Learning to Adapt to Label-Scarce Image Domain via Angular Distance-Based Feature Alignment

YOONHYUNG KIM
Electronics and Telecommunications Research Institute (ETRI), Daejeon 34129, South Korea
e-mail: yhkim1127@etri.re.kr
This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) Grant through the Korea Government (MSIT), Development of Semi-Supervised Learning Language Intelligence Technology and Korean Tutoring Service for Foreigners, under Grant 2019-0-00004.

ABSTRACT Most recent domain adaptation (DA) methods deal with unsupervised setup, which requires numerous target images for training. However, constructing a large-scale image set of the target domain is occasionally much harder than preparing a smaller number of image and label pairs. To cope with the problem, a great attention is recently paid to supervised domain adaptation (SDA), which takes an extremely small amount of labeled target images for training (e.g., at most three examples per category). In the SDA setup, adapting deep networks towards target domain is very challenging due to the lack of target data, and we tackle this problem as follows. Given labeled images from source and target domains, we first extract deep features and project them to hyper-spherical space via l2-normalization. Afterwards, an additive angular margin loss is embedded so that deep features of both domains are compactly grouped on the basis of shared class prototypes. To further relieve domain discrepancy, a pairwise spherical feature alignment loss is incorporated. All of our loss functions are defined in the hyper-spherical space, and the advantage of each ingredient is analyzed in the literature. Comparative evaluation results demonstrate that the proposed approach is superior to existing SDA methods, achieving 60.7% (1-shot) and 64.4% (3-shot) average accuracies for the DomainNet benchmark dataset using the ResNet-34 backbone. In addition, by applying a semi-supervised learning scheme to a network initialized by our SDA method, we achieve the state-of-the-art performance on semi-supervised domain adaptation (SSDA) as well.

INDEX TERMS Supervised domain adaptation, semi-supervised domain adaptation, transfer learning, deep learning.

I. INTRODUCTION
Deep convolutional neural networks (DCNNs) have shown promising results on various visual recognition tasks. However, due to the strong dependency on training data, the performance of DCNNs is usually degraded when tested to a shifted domain [1]. Preparing large-scale training resources for every target domain could be a simple solution, but it is time-consuming and costly. To resolve this problem, domain adaptation (DA) has been actively studied in recent years [1], [2].

The goal of DA is to overcome the deficiency of training resources in a target domain by leveraging knowledge from resource-sufficient a source domain. Depending on the type of target data available for training, DA schemes can be roughly categorized into unsupervised, semi-supervised, and supervised approaches, as indicated in Table 1. Most of previous studies on DA are devoted to unsupervised domain adaptation (UDA) [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], which assumes that a large number of unlabeled target images are available for training. The assumption behind UDA is that the expense of assigning labels is much larger than that of collecting images from a target domain. Similarly, semi-supervised domain adaptation (SSDA) [14], [15], [16], [17] assumes that a few amount of labeled target images are given for training, while leaving the rest of target images as unlabeled. However, mining a large number of images from a target domain is occasionally very hard, allowing only a
few amount of image and label pairs for training (e.g., one or three examples per category). To cope with those problems, supervised domain adaptation (SDA) is addressed by several previous works [18], [19], [20].

Previous SDA methods are focused on matching feature representations by using semantic priors. In [18] and [20], pairwise distance and similarity metrics are introduced to facilitate feature alignment based on the class equivalency of feature pairs. This pairwise alignment scheme is rebuilt as an adversarial learning model in [19]. Although these methods show promising results on SDA, the principle for feature alignment is limited to the pairwise comparison scheme. Based on this observation, we aim to develop a new approach for SDA beyond the existing methodology.

In this paper, we propose a new SDA framework for image classification. Given labeled images from both domains, we aim to learn highly discriminative representations by encouraging between-class variations and penalizing cross-domain variations of features in a spherical space. Firstly, we extract deep features and normalize them to unit vectors via $l^2$-normalization. Geometrically, this procedure is same as projecting a feature vector from the Euclidean space to a hyper-spherical (or spherical for simplicity) space. In this setup, a feature is solely represented by its direction, and a class prediction score is computed based on the angular distance from normalized class prototypes. An illustration that describes the difference between Euclidean and spherical feature spaces is given in Fig. 1(a) and (b). Compared to the Euclidean space, features in the spherical space have relatively lower degree of freedom. This characteristic of spherical feature space relieves the risk of overfitting [21], and implicitly enforces matching feature distributions of both domains, as illustrated in Fig. 1(b).

Secondly, inspired by [22], an additive angular margin loss is embedded as a classification loss function. Compared to the vanilla softmax loss, the additive angular margin loss encourages the smaller disparity between a feature and its corresponding class prototype by inserting a non-negative margin penalty to the prediction. In our method, the additive scheme is extended to SDA by applying an angular margin for each domain. Thirdly, to further alleviate the domain discrepancy, we propose a pairwise spherical feature alignment loss, which operates in the hyper-spherical feature space. Compared to training with a vanilla softmax loss function in the spherical space (Fig. 1(b)), our proposed loss function can generate more compact feature distributions as described in Fig. 1(c). Our method can be generically applied for various deep networks from CNNs to recent vision Transformer (ViT) [23] without requiring any domain classifier. Extensive experimental results show that the proposed SDA method is superior to the existing methods on various datasets. Furthermore, our SDA method can be extended to semi-supervised domain adaptation (SSDA) by applying a semi-supervised learning (SSL) scheme to a network initialized by our SDA method. Experimental results demonstrate that this extended version of our method surpasses recent state-of-the-art methods which are specifically designed for SSDA.

The contributions of this paper can be summarized as follows:

- We propose a novel SDA scheme which conducts feature alignment in a spherical space. In contrast to the Euclidean space in Fig. 1(a), the spherical space can reduce the risk of overfitting by enforcing feature alignment as illustrated in Fig. 1(b).
- For SDA, we propose a domain-wise additive margin loss based on the spherical feature embedding model. In addition, we propose a pairwise spherical feature alignment loss, which aligns feature distributions based on the class equivalency of feature pairs. By incorporating the proposed losses, the spherical features can be compactly grouped on the basis of class prototypes, as depicted in Fig. 1(c).
- For SSDA, we show that a deep network pre-trained by our SDA scheme can be further extended to SSDA by applying a semi-supervised learning scheme to target labeled and unlabeled data.
- By means of extensive experiments, we demonstrate that state-of-the-art results are achieved both on SDA and SSDA.

II. RELATED WORK

A. DOMAIN ADAPTATION

Based on the configuration of target data given for training, DA methods for image classification can be categorized into unsupervised, semi-supervised, and supervised schemes. A comparison on the three schemes is shown in Table 1. Beginning with a pioneering work [3], various UDA methods are proposed [4], [5], [6], [7], [8], [9], [10], [11], [12], [13] for learning domain-invariant feature representations. In addition, based on the observation that the class sets of
source and target domains could be different each other, several studies on partial [24], [25], open-set [26], [27], [28], universal [29], [30], and lifelong [31], [32] DA schemes are proposed. These methods, which are built upon the unsupervised setup, assume that a sufficient amount of target images are available for training whereas they are fully unlabeled. With a practical perspective, several SSDA schemes [14], [15], [16], [17] are recently proposed for the semi-supervised setup that a small portion of target images are annotated. A common assumption of UDA and SSDA is that relatively large amount of target images are prepared for training.

To cope with cases that only a few target images are available with labels, several SDA methods [18], [19], [20] are proposed. In [18], a pairwise loss function, named the classification and contrastive semantic alignment (CCSA) loss, is proposed to match features across domains based on the class equivalency of image pairs. This pairwise alignment scheme is reconstituted by means of the adversarial learning approach (FADA [19]) and the stochastic neighborhood embedding method via a modified Hausdorffian distance (d-SNE [20]). Apart from the pairwise loss functions, the previous SDA methods commonly adopt the vanilla softmax loss, which is an ordinary classification loss, equally for each domain. Unlike these methods, we introduce a new loss function, which is motivated by a margin-based metric learning scheme [22], and apply the loss in a domain-adaptive manner.

### B. FEW-SHOt LEARNING

The goal of few-shot learning [33], [34], [35], [36] is to learn discriminative feature representations for novel classes with a few training samples. Our work is partially related to few-shot learning since a few training samples are given for the target task. However, the problem statements of the two topics are different each other as follows: few-shot learning aims to adapt to novel classes within a fixed domain, whereas the goal of SDA is to adapt to a novel domain with a fixed set of classes. Chen et al. [21] introduce a baseline feature embedding scheme, and empirically demonstrate that the l2-normalization-based embedding scheme can be generically applied to various few-shot setups. This approach is adopted for feature alignment in several DA methods [10], [14], [16], yet the l2-normalization is applied to the feature space only. In our work, this embedding scheme is further extended to fully hyper-spherical space by applying l2-normalization to class prototypes (i.e., classifiers) as well. In this setup, we propose three loss functions for few-shot DA, as addressed in Section IV.

### C. MARGIN-BASED DEEP METRIC LEARNING

A metric is a function that defines a distance between each pair of elements in a set, and the goal of metric learning is to learn a task-specific metric in a data-driven manner. Recently, by letting the deep feature space as a metric subspace, deep metric learning has been explored for various tasks such as image retrieval [37], [38], person re-identification [39], [40], and face verification [22], [41], [42]. Learning a discriminative metric subspace is very challenging if image pairs are similarly structured as in face verification. To address this issue, margin-based methods are proposed, which insert a non-negative margin that penalizes the distance between predictions and their class centres. Inserting a margin makes deep features of a same class be compactly grouped by pushing them farther from decision boundaries. Several examples on margin-based methods for face verification are SphereFace [41], CosFace [42], and ArcFace [22]. In our work, we extend ArcFace to domain adaptation to reduce the intra-class and cross-domain variations and to enlarge the between-class variations.

### III. BACKGROUND: SPHERICAL FEATURE SPACE

An image classification network consists of a feature extractor $F(\cdot; \psi_F)$ and a classifier $C(\cdot; \psi_C)$, where $\psi_F$ and $\psi_C$ represent their corresponding weights. For an $i$-th input image $x_i$, its feature vector is embedded as $f_i = F(x_i; \psi_F) \in \mathbb{R}^d$, where $d$ indicates the feature dimension. Following [21], the classifier $C$ is modeled as a task-specific fully connected layer $W \in \mathbb{R}^{d \times K}$. Here, $K$ indicates the number of image categories, and the bias parameters of the classifier are fixed to zero for simplicity [41]. In this setup, the matrix $W$ can be understood as a set of class prototypes $W = [W_1, \ldots, W_K]$, where $W_k \in \mathbb{R}^d$ represents a weight vector for the $k$-th class prototype. In the Euclidean space, a classification logit vector $z_i$ is obtained as $z_i = C(f_i; \psi_C) = W^T f_i$, where $z_i = [z_{i,1}, \ldots, z_{i,K}]$.

To apply a spherical feature space, we apply the l2-normalization to both features and weight vectors as follows:

$$\bar{f}_i = \frac{f_i}{\|f_i\|_2}, \quad (1)$$

$$\bar{W}_k = \frac{W_k}{\|W_k\|_2}. \quad (2)$$

Given l2-normalized feature $\bar{f}_i$ and weight vector $\bar{W}_k$, a classification logit $\bar{z}_{i,k}$ is obtained as follows:

$$\bar{z}_{i,k} = \bar{W}_k^T \bar{f}_i = \frac{\|W_k\|_2}{\|W_k\|_2} \|\bar{f}_i\|_2 \cos \theta_{k,i,k} = \cos \theta_{k,i,k}, \quad (3)$$

where $\theta_{k,i,k} = \arccos(\bar{z}_{i,k})$ indicates the angle between the normalized feature of the $i$-th image and the $k$-th normalized
class prototype. Thus, class prediction in the spherical feature space solely depends on the angle between $\mathbf{f}_i$ and $\mathbf{W}_k$.

IV. PROPOSED METHOD

A. PROPOSED SDA SCHEME

1) PROBLEM FORMULATION

In SDA, we are given a set of labeled source images $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{N_s}$ and a set of labeled target images $\mathcal{D}_t = \{(\mathbf{x}_j^t, y_j^t)\}_{j=1}^{N_t}$ for training. The amount of the target data $N_t$ is very small (e.g., at most three examples per category), and is usually much smaller than that of source data (i.e., $N_t \ll N_s$).

2) ADDITIVE ANGULAR MARGIN LOSS

In the hyper-spherical feature space, a vanilla softmax function on the source data ($\mathcal{D}_s$) is represented as follows:

$$S(y_i^s|\mathbf{x}_i^s) = \frac{e^{r \cos(\theta_{y_i^s, \mathbf{x}_i^s})}}{\sum_{k=1}^{K} e^{r \cos(\theta_{y_i^s, \mathbf{x}_k^s})}}\quad e^{r \cos(\theta_{y_i^s, \mathbf{x}_k^s})} = \frac{e^{r \cos(\theta_{y_i^s, \mathbf{x}_k^s})}}{\sum_{k=1}^{K} e^{r \cos(\theta_{y_i^s, \mathbf{x}_k^s})}}$$

where $\cos(\theta_{y_i^s, \mathbf{x}_k^s})$ indicates a classification logit which represents a similarity between the feature of the $i$-th source sample ($\mathbf{x}_i^s$) and the $k$-th class prototype. In Eq. (4), $r$ denotes a scaling parameter that controls the magnitude of radius. This widely-used classification loss, however, does not explicitly regularize features to have higher intra-class similarity and between-class variations. Inspired by ArcFace [22], we insert an additive margin to enforce features to be compactly grouped on the basis of their corresponding class prototypes. In our work, source and target data are separately embedded with domain-dependent margin parameters as follows:

$$S_{arc}^s(y_j^s|\mathbf{x}_j^s) = \frac{e^{r \cos(\theta_{y_j^s, \mathbf{x}_j^s} + m_s)}}{e^{r \cos(\theta_{y_j^s, \mathbf{x}_j^s})} + \sum_{k=1, k \neq y_j^s}^{K} e^{r \cos(\theta_{y_j^s, \mathbf{x}_k^s})}},$$

$$S_{arc}^t(y_j^t|\mathbf{x}_j^t) = \frac{e^{r \cos(\theta_{y_j^t, \mathbf{x}_j^t} + m_t)}}{e^{r \cos(\theta_{y_j^t, \mathbf{x}_j^t})} + \sum_{k=1, k \neq y_j^t}^{K} e^{r \cos(\theta_{y_j^t, \mathbf{x}_k^t})}},$$

where $m_s$ and $m_t$ are margin parameters for source and target domains, respectively. The margin parameters are non-negative constants that induce features get closer to their corresponding class prototypes. In our work, we set $m_s < m_t$ to impose stronger regularization on the target domain. Discussions on the magnitude of the margin parameters are given in Section V-C1. By adopting Eq. (5) and (6), cross-entropy losses for each domain are given as follows:

$$L_s = \mathbb{E}_{(\mathbf{x}_i^s, y_i^s) \in \mathcal{D}_s}[-\log(S_{arc}^s(y_i^s|\mathbf{x}_i^s))],$$

$$L_t = \mathbb{E}_{(\mathbf{x}_j^t, y_j^t) \in \mathcal{D}_t}[-\log(S_{arc}^t(y_j^t|\mathbf{x}_j^t))].$$

FIGURE 2. Overall framework of the proposed SDA scheme. For a training iteration, mini-batch images are sampled from each domain, and they are embedded to spherical features ($\mathbf{f}_i^s$, $\mathbf{f}_i^t$). Afterwards, for each domain, an additive angular margin loss is embedded ($L_s$, $L_t$) as a classification loss. Here, $m_s$ and $m_t$ are additive margins for source and target domains, respectively. We set $m_s < m_t$ to impose stronger regularization on the target domain. In addition, for each pair of source (blue) and target (red) features, a pairwise spherical feature alignment loss is embedded based on the class equivalency of the pairs. For feature pairs of same class (blue and red circles, $y_j^s = y_j^t$), a pairwise similarity loss ($L_{ps}$) is embedded. For feature pairs of different class (blue triangle and red circle, $y_j^s \neq y_j^t$), a pairwise dissimilarity loss ($L_{pd}$) is embedded. Best viewed in color.
where $L_s$ and $L_t$ are additive angular margin losses on source and target domains, respectively.

3) **PAIRWISE SPHERICAL FEATURE ALIGNMENT LOSS**

To further match feature distributions of the two domains, we propose a pairwise spherical feature alignment scheme. Suppose a pair of normalized features $(\tilde{f}_i, \tilde{f}_j)$ is given on the spherical space, which are driven from the $i$-th source and the $j$-th target images, respectively. The similarity between the two normalized features $\tilde{f}_i$ and $\tilde{f}_j$ is obtained by a dot product as follows:

$$ (\tilde{f}_i)^T \tilde{f}_j = \cos(\phi_{x_i,x_j}), \quad (9) $$

where $\phi_{x_i,x_j}$ denotes the angle between the normalized features of $x_i$ and $x_j$. Our goal is to enhance the similarity of a pair sharing a same class. To this end, we define a metric $M_{ps}(\cdot, \cdot)$ as follows:

$$ M_{ps}(x_i, x_j) = \frac{1}{2} \left[ 1 - \cos(\phi_{x_i,x_j}) \right]. \quad (10) $$

On the contrary, we aim to penalize pairwise similarity of features of different classes, and a metric $M_{pd}(\cdot, \cdot)$ for this objective is defined as follows:

$$ M_{pd}(x_i, x_j) = \max \left( \frac{1}{2} \left[ 1 + \cos(\phi_{x_i,x_j}) \right] - \epsilon, 0 \right), \quad (11) $$

where $\epsilon$ is a constant that relaxes the objective. Setting $\epsilon = 0$ enforces every feature pair to have an opposite direction, which is infeasible in multiclass classification. In our work, we set $\epsilon$ to 0.5 for all experiments. By taking Eq. (10) and (11), pairwise loss functions are given as follows:

$$ L_{ps} = \mathbb{E}_{(x_i, x_j)^{\prime} \in \mathcal{D}_s, (x_i^{\prime}, x_j^{\prime}) \in \mathcal{D}_t} M_{ps}(x_i, x_j), \quad (12) $$

$$ L_{pd} = \mathbb{E}_{(x_i, x_j)^{\prime} \in \mathcal{D}_s, (x_i^{\prime}, x_j^{\prime}) \in \mathcal{D}_t} M_{pd}(x_i, x_j), \quad (13) $$

where $L_{ps}$ and $L_{pd}$ are pairwise similarity and dissimilarity losses, respectively.

4) **OVERALL OBJECTIVE FUNCTION**

The overall objective function is given as follows:

$$ L = L_s + L_t + \alpha (L_{ps} + L_{pd}), \quad (14) $$

where $\alpha$ is a balancing constant and set to 0.1 in our implementation. The overall pipeline of the proposed SDA scheme is illustrated in Fig. 2. For every iteration, $n_b$ images are randomly sampled from each domain (thus, $2n_b$ images for one iteration). The training procedure is continued until the overall loss is converged.

5) **FINE-TUNING ON TARGET DATA**

In SDA, the amount of source data is generally much larger than that of target data. As a result, the expectation of empirical risks on source data remains larger than that of target data after convergence. This makes a deep network more biased to the source domain, which is sub-optimal for our task. To further align to the target domain, we conduct fine-tuning with target data by means of $L_t$. Although the amount of the target data is small, this fine-tuning stage is resistant to overfitting since the spherical feature embedding scheme is inherently robust to few-shot setups. This simple method can enhance the final accuracies, as analyzed in Section V-C1.

**B. EXTENSION TO SSDA**

In SSDA, the amount of source data is generally much smaller than that of target data, which is sub-optimal for our data still after convergence. This makes a deep network more biased to the source domain, which is sub-optimal for our task. To further align to the target domain, we conduct fine-tuning with target data by means of $L_t$. Although the amount of the target data is small, this fine-tuning stage is resistant to overfitting since the spherical feature embedding scheme is inherently robust to few-shot setups. This simple method can enhance the final accuracies, as analyzed in Section V-C1.

**B. EXTENSION TO SSDA**

In SSDA, the amount of source data is generally much smaller than that of target data, which is sub-optimal for our data still after convergence. This makes a deep network more biased to the source domain, which is sub-optimal for our task. To further align to the target domain, we conduct fine-tuning with target data by means of $L_t$. Although the amount of the target data is small, this fine-tuning stage is resistant to overfitting since the spherical feature embedding scheme is inherently robust to few-shot setups. This simple method can enhance the final accuracies, as analyzed in Section V-C1.

**V. EXPERIMENTS**

**A. EXPERIMENTAL SETTINGS**

1) **DATASETS**

For comparative evaluations, we used the three benchmark datasets as follows. **DomainNet** [49] is a large-scale benchmark dataset for domain adaptation, which involves six domains with 345 classes. **Office** [50] involves three domains (Amazon, Webcam, and DSLR) with 31 classes. **Office-Home** [51] contains four domains (Real, Clipart, Art, and Product) with 65 classes. We used DomainNet as the primary benchmark dataset for comparative evaluation on SDA and SSDA. For further validation, we used the Office dataset for evaluation on SDA, and the Office-Home dataset on SSDA.

The setup for comparative evaluation on each dataset is as follows. For DomainNet, we addressed seven adaptation scenarios, which are derived from four domains (Real, Clipart, Painting, and Sketch) with 126 classes, following [14]. For Office, six adaptation scenarios are addressed [18], [19], [20].
TABLE 2. SDA results on the DomainNet dataset in terms of accuracy (%). ResNet-34 is adopted as the backbone architecture.

| Method  | Target setup | R→C | R→P | P→C | C→S | S→R | R→S | P→R | Avg. |
|---------|--------------|------|------|------|------|------|------|------|------|
| Source only | -          | 54.7 | 58.5 | 53.6 | 49.1 | 47.8 | 42.3 | 69.0 | 53.6 |
| CCSA [18]  | 1-shot   | 60.8 | 61.5 | 59.3 | 52.3 | 57.5 | 49.7 | 71.5 | 58.9 |
| FADA [19]  |          | 59.1 | 61.2 | 56.8 | 54.2 | 56.5 | 49.4 | 71.1 | 58.3 |
| d-SNE [20] |          | 57.1 | 62.2 | 59.2 | 53.5 | 57.1 | 49.3 | 72.0 | 58.6 |
| Proposed   |          | 60.9 | 63.9 | 60.5 | 54.7 | 57.9 | 53.5 | 73.8 | 60.7 |

TABLE 3. SDA results on the Office dataset with 3-shot setup (%). VGG-16 is adopted as the backbone architecture.

| Method  | A→D | A→W | D→A | D→W | W→A | W→D | Avg. |
|---------|------|------|------|------|------|------|------|
| Source only | 61.20 | 62.30 | 58.50 | 80.10 | 51.60 | 95.60 | 68.20 |
| CCSA [18]  | 89.00 | 88.20 | 71.80 | 96.40 | 72.10 | 97.60 | 95.80 |
| FADA [19]  | 88.20 | 88.10 | 68.10 | 96.40 | 71.10 | 97.50 | 94.90 |
| d-SNE [20] | 91.44 | 90.13 | 71.06 | 97.10 | 71.74 | 97.46 | 96.49 |
| Proposed   | 93.83 | 91.31 | 73.27 | 97.15 | 73.16 | 98.52 | 97.87 |

For Office-Home, five adaptation scenarios are selected [17]. All adaptation scenarios in our experiments are given one or three examples per category in a target domain (1-shot and 3-shot, respectively). For fair comparison, we adopted the labeled target image lists which are released in [14] for the DomainNet and the Office-Home datasets. For the Office dataset, we randomly selected source and target data, following [18], [19], and [20]. For both datasets, all available data in source domain are given for training. Meanwhile, in the Office dataset, the number of source data is 20 in Amazon domain, and 8 in Webcam and DSLR domains, following [18], [19], [20]. We employed ResNet-34 [52] for the DomainNet and the Office-Home datasets, and VGG-16 [53] for the Office dataset. For all experiments in SDA, we adopted ImageNet pre-trained models for weight initialization.

2) BASELINE METHODS FOR COMPARISON
We conducted comparative evaluations with three SDA methods and four SSDA methods. Baseline methods on SDA are CCSA [18], FADA [19], and d-SNE [20]. Baseline methods on SSDA are MME [14], SagNet [15], Meta-MME [17], and APE [16]. It is worth noting that these methods are specifically designed solely for SSDA, whereas our SSDA method is a simple extension of the proposed SDA scheme. All numerical measurements of other methods in this paper were directly quoted from the original papers, except the SDA results on the DomainNet dataset, which are our reproduced measurements.

3) IMPLEMENTATION DETAILS
For every iteration of the proposed SDA scheme, we randomly sampled 24 labeled images from the target domain. The angular margin parameters are given as $m_s = 0.1$ and $m_t = 0.25$ for source and target domains, respectively. The scaling parameter $r$ is set to 2 in the Office dataset and 20 in the other datasets. The maximum number of iteration is 50k, and a training phase is early stopped based on validation accuracies. For every iteration of the SSDA extension, we randomly sampled 12 labeled images and 144 unlabeled images from the target domain. The maximum iteration number of SSDA is 10k. Unlike [43], we employed a cross-entropy loss for mixup sets of labeled and unlabeled images. The other setups for the SSDA extension are identical to those in [43]. Both for SDA and SSDA, we used the Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 0.01, a momentum of 0.9, and a weight decay of $5 	imes 10^{-4}$. All experiments in this paper were implemented in PyTorch [54] by using a single NVIDIA RTX A5000 GPU.

B. EXPERIMENTAL RESULTS
1) RESULTS ON SDA
In Table 2, comparative evaluation results on the DomainNet dataset are reported. On average, the proposed SDA method outperforms the previous state-of-the-art by 1.8% and 2.2% on 1-shot and 3-shot setups, respectively. Among the seven adaptation scenarios, our SDA scheme leads to the largest performance gain over previous methods on the $R \rightarrow S$ scenario, surpassing by 3.8% and 3.9% on 1-shot and 3-shot setups, respectively. It is worth noting that the $R \rightarrow S$ scenario involves a large domain shift, and the accuracy without adaptation (Source only) on this scenario is the lowest among the seven adaptation scenarios. This indicates that the advantage of our SDA scheme becomes prominent for challenging scenarios involving a large domain disparity. The results on the Office dataset are reported in Table 3. Following [18], [19], and [20], we report an average accuracy with deviations of three runs for each adaptation scenario. On average, our SDA scheme surpasses the previous state-of-the-art method by 1.38%. The accuracy gains of our SDA scheme are relatively

VOLUME 10, 2022
TABLE 4. SSDA results on the DomainNet dataset in terms of accuracy (%). ResNet-34 is adopted as the backbone architecture.

| Method          | Target setup | R→C | R→P | P→C | C→S | S→P | R→S | P→R | Avg. |
|-----------------|--------------|-----|-----|-----|-----|-----|-----|-----|------|
| Source only     | -            | 54.7| 58.5| 53.6| 49.1| 47.8| 42.3| 69.0| 53.6 |
| MME [14]        | 1-shot       | 70.0| 67.7| 69.0| 56.3| 64.8| 61.0| 76.1| 66.4 |
| APE [16]        | 70.4| 70.8| 72.9| 56.7| 64.5| 63.0| 76.6| 67.6 |
| SagNet [15]     | 72.3| 69.0| 70.8| 61.7| 66.9| 64.3| 75.3| 68.6 |
| Proposed        | 76.4| 70.1| 74.9| 65.3| 69.1| 66.5| 79.1| 71.6 |

TABLE 5. SSDA results on the Office-home dataset with 3-shot setup (%). ResNet-34 is adopted as the backbone architecture.

| Method       | R→P | P→A | P→C | A→C | C→A | Avg. |
|--------------|-----|-----|-----|-----|-----|------|
| Source only  | 47.6| 39.9| 48.7| 46.1| 48.5| 46.2 |
| MME [14]     | 64.6| 64.5| 65.5| 63.6| 67.2| 65.1 |
| Meta-MME [17] | 65.2| 64.5| 66.7| 63.7| 67.5| 65.4 |
| APE [16]     | 66.4| 65.2| 66.1| 63.9| 66.6| 65.6 |
| Proposed     | 66.4| 66.0| 68.6| 64.4| 69.7| 67.0 |

TABLE 6. Ablation study results on the proposed SDA loss functions. Average accuracies on the seven adaptation scenarios in DomainNet (%).

| SDA loss function | 1-shot | 3-shot |
|-------------------|--------|--------|
| Vanilla softmax loss - Euclidean | 54.0 | 57.4 |
| Vanilla softmax loss - Spherical | 56.5 | 59.7 |
| Additive angular margin loss ($L_s$, $L_t$) | 59.2 | 62.5 |
| Pairwise alignment loss ($L_{pa}$) | 59.8 | 63.7 |
| Fine-tuning with $L_t$ | 60.7 | 64.4 |

2) RESULTS ON SSDA

We report the SSDA results on the DomainNet dataset in Table 4. Our SSDA extension achieves state-of-the-art accuracies except one scenario (R→P with 1-shot). On average, our method surpasses the other SSDA methods by 3.0% and 1.9% on 1-shot and 3-shot setups, respectively. The proposed SSDA scheme is particularly robust to 1-shot setups, and the average accuracy of our method with the 1-shot setup is superior or competitive to the 3-shot accuracies of the previous methods. The results on the Office-Home dataset is reported in Table 5. On average, our SSDA extension surpasses previous methods by 1.4%. It is worth noting that the previous methods [14], [15], [16], [17] are specifically designed for SSDA, whereas our SSDA method is a semi-supervised extension of the proposed SDA scheme.

C. ANALYSIS AND DISCUSSIONS

1) ABLATION STUDY ON SDA

To investigate the impact of each proposed module, we conducted ablation studies on the seven adaptation scenarios in DomainNet, and the results are reported in Table 6. When a vanilla softmax is employed, the spherical feature space leads to better results than the Euclidean feature space, demonstrating its robustness to few-shot setups. Based on the spherical feature space, the proposed additive angular margin losses ($L_s$, $L_t$) considerably enhance the performance. In addition, the pairwise spherical feature alignment losses ($L_{pa}$) and the following fine-tuning stage further improve the performance.

In our proposed additive angular margin loss for SDA, the margin parameters $m_s$ and $m_t$ are incorporated with the default values of 0.1 and 0.25, respectively. To investigate the impact of each margin parameter, we report ablation study results in Table 8. As we can see in the table, the margin parameters of both domains contribute to the performance of SDA. On the other hand, the effectiveness of the angular margin loss is significantly degraded when the magnitude of $m_s$ becomes larger than that of $m_t$, as indicated in the last row in Table 8. This implies that the magnitude of a margin parameter is closely related to the priority (or importance) of data in a certain domain. Based on this observation, we always set $m_s < m_t$ to make a deep network more focused on learning with target domain data.

2) VALIDATION ON VISION TRANSFORMER (ViT)

To further investigate the robustness of our SDA method across various backbone architectures, we report comparative evaluation results on vision Transformer (ViT) [23]. Unlike CNN-based models (e.g., ResNet and VGGNet), ViT is free from convolution operators, and is composed of multiple self-attention blocks [55]. For experiment, we adopt the ViT base model with 16 self-attention blocks which is pre-trained on ImageNet-21k [56]. Table 7 shows the results. In SDA, our proposed method outperforms other previous works on the DomainNet dataset. This result demonstrates that our SDA scheme can be broadly applied across various backbones from convolutional (CNNs) to self-attention models (Transformers).

3) IMPACTS OF SDA PRE-TRAINING ON SSDA EXTENSION

As explained in Section IV-B, the SSDA extension scheme assumes that a network is initialized by our SDA scheme. To investigate the impact of the initialization method in the

VOLUME 10, 2022
Y. Kim: Learning to Adapt to Label-Scarce Image Domain via Angular Distance-Based Feature Alignment

**FIGURE 3.** t-SNE visualization results of the $S \rightarrow P$ scenario in the DomainNet dataset. Each two-dimensional point in the figure represents a feature vector which is obtained by the t-SNE visualization scheme [57]. ResNet-34 is adopted as the backbone architecture. Best viewed in color.

**TABLE 7.** SDA results in terms of the average accuracy on the seven adaptation scenarios in DomainNet. Vision Transformer (ViT) is adopted as the backbone architecture.

| Method      | Accuracy (%) |
|-------------|--------------|
| Source only | 62.1         |
| Target setup |             |
| 1-shot      | 66.1         |
| 3-shot      | 68.7         |
| CCSA [18]   | 65.4         |
| FADA [19]   | 68.1         |
| d-SNE [20]  | 66.9         |
| Proposed    | **67.8**     |
|             | **71.9**     |

SSDA extension stage, we conducted an ablation study on SSDA by varying SDA pre-training methods. The results are reported in Table 9. Without SDA pre-training (i.e., initializing with an ImageNet pre-trained model), the SSDA results are far lower than those with SDA pre-training. This indicates that the pre-training stage is highly demanded for semi-supervised learning with a small number of labeled data. By comparing the two SSDA results, which are driven by SDA pre-trained weights, it can be confirmed that our SDA scheme leads to higher SSDA accuracies than applying a vanilla softmax loss. This implies that the SDA pre-training stage has a significant impact on the performance of SSDA.

4) **QUALITATIVE ANALYSIS VIA t-SNE VISUALIZATION**

To qualitatively analyze the impact of our proposed methods on feature alignment, we present the t-SNE [57] visualization results in Fig. 3. t-SNE (t-distributed Stochastic Neighborhood Embedding) is a statistical method for visualizing high-dimensional data by projecting them into the two or three dimensional space [57]. In our work, we applied the t-SNE scheme to visualize the high-dimensional feature vectors (e.g., 512-dimension for ResNet-34) in the two-dimensional space. On the upper row in Fig. 3, the visualization results showing source and target features are presented. Comparing the three learning schemes (i.e., Source only, SDA, and SSDA), we can see that the features from the two distinct domains are aligned by our proposed DA methods. On the other hand, the target features without applying DA scheme (Source only) are not well complied with the source features. The benefits of our DA schemes become more apparent when we focus on target features, as shown on the second row in Fig. 3. For the 15 categories that randomly selected from the DomainNet dataset, each target feature are colorized based on its category. By comparing target features of the three methods, we can figure out that our DA schemes enhance the discriminative ability in target domain.
TABLE 8. Ablation study results on the additive margin parameters ($m_s$ and $m_t$). Average accuracies on the seven adaptation scenarios in DomainNet (%). Note that the measurements below are obtained by the additive angular margin loss only ($L_s$ and $L_t$).

| $m_s$ (0.1) | $m_t$ (0.25) | 1-shot | 3-shot |
|------------|-------------|--------|--------|
| -          | -           | 56.5   | 59.7   |
| √          | -           | 57.1   | 60.4   |
| -          | √           | 58.8   | 61.9   |
| √          | √           | 59.2   | 62.5   |

TABLE 9. A comparison on SSDA results depending on SDA pre-training schemes. Average accuracies on the seven adaptation scenarios in DomainNet (%).

| SDA pre-training scheme | 1-shot | 3-shot |
|-------------------------|--------|--------|
| Without SDA pre-training | 34.5   | 42.7   |
| Vanilla softmax - Spherical | 69.8   | 71.7   |
| Proposed SDA scheme      | 71.6   | 73.6   |

5) COMPARISON WITH UDA

In this subsection, we introduce a comparative analysis on SDA and UDA. As indicated in Table 1, SDA assumes that a few amount of image and label pairs are given, whereas UDA assumes that relatively larger amount of unlabeled images are accessible in target domain. Thus, SDA would be more practical than UDA if the cost of collecting target images is larger than that of assigning labels to target images. To provide a quantitative comparison on SDA and UDA, we report average accuracies of the 12 adaptation scenarios in Office-Home (ResNet-50).

TABLE 10. Comparison with UDA methods. Each measurement is an average accuracy (%) of the 12 adaptation scenarios in Office-Home (ResNet-50).

| Method     | Avg.  | Method   | Avg.  |
|------------|-------|----------|-------|
| Source only | 46.1  | SymNets [9] | 67.6  |
| DANN [3]   | 57.6  | RWOT [13]  | 67.7  |
| SR [7]     | 61.5  | CDAN+SagNet [15] | 65.3  |
| CDAN [4]   | 65.8  | SymNet+SagNet [15] | 68.0  |
| BSP [58]   | 66.3  | Proposed SDA (1-shot) | 63.6  |
| SAPN [8]   | 67.3  | Proposed SDA (3-shot) | 68.8  |

REFERENCES

[1] S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, Oct. 2009.

[2] W. Mei and D. Wei, “Deep visual domain adaptation: A survey,” Neurocomputing, vol. 312, pp. 135–153, Jul. 2018.

[3] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in Proc. Int. Conf. Mach. Learn. (ICML), 2015, pp. 1180–1189.

[4] M. Long, Z. Cao, J. Wang, and M. Jordan, “Conditional adversarial domain adaptation,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2018, pp. 1640–1650.

[5] R. Valpi, P. Moretto, S. Savarese, and V. Murino, “Adversarial feature augmentation for unsupervised domain adaptation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 5495–5504.

[6] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, “Maximum classifier discrepancy for unsupervised domain adaptation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 3723–3732.

[7] G. French, M. Mackiewicz, and M. Fisher, “Self-ensembling for visual domain adaptation,” 2017, arXiv:1706.05208.

[8] R. Xu, G. Li, J. Yang, and L. Lin, “Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 1426–1435.

[9] Y. Zhang, H. Tang, K. Jia, and M. Tan, “Domain-symmetric networks for adversarial domain adaptation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 5026–5035.

[10] X. Gu, J. Sun, and Z. Xu, “Spherical space domain adaptation with robust pseudo-label loss,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 9101–9110.

[11] L. Hu, M. Kan, S. Shan, and X. Chen, “Unsupervised domain adaptation with hierarchical gradient synchronization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 4034–4052.

[12] H. Tang, K. Chen, and K. Jia, “Unsupervised domain adaptation via structurally regularized deep clustering,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 8725–8735.

[13] R. Xu, P. Liu, L. Wang, C. Chen, and J. Wang, “Reliable weighted optimal transport for unsupervised domain adaptation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 4394–4403.

[14] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, “Semi-supervised domain adaptation via minimax entropy,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 8050–8058.

[15] H. Nam, H. Lee, J. Park, W. Yoon, and D. Yoo, “Reducing domain gap by reducing style bias,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 8690–8699.

[16] T. Kim and C. Kim, “Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2020, pp. 591–607.

[17] D. Li and T. Hospedales, “Online meta-learning for multi-source and semi-supervised domain adaptation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2020, pp. 382–403.
S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, “Unified deep supervised domain adaptation and generalization,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 5715–5725.

S. Motiian, Q. Jones, S. M. Iramnesh, and G. Doretto, “Few-shot adversarial domain adaptation,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2017, pp. 6670–6680.

X. Xu, X. Zhou, R. Venkatesan, G. Swaminathan, and O. Majumder, “d-SNE: Domain adaptation using stochastic neighborhood embedding,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, pp. 2497–2506.

W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, “A closer look at few-shot classification,” 2019, arXiv:1904.04232.

J. Deng, J. Guo, N. Xue, and S. Zaferiou, “ArcFace: Additive angular margin loss for deep face recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4690–4699.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, and J. Uszkoreit, “An image is worth 16x16 words: Transformers for image recognition at scale,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2021, pp. 1–22.

Z. Cao, M. Long, J. Wang, and M. I. Jordan, “Partial transfer learning with selective adversarial networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 2724–2732.

Z. Cao, K. You, M. Long, J. Wang, and Q. Yang, “Learning to transfer examples for partial domain adaptation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 2985–2994.

P. P. Busto and J. Gall, “Open set domain adaptation,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 754–763.

K. Saito, S. Yamamoto, Y. Ushiku, and T. Harada, “Open set domain adaptation by backpropagation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 153–168.

H. Liu, Z. Cao, M. Long, J. Wang, and Q. Yang, “Separate to adapt: Open set domain adaptation via progressive separation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 2927–2936.

K. You, M. Long, Z. Cao, J. Wang, and M. I. Jordan, “Universal domain adaptation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 2720–2729.

K. Saito, D. Kim, S. Sclaroff, and K. Saenko, “Universal domain adaptation through self supervision,” 2020, arXiv:2002.07953.

M. Irfan, Z. Jiangbin, M. Iqbal, and M. H. Arif, “A novel lifelong learning model based on cross domain knowledge extraction and transfer to classify underwater images,” Inf., vol. 552, pp. 80–101, Apr. 2021.

M. Irfan, Z. Jiangbin, M. Iqbal, Z. Masood, and M. H. Arif, “Knowledge extraction and retention based continual learning by using convolutional autoencoder-based learning classifier system,” Inf., vol. 591, pp. 287–305, Apr. 2022.

G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” in Proc. ICML Deep Learning Workshop, vol. 2, 2015, pp. 1–30.

J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2017, pp. 4077–4087.

Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, “Meta-transfer learning for few-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 403–412.

L. Yang, L. Li, Z. Zhang, X. Zhou, E. Zhou, and Y. Liu, “DPGN: Distribution propagation graph network for few-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 13390–13399.

Y. Duan, W. Zheng, X. Lin, J. Lu, and J. Zhou, “Deep adversarial metric learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 2780–2789.

W. Zheng, J. Lu, and J. Zhou, “Deep metric learning via adaptive learnable assessment,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 2960–2969.

D. Yi, Z. Lei, S. Liao, and S. Z. Li, “Deep metric learning for person re-identification,” in Proc. 22nd Int. Conf. Pattern Recognit., Aug. 2014, pp. 34–39.

S. Bak and P. Carr, “One-shot metric learning for person re-identification,” in Proc. IEEE/CVF Conf. Pattern Recognit. (CVPR), Jul. 2017, pp. 2990–2999.

W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “SphereFace: Deep hypersphere embedding for face recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 212–220.

H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, “CosFace: Large margin cosine loss for deep face recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 5265–5274.

D. Bethelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, “MixMatch: A holistic approach to semi-supervised learning,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2019, pp. 5049–5059.

S. Laine and T. Aila, “Temporal ensembling for semi-supervised learning,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2017, pp. 1–13.

A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2017, pp. 1195–1204.

D.-H. Lee, “Pseudo-Label: The simple and efficient semi-supervised learning method for deep neural networks,” in Proc. Workshop Challenges Represent. Learn. (ICMLW), 2013, vol. 3, no. 2, p. 896.

H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2018, pp. 1–13.

Y. Grandvalet and Y. Bengio, “Semi-supervised learning by entropy minimization,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2005, pp. 529–536.

X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, “Moments matching for multi-source domain adaptation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 1406–1415.

K. Saenko, B. Kulis, M. Fritz, and T. Darrell, “Adapting visual category models to new domains,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2010, pp. 213–226.

H. Venkateswarar, J. Eusebio, S. Chakraborty, and S. Panchanathan, “Deep hashing network for unsupervised domain adaptation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 5018–5027.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” in Proc. NIPS, 2017, pp. 1–4.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 1–11.

T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, “ImageNet-21K pretraining for the masses,” 2021, arXiv:2104.10972.

J. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol. 9, pp. 2579–2605, Nov. 2008.

X. Chen, S. Wang, M. Long, and J. Wang, “Transferability vs. Discriminability: Batch spectral penalization for adversarial domain adaptation,” in Proc. Int. Conf. Mach. Learn. (ICML), 2019, pp. 1081–1090.

Y. Kim: Learning to Adapt to Label-Scarce Image Domain via Angular Distance-Based Feature Alignment