Cross-domain few-shot learning via adaptive transformer networks

Naeem Paeedeh [a],[1], Mahardhika Pratama [a],[*],[1], Muhammad Anwar Ma'sum [a], Wolfgang Mayer [a], Zehong Cao [a], Ryszard Kowlczyk [a],[b]

[a] *STEM, University of South Australia, Adelaide, Australia*
[b] *Systems Research Institute, Polish Academy of Sciences, Poland*

ARTICLE INFO

ABSTRACT

Most few-shot learning works rely on the same domain assumption between the base and the target tasks, hindering their practical applications. This paper proposes an adaptive transformer network (ADAPTER), a simple but effective solution for cross-domain few-shot learning where there exist large domain shifts between the base task and the target task. ADAPTER is built upon the idea of bidirectional cross-attention to learn transferable features between the two domains. The proposed architecture is trained with DINO to produce diverse, and less biased features to avoid the supervision collapse problem. Furthermore, the label smoothing approach is proposed to improve the consistency and reliability of the predictions by also considering the predicted labels of the close samples in the embedding space. The performance of ADAPTER is rigorously evaluated in the BSCD-FSL benchmarks in which it outperforms prior arts with significant margins.

1. Introduction

The advent of deep learning (DL) technologies has revolutionized the field of Artificial Intelligence (AI), making possible for complex data to be processed in an end-to-end manner without a manual feature extraction step. Although DL technologies have delivered state-of-the-art performances in various application domains, their successes are mainly attributed to large sample sizes hindering their applications in resource-constrained environments and being hardly applicable in the low sample setting.

The few-shot learning problem is motivated by the issue of data scarcity, where the main goal is to adapt quickly to new classes with extremely few samples. Despite its progress, most few-shot learning solutions assume that the base task and the target task are drawn from the same domain and often fail when there exist extreme discrepancies between the base task and the target task [1,2], i.e., they are inferior to naive fine-tuning strategies in the cross-domain cases. Extreme discrepancies here refer to different image types, such as ImageNet as the base task and satellite or X-ray images as the target task [2].

We follow the same setting as [1–3], where such a problem is overcome with moderate amounts of unlabelled samples from the target domain. The existence of unlabelled samples makes it possible to apply self-supervised learning techniques to avoid the supervision collapse

problem [4], where it loses necessary information for knowledge transfer. That is, the self-supervised learning step is capable of extracting meaningful features for generalizations beyond the base classes. The self-supervised learning strategy alone is not sufficient because it calls for a large number of unlabelled samples. Although unlabelled samples are given, the cross-domain few-shot learning problems distinguish themselves from the classical domain adaptation problem [5] because the base and the target task possess disjoint classes. The combination of labelled samples of the base task and unlabelled samples of the target task is a feasible solution because the labelled samples allow for the construction of common features while generalization beyond the base classes is availed by the unlabelled samples [3].

To the best of our knowledge, there exist several prior works in this domain where [2] adopts the concept of self-training for unlabelled samples and SimCLR [6] for representation learning while [3] adopts the concept of consistency regularization for weakly augmented unlabelled images and strongly augmented unlabelled images. The task adversarial augmentation method is proposed in [7], where the adversarial learning concept is implemented to construct the virtual tasks presenting the worst-case problem. [8] proposes the feature-wise transformation layers simulating different feature distributions across different domains. Knowledge Transduction (KT) is proposed in [9]

---

* Corresponding author.
E-mail addresses: naeem.paeedeh@mymail.unisa.edu.au (N. Paeedeh), dhika.pratama@unisa.edu.au (M. Pratama), muhammad_anwar.masum@mymail.unisa.edu.au (M.A. Ma'sum), Wolfgang.Mayer@unisa.edu.au (W. Mayer), Jimmy.Cao@unisa.edu.au (Z. Cao), Ryszard.Kowalczyk@unisa.edu.au (R. Kowlczyk).
[1] Authors share equal contributions.

where it relies on the feature adaptation module and the stochastic image transformation. [10] proposes the dual adaptive representation alignment approach and [11] addresses the cross-domain few-shot learning problem with the style adversarial training method. This problem remains an open problem because these works do not yet explore any domain alignment steps. The generalization power of the target task is upper bounded by the distance of the base task and the target task [12] in addition to the model's performances in recognizing the base task, which holds even in the case of mutually exclusive tasks [13]. A recent finding in [14] demonstrates significant performance gains with the pre-training step of the base task. Besides, all algorithms are crafted from the convolutional neural network's (CNN) solution, performing sub-optimally compared to that of the transformer approaches. In summary, two research gaps are to be attacked in this paper: (1) the issue of domain shifts remains an open problem in the literature; (2) existing approaches are still crafted under the convolutional neural network backbone.

This paper proposes an adaptive transformer network (ADAPTER) for cross-domain few-shot learning problems. It is built upon a compact transformer (CCT) backbone [15]. The CCT is lighter than the ViT [16] because of the convolutional tokenizer. Our model is structured under the bidirectional cross-attention transformer [17], making use of the cross-attention layer [18] to match the target queries and the base-task keys for smooth knowledge transfers. This strategy induces blurry task boundaries, facilitating seamless knowledge transfers. Our experiments follow the BSCD-FSL benchmark protocol [1] where the target task not only features different domains of the base task but also presents different image types, e.g., internet images vs X-ray images. To make this problem feasible, a model gains access to unlabelled samples of the target task.

ADAPTER is simple but highly effective, where the first phase is to perform the self-supervised training step of the base task via the bidirectional transformer structure using the DINO strategy [19]. The DINO strategy is selected because it is designed specifically for the transformer backbone. The self-supervised learning strategy extracts general features to prevent the supervision collapse problem [4,14], while bidirectional features function as implicit domain alignment steps to dampen the gaps of the base task and the target task [17]. This structure also enables both base samples and target samples to be utilized simultaneously under a single training stage, thus reducing the number of training stages. We do not apply any supervised training step because such an approach results in the overfitting problem of the base classes [20].

The self-supervised learning step using DINO under the bidirectional transformer architecture is followed by the supervised learning step of the support set comprising very few labelled samples to create a new classification head. Since a low sample quantity causes biased classifier responses, the label smoothing mechanism is undertaken to correct the classifier's logits, thus improving generalizations. The label smoothing mechanism is developed from the notion of label propagation [21] trading off spatial locations of an image in the embedding space and original predictions. It combines both distance measures and the classifier's logits in inferring the class labels. Unlike [2], no self-training mechanism is applied in our framework because the projection of the target task to the base label space leads to sub-optimal solutions.

This paper puts forward at least four major contributions:

1. ADAPTER is proposed in this paper as a simple but effective approach to address the cross-domain few-shot learning problems.
2. This paper proposes the self-supervised learning step via DINO [19] under the bidirectional transformer architecture, making it possible for base features and target features to be exploited simultaneously.
3. The label smoothing mechanism is developed to address biased responses of classifiers due to low sample sizes of the support set.

4. Source codes of ADAPTER are provided[2] to assure reproducibility and convenient further study.

Our novelty is present in two aspects: (1) the proposed approach applies the bidirectional transformer architecture built upon the quadruple transformer block and the compact transformer (CCT) backbone for the cross-domain few-shot learning problems; (2) the self-supervised learning strategy via DINO is implemented under the bidirectional transformer architecture allowing the base task and the target task to be exploited simultaneously. Such mechanism functions as a domain alignment step minimizing the distance of the base task and the target task.

The advantage of ADAPTER lies in its simplicity for cross-domain few-shot learning problems where such problem is handled with the self-supervised learning phase via DINO under the bidirectional transformer architecture coupled with the label smoothing strategy. Empirically, the advantage of ADAPTER is also evident where our approach outperforms prior arts in the BSCD-FSL benchmark with $1-9\%$ gaps in 14 out of 16 experiments.

2. Related works

Few Shot Learning is a learning problem to address the issue of data scarcity in dynamic environments, making it possible for a model to adapt quickly to new environments with extremely few samples. Existing works are categorized into two groups: metric-based and adaptation-based approaches. MatchingNet [22] is perceived as a pioneering metric-based approach where cosine similarity measures outputs classification outcomes. The relation network [23] is another early example of the metric-based approach, learning its own similarity metric. Prototypical Networks (ProtoNet) [24] is a popular few-shot learning approach making use of the prototype concept of the support set to produce the classification decision. On the other hand, model-agnostic meta-learning [25] is a prominent adaptation-based approach finding an initialization strategy through the meta-learning approach, enabling fast adaptations to new concepts with few gradient steps. [26] exemplifies the adaptation-based approach for few-shot learning benefiting from the meta-learning concept and the parameter transformation strategy. All these works have been successful when the base task and the target task are from the same domain but fail in the case of cross-domain few-shot learning [2].

BSCD-FSL benchmark [1] has been proposed as a challenging problem of few-shot learning where large domain gaps exist between the base and the target task. They also demonstrate the failure cases of popular few-shot learning approaches performing even worse than the naive fine-tuning strategy. [2] presents the STARTUP method derived from the combination of self-training and self-supervised learning techniques. [3] enhances this method with the weak and strong augmentation concept claimed as a self-supervised learning approach and removes the self-training step. [7] puts forward the task adversarial augmentation method to create the virtual tasks featuring the worst-case problem, and [8] proposes the feature-wise transformation layers simulating various feature distributions under different domains. In [9], the knowledge transduction approach is proposed while [10] presents the dual adaptive representation alignment approach. [11] puts forward the style adversarial training approach for cross-domain few-shot learning problem. These approaches still neglect the domain alignment step, which might boost the performance even further. Besides, these methods are still based on the CNN structure, which might be sub-optimal compared to the transformer solution. The transformer solutions for few-shot learning have appeared in [4,27]. We argue that these methods do not yet utilize the full potential of the ViT architecture [16]. In addition, they are tested with the meta-dataset [28]
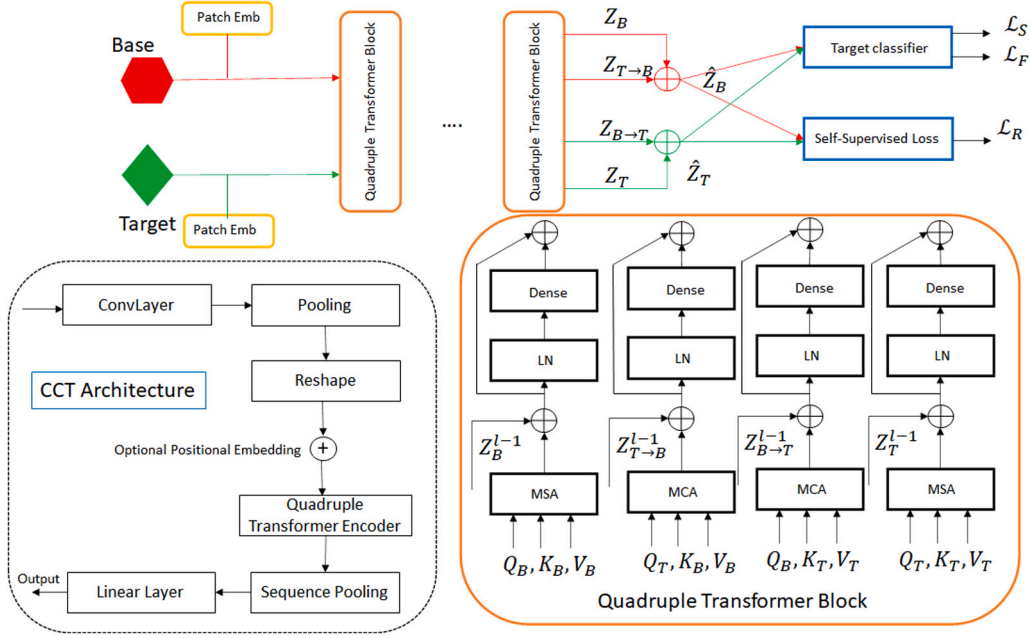
---

2 https://github.com/Naeem-Paeedeh/ADAPTER

Fig. 1. ADAPTER learns the base task B and the unlabelled samples of the target task U in a self-supervised manner using the DINO method and bidirectional features of the quadruple transformer block. A target classification head is created using a few labelled samples of the target task S. The label smoothing procedure via the label propagation method is carried out to refine the model's predictions.

rather than the BSCD-FSL dataset. This paper aims to cope with two research issues: (1) the domain alignment step remains an open research problem in the cross-domain few-shot learning problem; (2) existing approaches are still built upon the convolutional neural network structure.

Unsupervised Domain Adaptation is closely related to our work where the goal is to adapt a model to a new domain without any supervision of the target domain [5,12,13,29]. Recently, transformer-based approaches have been proposed for unsupervised domain adaptation [17,18,30]. Our work differs from them because the base task and the target task present distinct target classes. To the best of our knowledge, we are the first to explore the application of domain alignment for cross-domain few-shot learning under the transformer backbone, taking advantage of the presence of moderate unlabelled samples of the target task in addition to the self-supervised strategy.

3. Problem formulation

The cross-domain few-shot learning problem aims to classify unknown classes of the target task $\mathcal{D}_T$ over $\mathcal{X}_T \times \mathcal{Y}_T$ based on very few labelled samples $S = \{x_i, y_i\}_{i=1}^{N_S} \backsim \mathcal{D}_T$ known as the support set where $x_i \in \mathcal{X}_T$ and $y_i \in \mathcal{Y}_T$ respectively denote an ith input image and its corresponding label. An evaluation is performed in the unlabelled query set $Q = \{x_i\}_{i=1}^{N_Q} \backsim \mathcal{D}_T$ drawn from the same domain as the support set. Before the few-shot learning phase begins, one is given access to a base task $\mathcal{X}_B \times \mathcal{Y}_B \in \mathcal{D}_B$ where there exist plenty labelled samples $B = \{x_i, y_i\}_{i=1}^{N_B} \backsim \mathcal{D}_B$. The unique aspect of this problem lies in the existence of an extreme gap between the base task and the target task, i.e., completely distinct image space and label space. To make this problem feasible, unlabelled samples of the target task are supplied $U = \{x_i\}_{i=1}^{N_u} \backsim \mathcal{D}_T$ independent from the support set S and the query set Q. In a nutshell, a model is developed through two learning phases: representation learning via B, U and few-shot learning via S. The final evaluation is done with Q.

4. ADAPTER: Network structure

ADAPTER is developed with the bidirectional cross-attention transformer [17] modifying the CCT backbone [15] with bidirectional cross-attention layers [18]. The CCT backbone differs from the original ViT [16] with the presence of a convolutional tokenizer comprising convolutional and pooling layers downsizing the patch size thus improving scalability. The network structure of ADAPTER putting forward the quadruple transformer block is portrayed in Fig. 1. That is, it relies on the self-attention layer processing the single domain features, i.e., base and target and the cross-attention layer mining the cross-domain features, i.e., base→target and target→base. Both of them capture long-ranges dependencies of features of interests.

As with the conventional ViT, an input image $x_i \in \mathfrak{R}^{H \times W \times C}$ is converted into a sequence of image patches $\hat{x}_i \in \mathfrak{R}^{N \times (P^2 C)}$ where H, W, C are respectively the heights, weights and number of channels of the original images while (P, P) is the resolution of image patches and $N = \frac{HW}{P^2}$ is the number of image patches. $\hat{x}_i$ is transformed respectively into queries, keys and values $Q \in \mathfrak{R}^{N \times d_k}, K \in \mathfrak{R}^{N \times d_k}, V \in \mathfrak{R}^{N \times d_v}$ using linear projectors $\Psi_Q \in \mathfrak{R}^{d_k}, \Psi_K \in \mathfrak{R}^{d_k}, \Psi_V \in \mathfrak{R}^{d_v}$ where d_k, d_v stand for the dimensions of linear projectors. The self-attention layer operates in a single domain receiving x_i^s of the base task $\mathcal{D}_B$ or x_i^T of the target task $\mathcal{D}_T$.

$$Attn_B(x_i^B) = softmax(\frac{Q_B K_B'}{\sqrt{d_k}}) V_B \tag{1}$$

$$Attn_T(x_i^T) = softmax(\frac{Q_T K_T'}{\sqrt{d_k}}) V_T \tag{2}$$

where $'$ denotes the transpose operation. The self-attention layer performs a weighted sum of the values where its weights offer the compatibility degrees between the queries and the keys. In other words, relevant input patches can be favoured over others. The cross-attention layer distinguishes itself from the self-attention layer where it explores matching degrees of base task and target task rather than self similarities. That is, the queries $Q \in \mathfrak{R}^{M \times d_k}$ are obtained from different tasks of those the keys $K \in \mathfrak{R}^{N \times d_k}$ and the values $V \in \mathfrak{R}^{N \times d_v}$. This strategy

eases the alignment of the two tasks because similar patches across the two tasks should contribute more to the outputs than those of dissimilar input patches.

$$Attn_{T \to B}(x_i^T, x_i^B) = softmax(\frac{Q_T K_B'}{\sqrt{d_k}})V_B \tag{3}$$

$$Attn_{B \to T}(x_i^B, x_i^T) = softmax(\frac{Q_B K_T'}{\sqrt{d_k}})V_T \tag{4}$$

The weights assigned to the values are higher in the case of similar samples than those of dissimilar samples [18]. The cross-attention layer generates mixed features blurring the boundaries of the tasks, thus aligning their features [17].

The self-attention layer or the cross-attention layer is followed by a normalization layer, and a stack of dense layers parameterized with Ψ_W using the GeLU activation functions. In addition, two residual connections are integrated here to prevent the vanishing/exploding gradient problems where the first part combines the outputs of self-attention or cross-attention layers with the outputs of a previous quadruple transformer block while the second part adds the outputs of the Multi-Layer Perceptron (MLP) layer with the outputs of the self-attention or cross-attention layer as per the original ViT. We follow the original CCT backbone [15] with a stack of quadruple transformer blocks forming a transformer encoder whose output is aggregated to the sequence pooling operator before going to a linear layer for predictions.

5. ADAPTER: Learning policy

ADAPTER consists of two learning phases: the representation learning phase and the few-shot learning phase where the representation learning phase performs the domain alignment step via the self-supervised training strategy of bidirectional features to generalize beyond known base classes using B and U while the few-shot learning phase adapts the label space of a model using extremely few labelled samples of S. Our model is structured as the bidirectional cross-attention transformer where $\zeta_\Psi(.)$ stands for the feature extractor consisting of a stack of the quadruple transformer blocks and $\xi_\Phi(.)$ denotes a classifier. $\Psi = \{\Psi_Q, \Psi_K, \Psi_V, \Psi_W\}$ stand for the parameters of the feature extractor and Φ is the parameters of the classifier. The feature extractor generates the base features $Z_B = \zeta_\Psi(x_i^B)$, the target-base features $Z_{T \to B} = \zeta_\Psi(x_i^T, x_i^B)$, the base-target features $Z_{B \to T} = \zeta_\Psi(x_i^B, x_i^T)$ and the target features $Z_T = \zeta_\Psi(x_i^T)$ while the classifier converts the augmented feature vector to the label space of the target task $\xi_\Phi(\hat{Z}) \in \Re^D \to \Delta_{M_T}$. M_T, D denote the dimension of target classes and embedding space.

5.1. Representation learning phase

The representation learning phase starts with the self-supervised learning step of the base and target tasks simultaneously $B \backsim D_B, U \backsim D_T$ using the quadruple transformer block as an implicit domain adaptation step. This strategy follows the finding of [14] where the pre-training phase with an external dataset significantly boosts the performances of few-shot learners and differs from existing cross-domain few-shot learning methods [2,3] where such approaches are applied to the target domain. This is applied under the bidirectional architecture utilizing both base and target features, thus functioning as an implicit domain adaptation phase. In addition, no supervised step is applied on the base task because it leads to the overfitting problem of base classes [20]. The self-supervised learning step produces general features which can be easily adapted to another domain and prevents the supervision collapse problem. We leverage the DINO method here [19] because it is designed specifically for the transformer backbone.

The DINO method utilizes the knowledge distillation approach [31] between the teacher network $\xi_{\Phi_t}(.)$ and the student network $\xi_{\Phi_s}(.)$ using the multi-crop strategy resulting in different distorted views of an image V. That is, two global views of an image, x_1^g and x_2^g, are fed to the teacher only while all crops are passed to the student network. This is done with augmented feature space via the quadruple transformer block $\hat{Z} = [Z_B, Z_{T \to B}, Z_{B \to T}, Z_T]$ where $Z_{T \to B}$ and $Z_{B \to T}$ stand for base-dominant features and target-dominant features respectively. The target features are aligned to the base where the target query Q_T is matched with the base keys K_B and values V_B for base-dominant features and vice versa for target-dominant features. The loss function is expressed as:

$$\mathcal{L}_R = \mathbb{E}_{(x_i^B) \backsim D_B \cup (x_i^T) \backsim D_T} [\sum_{x \in \{x_1^g, x_2^g\}} \sum_{\tilde{x} \in V} l(\xi_{\Phi_t}(\zeta_{\Psi_t}(x_i^T, x_i^B))), \xi_{\Phi_s}(\zeta_{\Psi_s}(\tilde{x}_i^T, \tilde{x}_i^B))] \tag{5}$$

where $l(.)$ denotes the cross-entropy loss function and $x \neq \tilde{x}$. Note that different temperatures, τ_s, τ_t, are used in the cross entropy loss function. The same architecture is implemented for both students and teachers. (5) is minimized to adjust the student network, i.e., the stop gradient technique is applied to the teacher. The momentum encoder technique following the exponential moving average update rule is adopted to adjust the teacher network. Let $\theta = [\Phi, \Psi]$, $\theta_t = \lambda \theta_t + (1 - \lambda)\theta_s$ where λ is dynamically adjusted with the cosine scheduler. As with DINO [19], the teacher outputs are centred and sharpened to avoid model collapse. The teacher parameters are copied and passed to the few-shot learning phase.

5.2. Few-shot learning phase

The few-shot learning phase is carried out by mining few labelled samples of the target domain S to construct a new classification head. First, the new classification head is adopted for the classifier $\xi_\Phi(.)$: $\mathcal{X}_T \to \Re^{M_T}$ to adapt to the label space $\mathcal{Y}_T$. Scarcely labelled samples S are learned in the supervised manner via the cross-entropy loss function.

$$\mathcal{L}_S = \mathbb{E}_{(x_i^T, y_i^T) \backsim D_T}[l(\xi_\Phi(\zeta_\Psi(x_i^T), y_i^T))] \tag{6}$$

where only self-attention features are used due to the sample size constraint. Since the base task and the target task share no overlapping label spaces $\mathcal{Y}_B \cap \mathcal{Y}_T = \emptyset$, the pseudo-labelling strategy w.r.t the projection of the base task as done in [2] might be sub-optimal. Because of a tiny sample size in S, this step only does not suffice and often suffers from biased classifier responses.

The label smoothing step is implemented here by applying the label propagation strategy [21] enforcing the consistency of the predictions. That is, neighbouring samples must be labelled similarly or be smooth. To this end, a graph $G = (V, E)$ is created where the nodes V are defined by all images in Q and E stands for the edges weighted by an adjacency matrix $A \in \Re^{N_Q \times N_Q}$ whose elements are calculated as a proximity of two images in the embedding space $a_{i,j} = \exp\frac{(-\|\zeta_\Psi(x_i^U, x_i^U) - \zeta_\Psi(x_j^U, x_j^U)\|^2)}{2\sigma^2}, j \neq i$ and $a_{i,i} = 0$. σ denotes the standard deviation of the embedded images. The adjacency matrix is normalized as per $\hat{A} = D^{-1/2}AD^{-1/2}$ in which D is a diagonal matrix whose the $(i, i) - th$ element denotes the sum of the ith row of A. The label propagation is iterated until convergence $\mathcal{F}^*$.

$$\mathcal{F}_{t+1} = \alpha \hat{A} \mathcal{F}_t + (1 - \alpha)\overline{Y} \tag{7}$$

where $\mathcal{F} \in \Re^{N_Q \times M_T}$ denotes the label assignment matrix mapping the input space to the label space $\mathcal{F} : \mathcal{X}_T \to \Re^{M_T}$ and $\overline{Y} \in \Re^{N_Q \times M_T}$ stands for the initial label assignment matrix set as those of pseudo labels. The elements of $\overline{Y}$ is set as a one-hot encoded form of a pseudo label $\bar{y}_i$. $\alpha \in [0, 1]$ is a hyper-parameter controlling a trade-off between the smoothness constraint, i.e., a desirable classification decision should not deviate too much across neighbouring points, and the fitting constraint, i.e., a desirable classification decision should not

Algorithm 1: PyTorch style pseudocode for ADAPTER.

```
def representation_learning(student):
    teacher = deepcopy(student)
    # iter_target: An iterator that loads the next mini-batch from the target dataset
    optimizer = torch.optim.AdamW(studend.parameters())

    for i in range(epochs):
        for xb in data_loader_base:
            xt = next(iter_target)
            # yt and ys prefixes are the outputs of the teacher and students.
            # b, tb, t, and bt suffixes denote the base, target-base, target, and base-target features
            # Inputs are the 2 global views
            yt_b, yt_tb, yt_t, yt_bt = teacher(xb[:2], xt[:2])
            ys_b, ys_tb, ys_t, ys_bt = student(xb, xt)
            loss_base = dino_loss_base(cat(ys_b, ys_tb), cat(yt_b, yt_tb))
            loss_target = dino_loss_target(cat(ys_t, ys_bt), cat(yt_t, yt_bt))
            loss = loss_base + loss_target
            loss.backward()
            optimizer.step()
            # Updates the teacher's parameters with an exponential moving average of momentum, centering,
            # and sharpening
            update_teacher(teacher)
    teacher_cct = unwrap(teacher)    # Extracts only the CCT backbone
    return teacher_cct

def evaluation(cct):
    for support_set, query_set in episodic_data_loader:
        cct.reset()
        classifier = MLP(4*embed_dim, n_hidden, n_ways)
        optimizer = SGD(classifier.params())
        for i in range(n_epochs):
            for xs, ls in support_set:  # for each sample and label mini-batch
                features_s = cct(xs)
                ys = classifier(cat(features_s, features_s, features_s, features_s))
                loss = cross_entropy_loss(ys, ls)
                loss.backward()
                optimizer.step()
        xq, lq = query_set  # Images and labels of the query set
        features_q = cct(xq)
        logits_q = classifier(features_q)
        smoothed_logits = LabelPropagation(features_q, logits_q)
        list_correct.append(correct_predictions(smoothed_logits, lq))
    calculate_overall_accuracy(list_correct)

def ADAPTER():
    cct = CCT_14_7x2_with_quadruple_blocks() # Initializing a CCT-14/7x2 with quadruple blocks
    # head_base: MLP head for the base domain
    # head_target: MLP head for the target domain
    # Wrapping the cct, head_base, and head_target wrapped in one network object
    student = wrap(cct, head_base, head target)
    teacher = representation_learning(student)
    evaluation(teacher)
```

differ from its initial label assignment [21]. A closed-loop solution can be derived for (7):

$$\mathcal{F}^* = (I - \alpha\hat{A})^\dagger \overline{Y} \tag{8}$$

where † is the pseudo-inverse symbol. A final prediction is taken from the label assignment matrix $\hat{y}_i = \arg\max_{m \in M_B} \mathcal{F}_i^*, \{x_i, \overline{y}_i\} \in P$. Note that no label of evaluation samples in Q is leaked for predictions. For the testing phase on Q, ADAPTER operates in the target task without any base samples. The double target strategy [17] is implemented where the cross-attention layer executes target features rather than cross features, thus leading to $\hat{Z} = [Z_T, Z_T, Z_T, Z_T]$. Pseudo-code of ADAPTER is presented in Algorithm 1.

6. Time complexity analysis

We utilize a CCT with quadruple blocks in our experiments. The most time-consuming parts of our model are self-attentions, cross-attentions, MLP projections in the blocks, sequence pooling, and MLP head. In the following, we calculate the time complexity for each component. At last, we can find the total time complexity for the CCT.

First, we have four combinations for self-attention and cross-attention. Each self-attention or cross-attention requires a linear projection to calculate query, key, and value. The final attention calculations need three consecutive matrix multiplications. Therefore, the time complexity for attentions is $O(nd^2)$, where n is the number tokens or sequence length and d is the dimension of each token or embedding. Second, for each MLP that has L_{MLP} layers and d_{MLP} neurons in hidden layers, we approximately have L_{MLP} matrix multiplications. The time complexity for MLP is $O(L_{\text{MLP}}d_{\text{MLP}}^2)$. For L_{CCT} layers or blocks, the time complexity of MLP is $O(L_{\text{CCT}}L_{\text{MLP}}d_{\text{MLP}}^2)$. We have a few convolution layers for patch embedding. According to [32], convolutions are also matrix multiplication operations. An L_{CNN} layers convolutional neural network (CNN) requires $O(L_{\text{MLP}}d_{\text{CNN}}^2)$ operations, where d_{CNN} is the dimension for the resultant matrix multiplication. Finally, the sequence pooling has two matrix multiplications. Each operation requires $O(nd)$ operations, hence it requires $O(L_{\text{CCT}}nd)$ operations for L_{CCT} layers.

Overall, the time complexity of a CCT model with L_{CCT} layers is $O(L_{\text{CCT}}nd^2) + O(L_{\text{CCT}}L_{\text{MLP}}d_{\text{MLP}}^2)$.

7. Experiments

This section details our numerical study in the BSCD-FSL benchmark [1] as well as our ablation study to analyse the effect of each learning module.

Table 1

5-way k-shot classification accuracy on Mini-ImageNet → BSCD-FSL. The * indicates the experiments that we retested on our machine.

Dataset	ChestX		ISIC		EuroSAT		CropDisease	
Methods	1-shot	5-shot	1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
ProtoNet	$21.32 \pm .37$	$24.72 \pm .43$	$29.58 \pm .57$	$42.49 \pm .58$	$55.32 \pm .88$	$76.92 \pm .67$	$52.94 \pm .81$	$81.84 \pm .68$
MatchingNet	$20.65 \pm .29$	$22.62 \pm .36$	$27.37 \pm .51$	$33.96 \pm .54$	$54.88 \pm .90$	$68.00 \pm .68$	$46.86 \pm .88$	$63.94 \pm .84$
Transfer	$22.60 \pm .39$	$26.51 \pm .43$	$32.12 \pm .59$	$43.88 \pm .57$	$58.14 \pm .83$	$80.09 \pm .61$	$68.78 \pm .84$	$89.79 \pm .52$
SimCLR(Base)	$22.37 \pm .42$	$26.63 \pm .46$	$32.43 \pm .56$	$44.04 \pm .55$	$58.28 \pm .90$	$80.83 \pm .64$	$61.58 \pm .88$	$83.44 \pm .61$
SimCLR	$23.59 \pm .44$	$29.56 \pm .49$	$31.45 \pm .59$	$42.18 \pm .54$	$62.63 \pm .87$	$82.76 \pm .59$	$69.22 \pm .93$	$89.31 \pm .53$
Transfer+SimCLR	$23.72 \pm .44$	$29.45 \pm .10$	$31.67 \pm .55$	$45.97 \pm .54$	$63.91 \pm .83$	$85.78 \pm .51$	$70.35 \pm .85$	$91.10 \pm .49$
STARTUP	$23.09 \pm .43$	$26.94 \pm .45$	$32.66 \pm .60$	$47.22 \pm .61$	$63.88 \pm .84$	$82.29 \pm .60$	$75.93 \pm .80$	$93.02 \pm .45$
STARTUP*	$23.29 \pm .42$	$27.11 \pm .47$	$33.96 \pm .63$	$47.89 \pm .61$	$63.56 \pm .85$	$82.85 \pm .58$	$75.53 \pm .81$	$92.90 \pm .44$
Dynamic Distillation Net.	$23.38 \pm .43$	$28.31 \pm .43$	$34.66 \pm .58$	$49.36 \pm .59$	$73.14 \pm .84$	$89.07 \pm .47$	$82.14 \pm .78$	$95.54 \pm .38$
StyleAdv-FT	22.64 ± 0.35	26.24 ± 0.35	35.76 ± 0.52	53.05 ± 0.54	72.92 ± 0.75	91.64 ± 0.43	80.69 ± 0.28	96.51 ± 0.28
KT	22.68 ± 0.60	26.79 ± 0.61	34.06 ± 0.77	46.37 ± 0.77	66.43 ± 0.93	82.53 ± 0.66	73.10 ± 0.87	89.53 ± 0.58
Dara	22.92 ± 0.40	27.54 ± 0.42	36.42 ± 0.64	56.28 ± 0.66	67.42 ± 0.80	85.84 ± 0.54	80.74 ± 0.76	95.32 ± 0.34
ADAPTER	$\mathbf{25.28 \pm 0.45}$	$\mathbf{31.00 \pm 0.48}$	$\mathbf{40.66 \pm 0.70}$	55.67 ± 0.62	$\mathbf{73.20 \pm 0.67}$	$\mathbf{92.04 \pm 0.34}$	$\mathbf{85.23 \pm 0.70}$	$\mathbf{96.61 \pm 0.32}$

Table 2

5-way k-shot classification accuracy on Tiered-ImageNet → BSCD-FSL. The * indicates the experiments that we retested on our machine.

Dataset	ChestX		ISIC		EuroSAT		CropDisease	
Methods	1-shot	5-shot	1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
Transfer	$22.46 \pm .41$	$26.33 \pm .45$	$29.76 \pm .55$	$41.27 \pm .58$	$58.07 \pm .86$	$81.34 \pm .53$	$69.94 \pm .87$	$90.12 \pm .49$
SimCLR(Base)	$22.28 \pm .40$	$25.96 \pm .44$	$31.03 \pm .55$	$42.91 \pm .55$	$62.14 \pm .89$	$81.85 \pm .59$	$62.45 \pm .90$	$84.11 \pm .60$
Transfer+SimCLR	$23.81 \pm .46$	$30.26 \pm .50$	$31.71 \pm .55$	$45.08 \pm .56$	$58.08 \pm .83$	$86.08 \pm .47$	$71.25 \pm .89$	$91.31 \pm .49$
STARTUP	$22.10 \pm .40$	$26.03 \pm .44$	$29.73 \pm .51$	$43.55 \pm .56$	$64.32 \pm .87$	$85.19 \pm .50$	$70.09 \pm .86$	$90.81 \pm .49$
Dynamic Distillation Net.	$22.70 \pm .42$	$27.67 \pm .46$	$33.87 \pm .56$	$47.21 \pm .56$	$72.15 \pm .75$	$89.44 \pm .42$	$84.41 \pm .75$	$95.90 \pm .34$
ADAPTER	$\mathbf{25.88 \pm .49}$	$\mathbf{31.85 \pm .49}$	$\mathbf{40.57 \pm .69}$	$55.65 \pm .61$	$70.63 \pm .66$	$\mathbf{91.43 \pm .35}$	$83.80 \pm .72$	$\mathbf{96.55 \pm .34}$

7.1. Datasets

The BSCD-FSL benchmark [1] comprises four target datasets: CropDisease [33], EuroSAT [34], ISIC [35] and ChestX [36] while the base datasets make use of the Mini-ImageNet dataset [22] and the Tiered-ImageNet dataset [37]. We follow the same setting as [2,3] where U is constructed from 20% samples of the new domain datasets while the rest are reserved for evaluation. We evaluate both the case of 5-way 1-shot and 5-way 5-shot configurations where 600 runs are performed. Top-1 accuracy and 95% confidence interval across 600 runs are reported.

7.2. Baseline algorithms

ADAPTER is compared with nine prior arts: ProtoNet [24], MatchingNet [22], SimCLR [6], STARTUP [2], Dynamic Distillation Networks [3], KT [9], Dara [10], StyleAdv-FT [11]. In addition, the numerical results of the transfer learning method as well as the transfer learning method + SimCLR are reported. Numerical results of other methods are obtained from [3] while we also offer the STARTUP numerical results executed under our computer. ADAPTER's code is developed from the STARTUP environments to ensure fair comparisons and made publicly available for convenient further studies.

7.3. Implementation details

We utilize a CCT-7/7 × 2 backbone with quadruple blocks for all experiments. We add two MLPs with three hidden layers as projection heads for DINO that receive the base-dominant features for the base domain and target-dominant features for the target domain. The Mini-ImageNet dataset is deployed for the base domain, and each dataset in the BSCD-FSL benchmark is deployed for the target domain. The batch size is set to 16. We use the AdamW optimizer to conform to the original implementation of DINO. The weight decay is set to 1e-5, and the weight decay end is set to 0.4. We set the number of epochs to 500 but stopped it at 200 epochs; hence, we could continue an experiment with the same setting for the scheduler without a jump in the learning rate value. For the tiered-ImageNet, we stopped the experiments at 80 epochs.

In the final phase, we throw away the projection heads of the DINO. Next, we freeze the backbone network and use a four-layer MLP classifier with ReLU activation functions for the hidden layer on top of the backbone. Since both inputs of the cross-attentions in the testing phase are from the target domain, the cross-attentions will behave like self-attentions. Therefore, we concatenate the same embeddings for the MLP classifier head. We generalize this idea by replicating the self-attention outputs a few times (four times in our experiments) as the final input of the MLP. The hidden layer size is four times the number of the input features of the MLP classifier. We utilized the SGD optimizer to train the classifier. The learning rate for fine-tuning is 0.01. Weight decay is 0.001. Momentum and dampening are 0.9 for the SGD. The batch size is 4. The number of episodes is 600. We set the number of epochs to 500 for the 1-shot tests and 100 for the 5-shot test. The self-supervised training phase experiments of Mini-ImageNet are conducted on a GeForce RTX 3060 graphic card with batch size 16, and the Tiered-ImageNet are performed on an RTX A5000 graphic card with batch size 32. All experiments of the few-shot learning phase, including the training of the classifier head, label smoothing, and evaluation, are performed on the RTX A5000 graphic card. Hyper-parameters of ADAPTER are listed in Table 5.

7.4. Numerical results

Table 1 reports our numerical results using the Mini-ImageNet as the base task. It is perceived that ADAPTER produces the most improved results with $1 - 9\%$ gaps compared to other algorithms across almost all cases. ADAPTER performs equally well for both 1-shot and 5-shot configurations, where the highest margin to its counterparts is observed in the ISIC problem. Note that our experiments are executed under the STARTUP environments to ensure fair comparisons. Table 2 reports our numerical results using the tiered-ImageNet as the base task. The same finding is drawn where ADAPTER outperforms its counterparts with noticeable margins, i.e., 3%–4% for the ChextX problem and 6%–9% for the ISIC problem. This pattern applies to both the one-shot and five-shot configurations. On the other hand, ADAPTER beats other algorithms in the EuroSAT problem and in the CropDisease problem under the 5-shot setting. ADAPTER is slightly behind the dynamic distillation in the EuroSAT and CropDisease problem under the 1-shot scenario.

Table 3
Settings in ablation studies. QB, SSRL, and FSL stand for quadruple block, self-supervised representation learning, and few-shot learning, respectively.

Component	Normal	w/o label prop.	Label prop w. sup. set	Supervised	Supervised, fine-tuned
QB for SSRL	✓	✓	✓	–	✓
QB for FSL	–	–	–	–	✓
Label smoothing (in FSL)	✓	–	✓	✓	✓
S in label smoothing	–	–	✓	–	–
Frozen backbone in FSL	✓	✓	✓	✓	–

Table 4
Ablation studies on different configurations for 5-way k-shot classification accuracy on Mini-ImageNet → BSCD-FSL.

Dataset	ChestX		ISIC		EuroSAT		CropDisease	
Methods	1-shot	5-shot	1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
ADAPTER	$25.28 \pm .45$	$31.00 \pm .48$	$40.66 \pm .70$	$55.67 \pm .62$	$73.20 \pm .67$	$92.04 \pm .34$	$85.23 \pm .70$	$96.61 \pm .32$
ADAPTER (w/o label prop.)	$25.17 \pm .47$	$30.53 \pm .50$	$39.67 \pm .67$	$55.14 \pm .64$	$71.34 \pm .69$	$91.10 \pm .35$	$83.00 \pm .78$	$95.50 \pm .38$
ADAPTER (label prop w. sup. set)	$25.10 \pm .46$	$30.65 \pm .49$	$41.06 \pm .68$	$55.73 \pm .62$	$72.86 \pm .74$	$91.56 \pm .36$	$84.01 \pm .79$	$96.41 \pm .32$
ADAPTER (supervised)	$21.33 \pm .39$	$22.78 \pm .42$	$35.23 \pm .68$	$47.47 \pm .67$	$57.86 \pm .85$	$73.37 \pm .65$	$66.45 \pm .93$	$83.34 \pm .60$
ADAPTER (supervised, fine-tuned)	$21.73 \pm .39$	$22.99 \pm .39$	$34.82 \pm .68$	$47.81 \pm .66$	$55.24 \pm .81$	$70.49 \pm .71$	$65.04 \pm .88$	$84.00 \pm .61$

7.5. Ablation studies

An ablation study is carried out to analyse the contribution of each learning module to the overall performance of ADAPTER. It is performed across all datasets, making use of the Mini-ImageNet as the base dataset.

Table 3 shows which components are used for a specific variation. The accuracies for each target dataset in 1-shot and 5-shot settings are displayed for each variation in Table 4. The first variation is to remove the label smoothing mechanism via the label propagation technique and indicated in the second row of Table 4. The absence of label smoothing mechanism decreases the performance of ADAPTER significantly by around $0.5\% - 1.2\%$. This finding confirms the label smoothing mechanism to overcome the biased responses of the classifiers due to very few samples in S.

The second configuration is to include S in the label smoothing mechanism in addition to Q. It is shown in the third row of Table 4. Note that S is removed afterwards and excluded from any accuracy calculations. This modification does not improve the results significantly and often worsens the results. Besides, this configuration also imposes extra computations to induce the adjacency matrix A.

Another variation of ADAPTER is attempted where it is trained on the base domain in a supervised manner rather than the self-supervised manner without any cross-attention mechanisms. An MLP classifier, after that, is trained on the support set. This configuration significantly compromises the generalization of ADAPTER and aligns with the finding of [20] where the supervised learning step results in the overfitting problem of the target classes. It hinders ADAPTER from generalizing beyond its base classes. This finding also confirms the advantage of bidirectional transformer architecture to dampen the gap between the base task and the target task.

In the last experiment, after training the backbone on Mini-ImageNet, the CCT backbone is fine-tuned simultaneously on Mini-ImageNet, and each target dataset with two distinct linear classifiers to exploit the bidirectional cross-attention mechanism. For evaluation, we feed the target features from self-attention to the classifier as both base and target features. Although the cross-attention improves the predictions in most cases, the few samples in S are not enough to train many parameters of the whole model in a few cases.

7.6. t-SNE analysis

The t-SNE plots of the features of the CCT backbone for the EuroSAT and CropDisease are shown in Fig. 2. The graphs show that the CCT backbone, simultaneously trained on both domains with bidirectional cross-attention, can successfully learn transferable and discriminative features of both domains. While the vanilla transformer is weak in dealing with low-resolution input images, the combination of convolutions
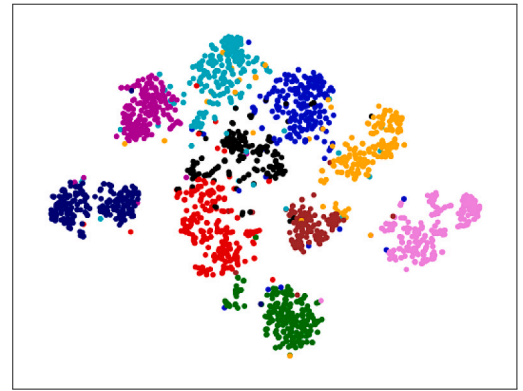
(a) EuroSAT
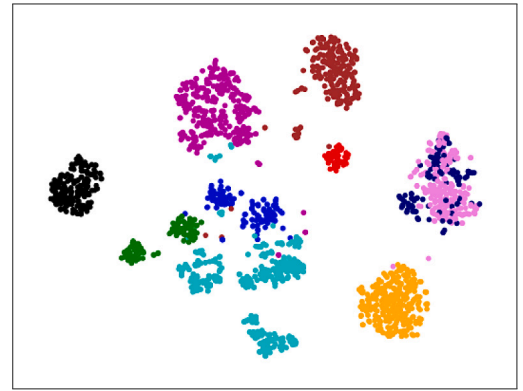
(b) CropDisease

Fig. 2. t-SNE plots of 10 classes from the test sets of EuroSAT and CropDisease datasets for 2000 samples.

and attention mechanisms does not exhibit such a disadvantage. In the second graph, for CropDisease, two classes have significant overlap. Moreover, the model sometimes maps a few samples from one cluster to another cluster. It explains another reason for the effectiveness of label smoothing that assigns the same label to a few labels in the vicinity of each other.

8. Conclusion

This paper proposes a simple but effective solution for cross-domain few-shot learning problems, adaptive transformer network (ADAPTER)

Table 5

The list of hyperparameters that are used in different phases of the experiments. SSRL, MI, TI, FSL, FT, and s+c stand for self-supervised representation learning, Mini-ImageNet, Tiered-ImageNet, few-shot learning, fine-tuning, and self-attention + cross-attention, respectively.

Phase	SSRL MI+BSCD-FSL	SSRL TI+BSCD-FSL	Supervised-MI	FSL 1-shot	FSL-5-shot	FT 1-shot	FT 5-shot
Batch size (MI+BSCD-FSL)	16	32	200	4	4	5	5
N. episodes	–	–	–	600	600	600	600
N. epochs	200	50	400	500	100	–	–
N. iterations	–	–	–	–	–	500	250
N. epochs in Cosine scheduler	500	500	–	–	–	–	–
Optimizer	AdamW	AdamW	SGD	SGD	SGD	SGD	SGD
Learning rate	2.5×10^{-4}	2.5×10^{-4}	0.025	0.01	0.01	0.01	0.01
Min LR	5×10^{-7}	5×10^{-7}	–	–	–	–	–
Momentum	0.9	0.9	0.9	0.9	0.9	0.9	0.9
Momentum for teacher	0.996	0.996	–	–	–	–	–
Dampening	–	–	0	0.9	0.9	0.9	0.9
Weight decay	1×10^{-5}	1×10^{-5}	1×10^{-5}	0.001	0.001	0.001	0.001
Weight decay (cosine scheduler)	0.4	0.4	–	–	–	–	–
Warmup epochs	10	10	–	–	–	–	–
N. hidden layers (MLP)	3	3	0 (linear)	3	3	0	0
Input dim. of the head	2×384 (s+c)	2×384 (s+c)	384 (self)	4×384 (self)	4×384 (self)	2×384 (s+c)	2×384 (s+c)
N. hidden neurons	2048	2048	–	6144	6144	–	–
N. local crops	8	8	–	–	–	–	–
Output dim. head	65 536	65 536	64	5	5	5	5
Frozen backbone in FSL	–	–	–	✓	✓	–	–
α (LP)	–	–	–	0.99	0.99	0.99	0.99
σ (LP)	–	–	–	50	50	50	50
rcond for pinv.				0.1	0.1	0.1	0.1
N. episodes (FSL)	–	–	–	600	600	600	600

constructed under the bidirectional cross-attention transformer architecture. ADAPTER is trained in a self-supervised manner utilizing the bidirectional cross-attention block aligning the labelled base task and the unlabelled target task. ADAPTER applies the label smoothing phase via the label propagation concept to further refine the model's predictions. Our experiments in the BSCD-FSL benchmark have demonstrated the advantage of ADAPTER, where it outperforms prior arts with $1 - 9\%$ margins in 14 out of 16 experiments across all target datasets in both 1-shot and 5-shot configurations. This finding is consistent regardless of the base dataset, Mini-ImageNet and Tiered-ImageNet.

Our future study is devoted to explore continual cross-domain few-shot learning where a model should be robust against the catastrophic forgetting problem while performing seamless knowledge transfer across the base and the target dataset.

CRediT authorship contribution statement

Naeem Paeedeh: Software, Methodology, Data curation, Conceptualization. **Mahardhika Pratama:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Investigation, Conceptualization. **Muhammad Anwar Ma'sum:** Software, Investigation, Data curation. **Wolfgang Mayer:** Writing – review & editing, Supervision. **Zehong Cao:** Writing – review & editing, Supervision. **Ryszard Kowlczyk:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link containing the link of my codes.

References

[1] Y. Guo, N.C.F. Codella, L. Karlinsky, J.R. Smith, T. Simunic, R.S. Feris, A new benchmark for evaluation of cross-domain few-shot learning, 2019, ArXiv, arXiv:1912.07200.

[2] C.P. Phoo, B. Hariharan, Self-training for few-shot transfer across extreme task differences, 2021, ArXiv, arXiv:2010.07734.

[3] A. Islam, C.-F. Chen, R. Panda, L. Karlinsky, R.S. Feris, R.J. Radke, Dynamic distillation network for cross-domain few-shot recognition with unlabeled data, in: NeurIPS, 2021.

[4] C. Doersch, A. Gupta, A. Zisserman, CrossTransformers: spatially-aware few-shot transfer, 2020, ArXiv, arXiv:2007.11498.

[5] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (2010) 1345–1359.

[6] T. Chen, S. Kornblith, M. Norouzi, G.E. Hinton, A simple framework for contrastive learning of visual representations, 2020, ArXiv, arXiv:2002.05709.

[7] H. Wang, Z. Deng, Cross-domain few-shot classification via adversarial task augmentation, in: International Joint Conference on Artificial Intelligence, 2021.

[8] H.-Y. Tseng, H.-Y. Lee, J.-B. Huang, M.-H. Yang, Cross-domain few-shot classification via learned feature-wise transformation, 2020, ArXiv, arXiv:2001.08735.

[9] P. Li, F. Liu, L. Jiao, S. Li, L. Li, X. Liu, X. Huang, Knowledge transduction for cross-domain few-shot learning, Pattern Recognit. 141 (2023) 109652, URL https://api.semanticscholar.org/CorpusID:258423748.

[10] Y. Zhao, T. Zhang, J. Li, Y. Tian, Dual adaptive representation alignment for cross-domain few-shot learning, IEEE Trans. Pattern Anal. Mach. Intell. 45 (2023) 11720–11732, URL https://api.semanticscholar.org/CorpusID:258462721.

[11] Y. Fu, Y. Xie, Y. Fu, Y. Jiang, StyleAdv: Meta style adversarial training for cross-domain few-shot learning, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), 2023, pp. 24575–24584, URL https://api.semanticscholar.org/CorpusID:257038175.

[12] S. Ben-David, J. Blitzer, K. Crammer, F.C. Pereira, Analysis of representations for domain adaptation, in: NIPS, 2006.

[13] Z. Fang, J. Lu, F. Liu, J. Xuan, G. Zhang, Open set domain adaptation: Theoretical bound and algorithm, IEEE Trans. Neural Netw. Learn. Syst. 32 (2021) 4309–4322.

[14] S.X. Hu, D. Li, J. Stuhmer, M. Kim, T.M. Hospedales, Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), 2022, pp. 9058–9067.

[15] A. Hassani, S. Walton, N. Shah, A. Abuduweili, J. Li, H. Shi, Escaping the big data paradigm with compact transformers, 2021, ArXiv, arXiv:2104.05704.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, 2021, ArXiv, arXiv:2010.11929.

[17] X. Wang, P. Guo, Y. Zhang, Domain adaptation via bidirectional cross-attention transformer, 2022, ArXiv, arXiv:2201.05887.

[18] T. Xu, W. Chen, P. Wang, F. Wang, H. Li, R. Jin, CDTrans: Cross-domain transformer for unsupervised domain adaptation, 2022, ArXiv, arXiv:2109.06165.

[19] M. Caron, H. Touvron, I. Misra, H. J'egou, J. Mairal, P. Bojanowski, A. Joulin, Emerging properties in self-supervised vision transformers, in: 2021 IEEE/CVF International Conference on Computer Vision, (ICCV), 2021, pp. 9630–9640.

[20] Y. Lu, L. Wen, J. Liu, Y. Liu, X. Tian, Self-supervision can be a good few-shot learner, in: European Conference on Computer Vision, 2022.

[21] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, in: NIPS, 2003.

[22] O. Vinyals, C. Blundell, T.P. Lillicrap, K. Kavukcuoglu, D. Wierstra, Matching networks for one shot learning, in: NIPS, 2016.

[23] F. Sung, Y. Yang, L. Zhang, T. Xiang, P.H.S. Torr, T.M. Hospedales, Learning to compare: Relation network for few-shot learning, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 1199–1208.

[24] J. Snell, K. Swersky, R.S. Zemel, Prototypical networks for few-shot learning, 2017, ArXiv, arXiv:1703.05175.

[25] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, 2017, ArXiv, arXiv:1703.03400.

[26] Q. Sun, Y. Liu, T.-S. Chua, B. Schiele, Meta-transfer learning for few-shot learning, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), 2019, pp. 403–412.

[27] L. Liu, W. Hamilton, G. Long, J. Jiang, H. Larochelle, A universal representation transformer layer for few-shot image classification, 2021, ArXiv, arXiv:2006.11702.

[28] E. Triantafillou, T.L. Zhu, V. Dumoulin, P. Lamblin, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol, H. Larochelle, Meta-dataset: A dataset of datasets for learning to learn from few examples, 2020, ArXiv, arXiv:1903.03096.

[29] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V.S. Lempitsky, Domain-adversarial training of neural networks, in: J. Mach. Learn. Res., 2016.

[30] J. Yang, J. Liu, N. Xu, J. Huang, TVT: Transferable vision transformer for unsupervised domain adaptation, 2021, ArXiv, arXiv:2108.05988.

[31] G.E. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, 2015, ArXiv, arXiv:1503.02531.

[32] N. Paeedeh, K. Ghiasi-Shirazi, Improving the backpropagation algorithm with consequentialism weight updates over mini-batches, Neurocomputing 461 (2021) 86–98.

[33] S.P. Mohanty, D.P. Hughes, M. Salathé, Using deep learning for image-based plant disease detection, Front. Plant Sci. 7 (2016).

[34] P. Helber, B. Bischke, A.R. Dengel, D. Borth, EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 12 (2017) 2217–2226.

[35] N.C.F. Codella, V.M. Rotemberg, P. Tschandl, M.E. Celebi, S.W. Dusza, D. Gutman, B. Helba, A. Kalloo, K. Liopyris, M.A. Marchetti, H. Kittler, A.C. Halpern, Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (ISIC), 2019, ArXiv, arXiv:1902.03368.

[36] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, R.M. Summers, Chestx-Ray8: Hospital-scale chest X-Ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, (CVPR), 2017, pp. 3462–3471.

[37] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J.B. Tenenbaum, H. Larochelle, R.S. Zemel, Meta-learning for semi-supervised few-shot classification, 2018, ArXiv, arXiv:1803.00676.