SEMI-SUPERVISED SINGLE DOMAIN GENERALIZATION WITH LABEL-FREE ADVERSARIAL DATA AUGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain generalization (DG) has attracted increasing attention recently, as it seeks to improve the generalization ability of visual recognition models to unseen target domains. DG leverages multiple source domains for model training, while single domain generalization (SDG) further restricts such setting by exploiting only a single source domain. Nevertheless, both DG and SDG assume that the source domains are fully labeled, which might not be practical in many real world scenarios. In this paper, we present a new problem, i.e., semi-supervised single domain generalization (SS-SDG), which aims to train a model with a partially labeled single source domain to generalize to multiple unseen testing domains. We propose an effective framework to address this problem. In particular, we design a label-free adversarial data augmentation strategy to diversify the source domain, and propose a novel multi-pair FixMatch loss to generalize classifiers to unseen testing domains. Extensive experiments on OfficeHome, PACS and DomainNet20 datasets show that our method surpasses the latest SDG and semi-supervised methods. Moreover, on PACS and DomainNet20, our method approaches the fully supervised ERM upper bound within 5% gap, but only uses less than 8% of the labels.

1 INTRODUCTION

Deep neural networks (DNNs) lead large success in the past decade in many fields, e.g., object detection and classifications. Many of the applications rely on the assumption that training and testing distributions are identical or close. However, in real scenarios, data acquiring always encounters the environment variance, i.e., the lighting changes from dawn to night, or the camera moves from one place to another. The environment variance inevitably brings in the domain shift for the captured training and testing data (Recht et al., 2019; Hendrycks & Dietterich, 2019). Closing this domain discrepancy has become one of the recent popular topics in the community.

Domain adaptation (DA) (Wang & Deng, 2018) and domain generalization (DG) (Zhou et al., 2021a; Wang et al., 2021a) are the major techniques to tackle this problem. DA methods jointly exploit the source and target domain data for model training, in which the methods attempt to align the feature space between the source and target domains. While DG methods solve a more challenging task, utilizing multiple labeled source domain data to learn towards a generalized model, to predict the target domain data which is unavailable in the training process. Compared to DA, DG relaxes the assumption on target domains and usually enjoys better model generalization ability. The differences between DA and DG are illustrated in Figure 1.

Though promising, DG still faces two major limitations. First, DG methods require multiple source domains for model training. In practice, it would be expensive or even infeasible to collect multiple source domains' data. To address this problem, single domain generalization (SDG) (Qiao et al., 2020) has been proposed recently. Different from DG, SDG aims to train a model with a single source domain data and generalize to multiple unseen target domains. Second, DG methods require that the source domains should be fully labeled, which is usually expensive and labor intense. Very recently, semi-supervised domain generalization (SS-DG) (Zhou et al., 2021b) is proposed to address this issue, which assumes only a small portion of the samples are labeled in source domains. Clearly, SDG and

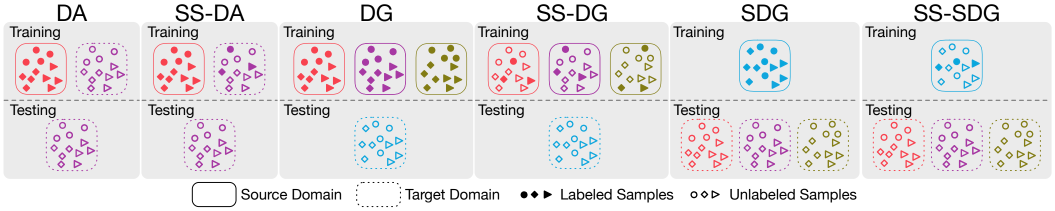

Figure 1: Problem setting differences among domain adaptation (DA), domain generalization (DG), single domain generalization (SDG), semi-supervised domain adaptation (SS-DA), semi-supervised domain generalization (SS-DG), and our newly introduced setting semi-supervised single domain generalization (SS-SDG) task. Different colors indicate different domains.

Table 1: Component-wise comparison of the proposed Semi-Supervised Single Domain Generalization (SS-SDG) to Domain Adaptation (DA), Domain Generalization (DG), Single Domain Generalization (SDG), Semi-supervised Domain Adaptation (SS-DA), Semi-supervised Domain Generalization (SS-DG). $\mathcal{D}_s$ and $\mathcal{D}_t$ denote the source domain and target domain, respectively.

Problem Setting	DA	DG	SDG	SS-DA	SS-DG	SS-SDG (ours)
Number of $\mathcal{D}_s > 1$?	×	✓	×	×	✓	×
$\mathcal{D}_s$ is full annotated?	✓	✓	✓	✓	×	×
Access of $\mathcal{D}_t$ in training process?	✓	×	×	✓	×	×

SS-DG separately tackle each of the two limitations. However, we desire a unified framework that can boost DG from both of the two perspectives.

Consequently, we propose a more practical yet unsolved problem, i.e., semi-supervised single domain generalization (SS-SDG). In this problem, we assume that only one source domain is available for model training, and it consists of a few labeled and abundant unlabeled samples. The relationships between SS-SDG and other related problems are illustrated in Figure 1. We also summarize the component-wise differences between the proposed SS-SDG and other settings in Table 1. While SDG is the most relevant setting to ours, most of existing SDG methods (Zhao et al., 2020; Qiao et al., 2020; Fan et al., 2021) are based on adversarial data augmentation (Volpi et al., 2018) and require the label information to generate new samples to enrich the diversity of source domain. As the result, these SDG methods would fail in our proposed SS-SDG setting which cannot provide sufficient and accurate label information for data augmentation. Our empirical results validate that existing SDG methods perform unsatisfying in our SS-SDG task.

To address the new challenging problem, we propose a novel label-free adversarial data augmentation framework to enrich the source domain diversity without label information, as well to leverage such generated data in a multi-pair FixMatch way to regularize for better training. Figure 2 illustrates the overall flowchart of our method. Inspired by self-supervised learning (SSL) (Yang et al., 2021), we design a label-free adversarial data augmentation strategy, which is an interactive feature extractor pre-train and adversarial sample generation approach without label information, to enhance the diversity of source domain data. Given source domain and the newly generated samples, we organize them into multiple training pairs and propose a novel multi-pair FixMatch (MPFM) loss to regularize the classifier training for better generalization ability to unseen testing domains.

Our contributions are thus summarized as:

- We introduce a new challenging domain generalization task, namely the semi-supervised single domain generalization (SS-SDG), addressing the scenario of training on a partially labeled single source domain while generalizing to multiple unseen target domains.
- We propose an efficient framework, introducing a novel label-free adversarial data augmentation strategy to enrich diversity of the single source domain diversity without label information, and a multi-pair FixMatch regularization to better utilize the diversified data towards a more generalized classifier for unseen target domains.
- We conduct extensive SS-SDG experiments on OfficeHome, PACS and DomainNet20, and achieve superior performance over the state-of-the-arts, e.g., approaching supervised ERM upper bound within 5% accuracy gap by only using less than 8% labels.

2 Related Work

In this section, we briefly introduce and discuss three closely related research topics to our work.

Domain Generalization (DG) aims at learning a generalized model with multiple source domains to adapt to an unseen target domain. Early DG methods (Ganin et al., 2016; Li et al., 2018b; Piratla et al., 2020) aim to learn a domain-invariant feature space by aligning distributions across all source domains. Following-up DG methods (Shankar et al., 2018; Zhou et al., 2020; Xu et al., 2021; Huang et al., 2021) explore how to generate extra synthetic data based on the source domain data, and use the joint data to recover the unseen target domain distribution. Recently, meta-learning based methods (Li et al., 2018a; Dou et al., 2019; Du et al., 2020) leverage the multiple source domains in a meta-train and meta-test manner and update the gradient in a more sophisticated way.

Single Domain Generalization (SDG) is a more challenging setting compared to domain generalization, which only uses one source domain to learn a model and generalize to multiple unseen target domains. A mainstream of recent SDG methods (Volpi et al., 2018; Zhao et al., 2020; Qiao et al., 2020; Li et al., 2021; Fan et al., 2021; Wang et al., 2021b) leverages data augmentation in diversifying the single source domain distribution for better generalization ability. Different from SDG, we propose a more practical yet challenging setting, i.e., the semi-supervised single domain generalization (SS-SDG), where labeled samples in the source domain are limited. Since the previous SDG methods' data augmentation highly relies on label information, when shifting to our SS-SDG setting, their methods' effectiveness can be degraded due to insufficient label information.

Semi-Supervised Learning (SSL) is a fundamental research topic in computer vision and machine learning, which seeks to learn a model with few labeled and a large amount of unlabeled data. SSL methods can be generally divided into three categories: (1) The pseudo-labeling (Lee et al., 2013) based methods (Berthelot et al., 2020; Xie et al., 2020b), utilize the intermediate model to predict the pseudo ground truth label and iteratively update the model with the pseudo labeled data. (2) The consistency constraint (Oliver et al., 2018) based methods (Tarvainen & Valpola, 2017; Miyato et al., 2019; Zhang & Qi, 2020), leverage the consistency across the same data with multiple augmentations to regularize the embedding learning. (3) The comprehensive methods (Sohn et al., 2020; Xie et al., 2020a; Zhang et al., 2021) combine both the pseudo-labeling and data augmentation for more performance boost. The canonical SSL setting would be impractical, as it assumes the training distribution and testing distribution are similar. To relax the assumption, a new semi-supervised domain generalization (SS-DG) (Zhou et al., 2021b) is proposed, i.e., StyleMatch (Zhou et al., 2021b), which tackles SS-DG by inducing style augmentation (Huang & Belongie, 2017) to extend source domain distribution in the FixMatch (Sohn et al., 2020) framework, and alleviates overfitting problem with stochastic classifier. Our proposed semi-supervised single domain generalization (SS-SDG), is a step further, with one source domain and a few labeled data to generalize to multiple target domains, in which the SS-DG methods, e.g., StyleMatch, are sub-optimal as source domain labeled data is extremely limited.

3 Our Approach

In this section, we start by introducing the SS-SDG problem with formal notations in Sec. 3.1. Then, we introduce the model pre-train (Figure 2 "Stage 1-1") in Sec. 3.2 including the contrastive learning loss design and the model parameter update. Further, we illustrate the new label-free adversarial data augmentation (Figure 2 "Stage 1-2") in Sec. 3.3. Finally, we explain our novel multi-pair FixMatch loss (Figure 2 "Stage 2") in Sec. 3.4 and propose the overall training objective.

3.1 Preliminaries

In semi-supervised single domain generalization (SS-SDG), a source domain is denoted as $\mathcal{D}_s = \{\mathcal{D}_s^l, \mathcal{D}_s^u\}$, where $\mathcal{D}_s^l = \{(x_s^{l,i}, y_s^i)\}_{i=1}^{m_s^l}$ is the portion of limited labeled samples and $\mathcal{D}_s^u = \{(x_s^{u,i})\}_{i=1}^{m_s^u}$ indicates abundant unlabeled samples. The multiple target domains are defined as $\mathcal{D}_t = \{\mathcal{D}_t^1, ..., \mathcal{D}_t^n\}$ where $\mathcal{D}_t^i = \{(x_{t,k}^i)\}_{i=1}^{m_t^i}$ indicates the i-th unseen target domain data. The source and target domains are sampled from different distributions but share the same label space. Our goal is to learn a model $\{\mathbf{G}, \mathbf{C}\}$, that can perform well on unseen multiple target domains $\mathcal{D}_t$ by

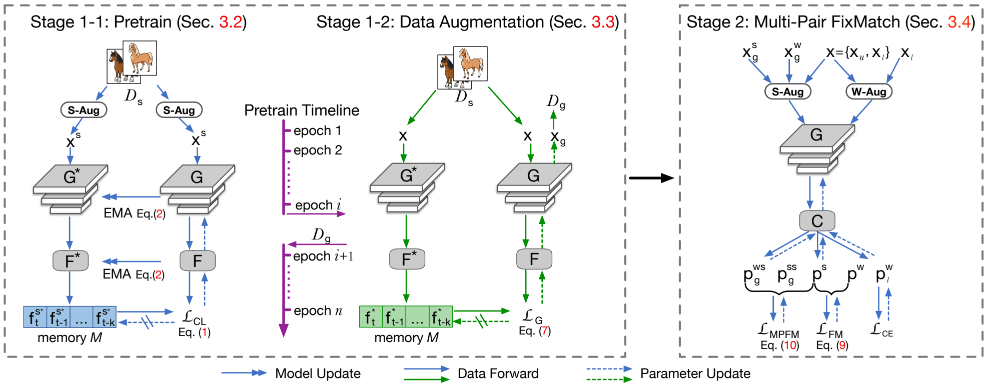

Figure 2: The proposed framework for SS-SDG. Our framework consists of two stages. At stage 1, we jointly train a feature extractor $\mathbf{G}$ in a self-supervised manner by excavating samples intrinsic supervision with contrastive learning (Stage 1-1), and generate new samples to enrich the diversity of source domain in an adversarial way (Stage 1-2) without label information. At stage 2, We propose a novel multi-pair FixMatch (MPFM) loss to better utilize the rich pair information for better generalization on unseen target domains.

utilizing partially labeled single source domain $\mathcal{D}_s$. In particular, we aim to learn a feature extractor $\mathbf{G} : x \to g$ that maps the input sample x into an embedding space, and train a classifier, $\mathbf{C} : g \to p$ that conducts classification by optimizing:

$$\min \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{x \in \mathcal{D}_t^i} \mathcal{L}(\mathbf{C}(\mathbf{G}(x)), y) \tag{1}$$

where $\mathcal{L}$ is a general classification loss term. Notice that y is known only at the testing phase for the target domains. $\mathbf{G}$ and $\mathbf{C}$ are learned only with the source domain data. We hereby introduce the learning process.

3.2 Self-supervision with Contrastive Learning

Previous SDG methods (Volpi et al., 2018; Zhao et al., 2020) may encounter degraded performance in the SS-SDG setting as the source domain data is with insufficient labels. Targeting on the lacking label venue, recent contrastive learning (He et al., 2020; Chen et al., 2020; Grill et al., 2020; Zbontar et al., 2021; Zhu et al., 2021) methods have shown promising results on unsupervised feature learning. They train the feature extractor by pulling the features from same sample under different augmented views close, and pushing the features from different samples apart.

Following the spirit, we propose to adopt contrastive learning framework to pre-train the feature extractor $\mathbf{G}$ with all source samples. As shown in Figure 2 "Stage 1-1: Pretrain", the contrastive framework consists of two feature extractors $\mathbf{G}^*$ and $\mathbf{G}$, two MLP projection heads $\mathbf{F}^*$ and $\mathbf{F}$, and a memory bank M used to store recent k samples' features by adopting the First-In-First-Out strategy to dynamically update it. Notice that $\mathbf{F}$ and $\mathbf{F}^*$ are the MLP projection heads only for the pre-train stage. They are different from our final classifier $\mathbf{C}$. We apply strong augmentation (i.e., RandAugment (Cubuk et al., 2020)) on source sample x twice, to obtain two strongly augmented inputs. Then by feeding these two inputs into $\mathbf{G}$ and $\mathbf{G}^*$, respectively, we compute the similarity between the augmented sample feature f_t^s and all features stored in memory bank and adopt the InfoNCE (Oord et al., 2018) loss to optimize the framework:

$$\mathcal{L}_{CL} = -\log \frac{\exp\left(f_t^s \cdot f_t^{s^*}/\tau\right)}{\sum_{i=0}^{k} \exp\left(f_t^s \cdot f_{t-i}^{s^*}/\tau\right)}, \tag{2}$$

where f_t^s represents the features of the t-th iteration inputs, and τ is a temperature hyper-parameter that controls the concentration level of the distribution (Hinton et al., 2015). To improve the consistency of features in memory bank, we employ the exponential moving average (EMA) strategy (Cai et al.,

2021) to update parameters (θ^*) and normalization factors (μ^* and σ^*) in $\mathbf{G}^*$ and $\mathbf{F}^*$ from $\mathbf{G}$ and $\mathbf{F}$:

$$\begin{aligned} \theta^* &\leftarrow \alpha\theta^* + (1-\alpha)\theta, \\ \mu^* &\leftarrow \alpha\mu^* + (1-\alpha)\mu, \\ \sigma^* &\leftarrow \alpha\sigma^* + (1-\alpha)\sigma, \end{aligned} \tag{3}$$

where α is a momentum coefficient close to 1, e.g., 0.999. The parameters θ in $\mathbf{G}$ and $\mathbf{F}$ are learned by standard SGD optimizer. There is no back-propagation through the $\mathbf{G}^*$ and $\mathbf{F}^*$. The $\mathbf{G}^*$ and $\mathbf{F}^*$ can be viewed as the smooth temporal ensemble of $\mathbf{G}$ and $\mathbf{F}$ along the training iterations.

3.3 Label-Free Adversarial Data Augmentation

While the self-supervised pre-train provides a good initialization, the problem of generalization to unseen target domains still exits. As we face the challenge of only using a single source domain to train $\mathbf{G}$ and $\mathbf{C}$, the thumb obstacle is the source data diversity. A highly concentrated source distribution can easily cause model overfitting. Many existing single domain generalization (SDG) methods (Volpi et al., 2018; Zhao et al., 2020; Qiao et al., 2020; Fan et al., 2021; Wang et al., 2021b) adopt the adversarial data augmentation fashion (Goodfellow et al., 2015) to complement the diversity of source domain. They formulate the SDG problem into a worst-case scenario (Sinha et al., 2018):

$$\min_{\psi} \sup_{\mathcal{D}_t} \left\{ \mathbb{E}[\mathcal{L}_{ce}(\psi; \mathcal{D}_t) : d(\mathcal{D}_t, \mathcal{D}_s) \leq \rho] \right\}, \tag{4}$$

where d represents a distance metric to evaluate the distribution similarity between source and target domains. ρ indicates the largest domain discrepancy between $\mathcal{D}_s$ and $\mathcal{D}_t$ in embedding space. ψ denotes model parameters optimized by cross-entropy loss $\mathcal{L}_{ce}$ with label information. The worst-case scenario (Equation 4) can be reformulated into a Lagrangian optimization problem with a fixed penalty parameter β:

$$\min_{\psi} \sup_{\mathcal{D}_g} \left\{ \mathbb{E}[\mathcal{L}_{ce}(\psi; \mathcal{D}_g)] - \beta d_W(\mathcal{D}_g, \mathcal{D}_s) \right\}, \tag{5}$$

where $\mathcal{D}_g$ indicates the generated domain from $\mathcal{D}_s$ and d_W denotes the Wasserstein metric (Volpi et al., 2018) applied to preserve the semantics of the generated samples. The overall loss function is formulated as:

$$\mathcal{L}_{SDG}(\psi; \mathcal{D}_s) = \mathcal{L}_{ce}(\psi; \mathcal{D}_s) - \beta d_W(\mathcal{D}_g, \mathcal{D}_s), \tag{6}$$

The new domain $\mathcal{D}_g$ is generated from $\mathcal{D}_s$ by maximizing $\mathcal{L}_{SDG}$ with label information under a small number of iterations: $x_{j+1} \leftarrow x_j + \eta \nabla_{x_j} \mathcal{L}_{SDG}(\psi; x_j)$.

Clearly, existing adversarial data augmentation based methods in SDG highly rely on label information used by $\mathcal{L}_{ce}$, whereas the limited labeled source data from SS-SDG setting impede the above SDG methods to generate plentiful new samples to diversify the source data distribution. By seamlessly combining the self-supervised signal introduced in Equation 2 and the worst-case scenario in Equation 5, we formulate our label-free adversarial data augmentation as:

$$\mathcal{L}_G(\theta; \mathcal{D}_s) = \mathcal{L}_{CL}(\theta; \mathcal{D}_s) - \beta d_W(\mathcal{D}_g, \mathcal{D}_s). \tag{7}$$

Following the same updating rule, as shown in Figure 2 “Stage 1-2”, we input each source sample into $\mathbf{G}^*$ and $\mathbf{G}$, respectively. We compute the similarity between the one passes $\mathbf{G}$ and all features stored in memory bank (a different memory bank from the pre-train stage). With a small number of iterations to maximize Equation 7 by:

$$x_{j+1} \leftarrow x_j + \eta \nabla_{x_j} \mathcal{L}_G(\theta; x_j) \tag{8}$$

In this way, we generate a new domain $\mathcal{D}_g$ to diversify source domain $\mathcal{D}_s$ without label information. The data augmentation phase is alternated with the pre-train phase along the training epochs.

3.4 Multi-Pair FixMatch Regularization

With the augmented $\mathcal{D}_g$ and original $\mathcal{D}_s$, there are rich information amongst the source domain data now. For instance, we can pair the strongly augmented data with the original data, or strongly augmented data with the weakly augmented data. Further, we can pair the generated data with original data as well as the strongly/weakly augmented data. Such multi-pair information clearly brings us a better chance to regularize the feature representation learning than the traditional original-to-augmented pairs. As FixMatch (Sohn et al., 2020) applies similar strong/weak augmentations

Algorithm 1 Our proposed Algorithm.

```
Require: $\mathcal{D}_s, \mathcal{D}_g, \mathcal{D}_g^w, \mathcal{D}_g^s$, initialized G, F, C.
Ensure: Learned G and classifier C
for $t = 1$ to $T_P$ do                                        ▷ $T_P$ #iterations
    Apply Eqn. 2 for Pre-train (Sec. 3.2)
    if Mod($t$,$Q$)=0 then                                     ▷ $Q$ iteration interval Upd
        Apply Eqn. 8 for Data Augmentation (Sec. 3.3)
    end if
end for
for $t = 1$ to $T_J$ do                                        ▷ $T_J$ #iterations
    Group $x_g^s, x_g^w, x, x_l$
    Apply S-Aug on $x_g^s, x_g^w, x$ and W-Aug on $x, x_l$
    Apply overall loss Eqn. 11 to train G, C
end for
```

as in our framework, and demonstrates state-of-the-art self-supervised training performance, we consider to formulate our multi-pair constraint into a consistent format, termed Multi-Pair FixMatch regularization.

Specifically, for strongly augmented data $\mathcal{D}_s^s = Aug_s(\mathcal{D}_s)$ and weakly augmented data $\mathcal{D}_s^w = Aug_w(\mathcal{D}_s)$, we generate pseudo-labels $\hat{y}^w = \arg\max{(p^w)}$ for weakly augmented sample $x^w \in \mathcal{D}_s^w$, where $p^w = \mathbf{C}(\mathbf{G}(x^w))$ is the output of classifier $\mathbf{C}$. FixMatch penalizes by assigning the pseudo-label $\hat{y}^w$ as the true label for strongly augmented sample $x^s \in \mathcal{D}_s^s$ with cross-entropy loss:

$$\mathcal{L}_{FM}(x) = -\mathbb{I}(\max(p^w) \geq \rho) \log p^s(\hat{y}^w), \tag{9}$$

where $p^s = \mathbf{C}(\mathbf{G}(x^s))$ and ρ is the threshold to decide whether $\mathcal{L}_{FM}(x)$ is applied on sample x.

Further, with augmented $\mathcal{D}_s^s$ and $\mathcal{D}_s^w$, we adopt our proposed label-free adversarial data augmentation (Figure 2 "Stage 1-2: Data Augmentation") to generate $\mathcal{D}_g^w$ and $\mathcal{D}_g^s$, respectively. This process is carried out at the same time as generating $\mathcal{D}_g$. As shown in Figure 2 "Stage 2: Multi-Pair FixMatch", we further apply strong augmentation on top of generated $\mathcal{D}_g^w, \mathcal{D}_g^s$ to derive $\mathcal{D}_g^{ws}, \mathcal{D}_g^{ss}$ as the input pairs for MPFM loss:

$$\mathcal{L}_{MPFM}(x) = -\mathbb{I}(\max(p^w) \geq \rho)(\log p_g^{ss}(\hat{y}^w) + \log p_g^{ws}(\hat{y}^w))/2, \tag{10}$$

where $p_g^{ss} = \mathbf{C}(\mathbf{G}(x_g^{ss}))$, $x_g^{ss} \in \mathcal{D}_g^{ss}$ and $p_g^{ws} = \mathbf{C}(\mathbf{G}(x_g^{ws}))$, $x_g^{ws} \in \mathcal{D}_g^{ws}$. The objective for updating our model ($\mathbf{G}$, $\mathbf{C}$) in stage1-2 is:

$$\mathcal{L}_{S2} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{FM} + \gamma \mathcal{L}_{MPFM}, \tag{11}$$

where λ,γ are balancing hyper-parameters and empirically set as 1 and 0.5, respectively. $\mathcal{L}_{CE}$ is cross-entropy loss over labeled source domain data. Our method procedure is summarized in Algorithm 1.

4 Experiments

In this section, we evaluate the effectiveness of our method as the following: Firstly, we introduce the Experimental settings. Next, we compare our method to other methods mainly from single domain generalization and semi-supervised learning. Then, we provide an extensive ablative study investigating each of our proposed modules. Last, we present the latent feature space visualization.

4.1 Experimental Settings

Datasets: We leverage three commonly used datasets from domain adaptation and generalization research field. **(1) PACS** (Li et al., 2017) is a recent challenging domain adaptation/generalization benchmark which shows larger domain discrepancy. It consists of seven object categories from four domains, namely art paintings, cartoon, sketch, and photo. For this dataset, we evaluate on two SS-SDG settings: 15 labeled samples per class (total 105 labels) and 25 labeled samples per class (total 175 labels). **(2) OfficeHome** (Venkateswara et al., 2017) contains four domains (art, clipart, product,

Table 2: mAP(%) on PACS. Named in column is source domain for training. Rest domains are the testing unseen domains. (A: Art painting, C: Cartoon, P: Photo, S: Sketch)

Method	labels: 105 (15 per class)					labels: 175 (25 per class)				
	A	C	P	S	Avg.	A	C	P	S	Avg.
ERM (sup.)	70.9	76.5	43.3	53.1	60.7	70.9	76.5	43.3	53.1	60.7
ERM	49.45	50.84	30.36	25.25	38.96	52.51	54.05	30.76	23.61	40.23
ENT-MIN	54.15	55.61	38.08	24.95	43.20	56.10	59.65	36.99	26.92	44.91
FixMatch	57.67	69.13	45.46	42.17	53.61	63.70	68.27	45.79	42.86	55.16
ADA	47.87	50.71	30.31	30.46	40.08	51.71	53.42	30.33	26.74	40.55
MEADA	48.79	52.81	34.23	30.42	41.56	51.97	54.54	32.27	28.19	41.74
Ours	**60.26**	**69.91**	**47.30**	**46.70**	**56.04**	**65.58**	**70.11**	**48.19**	**47.95**	**57.96**

and real world) with 65 classes. This is one of the canonical domain adaptation/generalization benchmarks. We design two SS-SDG settings on this dataset: 10 labeled samples per class (total 650 labels) and 15 labeled samples per class (total 975 labels). **(3) DomainNet20** is a subset of DomainNet (Peng et al., 2019). We self-construct the setting as picking 4 domains (clipart, painting, real, and sketch) and 20 classes out of the entire dataset. We adopt two SS-SDG settings: 15 labeled samples per class (total 300 labels) and 25 labeled samples per class (total 500 labels).

Baselines: We compare with two main streams of the state-of-the-art methods: (1) *Single domain generalization methods*, namely Adversarial Data Augmentation (ADA) (Volpi et al., 2018) and Maximum-Entropy Adversarial Data Augmentation (MEADA) (Zhao et al., 2020). (2) *Semi-supervised learning methods*, namely Entropy Minimization (ENT-MIN) (Yves Grandvalet, 2004) and FixMatch (Sohn et al., 2020). Besides, we consider another baseline general Empirical Risk Minimization (ERM) (Koltchinskii, 2011).

Evaluation Metrics: For each of the benchmarks, amongst all the domains defined, we iteratively take one domain as the source domain and test on all the rest domains. The mean average precision (mAP) is reported by averaging over 3 random splits of all the classes' average precision.

Implementation Details: At stage1 in Sec. 3.2 and 3.3, We adopt an ImageNet-pretrained ResNet18 as the feature extractor **G** and a 2-layer MLP head (hidden layer 512-d, with ReLU) as projection head **F**. We set memory bank size to 1600 and batch size to 32 in whole training process. At the model pre-train stage in Sec. 3.2, we use SGD optimizer with learning rate 0.0005, weight decay 0.0005 and momentum 0.9, and train for 1500 iterations. At data augmentation stage in Sec. 3.3, we adopt SGD optimizer with learning rate 50.0 and 15 iterations to maximize Equation 7. The data augmentation is involved only once at the 200-th iteration during model pre-train. At "Stage 2" in Sec. 3.4, we adopt SGD with learning rate 0.001, weight decay 0.0005, momentum 0.9, batch size 32 and train for 8500 iterations.

4.2 Performance Evaluation

PACS: Quantitative comparison is shown in Table 2. We group the methods on top three rows compared to ours on the last row. On the first row, notice that ERM (supervised) uses all the dataset labels for supervised learning hence serves as the upper bound, where it is neither in the 15 label per class setting nor in the 25 per class setting. We equip ERM with our semi-supervised domain generalization setting for the fair comparison. On the second row, there are two semi-supervised learning methods. One is canonical, the entropy minimization method, the other is recent top method FixMatch. On the third row, we present two cutting-edge domain generalization methods: Adversarial Data Augmentation (ADA) and Maximum-Entropy Adversarial Data Augmentation (MEADA). On 15 instance per class setting, we observe that our method consistently outperforms all the compared methods with significant margin. For example, we get 2.43% better than FixMatch and 14.48% better than MEADA in terms of "Avg". By checking the 25 labels per class setting, we see 2.4% and 15.92% performance gains on "Painting" compared with FixMatch and MEADA, respectively. Further, surprisingly our "Avg." number is approaching the upper bound ERM (supervised) which utilizes all the label information from the dataset within less than 4% gap, whereas our method only utilizes less than 5% (15 per class) or 8% (25 per class) of all the labels.

Table 3: mAP(%) on OfficeHome. Named in column is source domain for training. Rest domains are testing unseen domains. (A: Art, C: Clipart, P: Product, R: Real_world)

Method	labels: 650 (10 per class)					labels: 975 (15 per class)				
	A	C	P	R	Avg.	A	C	P	R	Avg.
ERM (sup.)	51.27	49.18	44.12	56.86	50.36	51.27	49.18	44.12	56.86	50.36
ERM	41.09	35.57	36.09	44.15	39.23	44.94	38.99	37.84	47.85	42.40
ENT-MIN	40.74	37.10	38.69	46.98	40.88	45.43	40.93	39.96	51.09	44.35
FixMatch	40.52	36.53	39.19	47.73	40.99	44.70	41.72	40.21	52.04	44.67
ADA	40.75	35.63	35.83	43.76	38.99	44.67	38.97	37.16	47.43	42.05
MEADA	40.74	35.80	35.86	44.15	39.14	44.97	39.41	37.14	47.71	42.31
Ours	**43.55**	**40.65**	**40.54**	**49.13**	**43.47**	**47.67**	**43.39**	**42.49**	**53.46**	**46.75**

Table 4: mAP(%) on DomainNet20. Named in column is source domain for training. Rest domains are the testing unseen domains. (C: Clipart, P: Painting, R: Real, S: Sketch)

Method	labels: 300 (15 per class)					labels: 500 (25 per class)				
	C	P	R	S	Avg.	C	P	R	S	Avg.
ERM (sup.)	59.57	64.74	55.57	55.82	58.93	59.57	64.74	55.57	55.82	58.93
ERM	44.87	52.27	39.38	42.34	44.72	48.81	54.21	44.56	47.47	48.76
ENT-MIN	45.22	53.78	46.84	42.36	47.05	50.93	53.14	49.59	49.73	50.83
FixMatch	51.45	55.27	50.17	49.56	51.61	54.23	57.42	50.35	53.26	53.72
ADA	45.28	52.11	39.69	43.10	45.05	48.93	53.68	44.82	48.74	49.04
MEADA	46.48	52.55	40.93	43.37	45.83	49.74	53.50	45.08	49.50	49.46
Ours	**53.31**	**57.52**	**52.61**	**52.91**	**54.09**	**56.21**	**59.35**	**53.38**	**56.18**	**56.28**

OfficeHome: Evaluation on OfficeHome is shown in Table 3. Similarly as PACS, we organize the experiment in the same layout, where the same three groups of the compared methods are listed on the top three rows with EMR (supervised) as the upper bound. In this evaluation, again we find that our method consistently and significantly outperforms all the compared methods on both of the two settings in Table 3. On 10 instance per class setting, we observe that our method consistently outperforms all the compared methods with significant margin, i.e., 2.8% better than second best on "Art" as source domain and 3.55% better than second best on "Clipart". The same trend is observed on 15 instances per class setting on the right column. Besides, as more instances per class is used in training, we see that the performance on the right column is generally higher than the left column setting as expected. Another interesting point is while our performance to upper bound gap is less than 7%, our setting only uses less than 18% labels of OfficeHome (10 instances per class) and 26.71% (15 instances per class). Furthermore, we still find that our label-free adversarial data augmentation based framework consistently and significantly outperforms ADA and MEADA with more than 10.0% advantage under different settings.

DomainNet20: The original DomainNet (Peng et al., 2019) is relatively a large scale domain adaptation benchmark with 6 domains and overall 345 categories. Since our setting is semi-supervised, where we only utilize a few labeled data, e.g., we constrain only 15 or 25 samples per class presenting their labels. In this way, for those categories that are with less than 15 images, we discard them from the training data. Meanwhile, to provide a similar-scale evaluation to other benchmarks such as OfficeHome and PACS, we randomly pick 4 out of 6 domains and 20 out of the 345 categories to form our training and testing data for demonstration. In Table 4, our method's performance across the 300-label and 500-label settings show clear advantage over the other methods. Compared to the most competitive opponent FixMatch, our method surpasses by 2.47% on 300-label "Avg." and 2.56% on 500-label "Avg.". Compared to the representative SDG methods, i.e., MEADA, we see 8.26% performance gain on 300-label "Avg." and 6.82% performance gain 500-label "Avg.". In this dataset, we observe that the ERM baseline performs at the same level as those single domain generalization methods, partially suggesting that this dataset is more challenging as label information other than domain discrepancy becomes more critical, where the self-supervised methods can benefit more. Notice that our method "Avg" presents very close performance to the upper bound ERM (supervised)

Table 5: Ablation study on PACS with 15 per class setting. Named in column is source domain. Rest domains are the testing unseen domains. (A: Art painting, C: Cartoon, P: Photo, S: Sketch)

PACS	$\mathcal{L}_{ce}$	$\mathcal{L}_{FM}$	$\mathcal{L}_{CL}$	$\mathcal{L}_{MPFM}$	A	C	P	S	Avg.
BL	✓				49.45	50.84	30.36	25.25	38.96
BL+S1	✓		✓		50.60	52.40	32.75	27.78	40.88
FM	✓	✓			57.67	69.13	45.46	42.17	53.61
FM+S1	✓	✓	✓		59.39	69.61	45.67	43.02	54.42
Ours	✓	✓	✓	✓	**60.26**	**69.91**	**47.30**	**46.70**	**56.04**

with less than 5% accuracy gap, but using only 7.7% (15 per class) or 12.96% (25 per class) of the overall labels, which further indicates the effectiveness of our self-supervised adversarial data augmentation design.

4.3 Ablation Study

We ablate our proposed core components in Table 5. We clearly decompose the loss terms and abbreviate the simple baseline (ERM) as "BL" and the strong baseline FixMatch as "FM". $\mathcal{L}_{CL}$ refers to our pre-training stage 1-1 and data augmentation stage 1-2. $\mathcal{L}_{MPFM}$ refers to that, in stage 1-2, we further conduct LFADA on weakly and strongly augmented data to generate data for the multi-pair loss. Compared to the naive ERM "BL", the strong baseline "FM" achieves much better performance. Our intention was not to show the progressive improvement on accuracy from "BL" to "FM" and then to "Ours". Indeed, we aim to demonstrate that proposed $\mathcal{L}_{MPFM}$ and $\mathcal{L}_{CL}$ can orthogonally improve the performance of baselines, e.g., comparing "BL" to "BL+S1", and "FM" to "FM+S1", where a clear boost of 1.8% and 0.8% is observed, respectively. Furthermore, by comparing "Ours" to "FM+S1" and "FM', we achieve another 1.6% and 2.4% accuracy gain. Each of the incremental combination demonstrates the effectiveness of the components, such as the label-free adversarial data augmentation and the Multi-Pair FixMatch, suggesting the LFADA is advantageous to handle semi-supervised single domain generalization (SS-SDG) problems.

4.4 Visualization

We visualize feature space extracted by models trained with our method and the baseline method, i.e., ERM, on PACS dataset where Sketch is the target domain. As shown in Figure 3, It appears that our model yields clear better separation of different categories. Meanwhile, we observe that for different shapes in the same color, with our method, the data points tend to cluster closer, while other methods leave those same classes but different domain data points separated. The closer clustering indicates better alignment of source domain to target domains. Our method presents clear lower domain gap than ERM, demonstrating that our method indeed continuously generalizes towards the unseen target domains.

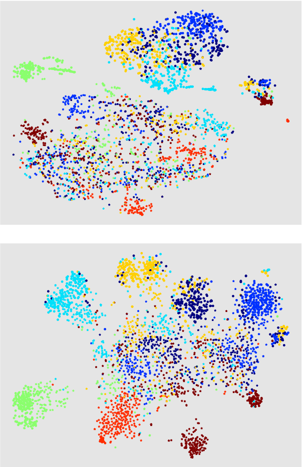

Figure 3: The t-SNE visualization of feature with ERM (up) and ours (down). Same color indicates the same category. Circle indicates source domain. Star indicates unseen target domains. Best viewed in color and zoomed in.

5 Conclusion

In this work, we propose to solve a new problem under a realistic setting, namely the semi-supervised single domain generalization, where number of domains for generalization is single and with only very limited label information of training data. We leverage expertise from self-supervised learning and propose a multi-pair FixMatch loss to mitigate the lack of label issue. Further, we newly introduce a label-free adversarial data augmentation to enrich the source domain distribution under insufficient label information scenario. Our augmentation is significantly different from traditional adversarial ones that heavily rely on label information. Extensive study across three challenging domain generalization benchmarks demonstrates our method's advantage, not only surpassing state-of-the-art methods, but also approaching the fully supervised upper bounds.

REFERENCES

David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *ICLR*, 2020.

Zhaowei Cai, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Zhuowen Tu, and Stefano Soatto. Exponential moving average normalization for self-supervised and semi-supervised learning. In *CVPR*, pp. 194–203, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pp. 1597–1607, 2020.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*, volume 32, pp. 6450–6461, 2019.

Yingjun Du, Jun Xu, Huan Xiong, Qiang Qiu, Xiantong Zhen, Cees GM Snoek, and Ling Shao. Learning to learn with variational information bottleneck for domain generalization. In *European Conference on Computer Vision*, pp. 200–216. Springer, 2020.

Xinjie Fan, Qifei Wang, Junjie Ke, Feng Yang, Boqing Gong, and Mingyuan Zhou. Adversarially adaptive normalization for single domain generalization. In *CVPR*, pp. 8208–8217, 2021.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pp. 9729–9738, 2020.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Fsdr: Frequency space domain randomization for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6891–6902, 2021.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pp. 1501–1510, 2017.

Vladimir Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: Ecole d'Eté de Probabilités de Saint-Flour XXXVIII-2008*, volume 2033. Springer Science & Business Media, 2011.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop*, volume 3, pp. 896, 2013.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, pp. 5542–5550, 2017.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018a.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *CVPR*, pp. 5400–5409, 2018b.

Lei Li, Ke Gao, Juan Cao, Ziyao Huang, Yepeng Weng, Xiaoyue Mi, Zhengze Yu, Xiaoya Li, and Boyang Xia. Progressive domain expansion network for single domain generalization. In *CVPR*, pp. 224–233, 2021.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2019.

Avital Oliver, Augustus Odena, Colin Raffel, Ekin D Cubuk, and Ian J Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *NeurIPS*, 2018.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, pp. 1406–1415, 2019.

Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In *ICML*, pp. 7728–7738. PMLR, 2020.

Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *CVPR*, pp. 12556–12565, 2020.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pp. 5389–5400. PMLR, 2019.

Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *ICLR*, 2018.

Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training. In *ICLR*, 2018.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. 2020.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, volume 30, 2017.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pp. 5018–5027, 2017.

Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *NeurIPS*, 2018.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Wenjun Zeng, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. *arXiv preprint arXiv:2103.03097*, 2021a.

Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312: 135–153, 2018.

Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. In *ICCV*, pp. 834–843, 2021b.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. In *NeurIPS*, 2020a.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, pp. 10687–10698, 2020b.

Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *CVPR*, pp. 14383–14392, 2021.

Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *arXiv preprint arXiv:2103.00550*, 2021.

Yoshua Bengio Yves Grandvalet. Semi-supervised learning by entropy minimization. In *NeurIPS2004*, 2004.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, 2021.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *NeurIPS*, 2021.

Liheng Zhang and Guo-Jun Qi. Wcp: Worst-case perturbations for semi-supervised deep learning. In *CVPR*, pp. 3912–3921, 2020.

Long Zhao, Ting Liu, Xi Peng, and Dimitris Metaxas. Maximum-entropy adversarial data augmentation for improved generalization and robustness. In *NeurIPS2020*, 2020.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *ECCV*, pp. 561–578. Springer, 2020.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *arXiv preprint arXiv:2103.02503*, 2021a.

Kaiyang Zhou, Chen Change Loy, and Ziwei Liu. Semi-supervised domain generalization with stochastic stylematch. *arXiv preprint arXiv:2106.00592*, 2021b.

Rui Zhu, Bingchen Zhao, Jingen Liu, Zhenglong Sun, and Chang Wen Chen. Improving contrastive learning by visualizing feature transformation. In *ICCV*, pp. 10306–10315, 2021.

A Appendix

This appendix provides details of datasets and additional experimental results. Section A.1 shows details about datasets used in our experiments. Section A.2 demonstrates the performance of our method with different hyper-parameter settings. Section A.3 visualizes the samples generated by our approach.

A.1 Datasets

Table 6 shows the overall descriptions of benckmark datasets, i.e., *OfficeHome* (Venkateswara et al., 2017), *PACS* (Li et al., 2017), and *DomainNet20*. Here, *DomainNet20* is a subset of *DomainNet* (Peng et al., 2019) created by ourselves, which contains 4 domains (Clipart, Painting, Real, and Sketch) with 20 categories (0-baseball bat, 1-binoculars, 2-bracelet, 3-diving board, 4-goatee, 5-hamburger, 6-hurricane, 7-knee, 8-parachute, 9-pickup truck, 10-pillow, 11-pizza, 12-sandwich, 13-saw, 14-scorpion, 15-speedboat, 16-square, 17-swing set, 18-tent, 19-trumpet). As explained in the main submission Sec. 4.2 "DomainNet20", the reason why we create a subset of original DomainNet is that, we find that many categories are with limited samples. When we consider the settings of 15 or 20 samples per class, not all the categories defined in DomainNet are valid. Moreover, to indicate the same behavior trend of our trained models, we choose to trim the dataset to at the same scale of the other two datasets, OfficeHome and PACS. In this way, we randomly select 4 domains and 20 categories to form DomainNet20.

Table 6: Statistics of the three benchmark datasets.

Datasets	Subsets	Classes	Samples
OfficeHome	Art Clipart Product Real World	65	2427 4365 4439 4357
PACS	Art Painting Cartoon Photo Sketch	7	2048 2344 1670 3929
DomainNet20	Clipart Painting Real Sketch	20	2971 2585 9673 4768

A.2 Additional Results of Our Approach

The overall loss function of our approach in "Stage 2" is:

$$\mathcal{L}_{S2} = \mathcal{L}_{ce} + \lambda\mathcal{L}_{FM} + \gamma\mathcal{L}_{MPFM}, \tag{12}$$

where λ and γ are hyper-parameters. In the following, we evaluate the performance of our approach with different hyperparameter settings.

A.2.1 Influence of λ and γ in Eq.(1)

We discuss the sensitivity of hyper-parameters λ and γ in Eq.(1) by evaluating them on *PACS* dataset where Sketch is selected as source domain. The results are shown in Fig. 4 and 5.

The value of hyper-parameter λ is selected from $\{0.1, 0.2, 0.4, 0.8, 1.0, 1.2, 1.5, 1.8, 2.0\}$ and the value of hyper-parameter γ is fixed to 0.5. We observe that our method is relatively stable in the range [0.1, 2.0] and gets better performance in the range [0.25, 0.75].

The value of hyper-parameter γ is selected from $\{0.1, 0.2, 0.4, 0.5, 0.8, 1.0, 1.2, 1.5, 1.8, 2.0\}$ and the value of λ is fixed to 1.0. We find that the classification accuracy is relatively stable in the range [0.75, 1.5]. When γ is out of the range [0.75, 1.5], the performance is slightly degraded.

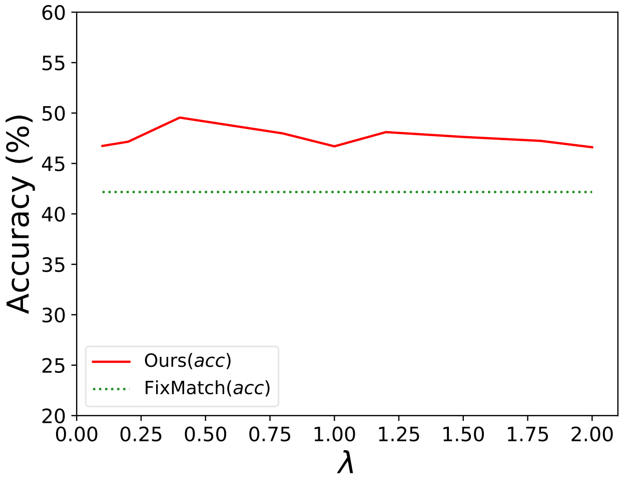

Figure 4: Accuracy versus different values of λ on *PACS* dataset by selecting Sketch as source domain, respectively.

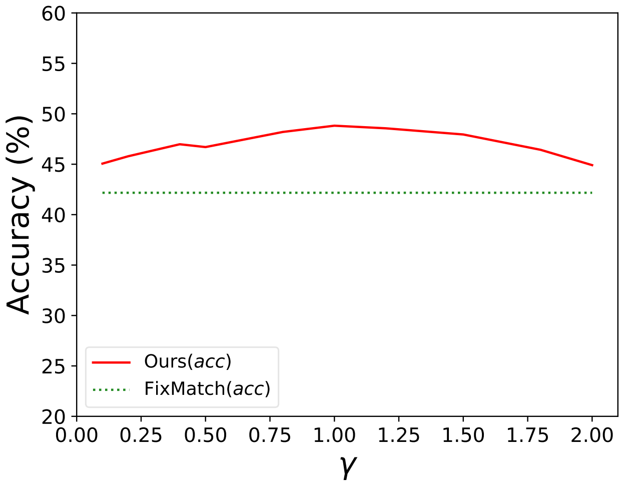

Figure 5: Accuracy versus different values of γ on *PACS* dataset by selecting Sketch as source domain, respectively.

A.2.2 Number of Augmentations in "Stage 1"

We study the effect of the hyper-parameter: the number of augmentations in "Stage 1", on *PACS* dataset where Sketch is adopted as source domain. We plot the accuracy curve under different augmentation times. As shown in Fig. 6, we find that the accuracy reaches the best value when augmenting twice and the accuracy is slowly getting worse with further increased the number of augmentation.

A.2.3 When to Do Augmentation in "Stage 1"

We explore the influence of when to augment in "Stage 1" on *PACS* dataset where Sketch is used as source domain. The experimental results are reported in Fig. 7. Here, the maximum iteration in "Stage 1" is set to 1500. It can be observed that the performance of our method is improving with augmentation being executed later in "Stage 1". Because a more well pre-trained model in "Stage 1" benefits more on our data augmentation process. After 800 iterations, the choice of different iterations

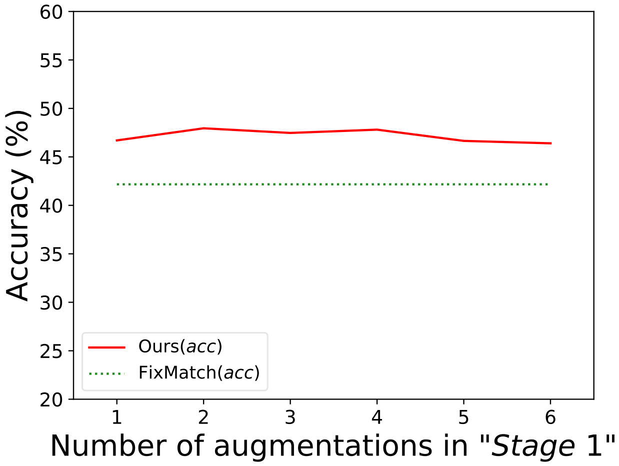

Figure 6: Accuracy versus different number of augmentations in “Stage 1” on *PACS* dataset by using Sketch as source domain.

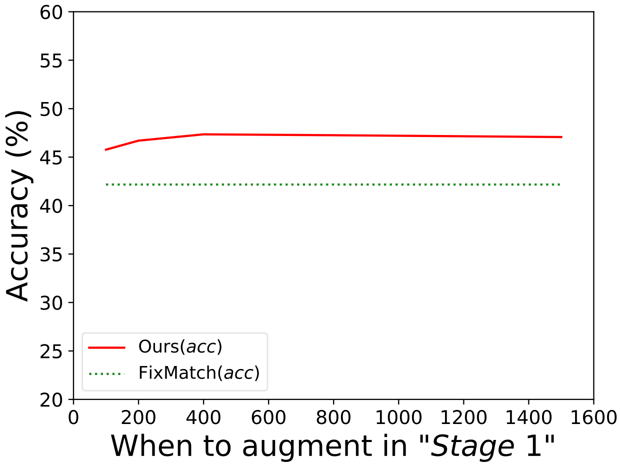

Figure 7: Accuracy versus when to augment in “Stage 1” on *PACS* dataset by choosing Sketch as source domain.

for data augmentation is very close, demonstrating that the pre-trained model in “Stage 1” after 800 iterations is already converged and thus does not change much.

A.2.4 Number of iteration for “Stage 1”

We investigate into on which iteration the pre-trained model halts and is then utilized in our label-free adversarial data augmentation training. Fig. 8, we observe that there are two peaks, one is at iteration 3000, the other is at iteration 6000. In general, our method is relatively insensitive to the pre-trained model, i.e., it is very stable to use the pre-trained model from different training iterations.

A.3 Visualization

We see in Fig. 9 besides the overlap of generated and original data, there is a clear portion of non-overlapped generated data (green dots on right). We also visualize that in fact the original and the generated samples are distant (color connected), which sufficiently guarantees the data diversity enrichment.

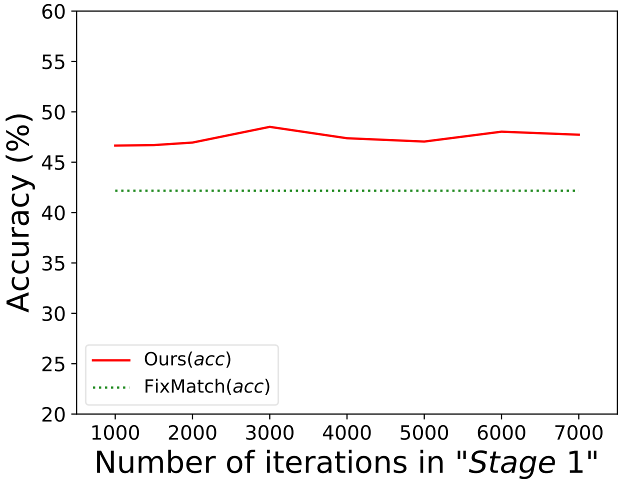

Figure 8: Accuracy versus different total iterations in “Stage 1” on *PACS* dataset by selecting Sketch as source domain.

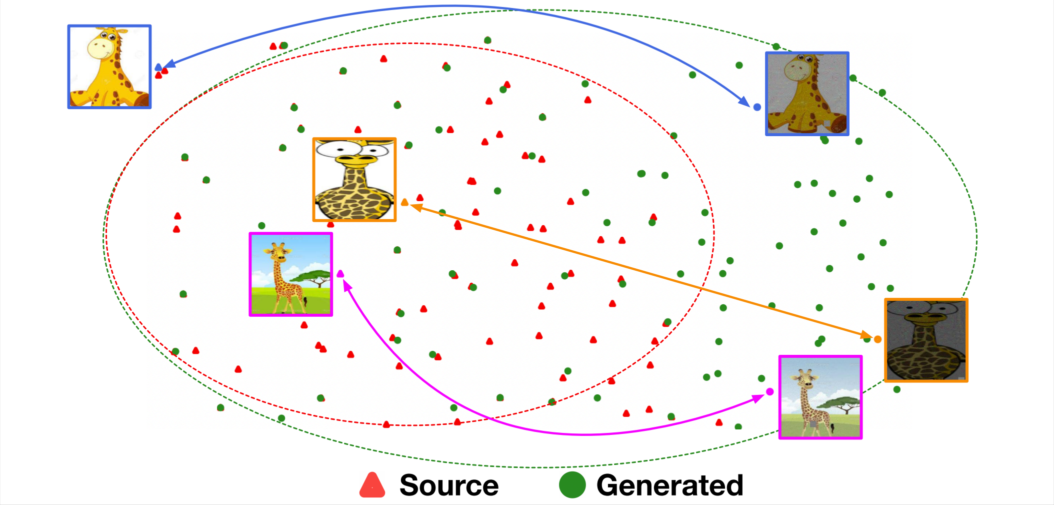

Figure 9: Visualization of features of source and generated samples (”Giraffe” category in Cartoon domain of PACS).