Semi-Supervised Generalized Source-Free Domain Adaptation (SSG-SFDA)

Jiayu An, Changming Zhao, and Dongrui Wu, *Fellow, IEEE*

2023 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-8867-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/IJCNN54540.2023.10191761

***Abstract*—Continual learning aims to learn on a sequence of new tasks while maintaining the performance on previous tasks. Source-free domain adaptation (SFDA), which adapts a pre-trained source model to a target domain, is useful in protecting the source domain data privacy. Generalized SFDA (G-SFDA) combines continual learning and SFDA to achieve outstanding performance on both the source and the target domains. This paper proposes semi-supervised G-SFDA (SSG-SFDA) for domain incremental learning, where a pre-trained source model (instead of the source data), few labeled target data, and plenty of unlabeled target data, are available. The goal is to achieve good performance on all domains. To cope with domain-ID agnostic, SSG-SFDA trains a conditional variational auto-encoder (CVAE) for each domain to learn its feature distribution, and a domain discriminator using virtual shallow features generated by CVAE to estimate the domain ID. To cope with catastrophic forgetting, SSG-SFDA uses soft domain attention to improve the sparse domain attention in G-SFDA. To cope with insufficient labeled target data, SSG-SFDA uses MixMatch to augment the unlabeled target data and better exploit the few labeled target data. Experiments on three datasets demonstrated the effectiveness of SSG-SFDA.**

***Index Terms*—Continual learning, source-free domain adaptation, semi-supervised learning, transfer learning**

I. Introduction

Continual learning [1], also known as lifelong learning [2], aims to learn on a sequence of tasks while keeping the performance on the previously learned tasks. Continual learning has three scenarios [3]:

1) Task incremental learning [4], where different tasks arrive at different moments. The algorithm knows clearly in both training and test what the current task is, so the model is usually trained using task-specific components, e.g., a separate network or output layer for each task. The challenge is to effectively share learned features across tasks, to improve the learning performance on one task using knowledge from other tasks, and to better trade-off performance and computational cost. An example is to train a single program to play different computer games, where it is always clear which game is being played.
2) Domain incremental learning (DIL) [5], where different domains with different data distributions arrive at different moments. It has some similarity with transfer learning [6]; however, transfer learning mainly focuses on the learning performance in the target domains, whereas DIL also needs to overcome catastrophic forgetting [7] to maintain the performance on previously learned domains. Different from task incremental learning, DIL does not know the domain ID at the test time. An example is learning to drive in all weather conditions [8].
3) Class incremental learning [9], where a sequence of tasks, each with different classes, arrive at different moments, and the algorithm needs to learn to distinguish among all classes. For example, in training, the algorithm first learns to classify desks and chairs in an office environment, and then flowers and trees in an outdoor environment; in test, it needs to classify all four classes without knowing the current environment (task) ID. A major challenge is learning to distinguish among classes not observed in the same task, especially when storing examples from previous tasks is not allowed.

Domain adaptation (DA) [10] aims to overcome the distribution shift between the source and target domains. It is very useful when the target domain does not have enough labeled data. It is different from continual learning, in that it focuses on the learning performance in the target domain only. Source-free domain adaptation (SFDA) [11], which adapts a pre-trained source model to the target domain, performs DA while protecting the source domain data privacy.

This paper considers semi-supervised generalized SFDA (SSG-SFDA) in DIL, where a pre-trained source model (instead of the source data, for privacy protection), few labeled target data, and plenty of unlabeled target data, are available. The goal is to achieve good classification performance in both source and target domains, without knowing the domain ID of an input.

SSG-SFDA has three challenges:

1) Domain-ID agnostic, which is usually true in real-world DIL. Generalized SFDA (G-SFDA) [5] solves it by saving a few data from each domain to train a domain discriminator.
2) Catastrophic forgetting [7], i.e., the performance of a model optimized for the target domain may have poor performance in the source domain. Replay, regularization, and parameter isolation are representative mitigation approaches [1].
3) Insufficient labeled target data, which easily result in overfitting in DA. This can be remedied by entropy minimization [12] or minimax entropy (MME) [13].

J. An, C. Zhao and D. Wu are with the Ministry of Education Key Laboratory of Image Processing and Intelligent Control, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China.

J. An and C. Zhao contributed equally to this work.

D. Wu is the corresponding author (e-mail: drwu@hust.edu.cn).

This paper proposes an SSG-SFDA approach to accommodate the above three challenges:

1) Domain-ID agnostic: SSG-SFDA trains a conditional variational auto-encoder (CVAE) [14] for each domain to learn its feature distribution, and a domain discriminator using virtual shallow features generated by CVAE to estimate the domain ID. SSG-SFDA does not need to save any data from the source domain, protecting their data privacy.
2) Catastrophic forgetting: SSG-SFDA uses soft domain attention to improve the sparse domain attention in G-SFDA [5]. Soft domain attention first uses the domain discriminator to output the domain probabilities of a test sample, and then weights the sparse domain attention vectors by their corresponding domain probabilities.
3) Insufficient labeled target data: SSG-SFDA uses MixMatch [15] to augment the unlabeled target data and better exploit the few labeled target data.

The remainder of this paper is organized as follows: Section II introduces related work. Section III proposes SSG-SFDA. Section IV presents the experimental results. Finally, Section V draws conclusions.

II. Related Work

A. Semi-Supervised DA

Many approaches have been proposed for semi-supervised DA, where few labeled and plenty of unlabeled target data are available.

Learning Invariant Representations and Risks [16] mimics unsupervised DA and derives an upper bound of the generalization error for semi-supervised DA. Deep co-training with Task Decomposition [17] decomposes semi-supervised DA into two sub-tasks, unsupervised DA across domains and semi-supervised learning in the target domain, and then performs co-training. Enhanced Categorical Alignment and Consistency Learning [18] performs unsupervised domain alignment, enhanced categorical alignment and consistency alignment to reduce the domain shift. Contrastive Learning for Domain Adaptation [19] uses contrastive learning to mitigate inter-domain and intra-domain variations. Minimax Entropy [13] and Prototypical Alignment and Consistency Learning [20] use prototypical alignment for DA. Bidirectional Adversarial Training [21], and Attract, Perturb, and Explore [22], augment the labeled target data by perturbations and then use them in adversarial training for DA. Adamatch [23] dynamically adjusts the pseudo-label confidence threshold to improves the pseudo-label quality. Consistent and Contrastive DA [24] extracts consistent representations from strongly and weakly augmented samples make use of the unlabeled samples.

B. Source-Free Domain Adaptation (SFDA)

SFDA adapts a pre-trained source model (instead of the source data) to the target domain.

Model Adaptation [25] uses generative adversarial networks [26] to generate target-style source data, and further imposes weighted clustering-based regularization constraints. Source Hypothesis Transfer (SHOT) [11] uses weighted k-nearest neighbors to update the pseudo-labels of the unlabeled target data, and then optimizes the information maximization loss. SHOT++ [27] combines SHOT and MixMatch [15]. It first uses SHOT to predict and assign pseudo-labels to the high-confidence target data, and then transforms SFDA into semi-supervised learning. Neighborhood Reciprocity Clustering (NRC) [28] uses the clustering hypothesis to capture the intrinsic structure of the target data, and proposes a self-regularization loss to reduce the negative impact of noisy neighbors.

C. Continual Learning

Continual learning focuses on mitigating catastrophic forgetting.

Incremental Classifier and Representation Learning [29] keeps representative data of previous tasks, and combines them with data from the current task to train the model. Learning without Forgetting [30] performs knowledge distillation by using the previous model's outputs as the previous task's soft labels.

Some approaches also adjust the model structure. Dynamically Expandable Network [31] calculates the relevance of the current and previous tasks to selectively expand the neural network's capacity. PathNet [32] consists of multiple layers, each with multiple modules. Each module, which can be any neural network type, tries to find the optimal route during training. Hard Attention to the Task [33] fixes the parameters of neurons relevant to previous tasks while training the current task. Meta-attention [34] sparsifies the multi-head attention and the fully connected layers of each ViT block.

D. G-SFDA

G-SFDA [5] is the most relevant approach to SSG-SFDA proposed in this paper. It assumes the target domain is completely unlabeled, and proposes local structure clustering to adapt the feature extractor to the target domain, and sparse domain attention to mitigate catastrophic forgetting. However, G-SFDA needs to save a few data in each domain to train a domain discriminator, raising privacy concerns. Additionally, it is not proposed for semi-supervised SFDA, where the target domain has a small number of labeled samples.

More specifically, G-SFDA considers unsupervised SFDA, where there are n_s labeled source domain samples $\{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^{n_s}$, and n_t unlabeled target domain samples $\{\boldsymbol{x}_i^t\}_{i=1}^{n_t}$. It trains a single neural network, which consists of a classifier g (the last fully connected layer of the network) and a feature extractor f (the rest of the network). Assume there are C classes. The output of the network is a class probability vector $p(\boldsymbol{x}) = [p_1(\boldsymbol{x}), \ldots, p_C(\boldsymbol{x})]^\intercal = g(f(\boldsymbol{x}))$.

To mitigate catastrophic forgetting and provide a good initialization for the target domain, G-SFDA trains a sparse domain attention vector $\mathcal{A}_s$ for the source domain, and $\mathcal{A}_t$ for the target domain:

$$\mathcal{A}_s = \text{sigmoid}(100\boldsymbol{e}_s), \quad \boldsymbol{e}_s \in \mathcal{R}^d, \tag{1}$$

$$\mathcal{A}_t = \text{sigmoid}(100\boldsymbol{e}_t), \quad \boldsymbol{e}_t \in \mathcal{R}^d, \tag{2}$$

where $\boldsymbol{e}_s$ and $\boldsymbol{e}_t$ are the outputs of the embedding layer for the source and target domains, respectively, d is the feature dimensionality, and the coefficient 100 makes $\mathcal{A}_s$ and $\mathcal{A}_t$ near binary (to impose sparsity) but still differentiable.

The embedding layer consists of a 1-layer neural network with $2d$ nodes, whose input is a fixed random vector. The outputs of the first d nodes form $\boldsymbol{e}_s$, and the last d nodes form $\boldsymbol{e}_t$. The embedding layer is optimized during training, together with f and g:

$$\begin{aligned}\mathcal{L}_{\text{source}} = &\sum_{i=1}^{n_s} \text{CE}(g(f(\boldsymbol{x}_i^s) \odot \mathcal{A}_s), y_i^s) \\ &+ \sum_{i=1}^{n_s} \text{CE}(g(f(\boldsymbol{x}_i^s) \odot \mathcal{A}_t), y_i^s) + \sum_{c=1}^{C} \text{KL}(\overline{p}_c^s || q_c) \quad (3)\end{aligned}$$

where CE is the cross-entropy, $\odot$ denotes element-wise multiplication, KL is the KL divergence, $\overline{p}_c^s = \frac{1}{n_s}\sum_{i=1}^{n_s} p_c(\boldsymbol{x}_i^s)$, and $q_c = \frac{1}{C}$.

With $\mathcal{A}_s$ and $\mathcal{A}_t$ learned during the training of the source domain, the outputs of the source and target domains become $g(f(\boldsymbol{x}) \odot \mathcal{A}_s)$ and $g(f(\boldsymbol{x}) \odot \mathcal{A}_t)$, respectively.

Additionally, G-SFDA uses local structure clustering (LSC) to cluster each target sample with its K semantically similar neighbors, adapting the model to the target domain without using source data:

$$\begin{aligned}\mathcal{L}_{\text{LSC}} = &-\frac{1}{n_t}\sum_{i=1}^{n_t}\sum_{k=1}^{K} \log\left[p(\boldsymbol{x}_i^t) \cdot p(\boldsymbol{x}_{i,k}^t)\right] \\ &+ \sum_{c=1}^{C} \text{KL}(\overline{p}_c^t || q_c), \end{aligned} \quad (4)$$

where $\boldsymbol{x}_{i,k}^t$ is a target sample whose $f(\boldsymbol{x}_{i,k}^t)$ is the k-th nearest neighbor of $f(\boldsymbol{x}_i^t)$ under the cosine similarity, and $\overline{p}_c^t = \frac{1}{n_t}\sum_{i=1}^{n_t} p_c(\boldsymbol{x}_i^t)$.

The first term in (4) enforces consistent predictions between each target sample and its K nearest neighbors, and the second term encourages prediction balance to avoid degenerated solution.

Let $\mathbb{1}_d \in \mathcal{R}^d$ be a vector of all ones. When training in the target domain, G-SFDA uses $(\mathbb{1}_d - \mathcal{A}_s)$ to mask the backward propagation gradients of f and g, so that the gradients related to the source domain are minimally affected:

$$W_{f_{\text{last}}} \leftarrow W_{f_{\text{last}}} - \frac{\partial \mathcal{L}_{\text{LSC}}}{\partial W_{f_{\text{last}}}} \odot (\mathbb{1}_d - \mathcal{A}_s), \quad (5)$$

$$W_g \leftarrow W_g - (\mathbb{1}_d - \mathcal{A}_s) \odot \frac{\partial \mathcal{L}_{\text{LSC}}}{\partial W_g}, \quad (6)$$

where $W_{f_{\text{last}}}$ are the weights of the last layer in f, and W_g are the weights in g.

III. SSG-SFDA

This section introduces our proposed SSG-SFDA, shown in Fig. 1.

A. Problem Setting

Consider semi-supervised SFDA, where there are n_l labeled target domain samples $\{(\boldsymbol{x}_i^t, y_i^l)\}_{i=1}^{n_l}$, and n_u unlabeled target domain samples $\{\boldsymbol{x}_i^u\}_{i=1}^{n_u}$. There are also pre-trained models from the source domain, including a feature extractor f, a classifier g, a source domain sparse domain attention vector $\mathcal{A}_s$, a target domain sparse domain attention vector $\mathcal{A}_t$, and a source domain CVAE decoder f_s^{Dec}.

Let M be the batch size, and $\mathcal{T}_l$ and $\mathcal{T}_u$ the index sets of labeled and unlabeled target data in a batch, i.e., $|\mathcal{T}_l| = |\mathcal{T}_u| = M$.

B. Model Architecture

SSG-SFDA in Fig. 1 consists of four parts:

1) Feature extractor f. We used ViT [35], which has demonstrated good transferability in unsupervised DA [36].
2) Domain discriminator f_{domain}, which estimates the domain ID of an input sample in domain-ID agnostic applications.
   SSG-SFDA first extracts shallow features from the 2nd block of f, and then generates virtual shallow features using f_s^{Dec}. Finally, SSG-SFDA minimizes the following discrimination loss:

$$\begin{aligned}\mathcal{L}_{\text{Dis}} = &\frac{1}{M}\sum_{m\in\mathcal{T}_l} \text{CE}([0,1]^\intercal, p_{\text{domain}}(f_{\text{shallow}}(\boldsymbol{x}_m^l))) \\ &+ \frac{1}{M}\sum_{m\in\mathcal{T}_u} \text{CE}([0,1]^\intercal, p_{\text{domain}}(f_{\text{shallow}}(\boldsymbol{x}_m^u))) \\ &+ \frac{1}{2M}\sum_{m=1}^{2M} \text{CE}([1,0]^\intercal, p_{\text{domain}}(f_s^{\text{Dec}}(\boldsymbol{z}_m))), \end{aligned} \quad (7)$$

   where

$$\begin{aligned}&p_{\text{domain}}(f_{\text{shallow}}(\boldsymbol{x})) \\ =&\text{softmax}(f_{\text{domain}}(f_{\text{shallow}}(\boldsymbol{x}))), \end{aligned} \quad (8)$$

   and CE is the cross-entropy between the domain-ID vector and the prediction probabilities of the domain discriminator. Note that $2M$ (instead of M) source domain samples are generated and used in (7) to balance the number of samples from the source and target domains.
3) Soft domain attention, which is used to overcome catastrophic forgetting.
   SSG-SFDA first uses f_{domain} to output the domain probabilities of an input sample, and then weights the sparse domain attention vectors to output the class probabilities:

$$p(\boldsymbol{x}) = g\left(f(\boldsymbol{x}) \odot \mathcal{A}\right), \quad (9)$$

   where

$$\mathcal{A} = p_{\text{domain}}^s(\boldsymbol{x}) \times \mathcal{A}_s + p_{\text{domain}}^t(\boldsymbol{x}) \times \mathcal{A}_t, \quad (10)$$

   in which $p_{\text{domain}}^s(\boldsymbol{x})$ and $p_{\text{domain}}^t(\boldsymbol{x})$ are elements in $p_{\text{domain}}(\boldsymbol{x})$ corresponding to the source and target domain probabilities, respectively. Note that both $\mathcal{A}_s$ and $\mathcal{A}_t$ has been trained on the source domain.

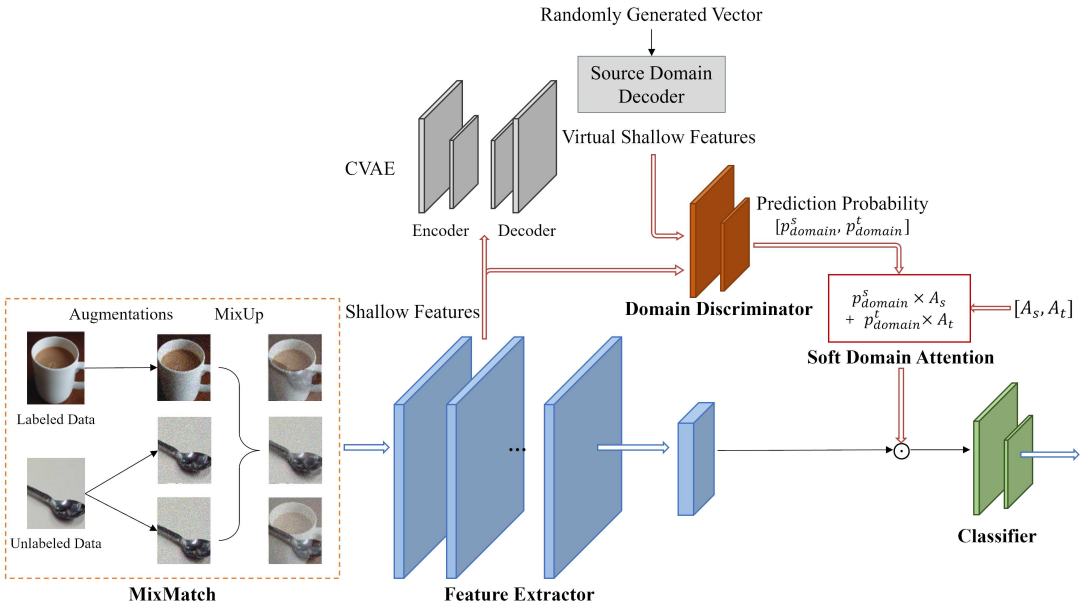

Fig. 1. Semi-supervised generalized source-free domain adaptation (SSG-SFDA).

4) Semi-supervised module MixMatch, which effectively exploits labeled and unlabeled target data.
MixMatch augments labeled and unlabeled target data one and Q times, respectively:

$$\hat{X}^l = \{(\text{augment}(\boldsymbol{x}^l_m), y_m)\}_{m\in\mathcal{T}_l}, \tag{11}$$

$$\hat{X}^u = \{(\text{augment}(\boldsymbol{x}^u_m)_q, p^u_m)\}_{m\in\mathcal{T}_u,\ q=1,\ldots,Q}, \tag{12}$$

where $\text{augment}(\boldsymbol{x})$ augments $\boldsymbol{x}$ by adding noise and then performing random cropping, and

$$p^u_m = \frac{1}{Q}\sum_{q=1}^{Q} p(\text{augment}(\boldsymbol{x}^u_m)_q)^\tau, \tag{13}$$

in which τ sharpens the probabilities. $Q = 2$ and $\tau = 2$ were used in our experiments.
Then, $\hat{X}^l$ and $\hat{X}^u$ are concatenated to form $W = [\hat{X}^l; \hat{X}^u]$. The rows of W are next randomly shuffled. Let

$$(\widetilde{\boldsymbol{x}}^l_m, p^l_m) = \text{MixUp}(\hat{X}^l_m,\ W_m), \tag{14}$$

$$(\widetilde{\boldsymbol{x}}^u_m, p^u_m) = \text{MixUp}(\hat{X}^u_m,\ W_{m+M}), \tag{15}$$

where $\text{MixUp}(X_m, W_m)$ is the MixUp [37] operation to calculate the random linear combination of X_m (the mth row of X) and W_m (the mth row of W).
The MixMatch loss is calculated as:

$$\mathcal{L}_{\text{MixMatch}} = \frac{1}{M}\sum_{m=1}^{M} \text{CE}\left(p^l_m, p(\widetilde{\boldsymbol{x}}^l_m)\right) + \frac{\lambda}{M}\sum_{m=1}^{M} \|p^u_m - p\left(\widetilde{\boldsymbol{x}}^u_m\right)\|_2^2, \tag{16}$$

where λ is a trade-off parameter, and $\lambda = 100$ was used in our experiments.

Finally, SSG-SFDA combines the above four modules, and optimizes the following loss function:

$$\mathcal{L} = \mathcal{L}_{\text{LSC}} + \mathcal{L}_{\text{Dis}} + \mathcal{L}_{\text{MixMatch}}. \tag{17}$$

In the case that more target domains will arrive later, SSG-SFDA also trains a CVAE [14] to learn the shallow feature distribution of the current target domain, so that we do not need to save any data from it.

CVAE consists of an encoder f^{Enc} and a decoder f^{Dec}. Its loss function is:

$$\mathcal{L}_{\text{CVAE}} = \frac{1}{2}\left(-\log \boldsymbol{v}^2 + \boldsymbol{\mu}^2 + \boldsymbol{v}^2\right) + \sum_{m=1}^{M} ||f_{\text{shallow}}(\boldsymbol{x}_m) - f^{\text{Dec}}(\boldsymbol{z}_m, y_m)||_2^2, \tag{18}$$

where f_{shallow} extracts shallow features from the 2nd block of f, $\boldsymbol{\mu}$ and $\boldsymbol{v}^2$ are the mean and variance of $\{f^{\text{Enc}}(f_{\text{shallow}}(\boldsymbol{x}_m), y_m)\}_{m=1}^M$, respectively, and $\boldsymbol{z}_m$ is a randomly generated vector. The first term in (18) encourages the shallow features to follow a Gaussian distribution, and the second term enforces the decoder to reconstruct the input data.

The pseudo-code of SSG-SFDA is shown in Algorithm 1.

C. SSG-SFDA for Multiple Target Domains

We just introduced SSG-SFDA for a single target domain. This subsection extends it to multiple target domains arriving sequentially.

At any time, SSG-SFDA only has access to data from the latest target domain, i.e., it does not save any data from any previous domains to protect their privacy. However, when the nth target domain arrives, SSG-SFDA already has $\mathcal{A}_s$ and $\{\mathcal{A}_{t,i}\}_{i=1}^{n-1}$, where $\mathcal{A}_{t,i}$ is the sparse domain attention vector for the ith target domain, from previous training.

Algorithm 1: SSG-SFDA for one target domain.

Input: $\{(\boldsymbol{x}_i^l, y_i^l)\}_{i=1}^{n_l}$, n_l labeled target domain samples;
$\{\boldsymbol{x}_i^u\}_{i=1}^{n_u}$, n_u unlabeled target domain samples;
Pre-trained models, including feature extractor f, classifier g, source domain sparse domain attention vector $\mathcal{A}_s$, target domain sparse domain attention vector $\mathcal{A}_t$, source domain CVAE decoder f_s^{Dec};
M, batch size.
Output: The classification model.

Initialize $\mathcal{A}$ in (10);
Initialize feature bank $\mathcal{F} = \{f(\boldsymbol{x}_i^l) \odot \mathcal{A}\}_{i=1}^{n_l}$;
while Adaptation **do**
- Sample M indices $\mathcal{T}_l$ and $\mathcal{T}_u$ from the labeled and unlabeled target data, respectively;
- Generate shallow features of samples in $\mathcal{T}_l$ and $\mathcal{T}_u$ using the first two blocks of f;
- Generate source domain virtual shallow features;
- Compute domain discriminator loss $\mathcal{L}_{\mathrm{Dis}}$ by (7);
- Compute the prediction probability $p(\boldsymbol{x})$ by (9);
- Compute the feature with soft domain attention $f(\boldsymbol{x}) \odot \mathcal{A}$ to update feature bank $\mathcal{F}$;
- Compute local structure clustering loss $\mathcal{L}_{\mathrm{LSC}}$ from $\mathcal{F}$ by (4);
- Compute MixMatch loss $\mathcal{L}_{\mathrm{MixMatch}}$ by (16);
- Compute $\mathcal{L}$ by (17);
- Mask the gradients of f and g by (5) and (6), respectively, and update the network except CVAE;
- Compute CVAE loss $\mathcal{L}_{\mathrm{CVAE}}$ by (18) and update CVAE;

end

For the nth target domain:

1) To cope with catastrophic forgetting to any previous domains, SSG-SFDA calculates the maximum of all previous domains' sparse domain attention vectors to replace $\mathcal{A}_s$ in the previous subsection:

$$\mathcal{A}' = \max(\mathcal{A}_s, \mathcal{A}_{t,1}, \cdots, \mathcal{A}_{t,n-1}), \tag{19}$$

where $\max(\cdot)$ is an element-wise operation.
2) When training the domain discriminator, SSG-SFDA first randomly selects a CVAE from previous domains to generate its virtual shallow features, then mixes them with shallow features of the nth target domain to train the domain discriminator.

IV. Experiments

This section presents experimental results to demonstrate the effectiveness of SSG-SFDA.

A. Datasets

Three datasets were used in our experiments:

1) Office-31 [38], which contains 31 classes and three domains (A: Amazon, W: Webcam, D: DSLR).
2) Office-Home [39], which contains 65 classes and four domains (R: Real, C: Clipart, A: Art, P: Product).
3) VisDA-2017 [40], which contains 12 classes and two domains. The source domain contains 152k synthetic images, and the target domain has 55k real object images.

B. Baselines

Source only (SourceOnly) was a baseline trained using source domain data only. Additional baselines included SHOT [11], SHOT++ [27], NRC [28] (ResNet101 as backbone) and G-SFDA [5] (ResNet50 as backbone). Note that we modified the latter four to use the labeled target data, by adding a supervised training loss.

C. Implementation Details

SourceOnly used ViT pre-trained on ImageNet as the feature extractor. SHOT, SHOT++ and NRC used ResNet-50 pre-trained on ImageNet as the feature extractor. G-SFDA used ResNet-50 or ViT pre-trained on ImageNet as the feature extractor. AdamW[1] with betas $(0.9, 0.999)$, initial learning rate 0.01, and weight decay 0.0001 was used to train the networks. The learning rate on VisDA-2017 was set 10 times smaller. Other settings followed their original papers.

The average classification accuracy across all domains was used as the performance measure. The floating-point operations per second (FLOPs) was used to measure the computational complexity of each model.

All code was implemented in PyTorch, and ran on a server with an NVIDIA RTX 3090 GPU and 112G RAM.

D. Results

Tables I-III compare SSG-SFDA with the five baselines on the three datasets in semi-supervised SFDA with only one target domain. The best accuracies are marked in bold. SSG-SFDA improved the average classification accuracy of G-SFDA by 1.1%, 1.2%, and 1.6% on Office-31, Office-Home and VisDA-2017, respectively.

Tables IV and V compare SSG-SFDA with three baselines in semi-supervised SFDA on the first two datasets with multiple target domains. SSG-SFDA improved the average classification accuracy of G-SFDA by 0.5% and 2.7% on Office-31 and Office-Home, respectively.

Compared with GSFDA, SSG-SFDA only adds a domain discriminator with two fully-connected layers, thus their FLOPs are about the same. Compared with approaches using ResNet as the backbone, SSG-SFDA's increase of FLOPs is obvious; however, this computational cost increase is offset by larger classification improvements.

In summary, SSG-SFDA achieved the highest average classification accuracies on all datasets in semi-supervised SFDA, with one or multiple target domains, demonstrating the effectiveness of SSG-SFDA.

[1] https://pytorch.org/docs/stable/generated/torch.optim.AdamW

TABLE I
CLASSIFICATION ACCURACIES (%) ON OFFICE-31 IN 3-SHOT SEMI-SUPERVISED SFDA.

Approach		A→D		A→W		D→A		D→W		W→A		W→D		Average			FLOPs (G)
		S	T	S	T	S	T	S	T	S	T	S	T	S	T	mean	
SHOT	ResNet	80.50	94.81	78.37	90.31	92.00	74.89	100.00	97.01	88.75	72.61	98.75	99.75	89.73	88.23	88.98	10.24
SHOT++	ResNet	84.40	96.05	84.04	91.31	86.00	75.40	100.00	97.86	88.75	73.13	98.75	99.75	90.32	88.92	89.62	10.24
NRC	ResNet	82.27	89.38	80.85	92.17	90.00	77.17	100.00	96.87	91.25	78.23	100.00	96.30	90.73	88.35	89.54	10.24
G-SFDA	ResNet	85.99	95.80	86.35	91.45	96.00	75.88	98.00	97.86	98.74	75.48	98.74	99.51	93.97	89.33	91.65	5.38
SourceOnly	ViT	92.91	85.93	92.91	87.61	100.00	79.92	100.00	98.58	100.00	81.09	100.00	100.00	**97.64**	88.86	93.25	22.03
G-SFDA	ViT	89.72	97.04	91.67	97.72	99.00	82.27	98.00	99.15	96.23	83.22	100.00	99.26	95.77	93.11	94.44	22.03
SSG-SFDA	ViT	89.36	100.00	90.96	97.86	99.00	85.24	100.00	100.00	99.37	85.50	100.00	99.51	96.45	**94.69**	**95.57**	22.03

TABLE II
CLASSIFICATION ACCURACIES (%) ON OFFICE-HOME IN 3-SHOT SEMI-SUPERVISED SFDA.

Approach		A→C		A→P		A→R		C→A		C→P		C→R		P→A	
		S	T	S	T	S	T	S	T	S	T	S	T	S	T
SHOT	ResNet	73.66	56.31	73.66	79.15	80.66	81.91	75.29	69.40	73.00	78.37	74.14	79.29	89.41	66.53
SHOT++	ResNet	72.43	57.41	69.96	79.57	81.07	82.94	73.23	70.88	62.47	79.45	65.45	80.11	85.14	68.46
NRC	ResNet	73.25	62.28	73.66	79.76	81.07	80.30	77.35	67.79	78.03	81.22	67.51	78.40	92.34	68.64
G-SFDA	ResNet	70.78	57.94	73.46	79.22	74.07	79.67	79.50	62.81	79.61	74.55	78.58	74.77	92.23	65.19
SourceOnly	ViT	85.80	65.78	85.80	84.19	85.80	88.08	88.32	80.24	88.32	85.27	88.32	86.42	96.28	76.08
G-SFDA	ViT	84.16	75.16	83.74	89.70	85.19	90.20	86.25	85.08	87.51	90.48	87.17	90.68	94.37	83.87
SSG-SFDA	ViT	83.95	79.33	84.36	91.59	86.21	91.09	86.03	86.69	87.63	91.82	87.97	90.63	95.50	84.45

Approach		P→C		P→R		R→A		R→C		R→P		Average			FLOPs (G)
		S	T	S	T	S	T	S	T	S	T	S	T	mean	
SHOT	ResNet	83.33	53.81	91.44	81.81	91.06	74.55	84.40	57.94	91.06	83.84	81.76	71.91	76.84	10.24
SHOT++	ResNet	77.48	55.20	86.94	82.46	90.60	76.34	78.44	59.26	88.53	84.94	77.65	73.09	75.37	10.24
NRC	ResNet	84.91	62.90	88.29	79.00	91.28	71.24	83.26	63.26	91.28	83.20	81.85	73.17	77.51	10.24
G-SFDA	ResNet	89.64	60.38	92.68	80.59	86.24	71.95	82.80	63.48	88.07	85.04	82.30	71.30	76.80	5.38
SourceOnly	ViT	96.28	63.33	96.28	87.72	92.78	81.14	92.78	66.74	92.78	88.90	**90.80**	79.49	85.14	22.03
G-SFDA	ViT	95.38	72.21	95.61	89.81	92.09	84.01	92.09	74.56	92.66	90.39	89.69	84.68	87.18	22.03
SSG-SFDA	ViT	95.61	78.18	96.06	90.53	92.20	86.07	91.97	77.46	93.23	92.67	90.06	**86.71**	**88.39**	22.03

TABLE III
CLASSIFICATION ACCURACIES (%) ON VISDA-2017 IN 3-SHOT SEMI-SUPERVISED SFDA.

Approach		plane		bicycle		bus		car		horse		knife		motorcycle	
		S	T	S	T	S	T	S	T	S	T	S	T	S	T
SHOT	ResNet	95.00	98.62	85.46	96.87	70.98	35.76	36.30	50.84	93.17	85.60	20.13	84.89	80.99	46.69
SHOT++	ResNet	99.79	96.65	98.13	89.37	38.30	83.32	71.84	94.42	96.93	97.27	22.83	95.51	46.15	91.61
NRC	ResNet	99.58	93.25	99.59	83.06	88.45	81.27	78.80	68.22	94.09	92.85	100.00	95.37	74.96	87.86
G-SFDA	ResNet	99.73	96.62	99.44	88.22	91.53	85.00	80.03	74.97	98.21	96.33	99.38	95.95	77.31	90.30
SourceOnly	ViT	99.93	99.12	100.0	55.67	99.29	77.60	99.36	75.42	100.0	92.83	99.86	65.78	99.77	93.60
G-SFDA	ViT	99.86	98.79	100.00	73.59	99.35	86.22	96.57	79.19	100.00	96.93	100.00	91.26	99.48	95.75
SSG-SFDA	ViT	98.75	98.68	99.86	90.24	98.46	88.78	95.62	93.54	100.00	98.14	100.00	97.49	97.18	94.55

Approach		person		plant		skateboard		train		truck		Average			FLOPs (G)
		S	T	S	T	S	T	S	T	S	T	S	T	mean	
SHOT	ResNet	80.99	95.45	90.26	99.44	83.80	100.00	90.48	80.10	54.43	53.35	73.50	77.30	75.40	10.24
SHOT++	ResNet	97.85	90.77	99.91	96.96	95.85	93.94	79.23	95.02	89.53	61.68	78.03	**90.54**	84.29	10.24
NRC	ResNet	97.05	82.19	97.96	91.20	96.98	82.66	89.30	87.60	91.84	49.27	92.38	82.90	87.64	10.24
G-SFDA	ResNet	98.17	80.44	99.22	95.56	98.90	92.23	87.79	89.84	88.30	47.09	93.17	86.04	89.61	5.38
SourceOnly	ViT	100.0	19.06	100.00	76.02	99.92	95.04	99.75	95.91	99.27	10.86	**99.76**	71.41	85.59	22.03
G-SFDA	ViT	100.00	65.35	100.0	89.40	99.83	97.59	99.12	95.65	99.48	25.48	99.47	82.93	91.20	22.03
SSG-SFDA	ViT	99.92	68.95	100.00	94.90	98.41	98.55	98.56	95.72	98.85	21.71	98.80	86.77	**92.79**	22.03

E. Ablation Study

Two key difference between SSG-SFDA and the baselines is the integration of MixMatch and the domain discriminator. This subsection studies their separate effects.

1) The Effect of MixMatch: Table VI compares SSG-SFDA with and without MixMatch in semi-supervised SFDA with one or multiple target domains. SSG-SFDA with MixMatch always outperformed SSG-SFDA without MixMatch, indicat-

TABLE IV
CLASSIFICATION ACCURACIES (%) IN 3-SHOT SEMI-SUPERVISED SFDA ON OFFICE-31 WITH MULTIPLE TARGET DOMAINS.

Approach		A→D→W				D→A→W				W→A→D				Average				FLOPs (G)
		S	T_1	T_2	mean	S	T_1	T_2	mean	S	T_1	T_2	mean	S	T_1	T_2	mean	
G-SFDA	ResNet	87.23	95.56	93.30	92.03	98.00	74.41	99.00	90.47	99.37	71.77	99.75	90.30	94.87	80.58	97.35	90.93	5.38
SourceOnly	ViT	**92.73**	85.19	85.61	87.84	**100.00**	81.31	98.58	93.30	**100.00**	81.86	100.00	93.95	**97.58**	82.79	94.73	91.70	22.03
G-SFDA	ViT	89.54	98.77	98.15	95.49	98.00	**84.07**	99.00	93.69	**100.00**	82.93	99.75	94.23	95.85	88.59	98.97	94.47	22.03
SSG-SFDA	ViT	89.54	**99.26**	**99.43**	**96.19**	**100.00**	82.82	**99.29**	**94.04**	**100.00**	**83.73**	**100.00**	**94.58**	96.63	**88.60**	**99.57**	**94.94**	22.03

TABLE V
CLASSIFICATION ACCURACIES (%) IN 3-SHOT SEMI-SUPERVISED SFDA ON OFFICE-HOME WITH MULTIPLE TARGET DOMAINS.

Approach		A→C→P→R					C→A→P→R					P→A→C→R				
		S	T_1	T_2	T_3	mean	S	T_1	T_2	T_3	mean	S	T_1	T_2	T_3	mean
G-SFDA	ResNet	73.46	58.20	76.32	76.86	71.21	76.86	59.86	71.91	69.56	69.55	90.65	65.64	54.82	79.17	72.57
SourceOnly	ViT	86.21	66.35	84.61	88.18	81.34	**88.55**	80.15	86.17	86.52	85.35	**96.40**	76.70	62.57	88.30	80.99
G-SFDA	ViT	82.92	70.67	89.77	90.22	83.40	81.56	84.86	87.44	88.95	85.70	93.13	81.59	69.14	89.00	83.21
SSG-SFDA	ViT	**86.42**	**77.03**	**90.67**	**91.30**	**86.33**	84.42	**85.66**	**91.23**	**90.87**	**88.00**	94.48	**84.50**	**79.09**	**88.66**	**86.53**

Approach		R→A→C→P					Average					FLOPs (G)
		S	T_1	T_2	T_3	mean	S	T_1	T_2	T_3	mean	
G-SFDA	ResNet	85.21	70.70	59.83	82.92	74.66	81.55	63.60	65.72	77.13	72.00	5.38
SourceOnly	ViT	92.09	80.73	65.80	89.14	81.94	**90.81**	75.98	74.79	88.04	82.40	22.03
G-SFDA	ViT	92.09	82.75	71.51	92.20	84.64	87.43	79.97	79.47	90.09	84.24	22.03
SSG-SFDA	ViT	**93.12**	**85.04**	**76.57**	**92.27**	**86.82**	89.64	**82.99**	**83.81**	**91.24**	**86.92**	22.03

TABLE VI
AVERAGE CLASSIFICATION ACCURACIES (%) OF SSG-SFDA WITHOUT MIXMATCH AND SSG-SFDA WITH MIXMATCH ON THE THREE DATASETS.

Dataset	Office-31	Office-Home	VisDA-2017
	Semi-supervised SFDA with one target domain		
SSG-SFDA without MixMatch	94.44	87.18	91.20
SSG-SFDA with MixMatch	**95.57**	**88.39**	**92.79**
	Semi-supervised SFDA with multiple target domains		
SSG-SFDA without MixMatch	94.47	84.24	–
SSG-SFDA with MixMatch	**94.94**	**86.92**	–

TABLE VII
AVERAGE CLASSIFICATION ACCURACIES (%) OF SSG-SFDA WITHOUT DOMAIN DISCRIMINATOR AND SSG-SFDA WITH DOMAIN DISCRIMINATOR ON THE THREE DATASETS.

Dataset	Office-31	Office-Home	VisDA-2017
	Semi-supervised SFDA with one target domain		
SSG-SFDA without domain discriminator	94.44	87.18	91.20
SSG-SFDA with domain discriminator	**95.13**	**87.60**	**92.30**
	Semi-supervised SFDA with multiple target domains		
SSG-SFDA without domain discriminator	94.47	84.24	–
SSG-SFDA with domain discriminator	**94.82**	**84.87**	–

ing that MixMatch helped exploit the labeled target data.

2) The Effect of Domain Discriminator: Table VII compares SSG-SFDA with and without domain discriminator in semi-supervised SFDA with one or multiple target domains. SSG-SFDA with domain discriminator always outperformed SSG-SFDA without domain discriminator, indicating that our proposed soft domain attention, which integrates a domain discriminator and sparse domain attention, made SSG-SFDA more suitable for domain-ID agnostic DIL.

V. CONCLUSION

This paper has integrated CVAE, MixMatch and G-SFDA to propose SSG-SFDA, which trains a model in the current domain without forgetting the knowledge learned in previous domains. To cope with domain-ID agnostic, SSG-SFDA trains a CVAE for each domain to learn its feature distribution, and a domain discriminator using virtual shallow features generated by CVAE to estimate the domain ID. To cope with catastrophic forgetting, SSG-SFDA uses soft domain attention to improve the sparse domain attention in G-SFDA. To cope with insufficient labeled target data, SSG-SFDA uses

MixMatch to augment the unlabeled target data and better exploit the few labeled target data. Experiments on three datasets demonstrated the effectiveness of SSG-SFDA.

ACKNOWLEDGEMENTS

This research was supported by STI 2030-Major Project 2021ZD0201300, and the Hubei Province Funds for Distinguished Young Scholars under Grant 2020CFA050.

REFERENCES

[1] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

[2] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[3] G. M. van de Ven, T. Tuytelaars, and A. S. Tolias, "Three types of incremental learning," *Nature Machine Intelligence*, pp. 1–13, 2022.

[4] D. Maltoni and V. Lomonaco, "Continuous learning in single-incremental-task scenarios," *Neural Networks*, vol. 116, pp. 56–73, 2019.

[5] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, and S. Jui, "Generalized source-free domain adaptation," in *Proc. Int'l Conf. on Computer Vision*, Virtual Event, Oct. 2021, pp. 8978–8987.

[6] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.

[7] J. Kirkpatrick, R. Pascanu, Rabinowitz *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[8] M. Jehanzeb Mirza, M. Masana, H. Possegger, and H. Bischof, "An efficient domain-incremental learning approach to drive in all weather conditions," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops*, New Orleans, LA, Jun. 2022, pp. 3000–3010.

[9] S. Mittal, S. Galesso, and T. Brox, "Essentials for class incremental learning," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Virtual Event, Jun. 2021, pp. 3513–3522.

[10] L. Zhang and X. Gao, "Transfer adaptation learning: A decade survey," *arXiv:1903.04687*, 2019.

[11] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," in *Proc. Int'l Conf. on Machine Learning*, Virtual Event, Jul. 2020, pp. 6028–6039.

[12] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. Advances in Neural Information Processing Systems*, Vancouver, Canada, Dec. 2004, pp. 529–536.

[13] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," in *Proc. Int'l Conf. on Computer Vision*, Seoul, Korea, Oct. 2019, pp. 8050–8058.

[14] C. Doersch, "Tutorial on variational autoencoders," *arXiv:1606.05908*, 2016.

[15] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Proc. Advances in Neural Information Processing Systems*, vol. 32, Vancouver, Canada, Dec. 2019.

[16] B. Li, Y. Wang, S. Zhang, D. Li, K. Keutzer, T. Darrell, and H. Zhao, "Learning invariant representations and risks for semi-supervised domain adaptation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Virtual Event, Jun. 2021, pp. 1104–1113.

[17] L. Yang, Y. Wang, M. Gao, A. Shrivastava, K. Q. Weinberger, W.-L. Chao, and S.-N. Lim, "Deep co-training with task decomposition for semi-supervised domain adaptation," in *Proc. Int'l Conf. on Computer Vision*, Virtual Event, Oct. 2021, pp. 8906–8916.

[18] K. Li, C. Liu, H. Zhao, Y. Zhang, and Y. Fu, "ECACL: A holistic framework for semi-supervised domain adaptation," in *Proc. Int'l Conf. on Computer Vision*, Virtual Event, Oct. 2021, pp. 8578–8587.

[19] A. Singh, "CLDA: Contrastive learning for semi-supervised domain adaptation," in *Proc. Advances in Neural Information Processing Systems*, vol. 34, Virtual Event, Dec. 2021, pp. 5089–5101.

[20] K. Li, C. Liu, H. Zhao, Y. Zhang, and Y. Fu, "Semi-supervised domain adaptation with prototypical alignment and consistency learning," *arXiv:2104.09136*, 2021.

[21] P. Jiang, A. Wu, Y. Han, Y. Shao, M. Qi, and B. Li, "Bidirectional adversarial training for semi-supervised domain adaptation." in *Proc. Int'l Joint Conf. on Artificial Intelligence*, Yokohama, Jap, Jan. 2020, pp. 934–940.

[22] T. Kim and C. Kim, "Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation," in *Proc. European Conf. on Computer Vision*, Glasgow, UK, Aug. 2020, pp. 591–607.

[23] D. Berthelot, R. Roelofs, K. Sohn, N. Carlini, and A. Kurakin, "Adamatch: A unified approach to semi-supervised learning and domain adaptation," in *Proc. Int'l Conf. on Learning Representations*, Virtual Event, Jun. 2022.

[24] M. I. Pérez-Carrasco, P. Protopapas, and G. Cabrera-Vives, "Con^{2}da: Simplifying semi-supervised domain adaptation by learning consistent and contrastive feature representations," in *Proc. Advances in Neural Information Processing Systems*, Virtual Event, Dec. 2021.

[25] R. Li, Q. Jiao, W. Cao, H.-S. Wong, and S. Wu, "Model adaptation: Unsupervised domain adaptation without source data," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, Jun. 2020, pp. 9641–9650.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[27] J. Liang, D. Hu, Y. Wang, R. He, and J. Feng, "Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8602–8617, 2022.

[28] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, and S. Jui, "Exploiting the intrinsic neighborhood structure for source-free domain adaptation," in *Proc. Advances in Neural Information Processing Systems*, Virtual Event, Dec. 2021, pp. 29 393–29 405.

[29] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, Jul. 2017, pp. 2001–2010.

[30] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[31] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," in *Proc. Int'l Conf. on Learning Representations*, Vancouver, Canada, Apr. 2018.

[32] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra, "Pathnet: Evolution channels gradient descent in super neural networks," *arXiv:1701.08734*, 2017.

[33] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *Proc. Int'l Conf. on Machine Learning*, Stockholm, Sverige, Jul. 2018, pp. 4548–4557.

[34] M. Xue, H. Zhang, J. Song, and M. Song, "Meta-attention for ViT-backed continual learning," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, New Orleans, LA, Jun. 2022, pp. 150–159.

[35] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int'l Conf. on Learning Representations*, Virtual Event, Apr. 2021.

[36] J. Yang, J. Liu, N. Xu, and J. Huang, "TVT: Transferable vision transformer for unsupervised domain adaptation," *arXiv:2108.05988*, 2021.

[37] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[38] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. European Conf. on Computer Vision*, Heraklion, Greece, Sep. 2010, pp. 213–226.

[39] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, Jul. 2017, pp. 5018–5027.

[40] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, "VisDA: The visual domain adaptation challenge," *arXiv:1710.06924*, 2017.