Complementary Domain Adaptation and Generalization for Unsupervised Continual Domain Shift Learning

Wonguk Cho¹, Jinha Park², and Taesup Kim¹
¹Graduate School of Data Science, ²Department of Electrical and Computer Engineering
Seoul National University
{wongukcho, jhpark410, taesup.kim}@snu.ac.kr

Abstract

Continual domain shift poses a significant challenge in real-world applications, particularly in situations where labeled data is not available for new domains. The challenge of acquiring knowledge in this problem setting is referred to as unsupervised continual domain shift learning. Existing methods for domain adaptation and generalization have limitations in addressing this issue, as they focus either on adapting to a specific domain or generalizing to unseen domains, but not both. In this paper, we propose Complementary Domain Adaptation and Generalization (CoDAG), a simple yet effective learning framework that combines domain adaptation and generalization in a complementary manner to achieve three major goals of unsupervised continual domain shift learning: adapting to a current domain, generalizing to unseen domains, and preventing forgetting of previously seen domains. Our approach is model-agnostic, meaning that it is compatible with any existing domain adaptation and generalization algorithms. We evaluate CoDAG on several benchmark datasets and demonstrate that our model outperforms state-of-the-art models in all datasets and evaluation metrics, highlighting its effectiveness and robustness in handling unsupervised continual domain shift learning.

1. Introduction

Machine learning algorithms have found extensive applications in various fields such as image recognition [19, 23], natural language processing [3, 9], and autonomous driving [5, 10]. Typically, these algorithms learn from a training dataset to build a model that performs a target task on new data. The assumption underlying these algorithms is that the training data and the test data are identically and independently distributed (IID) [18], drawn from the same distribution that is characterized by an environment or domain. However, this IID assumption is often not valid in real-world scenarios, as the environment in which the model is applied is more likely to change over time than remain fixed. This implies that the model will encounter new data from various domains over time, and its performance may decline if the data is from a domain that differs significantly from the one it was trained on.

To address this issue, two main approaches have been developed: domain generalization and domain adaptation. Domain generalization [60] is a method to enhance a model’s ability to generalize to unseen domains by training the model on labeled data from one or more domains, without assuming any prior knowledge of the test environment in which the model will be applied. However, collecting a large volume of labeled data from various domains for a particular task can be challenging in practice. Single-source domain generalization techniques [46] have been developed to address this issue, which rely solely on data from a single domain. Although more practical, these techniques generally have lower generalization abilities compared to multi-source domain generalization techniques.
On the other hand, domain adaptation [61] aims to enhance model performance only on the current target domain and does not prioritize performance on all other domains. In particular, unsupervised domain adaptation (UDA) [44] techniques leverage unlabeled data to adapt the model to a new target domain. However, domain adaptation methods fundamentally suffer from performance degradation on a new target domain before and during the adaptation process due to the lack of inherent mechanism to prepare for unseen domains.

In this paper, we tackle a challenging problem that simulates real-world scenarios where models face continual domain shifts and no labeled data is available for new domains. We refer to this problem as unsupervised continual domain shift learning. In this setting, the model must continually adapt to new domains (domain adaptation), while maintaining its generalization ability for upcoming and unseen domains (domain generalization), in an unsupervised manner. However, achieving both objectives simultaneously is not always feasible since they involve related but distinct goals. For instance, if the current target domain is vastly dissimilar from any other domains, none of the optimal solutions for adapting to the current target domain with DA would necessarily result in optimal generalization for performing well on other domains. Similarly, achieving optimal generalization for unseen domains through DG may not result in the best solution for the current target domain. Therefore, to address unsupervised continual domain shift learning, it is necessary to find a solution that resolves this trade-off between domain adaptation and generalization.

To address the trade-off between domain adaptation and generalization, we propose Complementary Domain Adaptation and Generalization (CoDAG), a learning framework that combines domain adaptation and domain generalization in a complementary manner. As shown in Fig. 1, our approach involves training two separate models: one for domain adaptation and the other for domain generalization. We use the domain adaptation model to adapt to the target domain, generating more accurate and reliable pseudo-labels for training the domain generalization model. In turn, the domain generalization model learns more generalized representations across multiple domains and provides the domain adaptation model with initializing parameters, enhancing its adaptability to a new domain. As a result, the domain adaptation and generalization models complement each other in our framework, leading to improved performance for both.

The main contribution of our framework, CoDAG, lies in the complementary manner in which we leverage existing domain adaptation and domain generalization methods to address unsupervised continual domain shift learning, a unique and challenging problem that has not been thoroughly explored. We deliberately apply existing methods to our framework, rather than introducing new ones, to underscore that the effectiveness of our framework is due to its complementary structure, not its individual components. Indeed, without requiring any models tailored for the present problem, our framework proved its merit by achieving SoTA performance against all baselines, including the one which is explicitly designed for this setting [37].

Finally, it is important to note that our work is one of the first attempts to explore the potential synergies between domain adaptation and domain generalization methods. We are breaking new ground by bridging the divide between the disparate fields of domain adaptation and generalization, which were primarily studied independently. This paradigm shift represents not just a novel approach, but one with profound practical implications.

Our contributions can be summarized as follows:

- We introduce a novel framework that combines domain adaptation and generalization models in a complementary manner, resulting in a synergistic process that enhances overall performance.
- Our method consistently outperforms state-of-the-art models across all datasets and metrics, demonstrating superior robustness with the lowest standard deviation across different orders in almost all cases.
- Our approach does not necessitate the use of models designed for the present problem, allowing seamless integration with existing domain adaptation and generalization algorithms for broader applications.

2. Related Work

Domain generalization Domain generalization (DG) is the process of training a model using labeled data from one or multiple domains, with the objective of achieving good generalization performance across unseen domains. Existing DG methods are based on domain-invariant learning [13, 16, 15, 29, 32, 40], meta-learning [2, 11, 26, 28], and data augmentation [52, 67]. To address practical scenarios, single-source DG methods [46, 50, 58, 65, 66] have been proposed, which use labeled data collected from a single domain. However, these techniques require a large amount of labeled data and can suffer from severe catastrophic forgetting when applied to scenarios with continual domain shift.

Unsupervised domain adaptation Unsupervised domain adaptation (UDA) [44] aims to improve the performance of a target model in scenarios where there is a domain shift between the labeled source domain and the unlabeled target domain. UDA methods often achieve distribution alignment through domain invariant feature transformation [34, 39, 43] or feature space alignment [12, 17, 55].
Typical domain adaptation techniques generally assume having access to the source data throughout the adaptation process, which is impractical in real-world scenarios. However, source-free domain adaptation (SFDA) methods [35, 36] adopt a model to the unlabeled target domain even when the source dataset is not available during the target adaptation process. Recent works on SFDA [1, 7, 31, 38, 63] have arisen, which use generative models to model the distribution of target data by creating pseudo-label refinement [1, 7], target-style images [31, 38], or variational inference for generating latent source features [63]. Nevertheless, these techniques are not intended to accumulate the knowledge acquired from continually shifting domains.

**Continual domain adaptation** Continual learning (CL) [8] focuses on avoiding catastrophic forgetting when learning new tasks by using regularization-based methods [53, 64] and replay-based methods [47]. Recent works [4, 51, 56] have adopted the ideas from continual learning to tackle the continual domain adaptation (CDA) problem. These methods involve distilling probability distributions at multiple levels from the previous models to solve the catastrophic forgetting problem [51], using sample replay buffers along with domain adversarial training [4], and utilizing a domain-specific memory buffer for each domain [56]. Despite the use of CDA methods, the model’s performance on the target domain is often poor before and during the adaptation process, although it may improve after sufficient adaptation on the new target domain. This can be particularly problematic when there is a significant domain shift. Our complementary framework overcomes this issue by leveraging the DG model’s generalization ability to initialize the DA model.

**3. Methodology**

**3.1. Unsupervised continual domain shift learning**

We adopt the problem setting introduced by [37] to address the challenge of unsupervised continual domain shift learning. Specifically, our approach works with \( T + 1 \) distinct domains \( D_t \) over \( t = 0, 1, \ldots, T \) for a given target task, which are sequentially encountered. Here, we tackle the \( K \)-way image classification problem as the target task, and the image space \( \mathcal{X} \) and the label space \( \mathcal{Y} \) are shared across all domains. For the sake of notational simplicity, here we use the notation \( D_t \) to denote both the \( t \)-th domain and the dataset sampled from it interchangeably.

In this setting, the first domain \( D_0 \) is regarded as the source domain \( S \) that is used to initially learn the target task. Different from other domains, it consists of labeled samples \( S = D_0 = \{(x_0^{(i)}, y_0^{(i)})\}_{i=1}^{N_0} \), where \( x_0^{(i)} \) and \( y_0^{(i)} \) denote input data and its corresponding label of the \( i \)-th sample and \( N_0 \) indicates the total number of samples in the source domain. After a model is initially trained with the source domain, it sequentially encounters a series of target domains \( T = \{D_1, D_2, \ldots, D_T\} \). In contrast to the source domain, the target domains do not provide any label information to the model, as we assume that the model encounters them after the deployment. Moreover, we consider more realistic settings that samples from previously experienced domains are not fully accessible but are partially available via a replay buffer with a limited capacity. Therefore, the model has to properly improve its generalization ability in an unsupervised manner by mainly using unlabeled samples \( D_t = \{x_t^{(i)}\}_{i=1}^{N_t} \) from the target domain \( D_t \) at each stage \( t \).

To deal with this problem setting, a model \( f(x; \theta) \) is used to map an input \( x \in \mathcal{X} \) to its corresponding label \( y \in \mathcal{Y} \), where \( \theta \) represents the set of model parameters to be learned. The model is expected to be initially trained with samples from the source domain \( S \) and then adapted to the target domains \( T \) in a continual manner (i.e., from \( D_1 \) to \( D_T \)). We denote the model parameters after training on the \( t \)-th domain as \( \theta_t^* \) and evaluate the corresponding model \( f(x; \theta_t^*) \) with \( x \) from different domains. There exist three main goals for this problem setting. (i) Firstly, we seek to achieve domain adaptation for the target domain \( D_t \in T \). (ii) Secondly, we aim to achieve domain generalization for
unseen target domains \(\{D_t\}_{1 \leq t \leq T}\). (iii) Finally, we aim to prevent catastrophic forgetting of knowledge gained from previously seen domains \(\{D_t\}_{0 \leq t < 1}\).

Most of the existing models for domain adaptation or generalization, which are able to handle unlabeled data, can be applied for unsupervised continual domain shift learning. A naive approach is to start with the model \(f(x; \theta_{t-1}^*\)) trained on domain \(D_{t-1}\) and update it with data from \(D_t\), resulting in a new model \(f(x; \theta_t^*)\) that is either adapted to \(D_t\) or generalized for unseen domains \(D_t\) for \(t' > t\), depending on its designed purpose. All the baselines used in our experiments reflect this naive approach.

However, achieving both DA and DG objectives in unsupervised continual domain shifts is challenging with a naive single-model approach, given the potential contradiction between adapting to a target domain (DA) and learning domain-invariant features (DG). For instance, when the current target domain is significantly different from any other domains, optimal solutions for domain adaptation may not lead to optimal generalization for other domains. Similarly, using domain generalization to attain optimal generalization for unseen domains does not necessarily yield the best solution for the current target domain.

### 3.2. CoDAG: Complementary Domain Adaptation and Generalization

To resolve this trade-off between DA and DG, we propose a novel framework called Complementary Domain Adaptation and Generalization (CoDAG). CoDAG is highly effective yet remarkably simple. Unlike other approaches that require complicated architectures or learning methods, our CoDAG framework relies on a straightforward and intuitive idea: maintaining two separate models for DA and DG. By having two distinct models optimized for their respective goals, the CoDAG framework eliminates the need for complicated techniques that attempt to balance DA and DG within a single model. Instead, our framework facilitates a synergistic relationship between the two models, where they complement each other, leading to improved performance for both.

To carry out this framework, DA and DG are conducted in an interleaved order with two separate models for DA and DG, and make each of them to be dependent on the other, as shown in Fig. 2. Specifically, we duplicate the model \(f\) to use one for DA, denoted as \(f_{DA}\), and the other for DG, denoted as \(f_{DG}\), with each model trained with DA and DG algorithms, respectively. This setting allows the models to effectively exchange knowledge and continually improve the performance for both.

As domain shifts occur continuously and there is no labeled information available from target domains, nor access to data from previously encountered domains, we have implemented a source-free unsupervised domain adaptation algorithm for our DA model. There are several options for this, but we choose a simple pseudo-labeling [25] along with regularization from SHOT [35] due to its simplicity. For the DG model, we simply apply an empirical risk minimization (ERM) method along with a RandMix augmentation technique [37] that generates diversified data, which boost the DG model’s ability to generalize to unseen domains. However, note that our proposed framework is model-agnostic, in the sense that it can be applied to any DA or DG algorithms that are suitable for the given problem setting. The implementation details of auxiliary methods and algorithms employed in this paper are provided in the Supplementary Material.

#### Source domain training

To initially learn a given target task, we train the DG model from scratch with the set of labeled samples from the source domain \(S (= D_0)\). This stage can be approached by a single-source domain generalization method. In that way, we simply minimize the following cross-entropy loss over the source domain data with enhanced data augmentation,

\[
L_{DG,0}(D_0) = \mathbb{E}_{(x,y) \in D_0} [L_{ce}(f_{DG}(R(x); \theta_{DG,0}), y)], \tag{1}
\]

where \(\theta_{DG,0}\) is the DG model parameters, \(L_{ce}\) is the cross-entropy loss for a classification setting, and \(R\) represents the RandMix augmentation [37].

#### Generalized initialization with DG for DA

In general, unsupervised source-free domain adaptation approaches involve initially training a source model with the source domain data and further updating the source model with unlabeled data from a new target domain. However, in our proposed framework, the DA model utilizes the parameters of the previous DG model for its initialization. This allows the DA model to leverage the DG model’s generalization ability to learn domain-invariant features and reduce domain-specific factors. As a result, we achieve efficient adaptation to a new target domain, even when there is a large gap between previously experienced domains and the new target domain (see Section 4.2 for experimental results).

To apply this approach for the current target domain \(D_t\), we first initialize the DA model \(f_{DA}\) with the parameters of the DG model trained with previously experienced domains, or \(\theta_{DG,t-1}^*\), treating it as a source model. Then, we freeze the classifier head in \(f_{DA}\) and only update the feature extractor part of it using information maximization and self-supervised pseudo-labeling with data from the current target domain \(D_t\). Accordingly, the loss to adapt to \(D_t\) is written as,

\[
L_{DA,t}(D_t) = \mathbb{E}_{x \in D_t} [L_{shot}(f_{DA}(x; \theta_{DA,t}))], \tag{2}
\]

where \(\theta_{DA,t}\) is the parameters of the DA model initialized with the optimal parameters of the DG model trained on the
previous domain $D_{t-1}$, or $\theta^{*}_{\text{DG},t-1}$, and $L_{\text{shot}}$ is the cross-entropy loss with pseudo-labels on target predictions along with regularization from SHOT [35].

**Pseudo-label generation with DA for DG** In our proposed framework, we simply use an empirical risk minimization (ERM) method along with an enhanced data augmentation method for training the DG model. Although labeled samples are necessary for this process, none of the target domains $T$ provide any labels. Thus, to make use of unlabeled samples from the current target domain $D_t$, we adopt a pseudo-label generation strategy [25] based on the highest prediction confidence of the DA model adapted to $D_t$. This involves applying the DA model to the unlabeled samples from $D_t$ to generate pseudo-labels, which are then used as training labels for the DG model on $D_t$.

Specifically, for each unlabeled sample $x$, we compute its pseudo-label $\hat{y}(x)$ as follows:

$$\hat{y}(x) = \arg\max_k \delta_k(f_{\text{DA}}(x; \theta^{*}_{\text{DA},t})), \quad (3)$$

where $\theta^{*}_{\text{DA},t}$ is obtained by optimizing $L_{\text{DA},t}$ in Eq. 2 and $\delta_k(\cdot)$ is the $k$-th element of a softmax output. By using the resulting pseudo-labels as training labels for the DG model on $D_t$, we construct a pseudo-labeled dataset $\tilde{D}_t = \{(x_t^{(i)}, \hat{y}(x_t^{(i)}))\}_{i=1}^{N_t}$. We then update the DG model with ERM in the same way as the source training in Eq. 1 with the following loss:

$$L_{\text{DG},t}(\tilde{D}_t) = \mathbb{E}_{(x,y) \in \tilde{D}_t} [L_{\text{ce}}(f_{\text{DG}}(R(x); \theta_{\text{DG},t}), y)], \quad (4)$$

where $\theta_{\text{DG},t}$ is initialized with the optimal parameters of the DG model trained on the previous domain $D_{t-1}$, or $\theta^{*}_{\text{DG},t-1}$.

**Learning from noisy labels** As the DA model is optimized to adapt to the current domain $D_t$, we assume that it can generate high-quality pseudo-labels. However, some of the labels may still contain errors due to the imperfectness of the DA method. Unfortunately, the errors in pseudo-labels (i.e., noisy labels [54]) can negatively affect the performance of the DG model. To alleviate this problem, we use an algorithm that can properly handle noisy labels to prevent the performance degradation of the DG model.

In this paper, we adopt an algorithm called Selective Negative Learning and Positive Learning (SelNLPL) [22] to reduce the risk of overfitting to noisy labels and improve performance of the DG model. We show the effectiveness of this approach in Sec. 4.2. Note that our approach offers greater flexibility that is not restricted to SelNLPL but rather can leverage any label noise-resilient methods [33, 48, 57].

**Forgetting alleviation** Forgetting alleviation is crucial in unsupervised continual domain shift learning, where limited access to data from previous domains makes it challenging to maintain performance on previous tasks. When encountering a new target domain, the performance of the DG model on previous domains tends to degrade, which is commonly referred to as catastrophic forgetting in continual learning.

To address this issue, we add a simple distillation loss term [20] $L_{\text{distill}}$ to the loss of the DG model in Eq. 4 to ensure that the model retains the knowledge gained from previous domains $\{D_t\}_{0 \leq t < T}$ while learning from the current target domain $D_t$, given by

$$L_{\text{distill}}(D_t) = \mathbb{E}_{x \in D_t} \left[ L_{\text{kl}}(q_t(x) \parallel p_t(x)) \right], \quad (5)$$

where $L_{\text{kl}}$ represents the KL divergence loss, and $q_t(x) = \delta(f_{\text{DG}}(R(x); \theta^{*}_{\text{DG},t-1}))$ and $p_t(x) = \delta(f_{\text{DG}}(R(x); \theta_{\text{DG},t}))$ are the predicted softmax probabilities from the previous and current DG models, respectively.

Another method we employ to prevent catastrophic forgetting is a replay buffer $M_t$ of size $M \ll N_t$ [49], which contains selected samples from previously experienced domains $\{D_t\}_{0 \leq t < T}$. We build the replay buffer based on the iCaRL approach [37, 47]. By using a replay buffer, along with the samples from $D_t$, additional $M$ selected samples from $D_0 \cup \{D_t\}_{1 \leq t < T}$ are available for training the DG model on $D_t$. This allows the DG model to not only improve its generalization ability but also prevent catastrophic forgetting. Then, our final loss to update the DG model on the current target domain $D_t$ is given by

$$L_{\text{DG},t}(\bar{D}_t) = L_{\text{DG},t}(\tilde{D}_t) + \alpha \cdot L_{\text{distill}}(\bar{D}_t), \quad (6)$$

where $\bar{D}_t = \tilde{D}_t \cup M_t$ represents $N_t + M$ samples available from the $t$-th domain with a replay buffer and $\alpha$ is a balancing hyperparameter.

Our findings suggest that utilizing a replay buffer leads to a significant improvement in performance, especially when dealing with previous domains. However, even without a replay buffer, our experiment shows that our model remains competitive against state-of-the-art models that are equipped with replay buffers (see Sec. 4.2 for experimental results).

**4. Experiments**

To validate the effectiveness of our framework, we compare our proposed framework, CoDAG, against state-of-the-art methods on three benchmark datasets: (i) PACS [27], (ii) Digits-five [14, 21, 24, 41], and (iii) DomainNet [45]. For a fair comparison, we adhere to the experimental setup from [37] in the following sections.

**Datasets** PACS consists of 4 distinct domains with 7 classes, including Photo (P), Art painting (A), Cartoon
Table 1. Comparison of the performance on the PACS, Digits-five, and DomainNet datasets for different state-of-art methods in TDA, TDG, FA, and All. The results are averaged over 10 different orders from each dataset. The results of the baseline models are referenced from [37]. The best results are highlighted in bold. Our method outperforms all the baselines across all datasets and evaluation metrics tested.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>SHOT+</th>
<th>SHOT++</th>
<th>Tent</th>
<th>AdaCon</th>
<th>EATA</th>
<th>L2D</th>
<th>PDEN</th>
<th>RaTP</th>
<th>CoDAG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>[36, 37]</td>
<td>[36]</td>
<td>[59]</td>
<td>[6]</td>
<td>[42]</td>
<td>[62]</td>
<td>[30]</td>
<td>[37]</td>
<td></td>
</tr>
<tr>
<td>PACS</td>
<td>TDA</td>
<td>81.9</td>
<td>84.4</td>
<td>78.7</td>
<td>79.9</td>
<td>80.3</td>
<td>78.8</td>
<td>77.8</td>
<td>84.7</td>
<td>87.6</td>
</tr>
<tr>
<td></td>
<td>±9.2</td>
<td>±8.0</td>
<td>±6.9</td>
<td>±5.9</td>
<td>±7.1</td>
<td>±5.6</td>
<td>±5.2</td>
<td>±5.1</td>
<td>±4.0</td>
<td>±4.0</td>
</tr>
<tr>
<td></td>
<td>TDG</td>
<td>54.9</td>
<td>56.0</td>
<td>65.8</td>
<td>65.2</td>
<td>64.1</td>
<td>65.8</td>
<td>64.4</td>
<td>70.6</td>
<td>72.2</td>
</tr>
<tr>
<td></td>
<td>±13.1</td>
<td>±10.9</td>
<td>±11.5</td>
<td>±10.5</td>
<td>±12.1</td>
<td>±9.6</td>
<td>±9.8</td>
<td>±9.1</td>
<td>±8.3</td>
<td>±8.3</td>
</tr>
<tr>
<td></td>
<td>FA</td>
<td>74.9</td>
<td>83.0</td>
<td>81.0</td>
<td>81.6</td>
<td>82.6</td>
<td>77.6</td>
<td>76.3</td>
<td>83.9</td>
<td>88.8</td>
</tr>
<tr>
<td></td>
<td>±8.1</td>
<td>±4.0</td>
<td>±6.2</td>
<td>±5.9</td>
<td>±7.0</td>
<td>±4.6</td>
<td>±4.0</td>
<td>±4.7</td>
<td>±3.0</td>
<td>±3.0</td>
</tr>
<tr>
<td></td>
<td>All</td>
<td>70.6</td>
<td>74.5</td>
<td>75.2</td>
<td>75.6</td>
<td>75.7</td>
<td>74.1</td>
<td>72.9</td>
<td>79.7</td>
<td>82.9</td>
</tr>
<tr>
<td></td>
<td>±9.2</td>
<td>±5.7</td>
<td>±7.8</td>
<td>±7.1</td>
<td>±8.6</td>
<td>±6.2</td>
<td>±5.9</td>
<td>±5.7</td>
<td>±4.8</td>
<td>±4.8</td>
</tr>
<tr>
<td>Digits-five</td>
<td>TDA</td>
<td>78.6</td>
<td>81.3</td>
<td>68.7</td>
<td>71.6</td>
<td>72.0</td>
<td>84.3</td>
<td>82.3</td>
<td>88.7</td>
<td>92.7</td>
</tr>
<tr>
<td></td>
<td>±13.2</td>
<td>±14.0</td>
<td>±11.0</td>
<td>±9.2</td>
<td>±9.8</td>
<td>±5.4</td>
<td>±5.8</td>
<td>±5.8</td>
<td>±1.7</td>
<td>±1.7</td>
</tr>
<tr>
<td></td>
<td>TDG</td>
<td>61.0</td>
<td>62.3</td>
<td>64.0</td>
<td>63.3</td>
<td>64.0</td>
<td>70.9</td>
<td>69.7</td>
<td>76.8</td>
<td>77.4</td>
</tr>
<tr>
<td></td>
<td>±14.9</td>
<td>±13.8</td>
<td>±13.6</td>
<td>±13.1</td>
<td>±12.9</td>
<td>±6.8</td>
<td>±7.0</td>
<td>±3.9</td>
<td>±4.3</td>
<td>±4.3</td>
</tr>
<tr>
<td></td>
<td>FA</td>
<td>58.2</td>
<td>64.5</td>
<td>66.1</td>
<td>72.2</td>
<td>73.0</td>
<td>76.5</td>
<td>74.0</td>
<td>85.0</td>
<td>87.1</td>
</tr>
<tr>
<td></td>
<td>±14.9</td>
<td>±13.3</td>
<td>±15.7</td>
<td>±11.2</td>
<td>±10.9</td>
<td>±3.8</td>
<td>±4.0</td>
<td>±2.2</td>
<td>±2.1</td>
<td>±2.1</td>
</tr>
<tr>
<td></td>
<td>All</td>
<td>65.9</td>
<td>69.4</td>
<td>66.2</td>
<td>69.1</td>
<td>69.6</td>
<td>77.2</td>
<td>75.3</td>
<td>83.5</td>
<td>85.7</td>
</tr>
<tr>
<td></td>
<td>±13.5</td>
<td>±12.9</td>
<td>±13.3</td>
<td>±11.0</td>
<td>±10.9</td>
<td>±4.8</td>
<td>±5.1</td>
<td>±2.1</td>
<td>±2.2</td>
<td>±2.2</td>
</tr>
<tr>
<td>DomainNet</td>
<td>TDA</td>
<td>66.0</td>
<td>66.9</td>
<td>53.6</td>
<td>62.2</td>
<td>62.5</td>
<td>56.2</td>
<td>55.6</td>
<td>65.4</td>
<td>71.0</td>
</tr>
<tr>
<td></td>
<td>±8.8</td>
<td>±8.7</td>
<td>±13.2</td>
<td>±7.7</td>
<td>±7.3</td>
<td>±6.2</td>
<td>±6.6</td>
<td>±5.1</td>
<td>±5.7</td>
<td>±5.7</td>
</tr>
<tr>
<td></td>
<td>TDG</td>
<td>47.3</td>
<td>48.1</td>
<td>47.7</td>
<td>51.3</td>
<td>52.1</td>
<td>50.7</td>
<td>49.3</td>
<td>55.2</td>
<td>56.2</td>
</tr>
<tr>
<td></td>
<td>±11.0</td>
<td>±10.7</td>
<td>±11.0</td>
<td>±10.0</td>
<td>±9.9</td>
<td>±9.1</td>
<td>±9.1</td>
<td>±7.4</td>
<td>±7.2</td>
<td>±7.2</td>
</tr>
<tr>
<td></td>
<td>FA</td>
<td>58.5</td>
<td>66.9</td>
<td>56.1</td>
<td>61.8</td>
<td>62.8</td>
<td>52.2</td>
<td>50.2</td>
<td>63.5</td>
<td>70.9</td>
</tr>
<tr>
<td></td>
<td>±8.3</td>
<td>±6.0</td>
<td>±14.5</td>
<td>±9.0</td>
<td>±8.8</td>
<td>±9.4</td>
<td>±9.5</td>
<td>±6.6</td>
<td>±6.6</td>
<td>±6.6</td>
</tr>
<tr>
<td></td>
<td>All</td>
<td>57.3</td>
<td>60.6</td>
<td>52.5</td>
<td>58.4</td>
<td>59.1</td>
<td>53.0</td>
<td>51.7</td>
<td>61.4</td>
<td>66.0</td>
</tr>
<tr>
<td></td>
<td>±8.9</td>
<td>±8.0</td>
<td>±12.4</td>
<td>±8.6</td>
<td>±8.3</td>
<td>±7.6</td>
<td>±7.8</td>
<td>±6.0</td>
<td>±6.2</td>
<td>±6.2</td>
</tr>
</tbody>
</table>

(C), and Sketch (S). Digits-five contains 5 different domains with 10 classes, 0 to 9, including MNIST (MT) [24], MNIST-M (MM) [14], SVHN (SN) [41], SYN-D (SD) [14] and USPS (US) [21]. DomainNet has an imbalance in class distribution, where some domains have limited images for certain classes. To address this issue, a subset of DomainNet is used by selecting the top 10 classes with most images in the whole dataset.

**Experimental settings** In the source domain, 80% of the data is randomly assigned as a training set, and the remaining 20% as a testing set. In target domains, all data is used for training and testing as we assume any label information is not available. The experiments are repeated three times using different seeds (2022, 2023, 2024), and the average performance is reported. For Digits-five, we use DTN [35] as a feature extractor, while for both PACS and DomainNet, we employ ResNet-50 [19]. The SGD optimizer is used with a batch size of 64 for all experiments. The size of replay buffer is set to 200 for all datasets. See the Supplementary Material for more details on the experimental settings and network architectures.

**Evaluation metrics**

(i) **TDA**: Target domain adaptation for each domain is the performance measured on the domain right after its training stage. The TDA of the $t$-th domain for $t = 0, \ldots, T$ is

\[
TDA_t = A(f(x; \theta^*_t), D_t),
\]

where $A(f(x; \theta^*_t), D_t)$ represents the test accuracy on the domain $D_t$, with the model $f(x; \theta^*_t)$ obtained after training on the domain $D_{t-1}$.

(ii) **TDG**: Target domain generalization for each domain is evaluated by the average performance on the domain prior to its training stage. The TDG of the $t$-th domain for $t = 0, \ldots, T$ is

\[
TDG_t = \frac{1}{t} \sum_{s=1}^{t} A(f(x; \theta^*_s), D_s).
\]
domain for \( t = 1, \ldots, T \) is given by

\[
TDG_t = \frac{1}{T} \sum_{t'=0}^{t-1} A(f(x; \theta_{t'}'), D_t).
\]  

(8)

(iii) FA: Forgetting alleviation for each domain is evaluated by the average performance on the domain after the model has been trained on subsequent domains. The FA of the \( t \)-th domain for \( t = 0, \ldots, T - 1 \) can be written as

\[
FA_t = \frac{1}{T-\hat{t}} \sum_{t'=\hat{t}+1}^{T} A(f(x; \theta_{t'}'), D_t).
\]  

(9)

For comparative analysis between different models, we use the averaged value of each metric over all domains for which the metric is defined. Additionally, we define the average of all the metrics as a composite score (All) to evaluate the overall performance of the models. To properly evaluate our proposed framework, CoDAG, we use the DA model \( f_{DA} \) to evaluate TDA, while the DG model \( f_{DG} \) is used to evaluate TDG and FA.

4.1. Effectiveness of the CoDAG framework

Comparing with state-of-the-art To compare the performance of our model with state-of-the-art methods, we use the experiment results of different methods reported in [37] as baselines. These methods include several state-of-the-art models from Source-Free DA (SHOT+ [36, 37] and SHOT++ [36]), Test-Time/Online DA (Tent [59], AdaCon [6], and EATA [42]), Single DG (L2D [62] and PDEN [30]), as well as Continual DG (RaTP [37]). To ensure consistency and fairness, we adopt the same backbone feature extractor with the baseline methods. In the present experiments, all baseline methods are equipped with the replay buffer of size 200.

We present an evaluation of our CoDAG framework against the baseline models on the three datasets (PACS, Digits-five, and DomainNet) using the four evaluation metrics (TDA, TDG, FA, and All). The results are averaged over 10 different orders from each dataset to ensure the robustness of our findings. The overall results presented in Table 1 demonstrate that our method consistently outperforms all the baseline models across all datasets and evaluation metrics.

Compared to RaTP [37], a continual domain generalization method that achieves the best performance among the baselines in most cases, our method demonstrates significantly higher performance in TDA and FA. This highlights the effectiveness of our domain adaptation stage based on the generalized initialization using the DG model. The more accurate pseudo-labels generated by the DA model also contribute to the DG model’s superior performance in FA.

On the DomainNet dataset, which is the most challenging among our benchmark datasets, the DA-specialized model SHOT++ [36] outperforms RaTP in terms of TDA and FA. However, our CoDAG consistently outperforms any DA-specialized models in terms of TDA and FA, while maintaining the improved generalization performance in terms of TDG.

Furthermore, Table 1 shows that our method achieves the lowest standard deviation across ten different orders in nearly all cases. Notably, even our lower bound performance \( (\mu - \sigma) \) surpasses the upper bound performance \( (\mu + \sigma) \) of other baselines in many cases. These findings demonstrate the robustness of our CoDAG framework in addressing the challenges of unsupervised continual domain shift learning.

Training curves In Fig. 3, we display the training curves of the DG model, each of which represents the accuracy of the model on its corresponding domain at different training
Table 3. The ablation study of SelNLPL conducted for all possible pairs of two domains \((D_0 \rightarrow D_1)\) from the PACS dataset. TDG was measured by the average performance on two unseen domains \((D_2\) and \(D_3\)), while FA was measured by the performance on the source domain \((D_0)\). Diff. denotes the result obtained by subtracting the performance without \((w/o)\) SelNLPL from the performance with \((w/)\) SelNLPL.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TDG</td>
<td>w/ SelNLPL</td>
<td>58.3</td>
<td>74.6</td>
<td>58.4</td>
<td>59.8</td>
<td>85.3</td>
<td>77.7</td>
<td>79.8</td>
<td>86.8</td>
<td>78.1</td>
<td>69.9</td>
<td>84.3</td>
<td>85.3</td>
<td>74.9</td>
</tr>
<tr>
<td></td>
<td>w/o SelNLPL</td>
<td>57.4</td>
<td>74.2</td>
<td>54.3</td>
<td>58.0</td>
<td>84.9</td>
<td>71.2</td>
<td>78.6</td>
<td>86.6</td>
<td>74.2</td>
<td>68.3</td>
<td>83.4</td>
<td>85.0</td>
<td>73.0</td>
</tr>
<tr>
<td>Diff.</td>
<td></td>
<td>+0.9</td>
<td>+0.4</td>
<td>+4.1</td>
<td>+1.8</td>
<td>+0.4</td>
<td>+4.6</td>
<td>+1.2</td>
<td>+0.2</td>
<td>+3.9</td>
<td>+1.6</td>
<td>+0.9</td>
<td>+0.3</td>
<td>+1.8</td>
</tr>
<tr>
<td>FA</td>
<td>w/ SelNLPL</td>
<td>98.5</td>
<td>98.2</td>
<td>95.7</td>
<td>94.1</td>
<td>93.2</td>
<td>83.0</td>
<td>90.4</td>
<td>92.6</td>
<td>85.8</td>
<td>85.7</td>
<td>91.2</td>
<td>89.7</td>
<td>91.5</td>
</tr>
<tr>
<td></td>
<td>w/o SelNLPL</td>
<td>98.5</td>
<td>97.2</td>
<td>93.0</td>
<td>93.2</td>
<td>92.6</td>
<td>76.7</td>
<td>89.4</td>
<td>91.3</td>
<td>84.7</td>
<td>85.1</td>
<td>89.3</td>
<td>88.4</td>
<td>89.9</td>
</tr>
<tr>
<td>Diff.</td>
<td></td>
<td>0.0</td>
<td>+1.0</td>
<td>+2.7</td>
<td>+0.9</td>
<td>+0.6</td>
<td>+6.3</td>
<td>+1.0</td>
<td>+1.3</td>
<td>+1.1</td>
<td>+0.6</td>
<td>+1.9</td>
<td>+1.3</td>
<td>+1.8</td>
</tr>
</tbody>
</table>

4.2. Further analysis

In this section, we perform additional analyses to investigate the key components of our method using the PACS dataset under different experimental settings.

Effectiveness of generalized initialization for DA

To evaluate the effectiveness of the generalized initialization approach for DA, we compare model performance between two different approaches for initializing \(\theta_{DA,t}\) of the DA model on \(D_t\): (1) initializing with the previous DG model using \(\theta_{DG,t-1}\) and (2) initializing with the previous DA model using \(\theta_{DA,t-1}\). The results presented in Table 2 show that initializing with the DG model significantly improves the performance in TDA. This exhibits the effectiveness of the generalized initialization, which enables the DA model to adapt more efficiently to a new domain, compared to relying on parameters specifically adapted to the previous domain. Furthermore, we observe that the improved TDA of the DA model has a positive impact on the performance of the DG model in FA and TDG by providing more accurate pseudo-labels.

Ablation study of SelNLPL

To understand SelNLPL’s contribution, we conduct experiments with and without SelNLPL in the training of our model for pairs of two different domains \((D_0 \rightarrow D_1)\). We then compare the resulting changes in TDG and FA. To isolate the effect of SelNLPL, we remove the replay buffer. The experiment results in Table 3 demonstrate that SelNLPL can improve model performance in both TDG and FA, which underscores the effectiveness of noise-resilient methods to amplify the complementary effect of the DA model on the DG model within our framework.

Ablation study of replay buffer

We perform ablation studies to assess the effectiveness of utilizing a replay buffer in CoDAG, by comparing the performance of the model with and without the buffer. The results presented in Fig. 4 clearly display that the absence of the buffer leads to a degradation in performance across all metrics, with the metric for FA being the most adversely affected. This indicates that the replay buffer plays a crucial role in preventing catastrophic forgetting.

Furthermore, our findings suggest that the replay buffer enhances the model’s ability to generalize by continually exposing it to multiple domains, thereby boosting its performance in TDG. This, in turn, leads to an improvement in TDA as the model can adapt to a new domain with more generalized initialization.

However, despite the decrease in performance after the removal of the buffer, our method without replay buffer still outperforms all other baselines with replay buffer, further confirming the effectiveness of our framework.
5. Conclusion

In this paper, we propose a learning framework that combines domain adaptation and generalization models in a complementary manner, which effectively addresses the challenge of unsupervised continual domain shift learning. Our method outperforms state-of-the-art performance across different datasets and evaluation metrics. It also achieves competitive results even without a replay buffer, demonstrating its effectiveness and robustness in real-world scenarios.

Our approach is model-agnostic, meaning it can be used with any domain adaptation and generalization algorithms suitable for a given problem. We envision extending our method to complex scenarios beyond a pre-defined set of domains, dynamically discovering domain shifts and adapting to new domains. This can increase its applicability to a broader range of scenarios, with potential contributions to practical applications in computer vision and other fields.

Acknowledgement

This work was supported by the Hyundai Motor Chung Mong-Koo Foundation, the New Faculty Startup Fund from Seoul National University, and IITP (RS-2023-00232046).

References


