IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DEEP LEARNING TECHNOLOGIES FOR INTERNET OF VIDEO THINGS

Received March 31, 2021, accepted April 16, 2021, date of publication April 21, 2021, date of current version April 29, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3074640

An Alternating Training Method of Attention-Based Adapters for Visual Explanation of Multi-Domain Satellite Images

HEEJAE KIM[1], KYUNGCHAE LEE[1], CHANGHA LEE[1], SANGHYUN HWANG[2], AND CHAN-HYUN YOUN[1], (Senior Member, IEEE)

[1]School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea
[2]Agency for Defense Development, Daejeon 34060, South Korea

Corresponding author: Chan-Hyun Youn (chyoun@kaist.ac.kr)

This work was supported in part by the Defense Challengeable Future Technology Program of the Agency for Defense Development, South Korea, and in part by the Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean Government (Core Technology Research for Self-Improving Integrated Artificial Intelligence System) under Grant 20ZS1100.

ABSTRACT Recently, satellite image analytics based on convolutional neural networks have been vigorously investigated; however, in order for the artificial intelligence systems to be applied in practice, there still exists several challenges: (a) *model explanability* to improve the reliability of the artificial intelligence system by providing the evidence for the prediction results; (b) *dealing with domain shift* among images captured by multiple satellites of which the specification of the image sensors is various. To resolve the two issues in the development of a deep model for satellite image analytics, in this paper we propose a multi-domain learning method based on attention-based adapters. As plug-ins to the backbone network, the adapter modules are designed to extract domain-specific features as well as improve visual attention for input images. In addition, we also discuss an alternating training strategy of the backbone network and the adapters in order to effectively separate domain-invariant features and -specific features, respectively. Finally, we utilize Grad-CAM/LIME to provide visual explanation on the proposed network architecture. The experimental results demonstrate that the proposed method can be used to improve test accuracy, and its enhancement in visual explanability is also validated.

INDEX TERMS Deep network parametrization, multi-domain learning, satellite image analytics, visual explanation.

I. INTRODUCTION

With the development of remote sensing technology, satellite imagery is now being utilized for a variety of applications such as environmental monitoring. For instance, South Korea's satellite Chollian[1] provides meteorological and ocean information around Korean peninsula; the imagery data is analyzed for typhoon forecasting, the detection of climate change, and so on.

However, for some analysis tasks, the decipherment of the satellite imagery should be conducted by human experts. According to SIA,[2] if a 100km^2 (10km $\times$ 10km) image is supposed to be deciphered in 300m $\times$ 200m units, the human experts should annotate about 1650 sub-images for the entire image; this requires at least 200 hours.

Hence, the need of automating the annotation process is recently emerging; to achieve this, there have been attempts based on the convolutional neural network (CNN), e.g., [1]–[3]. Nevertheless, in order for the artificial intelligence (AI) systems (based on deep networks) to be applied in practice, there still exists a room for improvement; in this paper we focus on the following two points:

Model Explanability. Since most of analytics applications for satellite imagery are highly critical in their accuracy (e.g., disaster prediction), wrong decision of the AI systems could cause significant problems. Hence, the reliability of the AI systems become extremely important in this case; with regard to the prediction results provided by the AI systems, it should

The associate editor coordinating the review of this manuscript and approving it for publication was Jinjia Zhou.

[1]https://nmsc.kma.go.kr.
[2]https://www.si-analytics.ai.

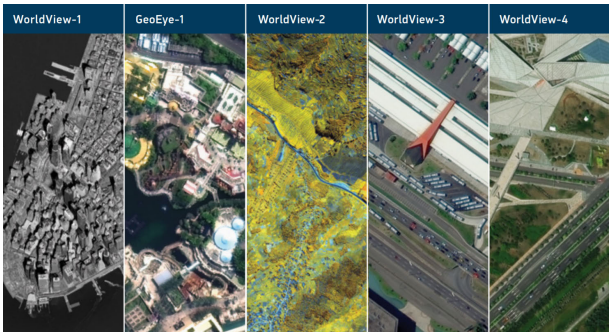


FIGURE 1. Multi-domain images provided by the Digital Globe[3] satellites [5]. The images lie on different domains according to operational altitude, spectral characteristics, and resolutions.

TABLE 1. Summary of four land-use aerial/satellite datasets from different domains.

Dataset	# Samples	# Classes	Spatial Resolution	Pixels
AID [8]	10000	30	$0.5 \sim 8$m	600×600
NWPU [7]	31500	45	$0.2 \sim 30$m	256×256
PatternNet [9]	30400	38	~ 0.8m	256×256
UC Merced [6]	2100	21	0.3m	256×256

be *explained how the results come out*. In this work, we particularly focus on visual interpretability of deep networks. In comparison with classical intelligent systems (such as rule-based) of which the decomposable pipelines allow each individual component to provide a natural intuitive explanation, a deep model could give better task performance, but its abstraction and complexity (arised from stacking lots of layers) make hard to interpret. To address it, Selvaraju *et al.* [4] proposed Grad-CAM as a scheme for the visual explanation to enhance localization of categories in images; our method is developed to make the best use of it.

Handling domain shift. If dealing with imageries from multiple satellites, we should consider *domain shift* among them. For instance, Fig. 1 show images captured by five different satellites; the domain shift occurs according to the specification of the image sensors. Also, Table 1 describes the details of four land-use aerial/satellite datasets, which were collected from different institutions [10], [11]. Given multi-domain images, we aim to build a deep network model to be suited well into the multiple source domains; this problem is referred to as *multi-domain learning* [12]. Here, the importance of constructing *universal representations* [13] stands out more than under domain adaptation of which the objective is to adapt the model from source domains to the (unlabeled) target domain.

In order to resolve the two issues in the development of a deep model for satellite image analytics, we propose a multi-domain learning method based on attention-based adapters. Plugged into the backbone network, the adapter modules are designed to improve visual attention for input images. In addition, given training data from multiple domains, the adapter modules capture domain-specific features of each domain, whereas domain-invariant features are extracted into the backbone network; to do so, we also introduce a training strategy to effectively split the domain-invariant and -specific features.

In summary, the main contributions of this paper are threefold. First, we propose a network architecture for multi-domain learning, based on attention-based adapters. As plug-ins to the backbone network, the adapter modules not only capture domain-specific features but also improve channel and spatial attention for input images. Second, we propose an alternating training strategy of the backbone and the adapters. In the approach, the training is performed by alternately freezing the two components; this allows to iteratively achieve effective separation of domain-invariant and -specific features. Third, we evaluate the proposed method on two kinds of multi-domain datasets, of which one includes aerial/satellite imagery. Through the experiments, we first demonstrate its effectiveness in classification performance, and the visual explanability is also validated.

II. PROBLEM DESCRIPTION

Fig. 2 describe our motivating scenario. In this figure, we assume that multiple satellites carry out remote sensing on a variety of regions. Also, we suppose that spatial resolutions and pixel sizes of the sensed images are various according to the satellites, which implies domain shift among them. Accordingly, each satellite would have images from different domains.

Periodically, each satellite transmits the captured images to the ground station; the received images are not only analyzed for the specific tasks (e.g., typhoon prediction, scene recognition), they can be also utilized to train machine learning models to automate the analysis process. In this work, we consider CNN-based deep networks for the automation process, and a situation is assumed when a sufficient amount of images has been collected in the ground station for training.

Along with human experts, the trained deep network is utilized to analyze upcoming images from the satellites; in addition, it also provides the visual evidence for the drawn results. Hence, the considered deep network model should have functionalities to support the explanability.

We now formulate a multi-domain problem for the above scenario. Given K domains (from the satellites), let $\mathcal{D}_k$ denote a distribution on each domain k ($1 \leq k \leq K$), where $\mathcal{D}$ stands for the mixture of all the domains. Here we consider image classification tasks; each domain distribution exists on the joint space $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are input and label spaces, respectively. We use $D_k = \{(\mathbf{x}_n^k, y_n^k)\}_{n=1}^{N_k}$ to denote a dataset from domain k, where $(\mathbf{x}_n^k, y_n^k) \sim \mathcal{D}_k$, and N_k is the number of the corresponding training samples. We also use $D = \bigcup_{k=1}^{K} D_k$ to denote the union set of the single domain datasets; correspondingly, N stands for the number of the total training samples in D.

In this formulation, the considered multi-domain learning problem has the following objective: to minimize the target

[3]https://www.maxar.com.

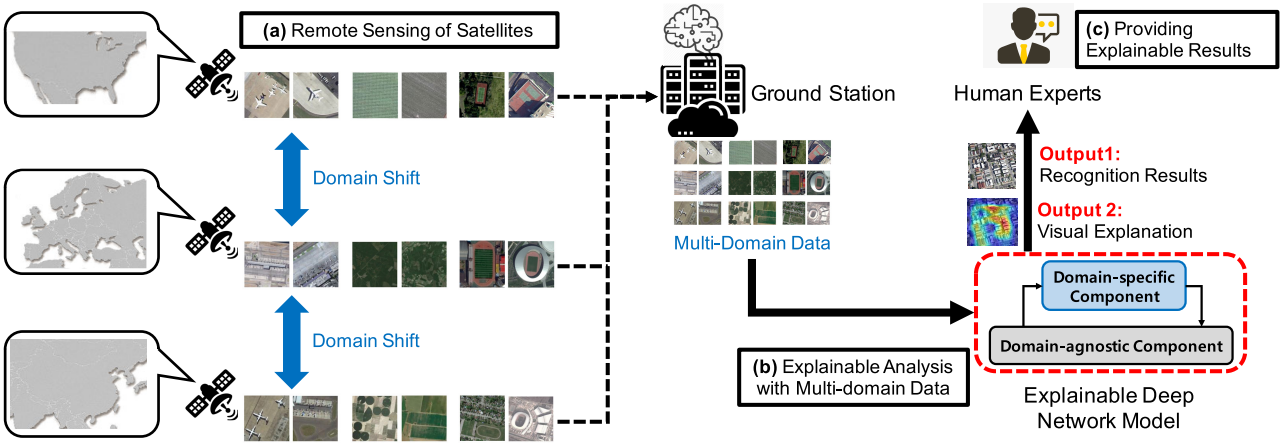


FIGURE 2. The considered scenario for explanable multi-domain learning with satellite images.

risk $\epsilon := \Pr_{(\mathbf{x},y)\sim\mathcal{D}}[h(\mathbf{x}) \neq y]$, where $h : \mathcal{X} \rightarrow \Delta\mathcal{Y}$ is the corresponding hypothesis and $\Delta\mathcal{Y}$ is the probability simplex over $\mathcal{Y}$. In addition, the hypothesis should be constructed to provide good visual attention so that it supports good visual explanability.

III. RELATED WORK

A. SATELLITE IMAGE ANALYTICS BASED ON DEEP LEARNING

First we review the existing literature on deep learning approaches to satellite image analytics. In [2], the authors proposed a method for fusing extracted features from multiple CNNs, in order to improve classification accuracy on the aerial/satellite datasets described in Table 1; Chaib *et al*. [15] also proposed a feature fusion method including a dimension reduction phase to speed up the classification task as well as to achieve better task performance. With regard to the similar datasets, Cheng *et al*. [16] presented a metric learning method to make features from different classes more discriminative. In [17], the authors proposed ARCNet that utilizes a recurrent attention mechanism to help discard non-critical information, which allows to focus on key region of images in training. In [18], the authors presented an architecture including pixel-set and temporal attention encoders in order to improve classification precision for satellite image time series.

In addition, there also have been studies to develop deep learning methods for addressing multi-domain issues on satellite image datasets, e.g., [11], [19]–[26]. Nevertheless, they mainly focus on *domain adaptation*; as mentioned in the introductory section, in this paper we deal with the related but different problem, *multi-domain learning*.

B. VISUAL EXPLANATION

Next we summarize the existing studies on providing visual explanation from deep networks. Class activation mapping (CAM) [27] is one of the first approaches to identify discriminative regions of input images from CNNs. By replacing fully-connected layers with convolutional layers and global average pooling [28], it achieves class-specific feature maps by weighting feature maps with respect to classes; there have been works that utilize the CAM for visual explanation of aerial/satellite imagery, such as [29]. However, the CAM has a drawback it is only applicable to CNN architectures utilizing global average pooling over the final convolutional feature maps; to overcome this limitation, Selvaraju *et al*. [4] proposed Grad-CAM (of which the detailed procedure is described in Section IV-C); consequently, we consider the Grad-CAM as the main explanation method in this paper. It is noted that Chattopadhay *et al*. presented Grad-CAM++ [30] as the extension of Grad-CAM, in order to improve object localization and occurrence explanation of multiple instances in a single image; since the difference between the two approaches lies in only how to compute neuron importance weights, our method also would be fitted well into the Grad-CAM++.

In addition to the above activation-based strategies, there also exist perturbation-based approaches to identify which part of input data contributes to the prediction. For instance, Ribeiro *et al*. [31] presented LIME that learns an interpretable model locally around the prediction. As a model-agnostic method, the LIME determines the parts to be highlighted by learning the weights of each perturbation of a sample; an evaluation using this approach will be also conducted in the experimental section. With the similar philosophy to the LIME, SHAP [32] computes the weights based on Shapley values. We also note that there exists an approach that utilizes integrated gradients from the baseline (e.g., the black image) to the input in order to satisfy sensitivity and implementation invariance [33].

C. MULTI-DOMAIN LEARNING

Given source domains, multi-domain learning aims to learn a model in order to improve the task performance across the domains. Here we summarize the existing studies on the multi-domain learning and describe their limitations.

One of the promising strategy for the multi-domain learning is to capture domain-invariant features to minimize domain discrepancy while guaranteeing domain-specific task

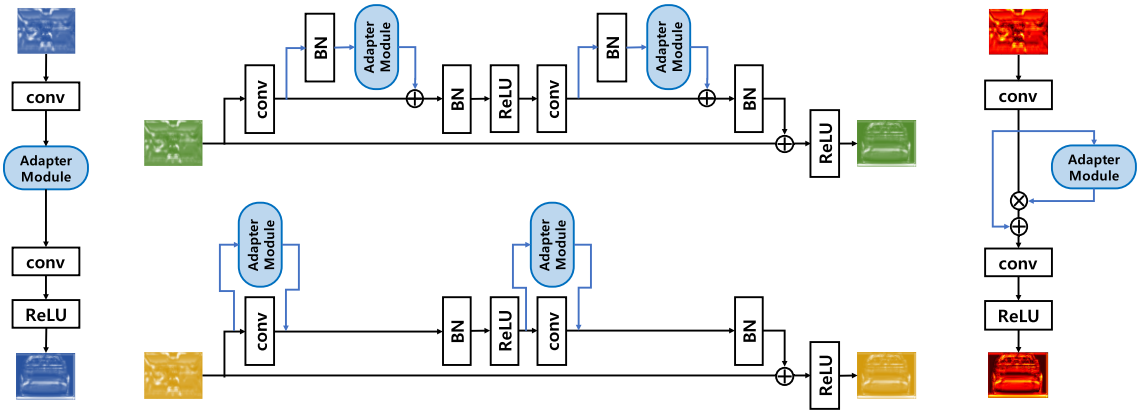


FIGURE 3. Various adapter modules for multi-domain learning. Left: series adapter [44]. Middle top: series residual adapter [12], [43], [44]. Middle bottom: parallel residual adapter [42]. Right: the proposed adapter. Note that the series and parallel residual adapters are mainly applied with ResNet [45]; BN denotes batch normalziation.

performance. To achieve this, In [34], the authors proposed domain separation networks for domain adaptation, consisting of private networks for each domain and the shared network. In their model, each of the private networks learn to extract domain-specific features, whereas the shared network is trained to capture domain-invariant features. As its extension, Liu *et al*. [35] modified the network architecture to apply to multi-domain learning; the authors introduced the joint adversarial loss to prevent the mixture of samples from different classes across domains during domain-invariant feature extraction, and they also proposed a method for the orthogonal regularization between private features across domains. In [36], the authors proposed a method based on multi-task learning; in their method, each domain-specific features are co-embedded into a common sparse space, and the co-embedded features are fused to extract domain-invariant features.

However, many of these approaches require large parameter spaces to extract the specific features of each domain. Instead, to reduce parameter sizes of the domain-specific components, lightweight adapter-based methods have been also introduced, in which domain-specific features are extracted into the adapter modules. In [13], the authors proposed to exploit the parameters in batch normalization [37] and instance normalization [38], [39] layers as adapter modules; the remaining core model parameters are shared for every domain. Also, expanding this, Berriel *et al*. [40] presented a budget-aware adapter that selects the most relevant feature channels for each domain; to reduce the computational complexity of the model, in their method, the channel selection is performed under a given complexity budget.

In [12], the authors proposed *series residual adapter modules* (mainly for ResNet [45]), which consist of a 1×1 filter bank in parallel with a skip connection (middle top panel in Fig. 3). The adapter modules are applied (as plug-ins) in each residual block of the main backbone network. If we denote an intermediate feature map as $\mathbf{F} \in \mathbb{R}^{H \times W \times C}$, the refined output becomes

$$\mathbf{F}' = \mathbf{F} + \mathrm{diag}(\pi) * \mathbf{F}, \tag{1}$$

where $\pi \in \mathbb{R}^{C \times C}$ is the adapter module, and $\mathrm{diag}(\cdot)$ is the operator to reshape the input matrix in a diagonal filter bank. It is noted that the controller modules in [41] can be thought as one of the series residual adapters. Also, Li *et al*. proposed a data driven method, called *covariance normalization*, in order to effectively reduce the size of the adapter parameters via two principal component analyzes (PCAs); the series residual adapters are primarily considered in their evaluation. Meanwhile, [42] presented *parallel residual adapter modules* (middle bottom panel in Fig. 3); that is, denoting the existing convolution filter bank as $\mathbf{f}$, the refined output is computed as

$$\mathbf{F}' = \mathbf{f} * \mathbf{E} + \mathrm{diag}(\pi) * \mathbf{E}, \tag{2}$$

where $\mathbf{E}$ is a feature map from the previous layer; note that $\mathbf{F} = \mathbf{f}(\mathbf{E})$. In addition, there have been methods that exploit *series adapter modules* (left panel in Fig. 3). For instance, Zhao *et al*. [44] proposed a neural architecture search (NAS)-based approach that adaptively discover the structure of the adapter modules for different domains to balance between the effectiveness and compactness; it is noted that, in addition to the series adapter, the series residual adapter is also considered in the adapter module selection for each layer.

Besides, there also exist studies that utilize other types of adapters. In [46], the authors presented an adaptive parameterization approach, in which adapter sizes are determined by the level of complexity of each domain. With regard to the location of adapters, Xiao *et al*. [47] exploited an adapter consisting of two convolutional layers only after the feature extraction network. It is additionally noted that, assuming no prior knowledge of domain labels, Deecke *et al*. [48] proposed a dynamic residual adapter as an adaptive gating mechanism to help account for latent domains.

Limitations: However, these approaches based on adapter modules have the two limitations. First, in the existing

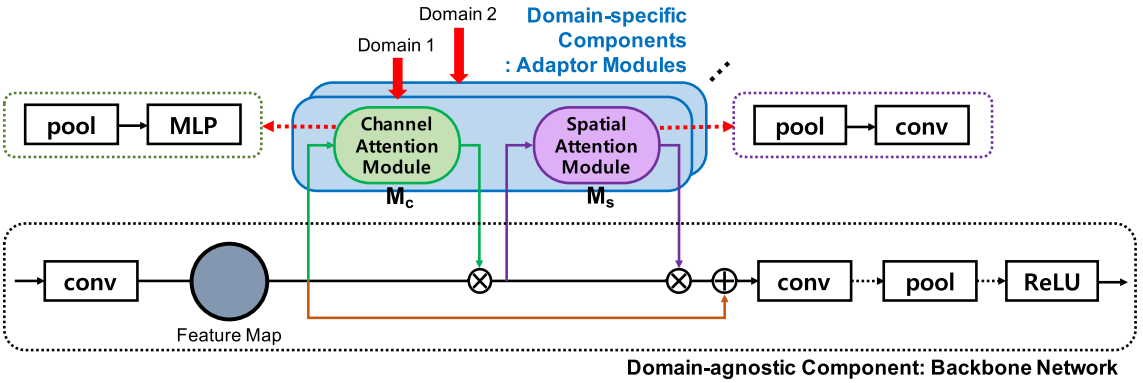


FIGURE 4. The proposed network architecture. A CNN is used as the domain-agnostic backbone, and we also use two attention modules for each convolutional layer as domain-specific components.

approaches, adapter modules were not designed to improve visual explanability. That is, even though the adapter modules can help to improve domain-specific task performance upon the domain-agnostic backbone, they do not consider additional process to motivate the improvement of the visual attention for input images. We note that Yang *et al.* [49] proposed a deep attention adapter based on the attention mechanism of ECA-Net [50]; however, it still has the problem in the separation of domain-invariant and -specific features, as described in the following paragraph.

Second, most of the adapter-based approaches mainly employ sequential learning process of domain-agnostic and -specific components. That is, in their approaches, (a) the domain-agnostic backbone is pre-trained on a general dataset (e.g., ImageNet [51]); (b) the domain-specific components are then trained by freezing the backbone parameters, for each single domain dataset. Even though Rebuffi *et al.* [12] presented the experimental results when applying their method to end-to-end learning, in order to train (or fine-tune) the backbone, they used a naïve training approach that samples mini-batches from each domain in a round robin fashion, which has a limitation on effectively separating domain-invariant and -specific features. In addition, the reviewed approaches are not suitable for training from scratch in common.

The Proposed Solution: To overcome the first limitation, we develop our adapter modules to improve visual explanability by investigating what and where to focus on input images. In the proposed adapter modules, we employ two attention processes with regard to channel and spatial information. Also, to resolve the second limitation, we propose an alternating training strategy for the separation of domain-invariant and -specific features. In each iteration, the domain-agnostic backbone and the domain-specific adapters are updated sequentially by freezing each other; the iterative freezing allows to effectively extract domain-specific features while the remaining features (i.e., domain-invariant) are captured by the backbone. Hence, the proposed strategy enables to train the network from scratch, unlike the reviewed adapter-based methods.

IV. A PROPOSED MULTI-DOMAIN LEARNING METHOD FOR BETTER VISUAL EXPLANATION

In this section, we propose our method for multi-domain learning to enhance the visual explanation. The proposed method consists of three phases, and the following sub-sections describe each of them; in the end of this section, the whole procedure is described in Algorithm 1.

A. PHASE 1: CONSTRUCTING ATTENTION-BASED ADAPTERS UPON THE BACKBONE NETWORK

For multi-domain learning, we take an approach to train multiple adapters with respect to each domain, in addition to the backbone network. $\mathbf{w}_k = \{\mathbf{w}, \mathbf{m}_k\}$ denotes model parameters for domain k, where $\mathbf{w}$ is the domain-agnostic component shared for all the domains, and $\mathbf{m}_k$ is each domain-specific component (i.e., adapter modules), respectively. In the proposed network architecture, a CNN (e.g., AlexNet [52], ResNet [45], etc.) is used as the backbone, and we adopt CBAM [14] for the domain-specific components, as illustrated in Fig. 4.

The CBAM consists of two sequential sub-modules: the channel and the spatial attention module, which are applied to each convolutional layer in order to adaptively refine the extracted features. In training, we use the two sub-modules as domain-specific components; thus, a set of them corresponds to $\mathbf{m}_k$ for each domain k. For an intermediate feature map $\mathbf{F} \in \mathbb{R}^{H\times W\times C}$ extracted from each convolutional layer, let $\mathbf{M_c} \in \mathbb{R}^{1\times 1\times C}$ and $\mathbf{M_s} \in \mathbb{R}^{H\times W\times 1}$ denote its 1-D channel attention map and 2-D spatial attention map, respectively. Then, the overall attention process is given as,

$$\mathbf{F}' = \mathbf{M_c}(\mathbf{F}) \otimes \mathbf{F}, \tag{3}$$

$$\mathbf{F}'' = \mathbf{M_s}(\mathbf{F}') \otimes \mathbf{F}', \tag{4}$$

where $\otimes$ is element-wise multiplication. Note that $\mathbf{F}''$ is the final refined output of the convolutional layer.

However, different from the original CBAM, in our attention process, the final output of each convolutional layer is computed as

$$\mathbf{F}'' = \mathbf{M_s}(\mathbf{F}') \otimes \mathbf{F}' + \mathbf{F}. \tag{5}$$

Algorithm 1 Overall Algorithm. Note That $\mathbf{w}_k = \{\mathbf{w}, \mathbf{m}_k\}$

Input: Training dataset D, the learning rate η, test images to be examined on the trained network
Output: $\mathbf{w}$, $\mathbf{m}_k$ for every domain k, and Grad-CAM results for the test images

/* Phase 1: Constructing attention-based adapters upon the backbone network*/
construct the adapters $\mathbf{m}_k$ upon $\mathbf{w}$ for every domain k
/* Phase 2: Alternating training of domain-agnostic and -specific components */
repeat
  for $k = 1, \cdots, K$ **do**
    compute $\mathcal{L}(\mathbf{w}_k; \xi \sim \mathcal{D})$
  end for
  /* Freeze $\{\mathbf{m}_k\}$ and update $\mathbf{w}$ */
  $\mathbf{w} \leftarrow \mathbf{w} - \eta \sum_{k=1}^{K} \frac{N_k}{N} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}_k; \xi \sim \mathcal{D})$
  for $k = 1, \cdots, K$ **do**
    compute $\mathcal{L}_k(\mathbf{w}_k; \zeta_k \sim \mathcal{D}_k)$
    /* Freeze $\mathbf{w}$ and update $\mathbf{m}_k$ */
    $\mathbf{m}_k \leftarrow \mathbf{m}_k - \eta \nabla_{\mathbf{m}_k} \mathcal{L}_k(\mathbf{w}_k; \zeta_k \sim \mathcal{D}_k)$
  end for
until convergence
/* Phase 3: Visual explanation on the trained network */
compute Grad-CAM on the trained network for the test images

Empirically, we identified that the modified attention process can improve the separation of domain-invariant and -specific features to the CNN backbone and the two attention modules, respectively.

Here, the channel attention map $\mathbf{M_c}(\mathbf{F})$ is constructed by analyzing the inter-channel relationship of features. That is, in order to find the channels that need to give more attention for an input image $\mathbf{x}$, the following operation is conducted: (a) Spatial information of the intermediate feature map is aggregated from average-pooling and max-pooling. (b) The both descriptors with each pooling operation, denoted as $\mathbf{F^c_{avg}}$ and $\mathbf{F^c_{max}}$ respectively, are then forwarded to multi-layer perceptron (MLP) with one hidden layer as

$$\begin{aligned}\mathbf{M_c}(\mathbf{F}) &= \sigma(MLP(AvgPool(\mathbf{F})) + MLP(MaxPool(\mathbf{F}))) \\ &= \sigma(\mathbf{W_1}(\mathbf{W_0}(\mathbf{F^c_{avg}})) + \mathbf{W_1}(\mathbf{W_0}(\mathbf{F^c_{max}}))), \end{aligned} \tag{6}$$

where σ denotes the sigmoid function, and $\mathbf{W_0}$ and $\mathbf{W_1}$ denote weights of the MLP layers.

Given the channel-refined features, the followed spatial attention module produces the spatial attention map $\mathbf{M_s}$ from inter-spatial relationship of them, of which the goal is to enhance where to focus on an input image. To achieve this, both average-pooling and max-pooling are firstly applied for the channel-refined feature map, as in the channel attention module; then, the concatenated descriptor $[\mathbf{F'^s_{avg}}; \mathbf{F'^s_{max}}]$ is forwarded for an additional convolutional operation f applied to create the spatial attention map, as follows:

$$\begin{aligned}\mathbf{M_c}(\mathbf{F'}) &= \sigma(f * [AvgPool(\mathbf{F'}); MaxPool(\mathbf{F'})]) \\ &= \sigma(f * [\mathbf{F'^s_{avg}}; \mathbf{F'^s_{max}}]). \end{aligned} \tag{7}$$

By applying the two attention processes, we can have improved intermediate feature maps for visual explanation, which contain better information of what and where to focus on input images. Also, in this work, we additionally assign the role of the extraction of domain-specific features to the adapter modules; accordingly, it can be achieved that the proposed network architecture provides both the improved visual explanability and domain robustness at the same time.

B. PHASE 2: ALTERNATING TRAINING OF DOMAIN-AGNOSTIC AND -SPECIFIC COMPONENTS

The proposed network architecture has been developed to extract domain-invariant features into the backbone network, whereas domain-specific features are extracted to the adapter modules. In addition, the parameters in the adapter modules should be able to have improved channel and spatial attention for better visual explanation. To achieve this, now we introduce a training approach for the domain-agnostic and -specific components based on *alternating optimization*, which is an iterative procedure to minimize the objective function jointly over all variables by alternating restricted minimizations over the individual subsets of variables [53].

In the training process, we consider the following objective:

$$\textbf{objective}: \min_{\mathbf{w}, \mathbf{m}_1, \cdots, \mathbf{m}_K} \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_k(\mathbf{w}_k), \tag{8}$$

where $\mathcal{L}_k(\mathbf{w}_k) = \mathbb{E}_{\zeta_k \sim \mathcal{D}_k}[\mathcal{L}_k(\mathbf{w}_k; \zeta_k)]$ is the objective function (e.g., cross-entropy loss) for domain k, and ζ_k denotes the unbiased randomness in the minibatch construction under $\mathcal{D}_k$.

As mentioned in the previous section, for the separation of domain-invariant and -specific features, the reference [12] considers a strategy of firstly training the shared component (i.e., $\mathbf{w}$ in our case) by sampling minibatches from each domain in a round robin fashion, then subsequently training each domain-specific components with the corresponding domain dataset.

Instead, our method is developed so that the domain-agnostic and -specific components are jointly trained in each iteration; it also allows to train them from scratch. As illustrated in Fig. 5, the domain-agnostic component $\mathbf{w}$ is firstly trained (from the merged training samples for all the domains) by fixing the current adapter modules for each domain k (i.e., $\mathbf{m}_k$); the $\mathbf{w}$ is updated by the averaged gradients for each domain according to the number of training samples. Subsequently, each of the $\mathbf{m}_k$ is trained (from the corresponding single domain samples) by freezing the updated $\mathbf{w}$, in order to extract domain-specific features. By doing this, as domain-specific features are extracted into the adapter modules, the remaining domain-invariant features becomes concentrated on the backbone.

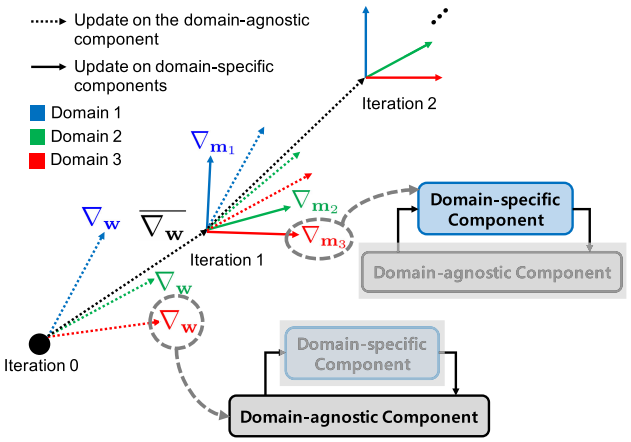


FIGURE 5. Illustration of the proposed alternating training strategy, which allows to separate domain-invariant and domain-specific features.

In summary, the update rule of the proposed alternating training in each iteration is described as follows:

1) **Freeze $\{\mathbf{m}_k\}$ and update $\mathbf{w}$.** Compute $\mathcal{L}(\mathbf{w}_k; \xi \sim \mathcal{D})$, where ξ denotes the unbiased randomness in the mini-batch construction under $\mathcal{D}$. Then, update $\mathbf{w}$ as $\mathbf{w} - \eta \sum_{k=1}^{K} \frac{N_k}{N} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}_k; \xi \sim \mathcal{D})$, where η is the learning rate.
2) **Freeze $\mathbf{w}$ and update $\{\mathbf{m}_k\}$.** Compute $\mathcal{L}_k(\mathbf{w}_k; \zeta_k \sim \mathcal{D}_k)$ for each domain k. Then, update $\mathbf{m}_k$ as $\mathbf{m}_k - \eta \nabla_{\mathbf{m}_k} \mathcal{L}_k(\mathbf{w}_k; \zeta_k \sim \mathcal{D}_k)$ for every domain k.

We now analyze the convergence of the proposed alternating training strategy. In the analysis, we use $\mathcal{L}(\mathbf{w}_k)$ to denote $\mathbb{E}_{\xi \sim \mathcal{D}}[\mathcal{L}(\mathbf{w}_k; \xi)]$. Also, we use $\mathbf{w}_k^{(t)}$ to denote the model parameters for domain k at t-th iteration; $\mathbf{w}_k^{(0)}$ stands for the initial parameters.

We first provide some assumptions and lemma for the convergence proof.

Assumption 1 (smoothness): $\mathcal{L}_k(\theta)$ is β-smooth, $\forall k$; $\mathcal{L}_k(\theta_1) \leq \mathcal{L}_k(\theta_2) + \langle \nabla \mathcal{L}_k(\theta_2), \theta_1 - \theta_2 \rangle + \frac{\beta}{2} \|\theta_1 - \theta_2\|^2, \forall \theta_1, \theta_2$.

Assumption 2 (bounded gradient of $\mathbf{w}$): There exists a constant G such that $\mathbb{E}\|\nabla_{\mathbf{w}} \mathcal{L}_k(\mathbf{w}_k^{(t)})\|^2 \leq G^2$, and the following holds: $\|\nabla_{\mathbf{w}} \mathcal{L}_k(\mathbf{w}_k^{(t)}) - \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}_{i \neq k}^{(t)})\| \leq \|\nabla_{\mathbf{w}} \mathcal{L}_k(\mathbf{w}_k^{(t)}) - \nabla_{\mathbf{w}} \mathcal{L}_k(\mathbf{w}_{i \neq k}^{(t)})\|, \forall k, i, t$.

Assumption 3 (bounded gradient of $\mathbf{m}_k$): There exists constants Q_k and R_k such that $\mathbb{E}\|\nabla_{\mathbf{m}_k} \mathcal{L}_k(\mathbf{w}_k^{(t)})\|^2 \leq Q_k^2$ and $\mathbb{E}\left\|\nabla_{\mathbf{m}_k} \mathcal{L}_k\left(\mathbf{w}_k^{(t)} - \frac{\eta}{K} \sum_{i=1}^{K} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}_i^{(t)})\right)\right\|^2 \leq R_k^2, \forall k, t$,

Lemma 1: If Assumption 2 is satisfied, the following holds under the proposed alternating training strategy:

$$\mathbb{E}\left\|\nabla_{\mathbf{w}} \mathcal{L}_k(\mathbf{w}_k^{(t-1)}) - \frac{1}{K} \sum_{i=1}^{K} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}_i^{(t-1)})\right\|^2 \leq \frac{4(K-1)G^2}{K}. \quad (9)$$

Proof: The proof is available in Appendix VI. ■

In order to establish the convergence bound of the alternating training strategy, we provide Theorem 1 as follows. In this theorem, for simplicity, the learning rate scheduling is not considered, and we suppose that N_k is identical for every domain k.

Theorem 1: If Assumption 1, 2, and 3 hold, the proposed alternating training strategy ensures the following for $0 < \eta \leq \frac{1}{\beta}$ and $T \geq 1$:

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\|\nabla \mathcal{L}_k(\mathbf{w}_k^{(t-1)})\|^2 \leq \frac{2}{\eta T}\left(\mathbb{E}[\mathcal{L}_k(\mathbf{w}_k^{(0)})] - \mathcal{L}_k^*\right) + \beta\delta\eta + \frac{16(K-1)G^2}{K} + 4Q_k^2 + 2R_k^2,^4 \quad (10)$$

where $\mathcal{L}_k^*$ is the optimal value for domain k, and δ is the smallest value such that $\mathbb{E}\left\|\frac{1}{K}\sum_{i=1}^{K} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}_i^{(t-1)}; \xi^{(t)}) + \nabla_{\mathbf{m}_k} \mathcal{L}_k\left(\mathbf{w}_k^{(t-1)} - \frac{\eta}{K}\sum_{i=1}^{K} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}_i^{(t-1)}; \xi^{(t)}); \zeta_k^{(t)}\right)\right\|^2 \leq \left(\mathbb{E}\left\|\frac{1}{K}\sum_{i=1}^{K} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}_i^{(t-1)}) + \mathbf{g}_k^{(t)}\right\|^2 + \delta\right)$.

Proof: Here we provide a sketch of the proof. The complete proof is available in See Appendix VI.

By Assumption 1, we have

$$\begin{aligned}\mathbb{E}[\mathcal{L}_k(\mathbf{w}_k^{(t)})] \leq\ & \mathbb{E}[\mathcal{L}_k(\mathbf{w}_k^{(t-1)})] \\ & + \mathbb{E}\left\langle \nabla \mathcal{L}_k(\mathbf{w}_k^{(t-1)}), \mathbf{w}_k^{(t)} - \mathbf{w}_k^{(t-1)} \right\rangle \\ & + \frac{\beta}{2} \mathbb{E}\|\mathbf{w}_k^{(t)} - \mathbf{w}_k^{(t-1)}\|^2. \end{aligned} \quad (11)$$

Then, letting $\mathbf{g}_k^{(t)} = \nabla_{\mathbf{m}_k} \mathcal{L}_k\left(\mathbf{w}_k^{(t-1)} - \frac{\eta}{K}\sum_{i=1}^{K} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}_i^{(t-1)})\right)$, we get

$$\begin{aligned}& \mathbb{E}\left\langle \nabla \mathcal{L}_k(\mathbf{w}_k^{(t-1)}), \mathbf{w}_k^{(t)} - \mathbf{w}_k^{(t-1)} \right\rangle \\ & \leq -\frac{\eta}{2}\left(\mathbb{E}\|\nabla \mathcal{L}_k(\mathbf{w}_k^{(t-1)})\|^2 \right. \\ & \quad + \mathbb{E}\left\|\frac{1}{K}\sum_{i=1}^{K} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}_i^{(t-1)}) + \mathbf{g}_k^{(t)}\right\|^2 \\ & \quad \left. - \frac{16(K-1)G^2}{K} - 4Q_k^2 - 2R_k^2, \right). \end{aligned} \quad (12)$$

Also we get

$$\begin{aligned}& \mathbb{E}\|\mathbf{w}_k^{(t)} - \mathbf{w}_k^{(t-1)}\|^2 \\ & \leq \eta^2\left(\mathbb{E}\left\|\frac{1}{K}\sum_{i=1}^{K} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}_i^{(t-1)}) + \mathbf{g}_k^{(t)}\right\|^2 + \delta\right). \end{aligned} \quad (13)$$

By combining the inequalities (12) and (13) into (11) and taking an average over $1 \leq t \leq T$, we finally have the inequality (10). ■

C. PHASE 3: VISUAL EXPLANATION ON THE TRAINED NETWORK

The final phase of our method is visual explanation for input images on the trained network in Phase 2. As a visual explanation tool, we utilize Grad-CAM [4].

Grad-CAM is a gradient-weighted class activation approach to improve visual explanability of the CNN. The earlier work CAM [27] utilizes the insight that the last convolutional layer is expected to have the best representation

[4] In (10), we use the expected squared gradient norm to characterize the convergence rate, as in [54]–[58].

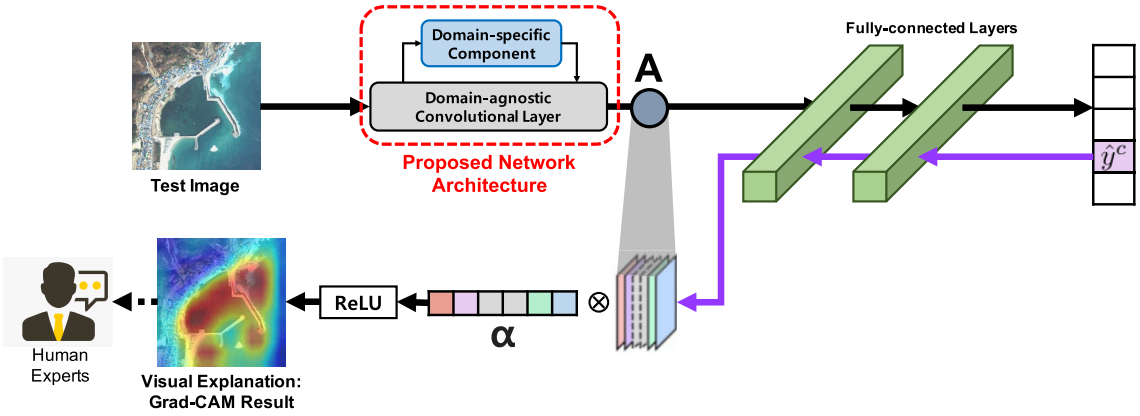


FIGURE 6. Illustration of providing visual explanation for input images on the trained network.

for semantic and spatial information; thus, the neurons in the last convolutional layer would be the most appropriate to capture class-specific information for an input image. Under the similar philosophy, the Grad-CAM distinctively utilizes the gradient information of the last convolutional layer to assign importance weights to the feature maps.

Firstly, Grad-CAM obtains the gradient of $\hat{y}^c$ (before the softmax), i.e., the score for class c, with regard to the feature map $\mathbf{F}^u$, where u denotes its channel index. The obtained gradients are then forwarded to global average pooling [28] to get the neuron importance weight as:

$$\alpha_u^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial \hat{y}^c}{\partial \mathbf{F}_{ij}^u}, \tag{14}$$

where height and weight dimensions are indexed by i and j, respectively.

After the importance weight α_k^c is obtained, the weighted linear combination of the feature maps is conducted as $ReLU(\sum_u \alpha_u^c \mathbf{F}^u)$; this result provides a coarse class-discriminative localization map that visualizes the important regions for the input image.

Until now, Grad-CAM has proven its effectiveness by being applied to various studies. As described in Fig. 6, in this work the Grad-CAM is exploited as a scheme to provide the visual evidence for the drawn results from the proposed network; this would assist human experts to make the final decision.

V. EVALUATION

In this section, we experimentally demonstrate the advantages of the proposed method via comparison with other approaches.

A. COMPARED METHODS AND EVALUATION METRICS

Throughout this section, we mainly compare the following three approaches for training from scratch; furthermore, several state-of-the-art methods are also considered for the evaluation.

- **Share:** A single network is trained using the merged training samples from all the domains. For the network model, the backbone CNN (i.e., $\mathbf{w}$) is used in training.
- **Share + CBAM:** Similar to the Share, a single network is trained with the merged training samples from all the domains; however, the proposed network architecture (described in Section IV-A) is utilized in training.
- **Share + CBAM + AT:** Similar to the Share + CBAM, the proposed network architecture is utilized in training; moreover, the alternating training (AT) strategy (described in Section IV-B) is additionally applied.

For evaluation metrics, we first consider test accuracy. The test accuracy is measured with test samples for each domain; the averaged one is computed as

$$\mathcal{A}(\mathbf{w}_1, \cdots, \mathbf{w}_K) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{A}_k(\mathbf{w}_k), \tag{15}$$

where $\mathcal{A}_k(\cdot)$ denotes test accuracy on domain k, which is calculated from test samples for each domain. Next, F1-score is utilized to consider imbalancedness of the evaluation datasets; we compute an averaged F1-score for all the domains as

$$F_1(\mathbf{w}_1, \cdots, \mathbf{w}_K) = \frac{1}{K} \sum_{k=1}^{K} \frac{2 \cdot \text{precision}(\mathbf{w}_k) \cdot \text{recall}(\mathbf{w}_k)}{\text{precision}(\mathbf{w}_k) + \text{recall}(\mathbf{w}_k)}, \tag{16}$$

where $\text{precision}(\mathbf{w}_k)$ and $\text{recall}(\mathbf{w}_k)$ are the precision and the recall values for domain k. Note that in the following the result values are reported as the mean $\pm$ standard deviation of ten independent runs with different random seeds. In addition, we also consider Grad-CAM [4] and LIME [31] visualization results for the evaluation; they will be compared qualitatively.

B. PACS DATASET

a: SETUP

PACS dataset [59] contains 9991 images of seven classes (dog, elephant, giraffe, guitar, horse, house, and person);

TABLE 2. Test accuracy and F1-score (%) on PACS dataset with respect to different adapter types for multi-domain learning.

		Art Painting	Cartoon	Photo	Sketch	**Average**
Serial [44]	Acc.	43.17 ± 3.58	69.23 ± 2.71	65.93 ± 2.51	76.39 ± 1.75	**63.68 ± 1.49**
	F1-score	41.26 ± 4.74	69.31 ± 3.43	58.75 ± 3.09	77.07 ± 2.00	**61.60 ± 1.48**
Serial Residual [12], [43], [44]	Acc.	53.37 ± 4.40	78.30 ± 2.06	73.95 ± 4.11	86.39 ± 1.93	**73.00 ± 1.94**
	F1-score	52.63 ± 4.73	78.98 ± 2.23	69.66 ± 4.21	87.94 ± 2.53	**72.30 ± 2.10**
Parallel Residual [42]	Acc.	51.41 ± 2.42	77.49 ± 2.56	73.41 ± 2.71	85.60 ± 1.98	**71.98 ± 1.32**
	F1-score	50.89 ± 2.50	77.83 ± 3.57	68.74 ± 2.43	86.15 ± 2.47	**70.90 ± 1.17**
Proposed	Acc.	54.93 ± 3.19	78.94 ± 2.72	74.97 ± 2.87	84.96 ± 2.10	**73.45 ± 1.49**
	F1-score	53.96 ± 4.27	79.38 ± 2.92	70.87 ± 2.78	86.54 ± 2.42	**72.69 ± 1.45**

TABLE 3. Test accuracy and F1-score (%) on PACS dataset with respect to different multi-domain training methods.

		Art Painting	Cartoon	Photo	Sketch	**Average**
Share	Acc.	39.66 ± 3.38	61.57 ± 4.07	58.80 ± 4.27	62.95 ± 2.74	**55.75 ± 1.66**
	F1-score	37.14 ± 3.04	61.18 ± 4.56	51.36 ± 4.22	55.32 ± 2.22	**51.25 ± 1.84**
Share + CBAM	Acc.	43.02 ± 3.35	67.91 ± 4.82	63.77 ± 4.90	69.85 ± 2.42	**61.14 ± 1.46**
	F1-score	40.98 ± 3.24	68.13 ± 4.84	56.96 ± 5.12	70.28 ± 3.85	**59.09 ± 1.75**
Share + CBAM + AT	Acc.	46.15 ± 3.50	70.64 ± 3.97	69.52 ± 4.55	75.75 ± 1.89	**65.51 ± 1.91**
	F1-score	43.99 ± 3.85	71.01 ± 4.41	64.18 ± 4.73	75.71 ± 2.50	**63.72 ± 2.01**

they belong to four different domains: Photo, Art Painting, Cartoon, and Sketch. Similar as in [59], we randomly divided each domain data into 90% as training data and 10% as test data. For the base network model, we used Caffenet, a variant of AlexNet [52]; the detailed architecture can be found in [60]. For training, we used pure SGD as an optimization method to minimize the cross-entropy loss, with a weight decay of 0.0005. The trainings were conducted for 2000 iterations with a minibatch size of 128. The initial rate of η was set to 0.01; we dropped the learning rate by 0.1 at 80% of the total training iterations. Also, we used random cropping and horizontal flipping in the data augmentation pipeline of all the compared strategies.

TABLE 4. Construction of the multi-domain satellite/aerial training samples.

Class	AID	NWPU	PatternNet	UC Merced
Airfield	360	1400	800	100
Anchorage	380	700	800	100
Beach	400	700	800	100
Dense Residential	410	700	800	100
Farm	370	1400	800	100
Flyover	420	700	800	100
Forest	250	700	800	100
Game Space	660	1400	1600	100
Parking Space	390	700	800	100
River	410	700	800	100
Sparse Residential	300	700	800	100
Storage Cisterns	360	700	800	100
Total	4710	10500	10400	1200

b: RESULTS

From Table 2, we compare test accuracy and F1-score on PACS dataset with respect to the four different types of domain adapters. In the experiments, the Share scheme has been applied for the training, and for fair comparison, we ignored batch normalization [37] layers in common. As shown in the table, the proposed adapter architecture outperforms the other three state-of-the-art architectures in view of both the evaluation metrics. Except for ours, it is seen that the series residual adapter shows the best performance values; the serial adapter provides noticeable lower outcomes compared to the other threes.

Following this, Table 3 depicts the effectiveness of the proposed alternating training strategy on PACS dataset. Here it is worth noting that we found in experiments that the proposed alternating training strategy does not provide a notable effect when the learning rate is relatively high, but its effect increases as lowering the learning rate; the table shows the results under the initial learning rate of 0.001. Under the low learning rate, we see from the table that the alternating training strategy provides the performance gain of ∼4.5 % in both averaged test accuracy and F1-score, compared to Share + CBAM. Also, from the results under Share + CBAM, we can see that the proposed network architecture produces better outcome than Share, even without the alternating training. In the case of Share, it not only has the lowest test accuracy and F1-score, but also the largest gap between the two evaluation metrics; particularly on Sketch, the Share yields noticeably greater gap than the two other approaches.

Figure 7 shows the Grad-CAM visualization results. From the figure, we can see that the proposed methods allow to focus better on the regions where the target object is located. With regard to Art Painting, it is observed that Share + CBAM + AT improves visual explanability by providing the attention on the characteristic part of the target object (e.g., the tongue of dog and the horns of giraffe). For the horse

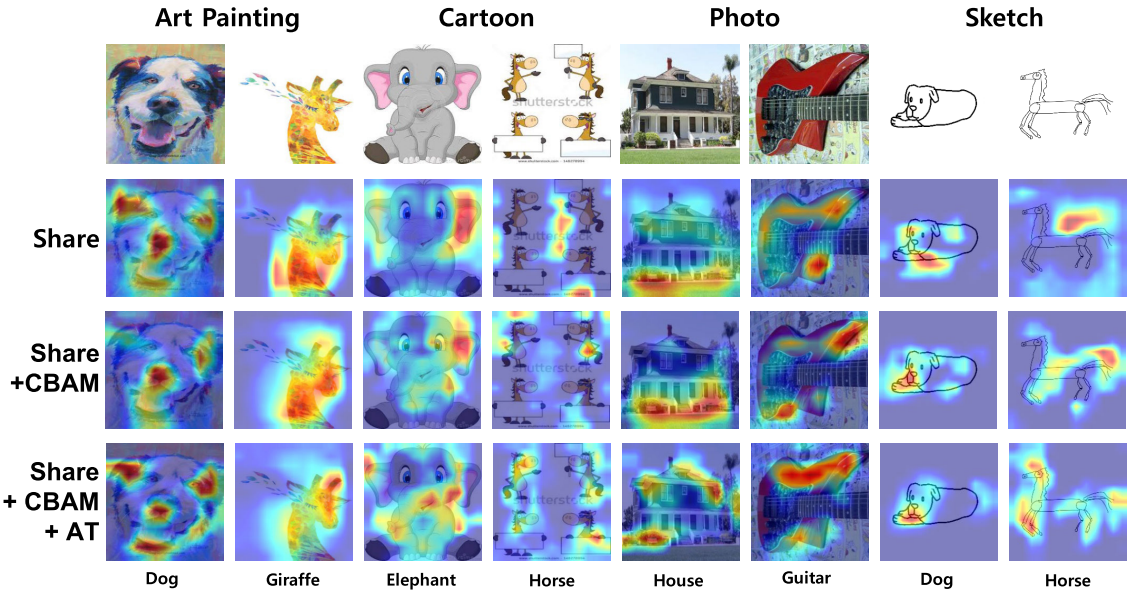


FIGURE 7. Selected Grad-CAM visualization results on PACS dataset. The Grad-CAM visualization is computed for the output of the last convolutional layer. The ground-truth label is shown on the bottom of images.

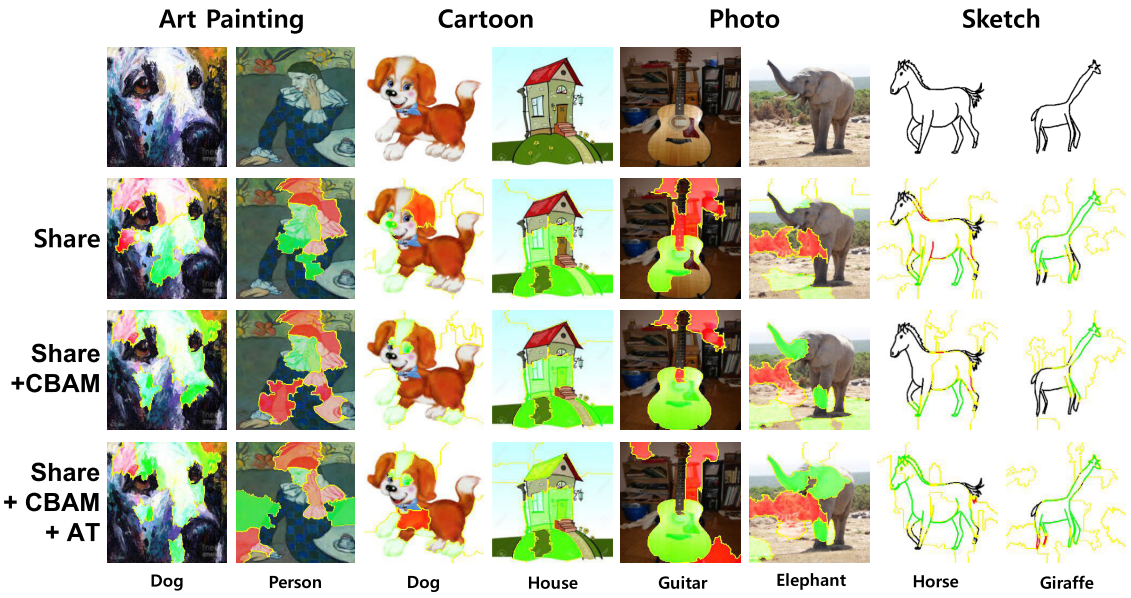


FIGURE 8. Selected LIME visualization results on PACS dataset. Positive and negative pixels are highlighted in green and red, respectively. The ground-truth label is shown on the bottom of images.

image of Cartoon, Share completely fails to find the target object regions, whereas the Share + CBAM + AT almost exactly highlights the important regions. In addition, we can also see that all the three strategies provide a similar level of visual explanation with respect to Photo, but regarding Sketch, Share + CBAM and Share + CBAM + AT give better results than Share.

Figure 8 shows the LIME visualization results. Similar to the Grad-CAM results, we see from the figure that our approaches allow LIME to find more appropriate superpixels for explaining instances. For instance, in the case of the guitar image of Photo, while Share fails to cover the entire body of the guitar, Share + CBAM and Share + CBAM + AT highlight the region almost accurately. Also, for the elephant image of Photo, the proposed approaches provide the appropriate superpixels for the nose and ear, distinct characteristics of an elephant, unlike Share. In addition, for the horse image of Sketch, it is seen that only Share + CBAM + AT has succeeded in capturing not only the legs but also the head of the horse.

TABLE 5. Test accuracy (%) on the multi-domain aerial/satellite dataset with respect to different adapter types for multi-domain learning.

		AID	NWPU	PatternNet	UC Merced	**Average**
Serial [44]	Acc.	62.72 ± 0.94	60.85 ± 1.02	89.82 ± 0.53	46.42 ± 2.04	**64.95 ± 0.90**
	F1-score	60.87 ± 0.87	61.21 ± 1.07	90.06 ± 0.50	43.23 ± 2.02	**63.84 ± 0.88**
Serial Residual [12], [43], [44]	Acc.	80.72 ± 1.41	82.81 ± 1.53	97.92 ± 0.41	73.83 ± 3.04	**83.82 ± 1.39**
	F1-score	79.97 ± 1.54	83.20 ± 1.60	97.94 ± 0.39	71.93 ± 3.57	**83.26 ± 1.54**
Parallel Residual [42]	Acc.	81.83 ± 1.50	84.31 ± 0.88	98.38 ± 0.22	74.75 ± 1.71	**84.82 ± 0.76**
	F1-score	81.26 ± 1.59	84.65 ± 0.88	98.40 ± 0.20	73.47 ± 1.85	**84.45 ± 0.83**
Proposed	Acc.	80.83 ± 0.98	83.65 ± 0.78	98.20 ± 0.25	74.00 ± 3.05	**84.17 ± 0.79**
	F1-score	79.98 ± 1.08	84.14 ± 0.74	98.21 ± 0.23	72.77 ± 3.22	**83.77 ± 0.83**

TABLE 6. Test accuracy (%) on the multi-domain aerial/satellite dataset with respect to different training methods.

		AID	NWPU	PatternNet	UC Merced	**Average**
Share	Acc.	78.87 ± 0.73	82.05 ± 0.45	97.82 ± 0.40	71.58 ± 1.48	**82.58 ± 0.39**
	F1-score	78.13 ± 0.77	82.71 ± 0.45	97.84 ± 0.40	70.42 ± 1.78	**82.27 ± 0.45**
Share + CBAM	Acc.	80.83 ± 0.98	83.65 ± 0.78	98.20 ± 0.25	74.00 ± 3.05	**84.17 ± 0.79**
	F1-score	79.98 ± 1.08	84.14 ± 0.74	98.21 ± 0.23	72.77 ± 3.22	**83.77 ± 0.83**
Share + CBAM + AT	Acc.	81.19 ± 1.30	84.38 ± 0.60	98.35 ± 0.30	76.83 ± 1.70	**85.19 ± 0.71**
	F1-score	80.49 ± 1.31	84.88 ± 0.69	98.38 ± 0.29	76.02 ± 1.91	**84.94 ± 0.80**

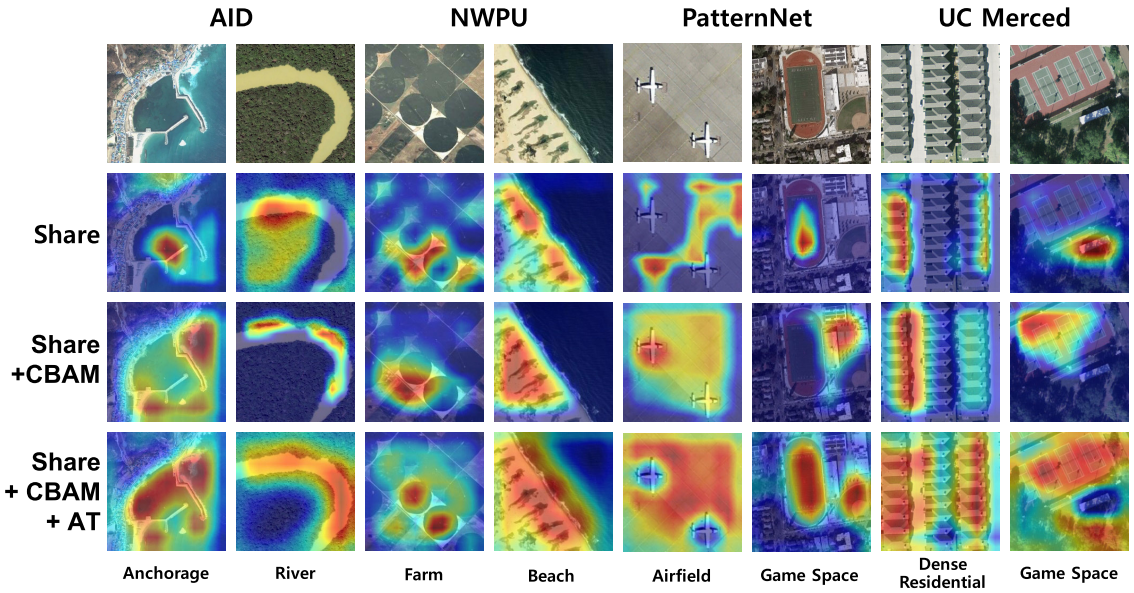


FIGURE 9. Selected Grad-CAM visualization results on the multi-domain aerial/satellite dataset. The Grad-CAM visualization is computed for the output of the last convolutional layer. The ground-truth label is shown on the bottom of images.

C. MULTI-DOMAIN AERIAL/SATELLITE DATASET

a: SETUP

For the experiments, we constructed a multi-domain aerial/satellite dataset with common class samples among AID [8] (600 × 600 pixel images with 0.5 ∼ 8m resolution), NWPU [7] (256 × 256 pixel images with 0.2 ∼ 30m resolution), PatternNet [9] (256 × 256 pixel images with ∼ 0.8m resolution), and UC Merced [6] (256 × 256 pixel images with 0.3m resolution), as depicted in Table 4. In the experiments, we resized the images to 225 × 225 pixel images. For the base network model, we used CaffeNet similarly as on the PACS dataset. For training, we used pure SGD as an optimization method to minimize the cross-entropy loss, with a weight decay of 0.0005. The trainings were conducted for 2000 iterations with a minibatch size of 128. The initial rate of η was set to 0.01; we dropped the learning rate by 0.1 at 80% of the total training iterations.

b: RESULTS

Table 5 depicts test accuracy and F1-score on the considered multi-domain aerial/satellite dataset with respect to domain adapter types. Unlike on the PACS dataset, where the serial

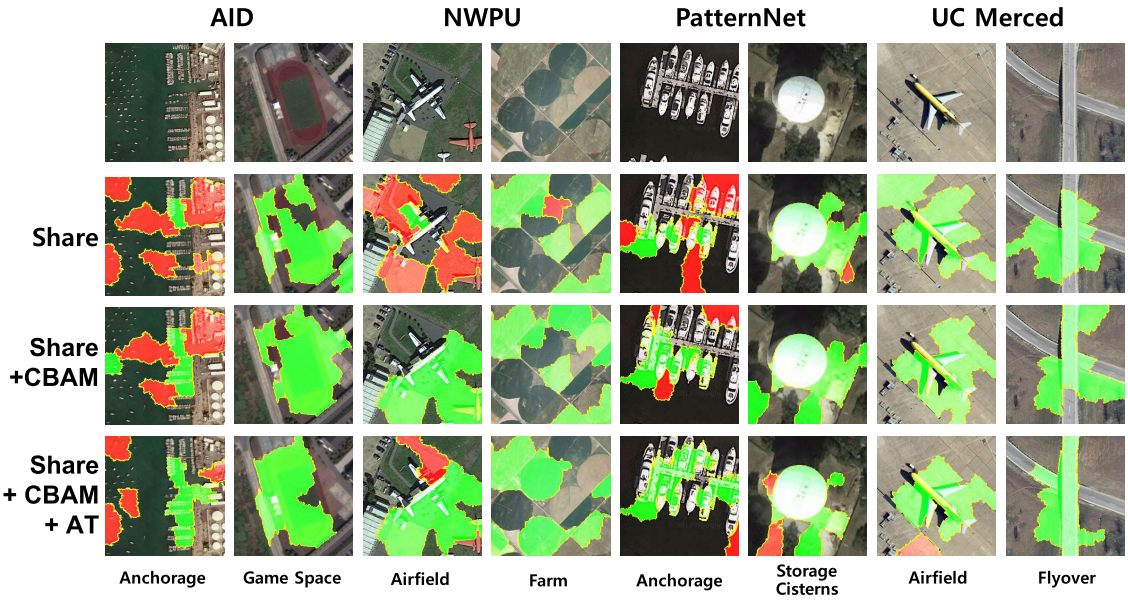


FIGURE 10. Selected LIME visualization results on the multi-domain aerial/satellite dataset. Positive and negative pixels are highlighted in green and red, respectively. The ground-truth label is shown on the bottom of images.

residual and the proposed adapter architectures provide better results than the others, we observe from the table that the parallel residual adapter yields the best performance among all the four types; it is seen that the proposed one is placed in the second rank. In addition, similar as in Table 2, the serial adapter shows the performance drop of ∼20% compared to the other architectures.

Next, Table 6 describes the task performance with respect to training methods on the multi-domain aerial/satellite dataset. From the table, it is seen that Share + CBAM + AT achieves the best test accuracy and F1-score among the three compared strategies. Unlike on the PACS dataset, we identified that the alternating training strategy gives the performance gain regardless of the learning rate; the table shows the results under the initial learning rate of 0.01, as stated in the experimental setup. In the case of Share + CBAM, we see that it outperforms Share also on this dataset, even without the alternating training; here we can observe the performance gain of ∼1.5% in both the evaluation metrics.

Figure 9 shows the Grad-CAM visualization results. Similar as on the PACS dataset, we see from the figure that the proposed method focuses better on the target object regions; in some cases, it provides almost perfect visual explanation (e.g., anchorage and river of AID, beach of NWPU, and game space of PatternNet). On the other hand, we have found that in rare cases, the Share + CBAM + AT also induces reverse attention, as shown in the airfield image of PatternNet; that is, for the image the region is highlighted except only the target object. However, even if the focused regions become reverse, the degree of explainability that human beings accept as color could be similar with the correct attention cases; the improvement in reverse attention also might be helpful to train classifiers, which would result in higher classification performance.

Figure 10 shows the LIME visualization results. From the figure, we observe that the target object region is well captured in the order of Share + CBAM + AT, Share + CBAM, and Share; this is seen clearly, particularly for the anchorage images. For the other cases, it is seen that the three strategies show the similar explanation performance; but, for the airfield image of NWPU, Share + CBAM has succeeded in highlighting the red airplane at the bottom right, while the others could not.

VI. CONCLUSION

In this paper, we proposed a multi-domain learning method for satellite image analytics, based on attention-based domain adapters. For the proposed method, we first introduced the architecture of the attention-based adapters, which improve channel and spatial attention for input images. The adapter modules are trained jointly with the backbone network; to do so, we also presented an alternating training strategy to effectively separate domain-invariant and -specific features into the backbone and the adapters, respectively. Finally, we exploited Grad-CAM to provide visual explanation on the proposed network architecture; in the evaluation of the proposed method, LIME was also considered for another explanation approaches. By the experiments with two multi-domain datasets, we demonstrated that the proposed method can not only improve task performance but also provide better visual explanation results. In this regard, we now interest in the related distributed learning settings where communication between workers is restricted (e.g., [61]); our future work lies in improving the alternating training strategy to fit well into the distributed environments.

APPENDIX A
THE PROOF OF LEMMA 1

$$\begin{aligned}
&\mathbb{E}\Big\|\nabla_{\mathbf{w}}\mathcal{L}_k(\mathbf{w}_k^{(t-1)}) - \frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)})\Big\|^2 \\
&= \mathbb{E}\Big\|\frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}_k(\mathbf{w}_k^{(t-1)}) - \frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)})\Big\|^2 \\
&= \frac{1}{K^2}\mathbb{E}\Big\|\sum_{i=1}^{K}\Big(\nabla_{\mathbf{w}}\mathcal{L}_k(\mathbf{w}_k^{(t-1)}) - \nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)})\Big)\Big\|^2 \\
&\le \frac{1}{K}\mathbb{E}\Big[\sum_{i=1}^{K}\|\nabla_{\mathbf{w}}\mathcal{L}_k(\mathbf{w}_k^{(t-1)}) - \nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)})\|^2\Big] \\
&\le \frac{1}{K}\mathbb{E}\Big[\sum_{k=1}^{K}\|\nabla_{\mathbf{w}}\mathcal{L}_k(\mathbf{w}_k^{(t-1)}) - \nabla_{\mathbf{w}}\mathcal{L}_k(\mathbf{w}_i^{(t-1)})\|^2\Big] \\
&\le \frac{1}{K}\mathbb{E}\Big[\sum_{i=1}^{K}\Big(2\|\nabla_{\mathbf{w}}\mathcal{L}_k(\mathbf{w}_k^{(t-1)})\|^2 + 2\|\nabla_{\mathbf{w}}\mathcal{L}_k(\mathbf{w}_i^{(t-1)})\|^2\Big)\Big] \\
&\le \frac{4(K-1)G^2}{K}, \qquad (17)
\end{aligned}$$

where the first inequality is obtained from $\|\sum_{i=1}^{K}\phi_i\|^2 \le K\sum_{i=1}^{K}\|\phi_i\|^2, \forall\phi_i \in \mathbb{R}^d$, the second and the last inequality comes from Assumption 2, and the third inequality follows from $\|\phi_1+\phi_2\|^2 \le 2\|\phi_1\|^2 + 2\|\phi_2\|^2$.

APPENDIX B
THE PROOF OF THEOREM 1

By Assumption 1, we have

$$\begin{aligned}
\mathbb{E}[\mathcal{L}_k(\mathbf{w}_k^{(t)})] \le\ & \mathbb{E}[\mathcal{L}_k(\mathbf{w}_k^{(t-1)})] \\
&+\mathbb{E}\big\langle\nabla\mathcal{L}_k(\mathbf{w}_k^{(t-1)}), \mathbf{w}_k^{(t)} - \mathbf{w}_k^{(t-1)}\big\rangle \\
&+\frac{\beta}{2}\mathbb{E}\|\mathbf{w}_k^{(t)} - \mathbf{w}_k^{(t-1)}\|^2. \qquad (18)
\end{aligned}$$

Then, letting $\mathbf{g}_k^{(t)} = \nabla_{\mathbf{m}_k}\mathcal{L}_k\Big(\mathbf{w}_k^{(t-1)} - \frac{\eta}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)})\Big)$, we get

$$\begin{aligned}
&\mathbb{E}\big\langle\nabla\mathcal{L}_k(\mathbf{w}_k^{(t-1)}), \mathbf{w}_k^{(t)} - \mathbf{w}_k^{(t-1)}\big\rangle \\
&= -\eta\mathbb{E}\Big\langle\nabla\mathcal{L}_k(\mathbf{w}_k^{(t-1)}), \frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)};\xi^{(t)}) \\
&\quad +\nabla_{\mathbf{m}_k}\mathcal{L}_k\Big(\mathbf{w}_k^{(t-1)} - \frac{\eta}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)};\xi^{(t)});\zeta_k^{(t)}\Big)\Big\rangle \\
&= -\eta\mathbb{E}\Big\langle\nabla\mathcal{L}_k(\mathbf{w}_k^{(t-1)}), \frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)}) + \mathbf{g}_k^{(t)}\Big\rangle \\
&= -\frac{\eta}{2}\Big(\mathbb{E}\|\nabla\mathcal{L}_k(\mathbf{w}_k^{(t-1)})\|^2 + \mathbb{E}\Big\|\frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)}) + \mathbf{g}_k^{(t)}\Big\|^2 \\
&\quad -\mathbb{E}\Big\|\nabla\mathcal{L}_k(\mathbf{w}_k^{(t-1)}) - \frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)}) + \mathbf{g}_k^{(t)}\Big\|^2\Big) \\
&\le -\frac{\eta}{2}\Big(\mathbb{E}\|\nabla\mathcal{L}_k(\mathbf{w}_k^{(t-1)})\|^2 + \mathbb{E}\Big\|\frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)}) + \mathbf{g}_k^{(t)}\Big\|^2 \\
&\quad -2\mathbb{E}\Big\|\nabla\mathcal{L}_k(\mathbf{w}_k^{(t-1)}) - \frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)})\Big\|^2 \\
&\quad -2\mathbb{E}\Big\|\mathbf{g}_k^{(t)}\Big\|^2\Big) \\
&\le -\frac{\eta}{2}\Big(\mathbb{E}\|\nabla\mathcal{L}_k(\mathbf{w}_k^{(t-1)})\|^2 + \mathbb{E}\Big\|\frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)}) + \mathbf{g}_k^{(t)}\Big\|^2 \\
&\quad -2\mathbb{E}\Big\|\nabla_{\mathbf{w}}\mathcal{L}_k(\mathbf{w}_k^{(t-1)}) - \frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)}) \\
&\quad +\nabla_{\mathbf{m}_k}\mathcal{L}_k(\mathbf{w}_k^{(t-1)})\Big\|^2 - 2R_k^2\Big) \\
&\le -\frac{\eta}{2}\Big(\mathbb{E}\|\nabla\mathcal{L}_k(\mathbf{w}_k^{(t-1)})\|^2 + \mathbb{E}\Big\|\frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)}) + \mathbf{g}_k^{(t)}\Big\|^2 \\
&\quad -4\mathbb{E}\Big\|\nabla_{\mathbf{w}}\mathcal{L}_k(\mathbf{w}_k^{(t-1)}) - \frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)})\Big\|^2 \\
&\quad -4\mathbb{E}\Big\|\nabla_{\mathbf{m}_k}\mathcal{L}_k(\mathbf{w}_k^{(t-1)})\Big\|^2 - 2R_k^2\Big) \\
&\le -\frac{\eta}{2}\Big(\mathbb{E}\|\nabla\mathcal{L}_k(\mathbf{w}_k^{(t-1)})\|^2 + \mathbb{E}\Big\|\frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)}) + \mathbf{g}_k^{(t)}\Big\|^2 \\
&\quad -\frac{16(K-1)G^2}{K} - 4Q_k^2 - 2R_k^2,\Big), \qquad (19)
\end{aligned}$$

where the first equality is obtained from that $\xi^{(t)}$ and $\zeta_k^{(t)}$ are independent to $\xi^{(1)},\cdots,\xi^{(t-1)}$ and $\zeta_k^{(1)},\cdots,\zeta_k^{(t-1)}$, respectively, the second equality is obtained from $\langle\phi_1,\phi_2\rangle = \frac{1}{2}(\|\phi_1\|^2 + \|\phi_2\|^2 - \|\phi_1-\phi_2\|^2), \forall\phi_1,\phi_2 \in \mathbb{R}^d$, the first inequality follows from $\|\phi_1+\phi_2\|^2 \le 2\|\phi_1\|^2 + 2\|\phi_2\|^2$, the second and the last inequality comes from Assumption 3 and Lemma 1.

Also we get

$$\begin{aligned}
&\mathbb{E}\|\mathbf{w}_k^{(t)} - \mathbf{w}_k^{(t-1)}\|^2 \\
&= \eta^2\mathbb{E}\Big\|\frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)};\xi^{(t)}) + \nabla_{\mathbf{m}_k}\mathcal{L}_k\Big(\mathbf{w}_k^{(t-1)} \\
&\quad -\frac{\eta}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)};\xi^{(t)});\zeta_k^{(t)}\Big)\Big\|^2 \\
&\le \eta^2\Big(\mathbb{E}\Big\|\frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)}) + \mathbf{g}_k^{(t)}\Big\|^2 + \delta\Big). \qquad (20)
\end{aligned}$$

By combining inequalities (19) and (20) into (18), we have, for $0 < \eta \le \frac{1}{\beta}$,

$$\begin{aligned}
\mathbb{E}[\mathcal{L}_k(\mathbf{w}^{(t)})] \le\ & \mathbb{E}[\mathcal{L}_k(\mathbf{w}^{(t-1)})] - \frac{\eta}{2}\mathbb{E}\|\nabla\mathcal{L}_k(\mathbf{w}_k^{(t-1)})\|^2 \\
&-\frac{\eta-\beta\eta^2}{2}\mathbb{E}\Big\|\frac{1}{K}\sum_{i=1}^{K}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_i^{(t-1)}) + \mathbf{g}_k^{(t)}\Big\|^2
\end{aligned}$$

$$+\frac{\beta\delta\eta^2}{2}+\frac{8\eta(K-1)G^2}{K}+2\eta Q_k^2+\eta R_k^2$$
$$\leq \mathbb{E}[\mathcal{L}_k(\mathbf{w}^{(t-1)})]-\frac{\eta}{2}\mathbb{E}\|\nabla\mathcal{L}_k(\mathbf{w}_k^{(t-1)})\|^2$$
$$+\frac{\beta\delta\eta^2}{2}+\frac{8\eta(K-1)G^2}{K}+2\eta Q_k^2+\eta R_k^2. \tag{21}$$

By dividing the inequality (21) with $\frac{\eta}{2}$, we have

$$\mathbb{E}\|\nabla\mathcal{L}(\mathbf{w}_k^{(t-1)})\|^2 \leq \frac{2}{\eta}\Big(\mathbb{E}[\mathcal{L}(\mathbf{w}_k^{(t-1)})]-\mathbb{E}[\mathcal{L}(\mathbf{w}_k^{(t)})]\Big)$$
$$+\beta\delta\eta+\frac{16(K-1)G^2}{K}+4Q_k^2+2R_k^2. \tag{22}$$

By taking an average over $1 \leq t \leq T$, finally we have

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla\mathcal{L}(\mathbf{w}_k^{(t-1)})\|^2$$
$$\leq \frac{2}{\eta T}\Big(\mathbb{E}[\mathcal{L}(\mathbf{w}_k^{(0)})]-\mathbb{E}[\mathcal{L}(\mathbf{w}_k^{(T)})]\Big)$$
$$+\beta\delta\eta+\frac{16(K-1)G^2}{K}+4\,Q_k^2+2\,R_k^2$$
$$\leq \frac{2}{\eta T}\Big(\mathbb{E}[\mathcal{L}(\mathbf{w}_k^{(0)})]-\mathcal{L}^*\Big)$$
$$+\beta\delta\eta+\frac{16(K-1)G^2}{K}+4Q_k^2+2R_k^2. \tag{23}$$

REFERENCES

[1] S. Han, D. Ahn, H. Cha, J. Yang, S. Park, and M. Cha, "Lightweight and robust representation of economic scales from satellite imagery," in *Proc. AAAI*, 2020, pp. 428–436.

[2] W. Xue, X. Dai, and L. Liu, "Remote sensing scene classification based on multi-structure deep features fusion," *IEEE Access*, vol. 8, pp. 28746–28755, 2020.

[3] C. Yeh, A. Perez, A. Driscoll, G. Azzari, Z. Tang, D. Lobell, S. Ermon, and M. Burke, "Using publicly available satellite imagery and deep learning to understand economic well-being in Africa," *Nature Commun.*, vol. 11, no. 1, pp. 1–11, Dec. 2020.

[4] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. ICCV*, Oct. 2017, pp. 618–626.

[5] S. K. Arora. (2018). *Getting Started With SpaceNet*. [Online]. Available: https://medium.com/@sumit.arora/getting-started-with-aws-spacenet-and-spacenet-dataset-visualization-basics-7ddd2e5809a2

[6] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst. (ACM GIS)*, 2010, pp. 270–279.

[7] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.

[8] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.

[9] W. Zhou, S. Newsam, C. Li, and Z. Shao, "PatternNet: A benchmark dataset for performance evaluation of remote sensing image retrieval," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 197–209, Nov. 2018.

[10] M. A. Shafaey, M. A.-M. Salem, H. M. Ebeid, M. N. Al-Berry, and M. F. Tolba, "Comparison of CNNs for remote sensing scene classification," in *Proc. ICCES*, Dec. 2018, pp. 27–32.

[11] M. Al Rahhal, Y. Bazi, T. Abdullah, M. Mekhalfi, H. AlHichri, and M. Zuair, "Learning a multi-branch neural network from multiple sources for knowledge adaptation in remote sensing imagery," *Remote Sens.*, vol. 10, no. 12, p. 1890, Nov. 2018.

[12] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," in *Proc. NeurIPS*, 2017, pp. 1–11.

[13] H. Bilen and A. Vedaldi, "Universal representations: The missing link between faces, text, planktons, and cat breeds," 2017, *arXiv:1701.07275*. [Online]. Available: http://arxiv.org/abs/1701.07275

[14] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. ECCV*, 2018, pp. 3–19.

[15] S. Chaib, H. Liu, Y. Gu, and H. Yao, "Deep feature fusion for VHR remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4775–4784, Aug. 2017.

[16] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, May 2018.

[17] Q. Wang, S. Liu, J. Chanussot, and X. Li, "Scene classification with recurrent attention of VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1155–1167, Feb. 2019.

[18] V. S. F. Garnot, L. Landrieu, S. Giordano, and N. Chehata, "Satellite image time series classification with pixel-set encoders and temporal self-attention," in *Proc. CVPR*, Jun. 2020, pp. 12325–12334.

[19] D. Lunga, H. L. Yang, A. Reith, J. Weaver, J. Yuan, and B. Bhaduri, "Domain-adapted convolutional networks for satellite image classification: A large-scale interactive learning workflow," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 3, pp. 962–977, Mar. 2018.

[20] K. A. Islam, V. Hill, B. Schaeffer, R. Zimmerman, and J. Li, "Semi-supervised adversarial domain adaptation for seagrass detection in multispectral images," in *Proc. ICDM*, Nov. 2019, pp. 1120–1125.

[21] S. Ji, D. Wang, and M. Luo, "Generative adversarial network-based full-space domain adaptation for land cover classification from multiple-source remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, early access, Sep. 15, 2020, doi: 10.1109/TGRS.2020.3020804.

[22] O. Tasar, Y. Tarabalka, A. Giros, P. Alliez, and S. Clerc, "StandardGAN: Multi-source domain adaptation for semantic segmentation of very high resolution satellite images by data standardization," in *Proc. CVPRW*, Jun. 2020, pp. 192–193.

[23] L. Yan, B. Fan, H. Liu, C. Huo, S. Xiang, and C. Pan, "Triplet adversarial domain adaptation for pixel-level classification of VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3558–3573, May 2020.

[24] J. Yang, H. Chen, Y. Xu, Z. Shi, R. Luo, L. Xie, and R. Su, "Domain adaptation for degraded remote scene classification," in *Proc. ICARCV*, Dec. 2020, pp. 111–117.

[25] J. Zhang, J. Liu, B. Pan, and Z. Shi, "Domain adaptation based on correlation subspace dynamic distribution alignment for remote sensing image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 7920–7930, Nov. 2020.

[26] O. Tasar, A. Giros, Y. Tarabalka, P. Alliez, and S. Clerc, "DAugNet: Unsupervised, multisource, multitarget, and life-long domain adaptation for semantic segmentation of satellite images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1067–1081, Feb. 2021.

[27] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. CVPR*, Jun. 2016, pp. 2921–2929.

[28] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. ICLR*, 2014, pp. 1–14.

[29] B. Vasu and A. Savakis, "Resilience and plasticity of deep network interpretations for aerial imagery," *IEEE Access*, vol. 8, pp. 127491–127506, 2020.

[30] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. WACV*, Mar. 2018, pp. 839–847.

[31] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?' Explaining the predictions of any classifier," in *Proc. KDD*, 2016, pp. 1135–1144.

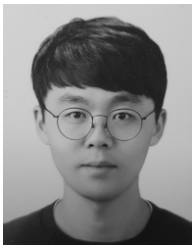
[32] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Proc. NeurIPS*, 2017, pp. 4768–4777.

[33] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. ICML*, 2017, pp. 3319–3328.

[34] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Proc. NeurIPS*, 2016, pp. 1–9.

[35] Y. Liu, X. Tian, Y. Li, Z. Xiong, and F. Wu, "Compact feature learning for multi-domain image classification," in *Proc. CVPR*, Jun. 2019, pp. 7193–7201.

[36] A.-A. Liu, N. Xu, W.-Z. Nie, Y.-T. Su, and Y.-D. Zhang, "Multi-domain and multi-task learning for human action recognition," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 853–867, Feb. 2019.
[37] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456.
[38] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016, *arXiv:1607.08022*. [Online]. Available: http://arxiv.org/abs/1607.08022
[39] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*. [Online]. Available: http://arxiv.org/abs/1607.06450
[40] R. Berriel, S. Lathuillére, M. Nabi, T. Klein, T. Oliveira-Santos, N. Sebe, and E. Ricci, "Budget-aware adapters for multi-domain learning," in *Proc. ICCV*, Oct. 2019, pp. 382–391.
[41] A. Rosenfeld and J. K. Tsotsos, "Incremental learning through deep adaptation," 2017, *arXiv:1705.04228*. [Online]. Available: http://arxiv.org/abs/1705.04228
[42] S.-A. Rebuffi, A. Vedaldi, and H. Bilen, "Efficient parametrization of multi-domain deep neural networks," in *Proc. CVPR*, Jun. 2018, pp. 8119–8127.
[43] Y. Li and N. Vasconcelos, "Efficient multi-domain learning by covariance normalization," in *Proc. CVPR*, Jun. 2019, pp. 5424–5433.
[44] H. Zhao, H. Zeng, X. Qin, Y. Fu, H. Wang, B. Omar, and X. Li, "What and where: Learn to plug adapters via NAS for multi-domain learning," 2020, *arXiv:2007.12415*. [Online]. Available: http://arxiv.org/abs/2007.12415
[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.
[46] A. Senhaji, J. Raitoharju, M. Gabbouj, and A. Iosifidis, "Not all domains are equally complex: Adaptive multi-domain learning," 2020, *arXiv:2003.11504*. [Online]. Available: http://arxiv.org/abs/2003.11504
[47] J. Xiao, S. Gu, and L. Zhang, "Multi-domain learning for accurate and few-shot color constancy," in *Proc. CVPR*, Jun. 2020, pp. 3258–3267.
[48] L. Deecke, T. Hospedales, and H. Bilen, "Latent domain learning with dynamic residual adapters," 2020, *arXiv:2006.00996*. [Online]. Available: http://arxiv.org/abs/2006.00996
[49] L. Yang, A. S. Rakin, and D. Fan, "DA^2: Deep attention adapter for memory-efficient on-device multi-domain learning," 2020, *arXiv:2012.01362*. [Online]. Available: https://arxiv.org/abs/2012.01362
[50] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. CVPR*, Jun. 2020, pp. 1–12.
[51] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, Jun. 2009, pp. 248–255.
[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NeurIPS*, 2012, pp. 1097–1105.
[53] J. C. Bezdek and R. J. Hathaway, "Some notes on alternating optimization," in *Proc. AFSS Int. Conf. Fuzzy Syst.*, in Lecture Notes in Computer Science, vol. 2275, 2002, pp. 288–300.
[54] S. Ghadimi and G. Lan, "Stochastic first- and zeroth-order methods for nonconvex stochastic programming," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2341–2368, Jan. 2013.
[55] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Proc. NeurIPS*, 2017, pp. 1–33.
[56] P. Jiang and G. Agrawal, "A linear speedup analysis of distributed deep learning with sparse and quantized communication," in *Proc. NeurIPS*, 2018, pp. 2530–2541.
[57] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *Proc. AAAI*, 2019, pp. 5693–5700.
[58] H. Yu, R. Jin, and S. Yang, "On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization," in *Proc. ICML*, 2019, pp. 7184–7193.
[59] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proc. ICCV*, Oct. 2017, pp. 5542–5550.
[60] *CaffeNet*. Accessed: Dec. 2015. [Online]. Available: https://github.com/BVLC/caffe/tree/master/models/bvlc_reference_caffenet
[61] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, 2017, pp. 1273–1282.



HEEJAE KIM received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2011, 2013, and 2021, respectively. He is currently a Postdoctoral Fellow with the Prof. Youn's Group, KAIST. His current research interests include learning with multi-domain data, explanability on deep models, and efficient distributed training of deep networks.

KYUNGCHAE LEE received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2018, and the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2019, where he is currently pursuing the Ph.D. degree with the Department of Electrical Engineering. His current research interests include deep learning based system control, reinforcement learning, and deep learning acceleration platform.

CHANGHA LEE received the B.S. degree in electronic engineering from Hanyang University, Seoul, South Korea, in 2018, and the M.S. degree in electronic engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2020, where he is currently pursuing the Ph.D. degree. His current research interests include deep learning acceleration platform, online learning, and energy prediction service platform.

SANGHYUN HWANG received the B.S. and M.S. degrees in electronic engineering from Kyungpook National University, Daegu, South Korea, in 1990 and 1992, respectively, and the Ph.D. degree in electronic engineering from the Kumoh National Institute of Technology (KIT), Gumi, South Korea, in 2016. After his M.S. degree, he developed mission computers and various intelligent computer softwares for aircraft with the Agency for Defense Development (ADD). His current research interest includes autonomous unmanned systems.

CHAN-HYUN YOUN (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electronics engineering from Kyungpook National University, Daegu, South Korea, in 1981 and 1985, respectively, and the Ph.D. degree in electrical and communications engineering from Tohoku University, Japan, in 1994. Before joining the University, from 1986 to 1997, he was the Leader of KT Telecommunications Network Research Laboratories, High-Speed Networking Team, where he had been involved in the research and developments of centralized switching maintenance system, high-speed networking, and ATM network. Since 2009, he has been a Professor with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. He was an Associate Vice-President of office of planning and budgets, KAIST, from 2013 to 2017. He is the Director of the Grid Middleware Research Center and the XAI Acceleration Technology Research Center, KAIST, where he is developing core technologies that are in the areas of high performance computing, edge-cloud computing, AI acceleration system and others. He wrote a book on *Cloud Broker and Cloudlet for Workflow Scheduling* (Springer, 2017). He served as a TPC Member for many international conferences. He was the General Chair of the 6th EAI International Conference on Cloud Computing (Cloud Comp 2015), KAIST, in 2015 and a Guest Editor of IEEE Wireless Communications in 2016.