our method in continual learning manner. We report the query accuracy of all learned tasks as the training progresses across tasks from Task 1 to Task 5 on two cases, USPS→MNIST (Fig. 7a) and Office-31 D→W (Fig. 7b). Fig. 7c and Fig. 7d show the query accuracy on the first task along with the training progresses across tasks from Task 1 to Task 5 on USPS→MNIST and Office-31 D→W respectively.

Appendix F. Effect of the episodic memory size

We explore the effect of episodic memory size on the performance of CLAMP. Table 15 shows additional results on the digit recognition experiment. We can see that the accuracy increases along with the increase in the episodic memory size until it reaches 50 exemplars per class. After that, the performance drops when expanding the size from 50 to 100. For digit recognition (MNIST→USPS), the accuracy approaches the highest of 85.02% when the exemplar size is 250. Considering computational efficiency and optimal performance, we opt for 50 exemplars per class as the episodic memory size for the digit recognition case. This analysis underscores

the delicate balance between memory size and performance, highlighting the need for episodic memory management in continual learning across many processes problem.

References

[1] J. Kirkpatrick, R. Pascanu, N.C. Rabinowitz, J. Veness, G. Desjardins, A.A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, R. Hadsell, Overcoming catastrophic forgetting in neural networks, Proc. Natl. Acad. Sci. 114 (2016) 3521–3526, https://api.semanticscholar.org/CorpusID:4704285.

[2] F. Mao, W. Weng, M. Pratama, E.K.Y. Yapp, Continual learning via inter-task synaptic mapping, arXiv:2106.13954 [abs], 2021, https://doi.org/10.1016/j.knosys.2021.106947, https://api.semanticscholar.org/CorpusID:233710658.

[3] A.A. Rusu, N.C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, R. Hadsell, Progressive neural networks, arXiv:1606.04671 [abs], 2016, https://doi.org/10.48550/arXiv.1606.04671, https://api.semanticscholar.org/CorpusID:15350923.

[4] M. Pratama, A. Ashfahani, E.D. Lughofer, Unsupervised continual learning via self-adaptive deep clustering approach, in: CSSL, 2021, https://api.semanticscholar.org/CorpusID:235658071.

[5] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C.H. Lampert, icarl: incremental classifier and representation learning, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 5533–5542, https://api.semanticscholar.org/CorpusID:206596260.

[6] H. Shin, J.K. Lee, J. Kim, J. Kim, Continual learning with deep generative replay, in: Neural Information Processing Systems, 2017, https://api.semanticscholar.org/CorpusID:1888776.

[7] S. Chandra, A. Haque, L. Khan, C.C. Aggarwal, An adaptive framework for multistream classification, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 2016, https://api.semanticscholar.org/CorpusID:16830931.

[8] Q. Lao, X. Jiang, M. Havaei, Y. Bengio, A two-stream continual learning system with variational domain-agnostic feature replay, IEEE Trans. Neural Netw. Learn. Syst. 33 (2021) 4466–4478, https://api.semanticscholar.org/CorpusID:232113812.

[9] H. Lin, Y. Zhang, Z. Qiu, S. Niu, C. Gan, Y. Liu, M. Tan, Prototype-guided continual adaptation for class-incremental unsupervised domain adaptation, https://doi.org/10.48550/arXiv.2207.10856, https://api.semanticscholar.org/CorpusID:251018211, 2022.

[10] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V.S. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. (2015), https://doi.org/10.48550/arXiv.1505.07818, https://api.semanticscholar.org/CorpusID:2871880.

[11] S. Li, W. Ma, J. Zhang, C.H. Liu, J. Liang, G. Wang, Meta-reweighted regularization for unsupervised domain adaptation, IEEE Trans. Knowl. Data Eng. 35 (2023) 2781–2795, https://api.semanticscholar.org/CorpusID:240536699.

[12] W. Zheng, J. Lu, J. Zhou, Deep metric learning via adaptive learnable assessment, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 2957–2966, https://api.semanticscholar.org/CorpusID:219631345.

[13] M.V. de Carvalho, M. Pratama, J. Zhang, C. Haoyan, E.K.Y. Yapp, Towards cross-domain continual learning, https://doi.org/10.48550/arXiv.2402.12490, https://api.semanticscholar.org/CorpusID:267759608, 2024.

[14] M.A. Ma'sum, M. Pratama, E.D. Lughofer, W. Ding, W. Jatmiko, Assessor-guided learning for continual environments, Inf. Sci. 640 (2023) 119088, https://doi.org/10.48550/arXiv.2303.11624, https://api.semanticscholar.org/CorpusID:257636622.

[15] A. Ashfahani, M. Pratama, Unsupervised continual learning in streaming environments, IEEE Trans. Neural Netw. Learn. Syst. 34 (2021) 9992–10003, https://api.semanticscholar.org/CorpusID:237572299.

[16] A. Rakaraddi, S.-K. Lam, M. Pratama, M.V. de Carvalho, Reinforced continual learning for graphs, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, https://api.semanticscholar.org/CorpusID:252089650.

[17] M.V. de Carvalho, M. Pratama, J. Zhang, Y. San, Class-incremental learning via knowledge amalgamation, in: ECML/PKDD, 2022, https://api.semanticscholar.org/CorpusID:252090326.

[18] T. Dam, M. Pratama, M.M. Ferdaus, S.G. Anavatti, H. Abbas, Scalable adversarial online continual learning, in: ECML/PKDD, 2022, https://api.semanticscholar.org/CorpusID:252089827.

[19] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J.G. Dy, T. Pfister, Learning to prompt for continual learning, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 139–149, https://api.semanticscholar.org/CorpusID:245218925.

[20] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, 2017, https://api.semanticscholar.org/CorpusID:6719686.

[21] J. Schmidhuber, Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-. hook, https://api.semanticscholar.org/CorpusID:264351059, 1987.

[22] K. Javed, M. White, Meta-learning representations for continual learning, arXiv:1905.12588 [abs], 2019, https://doi.org/10.48550/arXiv.1905.12588, https://api.semanticscholar.org/CorpusID:168169908.

[23] G. Gupta, K. Yadav, L. Paull, La-maml: look-ahead meta learning for continual learning, Adv. Neural Inf. Process. Syst. 33 (2020) 11588–11598, https://api.semanticscholar.org/CorpusID:220831405.

[24] Q.H. Pham, C. Liu, D. Sahoo, S.C.H. Hoi, Contextual transformation networks for online continual learning, in: International Conference on Learning Representations, 2021, https://api.semanticscholar.org/CorpusID:235614405.

[25] S. Chandra, A. Haque, L. Khan, C.C. Aggarwal, An adaptive framework for multistream classification, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 2016, https://api.semanticscholar.org/CorpusID:16830931.

[26] A. Haque, Z. Wang, S. Chandra, B. Dong, L. Khan, K.W. Hamlen, Fusion: an online method for multistream classification, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, https://api.semanticscholar.org/CorpusID:25322234.

[27] M. Pratama, M.V. de Carvalho, R. Xie, E.D. Lughofer, J. Lu, Atl: autonomous knowledge transfer from many streaming processes, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, https://api.semanticscholar.org/CorpusID:203902308.

[28] H. Du, L.L. Minku, H. Zhou, Multi-source transfer learning for non-stationary environments, in: 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1–8, https://api.semanticscholar.org/CorpusID:57721122.

[29] H. Du, L.L. Minku, H. Zhou, Marline: multi-source mapping transfer learning for non-stationary environments, in: 2020 IEEE International Conference on Data Mining (ICDM), 2020, pp. 122–131, https://api.semanticscholar.org/CorpusID:222262850.

[30] R. Xie, M. Pratama, Automatic online multi-source domain adaptation, arXiv:2109.01996 [abs], 2021, https://doi.org/10.48550/arXiv.2109.01996, https://api.semanticscholar.org/CorpusID:237420507.

[31] M.V. de Carvalho, M. Pratama, J. Zhang, E.K.Y. Yapp, Acdc: online unsupervised cross-domain adaptation, Knowl.-Based Syst. 253 (2021) 109486, https://api.semanticscholar.org/CorpusID:238259903.

[32] W. Weng, M. Pratama, C. Za'in, M.V. de Carvalho, A. Rakaraddi, A. Ashfahani, E.K.Y. Yapp, Autonomous cross domain adaptation under extreme label scarcity, IEEE Trans. Neural Netw. Learn. Syst. 34 (2022) 6839–6850, https://api.semanticscholar.org/CorpusID:249955297.

[33] P. Buzzega, M. Boschini, A. Porrello, D. Abati, S. Calderara, Dark experience for general continual learning: a strong, simple baseline, arXiv:2004.07211 [abs], 2020, https://doi.org/10.48550/arXiv.2004.07211, https://api.semanticscholar.org/CorpusID:215768806.

[34] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F.C. Pereira, J.W. Vaughan, A theory of learning from different domains, Mach. Learn. 79 (2010) 151–175, https://api.semanticscholar.org/CorpusID:8577357.

[35] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in: European Conference on Computer Vision, 2010, https://api.semanticscholar.org/CorpusID:7534823.
[36] H. Venkateswara, J. Eusébio, S. Chakraborty, S. Panchanathan, Deep hashing network for unsupervised domain adaptation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5385–5394, https://api.semanticscholar.org/CorpusID:2928248.
[37] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, K. Saenko, Visda: the visual domain adaptation challenge, arXiv preprint, arXiv:1710.06924, 2017, https://doi.org/10.48550/arXiv.1710.06924, https://api.semanticscholar.org/CorpusID:28698351.
[38] Z. Li, D. Hoiem, Learning without forgetting, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2016) 2935–2947, https://api.semanticscholar.org/CorpusID:4853851.
[39] F. Zenke, B. Poole, S. Ganguli, Continual learning through synaptic intelligence, Proc. Mach. Learn. Res. 70 (2017) 3987–3995, https://api.semanticscholar.org/CorpusID:10409742.
[40] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, T. Tuytelaars, Memory aware synapses: learning what (not) to forget, https://doi.org/10.48550/arXiv.1711.09601, 2018, pp. 139–154, https://api.semanticscholar.org/CorpusID:4254748.
[41] A. Chaudhry, P.K. Dokania, T. Ajanthan, P.H.S. Torr, Riemannian walk for incremental learning: understanding forgetting and intransigence, https://doi.org/10.48550/arXiv.1801.10112, https://api.semanticscholar.org/CorpusID:4047127, 2018.
[42] E. Belouadah, A.D. Popescu, Il2m: class incremental learning with dual memory, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 583–592, https://api.semanticscholar.org/CorpusID:204923710.
[43] F.M. Castro, M.J. Marín-Jiménez, N.G. Mata, C. Schmid, A. Karteek, End-to-end incremental learning, in: European Conference on Computer Vision, 2018, https://api.semanticscholar.org/CorpusID:50785377.
[44] A. Chaudhry, A. Gordo, P.K. Dokania, P.H.S. Torr, D. Lopez-Paz, Using hindsight to anchor past knowledge in continual learning, in: AAAI Conference on Artificial Intelligence, 2021, https://api.semanticscholar.org/CorpusID:210957697.
[45] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, B. Wang, Moment matching for multi-source domain adaptation, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2018, pp. 1406–1415, https://api.semanticscholar.org/CorpusID:54458071.
[46] X. Song, N. Wu, S. Song, V. Stojanovic, Switching-like event-triggered state estimation for reaction–diffusion neural networks against dos attacks, Neural Process. Lett. 55 (2023) 8997–9018, https://api.semanticscholar.org/CorpusID:257503064.
[47] Z. Zhuang, H. Tao, Y. Chen, V. Stojanovic, W. Paszke, An optimal iterative learning control approach for linear systems with nonuniform trial lengths under input constraints, IEEE Trans. Syst. Man Cybern. Syst. 53 (2023) 3461–3473, https://api.semanticscholar.org/CorpusID:254613055.
[48] R. Wang, Z. Zhuang, H. Tao, W. Paszke, V. Stojanovic, Q-learning based fault estimation and fault tolerant iterative learning control for mimo systems, ISA Trans. 142 (2023) 123–135, https://www.sciencedirect.com/science/article/pii/S0019057823003506.
[49] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (1998) 2278–2324, https://api.semanticscholar.org/CorpusID:14542261.
[50] J.J. Hull, A database for handwritten text recognition research, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1994) 550–554, https://doi.org/10.1109/34.291440, https://api.semanticscholar.org/CorpusID:8148915.