Unsupervised Continual Learning for Gradually Varying Domains

Abu Md Niamul Taufique Chowdhury Sadman Jahan Andreas Savakis
Rochester Institute of Technology, Rochester, NY, USA
{at7133,sj4654,andreas.savakis}@rit.edu

Abstract

In Unsupervised Domain Adaptation (UDA), a network is trained on a source domain and adapted on a target domain where no labeled data is available. Existing UDA techniques consider having the entire target domain available at once, which may not be feasible during deployment in realistic settings where batches of target data are acquired over time. Continual Learning (CL) has been dealing with data constrained paradigms in a supervised manner, where batches of labeled samples are sequentially presented to the network and the network continually learns from the new data without forgetting what was previously learned. Our method for unsupervised continual learning serves as a bridge between the UDA and CL paradigms. This research addresses a gradually evolving target domain fragmented into multiple sequential batches where the model continually adapts to the gradually varying stream of data in an unsupervised manner. To tackle this challenge, we propose a source free method based on episodic memory replay with buffer management. A contrastive loss is incorporated for better alignment of the buffer samples and the continual stream of batches. Our experiments on the rotating MNIST and CORe50 datasets confirm the benefits of our unsupervised continual learning method for gradually varying domains. The codes are available at https://github.com/abutaufique/ucl-gv.git.

1. Introduction

Deep neural networks have shown near human level capabilities in fundamental computer vision tasks such as image classification [3], object detection [43], object tracking [51], and semantic segmentation [55]. While humans can learn new information continuously without drastically forgetting the previously learned information [16, 35], neural networks show vulnerability when new tasks are added for learning by forgetting the previously learned knowledge, also known as catastrophic forgetting [14,22,23]. Deep networks also suffer when transferring the existing knowledge to learning a relevant task if there is a domain shift or concept drift in the training domain [36, 39].

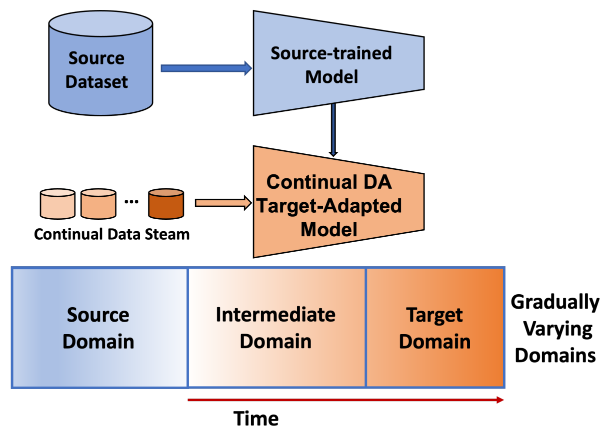

Figure 1. Proposed paradigm of Unsupervised Continual Learning for Gradually Varying domain adaptation (UCL-GV). The network is trained on a source domain and continually adapts using small incoming batches of data from a gradually varying target domain that has no labels.

The classification task becomes challenging when a network is trained on a source domain dataset and operates on a target domain which has different distribution and no labeled samples. The target domain represents a new environment where the model is deployed and the network needs to adapt using unsupervised domain adaptation (UDA) methods [12, 49]. In many real world settings, the target domain is gradually changing over time, such as the case of an autonomous vehicle's interaction with the surrounding environment which is continuously changing due to varying illumination (from day to night) or weather conditions (from sunny to cloudy) etc. [1, 2, 26, 33]. However, current UDA methods are not well-suited for operating in gradually changing environments where the target data are acquired in small batches of unlabeled samples. In this paper, we present an unsupervised continual learning solution to the novel paradigm of domain adaptation in gradually evolving domains illustrated in Fig. 1.

Existing approaches tackle the UDA task using subspace/manifold alignment between source and target features [8, 11, 38, 42, 52] or adversarial domain alignment [46, 53, 54]. In all of these approaches, a network is first trained on the source domain and then adapted to the target domain using all samples from both the source and target datasets. Simultaneous access to both source and target datasets is often not feasible, which has motivated source free domain adaptation methods [27,32,50] that do not need the source data during adaptation. While source free UDA methods conserve memory, they still require the entire target domain which is not suitable for cases where target data are acquired in small batches or when dealing with data that have gradually varying distributions.

Our unsupervised continual learning approach for gradually varying domains (UCL-GV) is designed to adapt to the target domain using small incoming batches, as the target domain distribution is gradually varying. The UCL-GV method has potential applications in edge AI [1, 2, 26, 33], where learning takes place under resource constraints with limited available memory or computational resources. The CL paradigm typically addresses these scenarios for supervised learning [29], while our proposed UCL-GV method deals with an unsupervised adaptation setting.

In recent years, CL attracted significant interest because of its realistic nature in model deployment [29]. In supervised CL, the learning problem is formulated in two major ways [16]: a) batch incremental learning, where the dataset is split into a certain number of batches and the algorithm encounters each data batch sequentially and runs for multiple epochs to learn the distribution of the samples [5, 6, 21], and b) streaming learning, where only a single instance of the data is fetched by the algorithm and this sample is only seen once [13, 15, 19]. Streaming learning is inherently more challenging than batch incremental learning. In this research, we access the gradually evolving domain through a sequence of batches, as illustrated in Fig. 1. We term this learning procedure as *batch streaming domain incremental learning*. Since the target domain is dynamic (gradually evolving) and each batch of data is presented only once to the network, fast adaptation is a key challenge for such domain adaptation.

To address these challenges, we investigate a novel paradigm for unsupervised continual learning that performs domain adaptation in gradually evolving domains, as depicted in Fig. 1. We propose UCL-GV, a novel method based on selectively storing samples in a buffer and replaying them when a new batch of samples is fetched. To mitigate the small domain shift between the existing buffer samples and the incoming batch samples, due to the gradually varying nature of the target data, we propose to perform alignment using a contrastive loss.

The contributions of our research are outlined below.

- We introduce the novel setting of unsupervised continual learning for domain adaptation in gradually varying domains based on batch streaming that bridges the gap between unsupervised domain adaptation and continual learning research.
- We use a First-In, First-Out (FIFO) buffer for replaying the episodic memory to aid the domain alignment for gradually evolving domains. The buffer samples with the incoming batch samples help achieve better clustering to compute robust pseudo labels for adaptation.
- We utilize a contrastive loss to improve the domain alignment between the existing buffer samples and the incoming batch samples from the gradually varying target domain.
- Our method significantly outperforms existing SOTA UDA methods that do not use any episodic memory.

2. Related Works

2.1. Continual Learning

Continual learning without catastrophic forgetting [37] is an inherent capability in humans. On the other hand, neural networks are mired in catastrophic forgetting whenever a network is required to learn a new task with limited data [22]. The CL community approaches this problem in multiple ways [15]: Elastic Weight Consolidation (EWC) [23], memory replay [48], distillation [10, 18], and fine tuning. In EWC, the previously learned kernel weights are regularized by a quadratic term during learning new information so that the changes in the weights are not drastic. Memory replay is another popular way of combating catastrophic forgetting, where samples are partially or fully stored after passing through the network and then replayed with the incoming data, so as to prevent the model from forgetting the already learned knowledge. Existing methods save samples in a memory buffer by storing raw samples [41, 47] or by storing sample embeddings [14]. With the availability of a new batch or stream of samples, either the entire set of samples [13] or a partial set of samples [5, 21] are replayed from the buffer. Approaches for CL under concept drift include [9, 24]. Matthias et. al. [9] proposed an evolving prototype estimation mechanism to continually learn under concept drift. In our UCL-GV method, we adopt a *selective store and replay* strategy for our unsupervised continual learning scenario, motivated by the strong performance of this scheme in the CL literature [5].

2.2. Continual Domain Adaptation

Existing DA research formulates the problem of continual domain adaptation in primarily two major ways: gradually evolving domain shift [1, 2, 20, 26], and sudden domain shift [40, 44] between the source and target domains.

FRIDA [40] formulated a multi-target continual UDA approach where each target domain is fully available to the model at once during adaptation, and the network continues to adapt to multiple target domains one after the other sequentially. A similar formulation is used by [44] where EWC is used to mitigate catastrophic forgetting.

The work in [26] proposed an UDA method for an evolving target domain. The sequential gradually varying data were split into three different domains: a source domain, an intermediate domain, and a target domain. The intermediate domain was introduced to represent the gradually evolving nature of the data, rather than having a drastic domain shift between the source and the target domains. A meta learning approach was proposed for continual adaptation. Following [26], the work in [7] proposed to perform domain adaptation without having the sequential indexes of the intermediate domains. ConDA [45] formulated the continual domain adaptation problem that considers small batches of samples arriving from the target domain with the network continually adapting to the incoming batches of samples. The ConDA method utilized a buffer mechanism to selectively store and replay samples, and sample mixup to improve the soft label estimation for the unlabeled target domain.

Our proposed setting of unsupervised continual learning for domain adaptation in gradually evolving domains has two major differences from [7, 26]. First, each batch of data from the intermediate and target domains are only seen once rather than multiple times as proposed in [7, 26]. Second, both the source data and the intermediate/target data are required during meta training, while ours is a more realistic source-free adaptation setting to address the constraints in data access or privacy concerns. In contrast to ConDA [45], which considers that the continual data come from a stationary target domain and the network can adapt over multiple epochs, our UCL-GV considers that the data come from a gradually evolving domain and is only allowed one epoch for adaptation. Only a batch of samples is presented to the network at one time, and these samples are discarded after that time step, except the ones which are selectively stored in the memory buffer. In accordance with batch streaming learning, we restrict adapting our model to only one epoch over the new batch and buffer samples, whenever a new batch of gradually evolving samples is received. These constraints make the scope of our research more challenging compared to existing UDA settings and more aligned with the current direction in CL research.

3. Method

For the UDA problem, we consider three domains as illustrated in Fig. 1: a source domain, an intermediate domain, and a target domain. The source domain, $\mathcal{D}_s$, has $\mathcal{C}_s$ classes with source data $\{x_s^i, y_s^i\}_{i=1}^{n_s}$ with n_s labeled samples, where $x_s \in \mathcal{X}_s$ with labels $y_s \in \mathcal{Y}_s$. As in [26], we further consider an unlabeled intermediate domain, D_{int}, that has $\mathcal{C}_{int}$ classes with $\mathcal{X}_{int}$ samples, and an unlabeled target domain, $\mathcal{D}_{tar}$, that has $\mathcal{C}_{tar}$ classes with samples $\mathcal{X}_{tar}$. By generalizing the notations, we combine the intermediate and target domain as $\mathcal{D}_t$ with unlabeled data $\mathcal{X}_t = \mathcal{X}_{int} \bigcup \mathcal{X}_{tar}$ with $\mathcal{C}_t = \mathcal{C}_{int} = \mathcal{C}_{tar} = \mathcal{C}_s$ classes. Here $x_t \in \mathcal{X}_t$ and $\{x_t^i\}_{i=1}^{n_t}$ with n_t is the total number of unlabeled samples and t is gradually varying, $t \in [0, 1]$. We further consider that $\mathcal{D}_t$ is split into m sequential batches $\mathcal{X}_t = \{\mathcal{X}_{t_1}, \mathcal{X}_{t_2}, \mathcal{X}_{t_3}, \cdots, \mathcal{X}_{t_m}\}$ where $t_1 < t_2 < t_3 < \cdots < t_m$ and each batch has n_{t_i} i.i.d. samples where $n_t = \sum_{i=1}^{m} n_{t_i}$. Since we consider a gradual domain adaptation, we assume that the domain change in continual batches is small, i.e., $\lim_{\Delta t \to 0} d(\mathcal{D}_t, \mathcal{D}_{t+\Delta t}) = 0$ for any domain distribution distance measurement method d [33].

The objective of UCL-GV is to train a model $f_s : \mathcal{X}_s \to \mathcal{Y}_s$, parameterized by θ_s, and continually adapt it on $\mathcal{D}_t$ so that the model $f_t : \mathcal{X}_{t_i} \to \mathcal{Y}_{t_i}$, parameterized by θ_t, provides better performance on $\mathcal{X}_{tar}$ when $i = m$, compared to $f_t : \mathcal{X}_t \to \mathcal{Y}_t$ with having only f_s and $\mathcal{X}_t$ during adaptation. The overall objective can also be represented in terms of the loss computation as follows [33].

$$\begin{aligned} &\min_{\theta_t} \mathbb{E}_{t \sim U(0,1)} \mathbb{E}_{(x_t, y_t) \sim \mathcal{D}_{tar}} \mathcal{L}(f_t(x_t), y_t) = \\ &\quad \min_{\theta_t} \int_0^1 \mathbb{E}_{(x_t, y_t) \sim \mathcal{D}_{tar}} \mathcal{L}(f_t(x_t), y_t) \, dt \end{aligned} \tag{1}$$

The architecture of UCL-GV is shown in Fig. 2. Inspired by [32], we initially train our source model $f_s(x) = h_s(g_s(x))$ on the source data. The model consists of two parts, a feature extractor with a backbone followed by a fully connected layer and a batch normalization layer denoted as g_s. The generated features are passed through the hypothesis layer that consists of a fully convolutional layer, followed by a weight normalization layer denoted as h_s. The source network is trained with a label smoothing loss. For the target model, $f_t(x) = h_t(g_t(x))$, the feature extractor model g_t is initialized with g_s and set as trainable, while the transferred hypothesis model $h_t = h_s$ is kept frozen throughout the adaptation procedure.

The unlabeled data from $\mathcal{D}_t$ are sequentially presented to the network and certain samples are selectively stored in a buffer after processing each incoming new batch, $\mathcal{X}_{t_i}$. At each step in time when a new batch is received, the existing buffer samples are added to the incoming batch samples for adaptation. This prevents the clusters from deviating too much from one batch to the next. The details of the buffer and buffer management strategies are provided next, in Sec. 3.1 and 3.2. Since the incoming samples are without labels, clustering is needed for pseudo-label assignment. However,

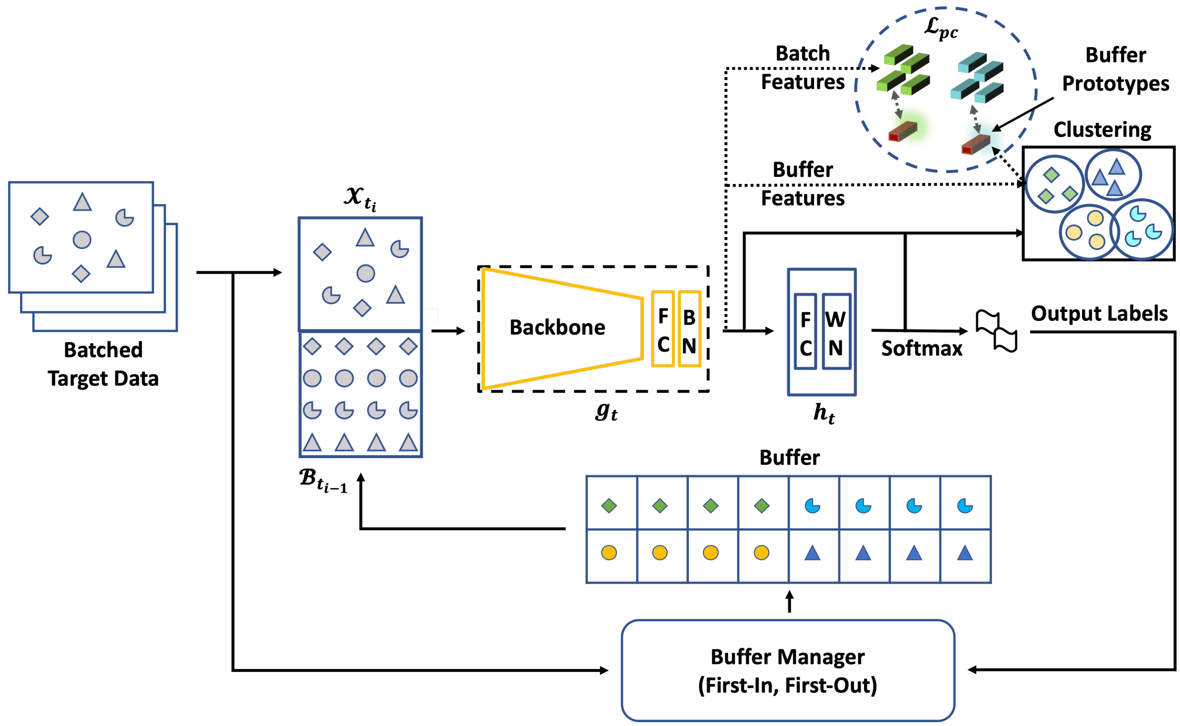

Figure 2. Proposed UCL-GV method for unsupervised continual learning for domain adaptation in gradually varying domains.

the clustering techniques utilized in [4, 32] primarily deal with samples from a stationary distribution and are not suitable for gradually varying domains. In this paper, we improve upon this clustering technique to incorporate samples from non-stationary distributions. Since the domain gap between the incoming batch samples and the buffer samples is small, we utilize contrastive alignment between the buffer prototypes (cluster centers) and the batch samples by minimizing the prototypical contrastive loss $\mathcal{L}_{pc}$, as shown in Fig. 2. The procedure is detailed in Sec. 3.4. It is important to note that the existing buffer samples and new incoming batch samples are fed through the network only once, i.e. only one epoch of the $\mathcal{B}_{t_{i-1}} \bigcup \mathcal{X}_{t_i}$ samples is allowed at each time step during adaptation. The total number of adaptation time steps is equal to the number of sequential incoming batches of data from $\mathcal{D}_t$, the combined intermediate and target domain.

3.1. Buffer

In our setting we consider closed-set domain adaptation where $\mathcal{C}_s = \mathcal{C}_t$ with the same classes in the source and target domains. We allocate equal number of samples from each class in the buffer $\mathcal{B}_t = \{\mathcal{B}_{t_1}, \mathcal{B}_{t_2}, \cdots, \mathcal{B}_{t_m}\}$ based on pseudo-label assignment on incoming target samples. This allows the class-wise data distribution to be considerably uniform throughout the adaptation process. The buffer stores raw samples for adaptation, and the buffer samples are managed by a buffer manager as described in the next subsection.

3.2. Buffer Manager

The buffer manager is responsible for populating the buffer with new samples while partially or fully dropping the existing samples depending on the number of batch and buffer sizes. We considered multiple buffer sample selection mechanisms that exist for the supervised CL paradigm. One popular scheme of sample selection is uniform random, where all the incoming batch samples are combined with the existing buffer samples and the samples to be stored for the next time step are randomly selected with uniform probability. Another option is to store all the previously encountered samples until the current batch, and then randomly select a subset with uniform probability for replay [5]. Existing research shows strong results based on this method. Minimum logit distance is another method where the samples are selected based on the distance to a decision boundary [6]. Some other mechanisms are also introduced in [15] such as choosing samples with minimum confidence, maximum loss, maximum time since last replay, and so on. However, we argue that most of the supervised buffer management strategies are not readily applicable to unsupervised continual learning, except the random selection technique. We tested several schemes for updating the buffer samples, such as selecting samples randomly with uniform probability, samples with high confidence, samples closer to the cluster center, and samples with first-in, first-out queue. We found that *first-in, first-out queue* performs slightly better than all of the other methods for gradually varying domain adaptation (demonstrated in Sec. 5.3). Intuitively, since the domain is gradually evolving, the estimated pseudo labels are the most appropriate when the domain shift within the available data is minimum. If the domain shift between existing buffer samples and the incoming batch samples is high, the estimated pseudo label quality degrades and hence the adaptation performance also degrades.

3.3. Clustering

At time t_i, the network utilizes a new batch of samples $\mathcal{X}_{t_i}$ and the existing buffer samples $\mathcal{B}_{t_{i-1}}$ from the previous time step. The combined data $\mathcal{X}_{t_i} \bigcup \mathcal{B}_{t_{i-1}}$ produces n_b i.i.d. minibatches that are passed through the feature extraction network g_t, and the features are accumulated to perform clustering. We adopted weighted k-means clustering encouraged from [4, 32, 45] that provides the pseudo labels and cluster centers.

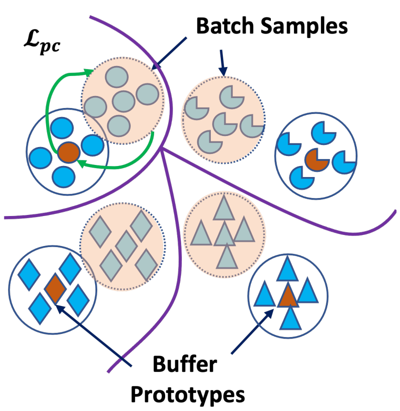

Figure 3. Application of contrastive loss using the buffer prototypes (cluster centers) and the batch samples, for better clustering.

3.4. Contrastive Alignment

Since the domain gap between two consecutive data batches is small (due to the gradually varying domains), we propose to align the feature representations of the incoming batch and buffer samples using a contrastive loss. Such alignment between the buffer and batch features complements the clustering process and generates better pseudo labels. We compute a cosine distance based contrastive loss from the buffer prototypes to the batch samples as shown in Fig. 3. The buffer prototypes (cluster centers) are computed with the current state of the feature extractor $\hat{g}_t$, using the pseudo-labels $\hat{y}_t \in \hat{\mathcal{Y}}_t$ for the samples in the buffer and the incoming batch samples, $\mathcal{B}_{t_{i-1}} \bigcup \mathcal{X}_{t_i}$, as follows [32].

$$\mathbf{z}_k = \frac{\sum_{x_t \in \mathcal{B}_{t_{i-1}}} \mathbb{1}(\hat{y}_t = k)\hat{g}_t(x_t)}{\sum_{x_t \in \mathcal{B}_{t_{i-1}}} \mathbb{1}(\hat{y}_t = k)} \tag{2}$$

In our experiments, $\mathbf{z}_k \in \mathbb{R}^{|\mathcal{C}_t| \times 256}$. The batch features are computed as follows.

$$\mathbf{z} = \hat{g}_t(x_t), \forall x_t \in \mathcal{X}_{t_i} \tag{3}$$

Both the batch features and the buffer features are normalized.

$$\hat{\mathbf{z}}_k = \frac{\mathbf{z}_k}{||\mathbf{z}_k||}, \hat{\mathbf{z}} = \frac{\mathbf{z}}{||\mathbf{z}||} \tag{4}$$

The normalized features are used to compute the prototypical contrastive (PC) loss $\mathcal{L}_{pc}$ [30, 31].

$$\mathcal{L}_{pc} = -\log \frac{\exp\left(\hat{\mathbf{z}}_i \cdot \hat{\mathbf{z}}_{k=\hat{y}_t^i}\right)}{\sum_{c=1}^{|\mathcal{C}_t|} \exp\left(\hat{\mathbf{z}}_i \cdot \hat{\mathbf{z}}_{k=c}\right)} \tag{5}$$

We minimize the PC loss in conjunction with the other loss functions.

3.5. Overall Loss Function

We adopt the Information Maximization (IM) [25, 32] loss, according to the formulation of [45] that minimizes the entropy $\mathcal{L}_{ent}$ and equal diversity loss $\mathcal{L}_{eq}$. With the pseudo labels computed in the overall clustering, we compute the cross-entropy loss below.

$$\mathcal{L}_{ce} = \mathbb{E}_{x_t \in \mathcal{B}_{t_{i-1}} \bigcup \mathcal{X}_{t_i}, \hat{y}_t \in \hat{\mathcal{Y}}_t} - \log \sigma_k(f_t(x_t)) \tag{6}$$

where, σ_k is the softmax function. The overall loss function is written as follows.

$$\mathcal{L}(g_t) = \mathcal{L}_{ent} + \gamma_1 \mathcal{L}_{eqdiv} + \gamma_2 \mathcal{L}_{ce} + \gamma_3 \mathcal{L}_{pc} \tag{7}$$

where γ_1, γ_2, and γ_3 are hyper-parameters. The overall process is presented in Algorithm 1.

4. Datasets and experiments

We used two datasets, rotating MNIST and CORe50, for evaluation. We adopt the **rotating MNIST** [26] which has 50,000 training and 10,000 test images. It is created to mimic an evolving domain where the first 20,000 images are used for training our source model and are rotated between $[0°, 10°]$. The next 30,000 images from the training set form the intermediate domain and are rotated between $[10°, 50°]$. The 10,000 test images are selected as the target domain and are rotated between $[50°, 60°]$. Following [32], we consider the entire target domain for evaluation after adaptation on

Algorithm 1: UCL-GV algorithm

Input : A source trained model $f_s = h_s \cdot g_s : \mathcal{X}_s \rightarrow \mathcal{Y}_s$, evolving data batches $\{\mathcal{X}_{t_1}, \mathcal{X}_{t_2}, \cdots \mathcal{X}_{t_m}\}$ from $\mathcal{D}_t$.

Output: A model continually adapted on $\mathcal{D}_t$ and the corresponding predicted labels for $\mathcal{X}_{tar}$.

Init. : Initialize the target network g_t with g_s and set the hypothesis network $h_t = h_s$ and keep it frozen during adaptation.

for $i \leftarrow 1$ **to** m **do**
  if $i = 1$ **then**
    $X \leftarrow \mathcal{X}_{t_i}$;
  else
    $X \leftarrow \mathcal{X}_{t_i} \bigcup \mathcal{B}_{t_{i-1}}$;
  end
  $\hat{Y} \leftarrow$ Compute psuedo labels for X;
  for $j \leftarrow 1$ **to** n_b **do**
    Get i.i.d batch samples from $(X, \hat{Y})$;
    Compute $\mathcal{L}_{ent}$, $\mathcal{L}_{eq}$, and $\mathcal{L}_{ce}$;
    if $i = 1$ **then**
      $\mathcal{L}_{pc} \leftarrow 0$;
    else
      $\mathcal{L}_{pc} \leftarrow$ Compute the PC loss using Equation (5);
    end
    Compute $\mathcal{L}(g_t)$ using Equation (7);
    Optimize g_t with $\mathcal{L}(g_t)$;
  end
  $\mathcal{B}_{t_i} \leftarrow$ Fill buffer using $g_t$and $(\mathcal{X}_{t_i}, \mathcal{B}_{t_{i-1}})$;
end

the intermediate and the target domains. Examples of the rotating MNIST dataset are shown in Fig. 4.

Further, we restructure **CORe50** [34] dataset to evaluate UCL-GV under the continually evolving domain adaptation setting. CORe50 dataset is specifically designed for CL research and has 50 domestic objects from 10 categories collected on 11 sessions. We found that choosing 8 sessions makes the dataset suitable for gradually varying domains where the backgrounds of the images vary gradually in appearance. Additionally, there are pose and illumination changes among various sessions. We used the samples from session 's1' as the source domain, 's2', 's3', and 's8' as the unlabeled intermediate domain, and 's9', 's11', 's4', and 's10' as the target domain where the samples are appended according to the order mentioned here. Examples of the CORe50 dataset are shown in Fig. 5.

The source model is trained with randomly sampled data from the entire source domain. Following the setting in [26], the intermediate domain is chosen to implement a

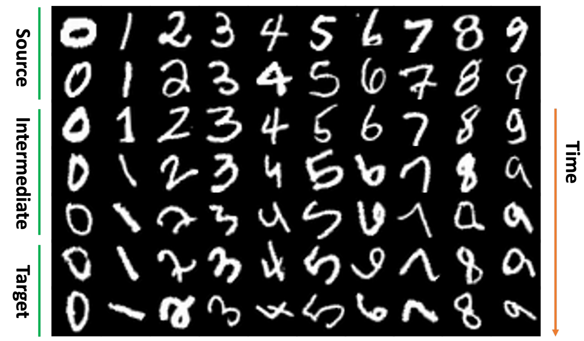

Figure 4. Rotating MNIST dataset.

gradual change, rather than a drastic change from the source domain to the target domain. The intermediate domain and the target domain are provided to the network sequentially, however, the classes are randomly mixed. For the rotating MNIST dataset, we utilize a LeNet backbone [28] with two convolutional layers. For the CORe50 dataset, we choose a ResNet18 backbone [17]. We normalize the rotating MNIST samples to have 0.5 mean and 0.5 standard deviation. CORe50 samples undergo resizing to 256×256 pixels, and random cropping to size 224×224, random horizontal flipping, and normalization for adaptation. The starting learning rate for rotating MNIST is 0.01 and for CORe50 is 0.001, and are varied according to the setup of [32]. We empirically set $\gamma_1 = 1$, $\gamma_2 = 0.6$, and $\gamma_3 = 3.0$ for rotating MNIST and $\gamma_2 = 0.1$, and $\gamma_3 = 1.0$ for the CORe50 dataset.

5. Results

5.1. Performance on Full Target Domain

We computed the domain adaptation performance with our baseline method [32] using the full target dataset, as shown in Table 1. For all settings, the model is evaluated only on the target dataset, $\mathcal{X}_{tar}$. The model with only source training (without adaptation on the intermediate or the target domain) evaluated on the target domain indicates the domain gap between the source and the target domain. On the rotating MNIST dataset, the low classification score of 45.16% of the source trained model indicates a large domain gap between the source domain and the target domain. On the other hand, the performance of the source trained model on CORe50 dataset is 74.59%, which shows a smaller domain gap between the source and the target domains. The CORe50 dataset contains slight changes among the three domains in the background.

The target-only model is the case where the model is trained on the source dataset and adapted to the target dataset, $\mathcal{X}_{tar}$ without any intermediate domain data. After adapting to the target domain with the baseline method,

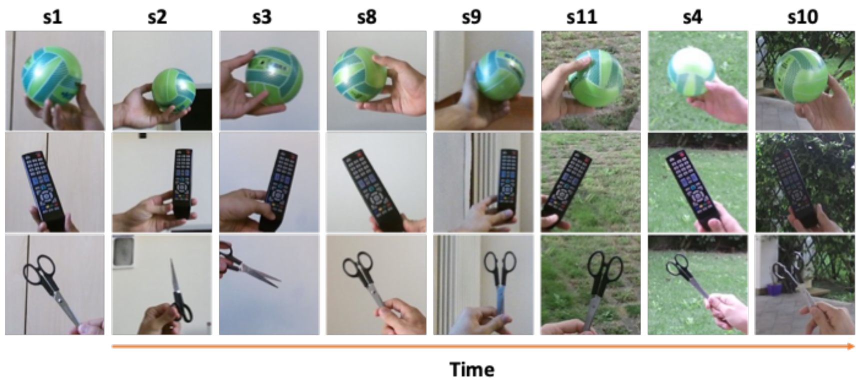

Figure 5. CORe50 [34] dataset in a gradual time varying setting.

Method	Adaptation domain	Domain availability	Rotating MNIST	CORe50
Baseline [32]	None (No adaptation)	Full	45.16	74.59
Baseline [32]	Target only	Full	67.88	90.19
Baseline [32]	Intermediate + Target	Full	96.20	91.49
Gradual ST [26]	Intermediate + Target	Continual	92.03	N/A
Baseline [32]	Intermediate + Target	Continual	94.20	87.14
UCL-GV	Intermediate + Target	Continual	95.66	89.07

Table 1. Percent accuracy of UCL-GV and comparison with other methods. The experiments on rotating MNIST are performed with a continual batch size of 128 and buffer size of 512. CORe50 experiments are performed with a continual batch size of 16 and buffer size of 32. All evaluations are conducted on the target domain $\mathcal{D}_{tar}$.

performance on both datasets improves significantly. For the rotating MNIST dataset, the performance improves by 22.72% and for the CORe50 dataset, the performance improves by 15.6%.

With the availability of the intermediate domain, the shift between the source and the adaptation domains is much smaller. This leads to significant performance gains compared to the target-only adapted baseline model, even for the cases of continual learning from small incoming batches.

5.2. Performance on Gradually Varying Domains

UCL-GV shows significant improvement over the existing baseline [32] and Gradual ST [26], as shown in Table 1. The results on Gradual ST [26] are obtained by running the publicly available codebase on our dataset settings. In the continual adaptation setting, the performance of the baseline [32] method degrades by 2% on the rotating MNIST dataset and by 4.35% on the CORe50 dataset, compared to the adaptation on the full intermediate and target domains simultaneously. UCL-GV outperforms Gradual ST by 3.63% and the baseline method by 1.46% on rotating MNIST dataset on the continual settings. On the CORe50 dataset, UCL-GV outperforms the continual baseline method by 1.93%.

To further illustrate the continual learning capability of our method, we evaluate the classification performance on all of the target samples $\mathcal{X}_{tar}$ of the rotating MNIST dataset after each incoming batch $\mathcal{X}_{t_i}$ from $\mathcal{D}_t$, as shown in Fig. 6. Our method shows consistent performance gains while learning on new batches of data.

5.3. Effects of Batch and Buffer Sizes

To understand the impact of batch and buffer sizes on continual adaptation, we conducted ablation studies on the rotating MNIST dataset. Fig. 7 shows the results obtained when varying the buffer size (top) and batch size (bottom). The results in Fig. 7 (top) also demonstrate the effectiveness of the first-in, first-out queue. Additionally, we observe that the performance increases with increase in the buffer size. This observation is consistent with the existing supervised streaming learning scenario [15]. Based on intuition, increasing the buffer size provides access to more samples, which improves unsupervised clustering and prototype representation from the buffer samples. However, this comes at the cost of larger memory footprint, and the buffer size selection will depend on the available system resources.

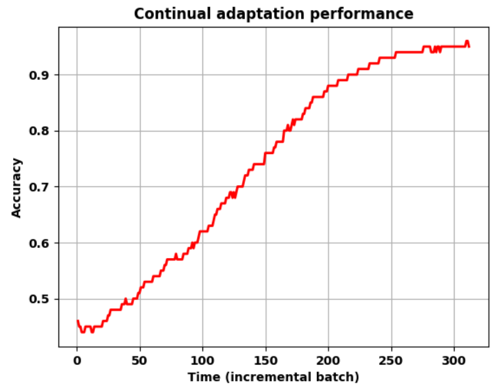

Figure 6. Performance of UCL-GV on the rotating MNIST target domain $\mathcal{D}_{tar}$ during continual adaptation on each incremental batch from the combined intermediate and target domain $\mathcal{D}_t$.

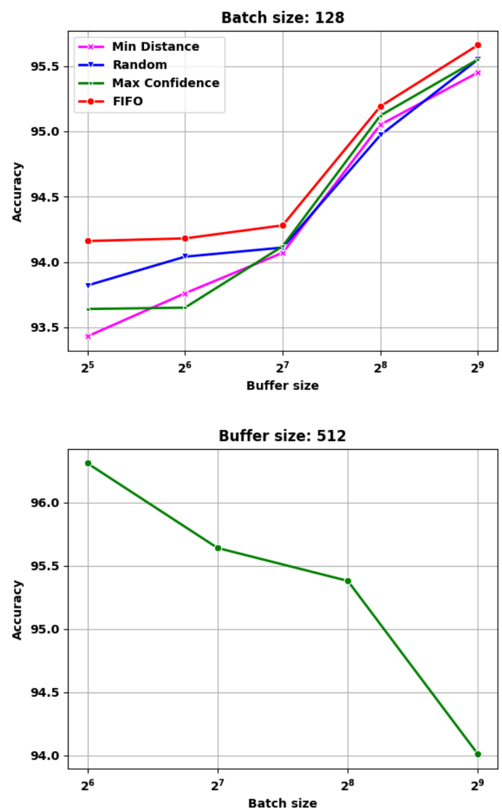

Figure 7. Impact of varying the buffer size (top) and batch size (bottom) of UCL-GV on the rotating MNIST dataset.

The results for various batch sizes, while the buffer size is kept fixed, are shown in Fig. 7 (bottom). When the incoming batch size is varied from 64 to 512, the performance degrades with increase in batch size, which may appear counter intuitive. However, since the target domain data is varying gradually, that is, the class-wise data distribution is continuously changing, having a larger batch size might cause overlap between different class distributions across the varying domain. This can potentially lead to incorrect pseudo-label assignments and eventually result in negative adaptation and lower performance.

5.4. Ablation Studies

We demonstrate the effectiveness of various aspects of UCL-GV by performing ablation studies on the rotating MNIST dataset. We performed each experiment 3 times and report the average in Table 2. The UDA baseline [32] method achieves 94.20% accuracy in continual adaptation across varying domains. After adding the buffer, we observe $\sim$1% improvement in performance, which corresponds to 14.8% reduction in error, validating the effectiveness of including the memory buffer. With the introduction of contrastive alignment between the buffer prototypes and the batch samples, the final performance of UCL-GV is 95.66%, which is a 1.46% total improvement over the baseline, or 25.2% reduction in error .

Method	Percent Accuracy
Baseline	94.20
Baseline+Buffer	95.06
UCL-GV: Baseline+Buffer+$\mathcal{L}_{pc}$	95.66

Table 2. Ablation studies of UCL-GV on the rotating MNIST dataset. Experiments are performed with a continual batch size of 128 and buffer size of 512.

6. Conclusion

We propose a novel method for unsupervised continual learning for domain adaptation in gradually varying domains. We formulate the adaptation problem in a batch streaming manner where the network needs to adapt to the incoming batches by leveraging the already learned information from the earlier batches. To aid gradual adaptation, we propose to utilize episodic memory replay by selectively storing samples in a first-in, first-out buffer and replay them with the next incoming batch. The domain alignment between the buffer and incoming batch samples is further improved by utilizing a contrastive loss. Our UCL-GV method outperforms SOTA methods on two datasets.

Acknowledgements

This research was supported in part by AFOSR grant number FA9550-22-P-0009 through Intelligent Fusion Technology and Air Force Research Laboratory contract number FA8650-18-C-1075 to Kitware.

References

[1] Adeleh Bitarafan, Mahdieh Soleymani Baghshah, and Marzieh Gheisari. Incremental evolving domain adaptation. *IEEE Transactions on Knowledge and Data Engineering*, 28(8):2128–2141, 2016. 1, 2

[2] Andreea Bobu, Eric Tzeng, Judy Hoffman, and Trevor Darrell. Adapting to continuously shifting domains. *International Conference on Learning Representations*, 2018. 1, 2

[3] Andy Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. In *International Conference on Machine Learning*, pages 1059–1071. PMLR, 2021. 1

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision*, pages 132–149, 2018. 4, 5

[5] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision*, pages 233–248, 2018. 2, 4

[6] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision*, pages 532–547, 2018. 2, 4

[7] Hong-You Chen and Wei-Lun Chao. Gradual domain adaptation without indexed intermediate domains. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[8] Sentao Chen, Mehrtash Harandi, Xiaona Jin, and Xiaowei Yang. Domain adaptation by joint distribution invariant projections. *IEEE Transactions on Image Processing*, 29:8264–8277, 2020. 2

[9] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8250–8259, 2021. 2

[10] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5138–5146, 2019. 2

[11] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2960–2967, 2013. 2

[12] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189. PMLR, 2015. 1

[13] Tyler L Hayes, Nathan D Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9769–9776. IEEE, 2019. 2

[14] Tyler L Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. Remind your neural network to prevent catastrophic forgetting. In *Proceedings of the European Conference on Computer Vision*, pages 466–483. Springer, 2020. 1, 2

[15] Tyler L Hayes and Christopher Kanan. Selective replay enhances learning in online continual analogical reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3502–3512, 2021. 2, 4, 7

[16] Tyler L Hayes, Giri P Krishnan, Maxim Bazhenov, Hava T Siegelmann, Terrence J Sejnowski, and Christopher Kanan. Replay in deep learning: Current approaches and missing biological elements. *Neural Computation*, 33(11):2908–2950, 2021. 1, 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6

[18] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 2

[19] T Ryan Hoens, Robi Polikar, and Nitesh V Chawla. Learning from streaming data with concept drift and imbalance: an overview. *Progress in Artificial Intelligence*, 1(1):89–101, 2012. 2

[20] Judy Hoffman, Trevor Darrell, and Kate Saenko. Continuous manifold based adaptation for evolving visual domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 867–874, 2014. 2

[21] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019. 2

[22] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1, 2

[23] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 1, 2

[24] Lukasz Korycki and Bartosz Krawczyk. Class-incremental experience replay for continual learning under concept drift. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3649–3658, 2021. 2

[25] Andreas Krause, Pietro Perona, and Ryan Gomes. Discriminative clustering by regularized information maximization. *Advances in Neural Information Processing Systems*, 23, 2010. 5

[26] Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In *International Conference on Machine Learning*, pages 5468–5479. PMLR, 2020. 1, 2, 3, 5, 6, 7

[27] Vinod K Kurmi, Venkatesh K Subramanian, and Vinay P Namboodiri. Domain impression: A source data free domain adaptation method. In *Proceedings of the IEEE/CVF Winter

Conference on Applications of Computer Vision, pages 615–625, 2021. 2

[28] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 6

[29] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information Fusion*, 58:52–68, 2020. 2

[30] Junnan Li, Caiming Xiong, and Steven CH Hoi. Learning from noisy data with robust representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9485–9494, 2021. 5

[31] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020. 5

[32] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020. 2, 3, 4, 5, 6, 7, 8

[33] Hong Liu, Mingsheng Long, Jianmin Wang, and Yu Wang. Learning to adapt to evolving domains. *Advances in Neural Information Processing Systems*, 33:22338–22348, 2020. 1, 2, 3

[34] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Conference on Robot Learning*, pages 17–26. PMLR, 2017. 6, 7

[35] Vincenzo Lomonaco, Lorenzo Pellegrini, Pau Rodriguez, Massimo Caccia, Qi She, Yu Chen, Quentin Jodelet, Ruiping Wang, Zheda Mai, David Vazquez, et al. Cvpr 2020 continual learning in computer vision competition: Approaches, results, current challenges and future directions. *Artificial Intelligence*, 303:103635, 2022. 1

[36] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363, 2018. 1

[37] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419, 1995. 2

[38] Breton Minnehan and Andreas Savakis. Deep domain adaptation with manifold aligned label transfer. *Machine Vision and Applications*, 30(3):473–485, 2019. 2

[39] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008. 1

[40] Sayan Rakshit, Anwesh Mohanty, Ruchika Chavhan, Biplab Banerjee, Gemma Roig, and Subhasis Chaudhuri. FRIDA—generative feature replay for incremental domain adaptation. *Computer Vision and Image Understanding*, page 103367, 2022. 2, 3

[41] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[42] Baochen Sun and Kate Saenko. Subspace distribution alignment for unsupervised domain adaptation. In *BMVC*, volume 4, pages 24–1, 2015. 2

[43] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14454–14463, 2021. 1

[44] Shixiang Tang, Peng Su, Dapeng Chen, and Wanli Ouyang. Gradient regularized contrastive learning for continual domain adaptation. In *Proceedings 35th of the AAAI Conference on Artificial Intelligence*, pages 2–13, 2021. 2, 3

[45] Abu Md Niamul Taufique, Chowdhury Sadman Jahan, and Andreas Savakis. ConDA: Continual unsupervised domain adaptation. *arXiv preprint arXiv:2103.11056*, 2021. 3, 5

[46] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017. 2

[47] Gido M van de Ven, Hava T Siegelmann, and Andreas S Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature Communications*, 11(1):1–14, 2020. 2

[48] Eli Verwimp, Matthias De Lange, and Tinne Tuytelaars. Rehearsal revealed: The limits and merits of revisiting samples in continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9385–9394, 2021. 2

[49] Ni Xiao and Lei Zhang. Dynamic weighted learning for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15242–15251, 2021. 1

[50] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8978–8987, 2021. 2

[51] Jianming Zhang, Juan Sun, Jin Wang, and Xiao-Guang Yue. Visual object tracking based on residual network and cascaded correlation filters. *Journal of Ambient Intelligence and Humanized Computing*, 12(8):8427–8440, 2021. 1

[52] Lei Zhang, Shanshan Wang, Guang-Bin Huang, Wangmeng Zuo, Jian Yang, and David Zhang. Manifold criterion guided transfer learning via intermediate domain generation. *IEEE Transactions on Neural Networks and Learning Systems*, 30(12):3759–3773, 2019. 2

[53] Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3801–3809, 2018. 2

[54] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5031–5040, 2019. 2

[55] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021. 1