# Learning to Learn Single Domain Generalization

Fengchun Qiao
University of Delaware
fengchun@udel.edu

Long Zhao
Rutgers University
lz311@cs.rutgers.edu

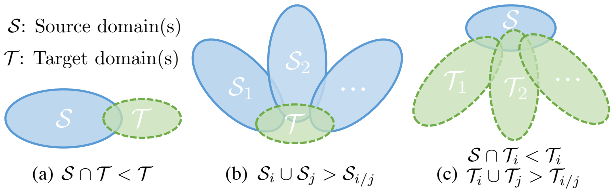
Xi Peng
University of Delaware
xipeng@udel.edu

## Abstract

*We are concerned with a worst-case scenario in model generalization, in the sense that a model aims to perform well on many unseen domains while there is only one single domain available for training. We propose a new method named adversarial domain augmentation to solve this Out-of-Distribution (OOD) generalization problem. The key idea is to leverage adversarial training to create "fictitious" yet "challenging" populations, from which a model can learn to generalize with theoretical guarantees. To facilitate fast and desirable domain augmentation, we cast the model training in a meta-learning scheme and use a Wasserstein Auto-Encoder (WAE) to relax the widely used worst-case constraint. Detailed theoretical analysis is provided to testify our formulation, while extensive experiments on multiple benchmark datasets indicate its superior performance in tackling single domain generalization.*

## 1. Introduction

Recent years have witnessed rapid deployment of machine learning models for broad applications [17, 42, 3, 60]. A key assumption underlying the remarkable success is that the training and test data usually follow similar statistics. Otherwise, even strong models (*e.g.,* deep neural networks) may break down on unseen or Out-of-Distribution (OOD) test domains [2]. Incorporating data from multiple training domains somehow alleviates this issue [21], however, this may not always be applicable due to data acquiring budget or privacy issue. An interesting yet seldom investigated problem then arises: Can a model generalize from one source domain to many unseen target domains? In other words, how to maximize the model generalization when there is only a single domain available for training?



(a) $\mathcal{S} \cap \mathcal{T} < \mathcal{T}$ (b) $\mathcal{S}_i \cup \mathcal{S}_j > \mathcal{S}_{i/j}$ (c) $\mathcal{S} \cap \mathcal{T}_i < \mathcal{T}_i$ $\mathcal{T}_i \cup \mathcal{T}_j > \mathcal{T}_{i/j}$

Figure 1. The domain discrepancy: (a) domain adaptation, (b) domain generalization, and (c) single domain generalization.

The discrepancy between source and target domains, also known as domain or covariate variant [48], has been intensively studied in *domain adaptation* [30, 33, 57, 24] and *domain generalization* [32, 9, 22, 4]. Despite of their various success in tackling ordinary domain discrepancy issue, however, we argue that existing methods can hardly succeed in the aforementioned single domain generalization problem. As illustrated in Fig. 1, the former usually expects the availability of target domain data (either labeled or unlabeled); While the latter, on the other hand, always assumes multiple (rather than one) domains are available for training. This fact emphasizes the necessity to develop a new learning paradigm for *single domain generalization*.

In this paper, we propose *adversarial domain augmentation* (Sec. 3.1) to solve this challenging task. Inspired by the recent success of adversarial training [35, 50, 49, 36, 24], we cast the single domain generalization problem in a worst-case formulation [44, 20]. The goal is to use single source domain to generate "fictitious" yet "challenging" populations, from which a model can learn to generalize with theoretical guarantees (Sec. 4).

However, technical barriers exist when applying adversarial training for domain augmentation. On the one hand, it is hard to create "fictitious" domains that are largely different from the source, due to the contradiction of semantic consistency constraint [11] in worst-case formulation. On the other hand, we expect to explore many "fictitious" domains to guarantee sufficient coverage, which may result in significant computational overhead. To circumvent these barriers, we propose to *relax the worst-case constraint* (Sec. 3.2) via a Wasserstein Auto-Encoder (WAE) [52] to encourage large domain transportation in the input space. Moreover, rather than learning a series of ensemble models [56], we organize adversarial domain augmentation via *meta-learning* [6] (Sec. 3.3), yielding a highly efficient model with improved single domain generalization.

[1] The source code and pre-trained models are publicly available at: https://github.com/joffery/M-ADA.

The primary contribution of this work is a meta-learning based scheme that enables single domain generalization, an important yet seldom studied problem. We achieve the goal by proposing adversarial domain augmentation, while at the same time, relaxing the widely used worst-case constraint. We also provide detailed theoretical understanding to testify our solution. Extensive experiments indicate that our method marginally outperforms state of the art in single domain generalization of benchmark datasets including *Digits*, *CIFAR-10-C* [14], and *SYTHIA* [37].

## 2. Related Work

**Domain discrepancy**: Domain discrepancy brought by domain or covariance shifts [48] severely degrades the model performance on cross-domain recognition. The models trained using Empirical Risk Minimization [16] usually perform poorly on unseen domains. To reduce the discrepancy across domains, a series of methods are proposed for unsupervised [33, 43, 7, 38, 39] or supervised domain adaptation [31, 57]. Some recent work also focused on few-shot domain adaptation [30] where only a few labeled samples from target domain are involved in training.

Different from domain adaptation, domain generalization aims to learn from multiple source domains without any access to target domains. Most previous methods either tried to learn a domain-invariant space to align domains [32, 9, 12, 21, 59] or aggregate domain-specific modules [29, 28]. Recently, Carlucci *et al.* [4] solved this problem by jointly learning from supervised and unsupervised signals from images. In data level, gradient-based domain perturbation [41] and adversarial training methods [56] are proposed to improve generalization. In particular, [56] is designed for single domain generalization and achieves better performance through an ensemble model. Compared to [56], we aim at creating large domain transportation for "fictitious" domains and devising a more efficient meta-learning scheme within a single unified model.

**Adversarial training**: Adversarial training [11] is proposed for improving model robustness against adversarial perturbations or attacks. Madry *et al.* [27] provided evidence that deep neural networks is capable of resistant to adversarial attacks through reliable adversarial training methods. Further, Sinha *et al.* [44] proposed principled adversarial training through the lens of distributionally robust optimization. More recently, Stutz *et al.* [47] pointed out that on-manifold adversarial training boosts generalization, and hence models with both robustness and generalization can be obtained at the same time. Peng *et al.* [35] proposed to learn robust models via perturbed examples. In our work, we generate "fictitious" domains through adversarial training to improve single domain generalization.

**Meta-learning**: Meta-learning [40, 51] is a long standing topic in how to learn new concepts or tasks fast with a few training examples. It has been widely used in optimization of deep neural networks [1, 23] and few-shot classification [15, 55, 46]. Recently, Finn *et al.* [6] proposed a Model-Agnostic Meta-Learning (MAML) procedure for few-shot learning and reinforcement learning. The objective of MAML is to find a good initialization which can be fast adapted to new tasks within few gradient steps. Li *et al.* [22] proposed a MAML-based approach to solve domain generalization. Balaji *et al.* [2] proposed to learn an adaptive regularizer through meta-learning for cross-domain recognition. However, neither of them is applicable for single domain generalization. Instead, in this paper, we propose a MAML-based meta-learning scheme to efficiently train models on "fictitious" domains for single domain generalization. We show that the learned model is robust to unseen target domains while it can also be easily leveraged for few-shot domain adaptation.

## 3. Method

We aim at solving the problem of single domain generalization: A model is trained on only one source domain $\mathcal{S}$ but is expected to generalize well on many unseen target domains $\mathcal{T}$. A promising solution of this challenging problem, inspired by many recent achievements [36, 56, 24], is to leverage adversarial training [11, 49]. The key idea is to learn a robust model that is resistant to out-of-distribution perturbations. More specifically, we can learn the model by solving a worst-case problem [44]:

$$\min_{\theta} \sup_{\mathcal{T}:D(\mathcal{S},\mathcal{T})\leq\rho} \mathbb{E}[\mathcal{L}_{\text{task}}(\theta;\mathcal{T})], \tag{1}$$

where $D$ is a similarity metric to measure the domain distance and $\rho$ denotes the largest domain discrepancy between $\mathcal{S}$ and $\mathcal{T}$. $\theta$ are model parameters that are optimized according to a task-specific objective function $\mathcal{L}_{\text{task}}$. Here, we focus on classification problems using cross-entropy loss:

$$\mathcal{L}_{\text{task}}(\mathbf{y},\hat{\mathbf{y}}) = -\sum_i y_i \log(\hat{y}_i), \tag{2}$$

where $\hat{\mathbf{y}}$ is *softmax* output of the model; $\mathbf{y}$ is the one-hot vector representing the ground truth class; $y_i$ and $\hat{y}_i$ represent the $i$-th dimension of $\mathbf{y}$ and $\hat{\mathbf{y}}$, respectively.

Following the worst-case formulation (1), we propose a new method, *Meta-Learning based Adversarial Domain Augmentation* (M-ADA), for single domain generalization. Fig. 2 presents an overview of our approach. We create "fictitious" yet "challenging" domains by leverage adversarial training to augment the source domain in Sec. 3.1. The task model learns from the domain augmentations with the assistance of a Wasserstein Auto-Encoder (WAE), which relaxes the worst-case constraint in Sec. 3.2. We organize the joint training of task model and WAE, as well as the domain augmentation procedure, in a learning to learn framework as

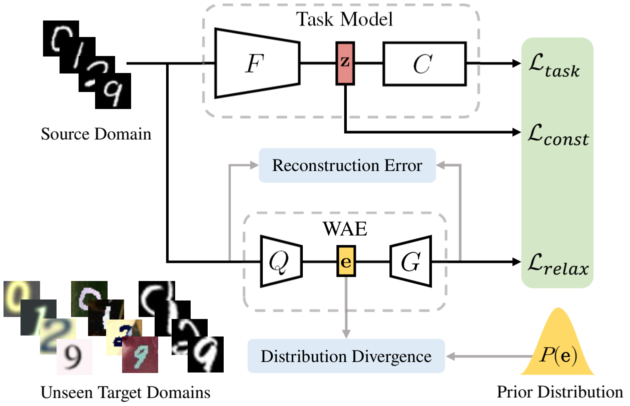


Figure 2. Overview of adversarial domain augmentation.

described in Sec. 3.3. Finally, we present theoretical analysis to prove the worst-case guarantee in Sec. 4.

### 3.1. Adversarial Domain Augmentation

Our goal is to create multiple augmented domains from the source domain. Augmented domains are required to be distributionally different from the source domain so as to mimic unseen domains. In addition, to avoid divergence of augmented domains, the worst-case guarantee defined in Eq. (1) should also be satisfied.

To achieve this goal, we propose Adversarial Domain Augmentation. Our model consists of a task model and a WAE shown in Fig. 2. In Fig. 2, the task model consists of a feature extractor $F : \mathcal{X} \rightarrow \mathcal{Z}$ mapping images from input space to embedding space, and a classifier $C : \mathcal{Z} \rightarrow \mathcal{Y}$ used to predict labels from embedding space. Let $\mathbf{z}$ denote the latent representation of $\mathbf{x}$ which is obtained by $\mathbf{z} = F(\mathbf{x})$. The overall loss function is formulated as follows:

$$\mathcal{L}_{\text{ADA}} = \underbrace{\mathcal{L}_{\text{task}}(\theta;\mathbf{x})}_{\text{Classification}} - \alpha \underbrace{\mathcal{L}_{\text{const}}(\theta;\mathbf{z})}_{\text{Constraint}} + \beta \underbrace{\mathcal{L}_{\text{relax}}(\psi;\mathbf{x})}_{\text{Relaxation}}, \tag{3}$$

where $\mathcal{L}_{\text{task}}$ is the classification loss defined in Eq. (2), $\mathcal{L}_{\text{const}}$ is the worst-case guarantee defined in Eq. (1), and $\mathcal{L}_{\text{relax}}$ guarantees large domain transportation defined in Eq. (7). $\psi$ are parameters of the WAE. $\alpha$ and $\beta$ are two hyper-parameter to balance $\mathcal{L}_{\text{const}}$ and $\mathcal{L}_{\text{relax}}$.

Given the objective function $\mathcal{L}_{\text{ADA}}$, we employ an iterative way to generate the adversarial samples $\mathbf{x}^+$ in the augmented domain $\mathcal{S}^+$:

$$\mathbf{x}^+_{t+1} \leftarrow \mathbf{x}^+_t + \gamma \nabla_{\mathbf{x}^+_t} \mathcal{L}_{\text{ADA}}(\theta, \psi; \mathbf{x}^+_t, \mathbf{z}^+_t), \tag{4}$$

where $\gamma$ is the learning rate of gradient ascent. A small number of iterations are required to produce sufficient perturbations and create desirable adversarial samples.

$\mathcal{L}_{\text{const}}$ imposes semantic consistency constraint to adversarial samples so that $\mathcal{S}^+$ satisfies $D(\mathcal{S}, \mathcal{S}^+) \leq \rho$. More specifically, we follow [56] to measure the Wasserstein distance between $\mathcal{S}^+$ and $\mathcal{S}$ in the embedding space:

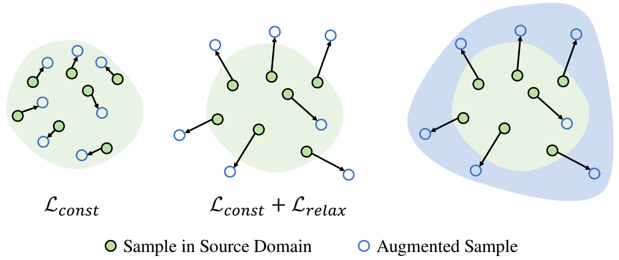


Figure 3. Motivation of $\mathcal{L}_{\text{relax}}$. **Left:** The augmented samples may be close to the source domain if applying $\mathcal{L}_{\text{const}}$. **Middle:** We expect to create out-of-domain augmentations by incorporating $\mathcal{L}_{\text{relax}}$. **Right:** This would yield an enlarged training domain.

$$\mathcal{L}_{\text{const}} = \frac{1}{2}\|\mathbf{z} - \mathbf{z}^+\|_2^2 + \infty \cdot \mathbf{1}\left\{\mathbf{y} \neq \mathbf{y}^+\right\}, \tag{5}$$

where $\mathbf{1}\{\cdot\}$ is the 0-1 indicator function and $\mathcal{L}_{\text{const}}$ will be $\infty$ if the class label of $\mathbf{x}^+$ is different from $\mathbf{x}$. Intuitively, $\mathcal{L}_{\text{const}}$ controls the ability of generalization outside the source domain measured by Wasserstein distance [54]. However, $\mathcal{L}_{\text{const}}$ yields limited domain transportation since it severely constrains the semantic distance between the samples and their perturbations. Hence, $\mathcal{L}_{\text{relax}}$ is proposed to relax the semantic consistency constraint and create large domain transportation. The implementation of $\mathcal{L}_{\text{relax}}$ is discussed in Sec. 3.2.

### 3.2. Relaxation of Wasserstein Distance Constraint

Intuitively, we expect the augmented domains $\mathcal{S}^+$ are largely different from the source domain $\mathcal{S}$. In other words, we want to maximize the domain discrepancy between $\mathcal{S}^+$ and $\mathcal{S}$. However, the semantic consistency constraint $\mathcal{L}_{\text{const}}$ would severely limits the domain transportation from $\mathcal{S}$ to $\mathcal{S}^+$, posing new challenges to generate desirable $\mathcal{S}^+$. To address this issue, we propose $\mathcal{L}_{\text{relax}}$ to encourage out-of-domain augmentations. We illustrate the idea in Fig. 3.

Specifically, we employ Wasserstein Auto-Encoders (WAEs) [52] to implement $\mathcal{L}_{\text{relax}}$. Let $V$ denote the WAE parameterized by $\psi$. $V$ consists of an encoder $Q(\mathbf{e}|\mathbf{x})$ and a decoder $G(\mathbf{x}|\mathbf{e})$ where $\mathbf{x}$ and $\mathbf{e}$ denote inputs and bottleneck embedding, respectively. Additionally, we use a distance metric $\mathcal{D}_{\mathbf{e}}$ to measure the divergence between $Q(\mathbf{x})$ and a prior distribution $P(\mathbf{e})$, which can be implemented as either *Maximum Mean Discrepancy* (MMD) or GANs [10]. We can learn $V$ by optimizing:

$$\min_{\psi} [\|G(Q(\mathbf{x})) - \mathbf{x}\|^2 + \lambda \mathcal{D}_{\mathbf{e}}(Q(\mathbf{x}), P(\mathbf{e}))], \tag{6}$$

where $\lambda$ is a hyper-parameter. After pre-training $V$ on the source domain $S$ offline, we keep it frozen and maximize

**Algorithm 1:** The proposed Meta-Learning based Adversarial Domain Augmentation (M-ADA).

```
Input: Source domain $\mathcal{S}$; Pre-train WAE $V$ on $\mathcal{S}$;
       Number of augmented domains $K$
Output: Learned model parameters $\theta$
for $k = 1, ..., K$ do
    Generate $\mathcal{S}_k^+$ from $\mathcal{S} \cup \{\mathcal{S}_i^+\}_{i=1}^{k-1}$ using Eq. (4)
    Re-train $V$ with $\mathcal{S}_k^+$
    Meta-train: Evaluate $\mathcal{L}_{\text{task}}(\theta; \mathcal{S})$ w.r.t. $\mathcal{S}$
    Compute $\hat{\theta}$ using Eq. (8)
    for $i = 1, ..., k$ do
        Meta-test: Evaluate $\mathcal{L}_{\text{task}}(\hat{\theta}; \mathcal{S}_i^+))$ w.r.t. $\mathcal{S}_i^+$
    end
    Meta-update: Update $\theta$ using Eq. (9)
end
```

the reconstruction error $\mathcal{L}_{\text{relax}}$ for domain augmentation:

$$\mathcal{L}_{\text{relax}} = \|\mathbf{x}^+ - V(\mathbf{x}^+)\|^2. \tag{7}$$

Different from Vanilla or Variation Auto-Encoders [45], WAEs employ the Wasserstein metric to measure the distribution distance between the input and reconstruction. Hence, the pre-trained $V$ can better capture the distribution of the source domain and maximizing $\mathcal{L}_{\text{relax}}$ creates large domain transportation. Comparison of different $\mathcal{L}_{\text{relax}}$ is also provided in the supplementary.

In this work, $V$ acts as a *one-class discriminator* to distinguish whether the augmentation is outside the source domain, which is significantly different from the traditional discriminator of GANs [10]. And it is also different from the domain classifier widely used in domain adaptation [24], since there is only one source domain available. As a result, $\mathcal{L}_{\text{relax}}$ together with $\mathcal{L}_{\text{const}}$ are used to "push away" $\mathcal{S}^+$ in input space and "pull back" $\mathcal{S}^+$ in the embedding space simultaneously. In Sec. 4, we show that $\mathcal{L}_{\text{relax}}$ and $\mathcal{L}_{\text{const}}$ are derivations of two Wasserstein distance metrics defined in the input space and embedding space, respectively.

### 3.3. Meta-Learning Single Domain Generalization

To efficiently organize the model training on the source domain $S$ and augmented domains $\mathcal{S}^+$, we leverage a meta-learning scheme to train a single model. To mimic real domain-shifts between the source domain $S$ and target domain $T$, at each learning iteration, we perform meta-train on the source domain $\mathcal{S}$ and meta-test on all augmented domains $\mathcal{S}^+$. Hence, after many iterations, the model is expected to achieve good generalization on the final target domain $\mathcal{T}$ during evaluation.

Formally, the proposed Meta-Learning based Adversarial Domain Augmentation (M-ADA) approach consists of three parts in each iteration during the training procedure: meta-train, meta-test and meta-update. In meta-train, $\mathcal{L}_{\text{task}}$ is computed on samples from the source domain $\mathcal{S}$, and the model parameters $\theta$ is updated via one or more gradient steps with a learning rate of $\eta$:

$$\hat{\theta} \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{task}}(\theta; \mathcal{S}). \tag{8}$$

Then we compute $\mathcal{L}_{\text{task}}(\hat{\theta}; \mathcal{S}_k^+)$ on each augmented domain $\mathcal{S}_k^+$ in meta-test. At last, in meta-update, we update $\theta$ by the gradients calculated from a combined loss where meta-train and meta-test are optimised simultaneously:

$$\theta \leftarrow \theta - \eta \nabla_\theta [\mathcal{L}_{\text{task}}(\theta; \mathcal{S}) + \sum_{k=1}^{K} \mathcal{L}_{\text{task}}(\hat{\theta}; \mathcal{S}_k^+)], \tag{9}$$

where $K$ is the number of augmented domains.

The entire training pipeline is summarized in Alg. 1. Our method has following merits. First, in contrast to prior work [56] that learns a series of ensemble models, our method achieves a single model for efficiency. In Sec. 5.4, we prove that M-ADA outperforms [56] marginally in terms of memory, speed and accuracy. Second, the meta-learning scheme prepares the learned model for fast adaptation: One or a small number of gradient steps will produce improved behavior on a new target domain. This enables M-ADA for *few-shot domain adaptation* as shown in Sec 5.5.

## 4. Theoretical Understanding

We provide a detailed theoretical analysis of the proposed Adversarial Domain Augmentation. Specifically, we show that the overall loss function defined in Eq. (3) is a direct derivation of a relaxed worst-case problem.

Let $c : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}_+ \cup \{\infty\}$ be the "cost" for an adversary to perturb $\mathbf{z}$ to $\mathbf{z}^+$ in the embedding space. Let $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+ \cup \{\infty\}$ be the "cost" for an adversary to perturb $\mathbf{x}$ to $\mathbf{x}^+$ in the input space. The Wasserstein distances between $\mathcal{S}$ and $\mathcal{S}^+$ can be formulated as: $W_c(\mathcal{S}, \mathcal{S}^+) := \inf_{M_\mathbf{z} \in \Pi(\mathcal{S}, \mathcal{S}^+)} \mathbb{E}_{M_\mathbf{z}} [c(\mathbf{z}, \mathbf{z}^+)]$ and $W_d(\mathcal{S}, \mathcal{S}^+) := \inf_{M_\mathbf{x} \in \Pi(\mathcal{S}, \mathcal{S}^+)} \mathbb{E}_{M_\mathbf{x}} [d(\mathbf{x}, \mathbf{x}^+)]$, where $M_\mathbf{z}$ and $M_\mathbf{x}$ are measures in the embedding and input space, respectively; $\Pi(\mathcal{S}, \mathcal{S}^+)$ is the joint distribution of $\mathcal{S}$ and $\mathcal{S}^+$. Then, the relaxed worst-case problem can be formulated as:

$$\theta^* = \min_\theta \sup_{\mathcal{S}^+ \in \mathcal{D}} \mathbb{E}[\mathcal{L}_{\text{task}}(\theta; \mathcal{S}^+)], \tag{10}$$

where $\mathcal{D} = \{\mathcal{S}^+ : W_c(\mathcal{S}, \mathcal{S}^+) \leq \rho, W_d(\mathcal{S}, \mathcal{S}^+) \geq \eta\}$. We note that $\mathcal{D}$ covers a robust region that is within $\rho$ distance of $\mathcal{S}$ in the embedding space and $\eta$ distance away from $\mathcal{S}$ in the input space under the Wasserstein distance measures $W_c$ and $W_d$, respectively.

For deep neural networks, Eq. (10) is intractable with arbitrary $\rho$ and $\eta$. Consequently, we consider its Lagrangian relaxation with fixed penalty parameters $\alpha \geq 0$ and $\beta \geq 0$:

$$\min_\theta \{\sup_{\mathcal{S}^+} \{\mathbb{E}[\mathcal{L}_{\text{task}}(\theta; \mathbf{x}^+)] - W_{c,d}\} = \mathbb{E}[\phi_{\alpha,\beta}(\theta, \psi; \mathbf{x})]\},$$

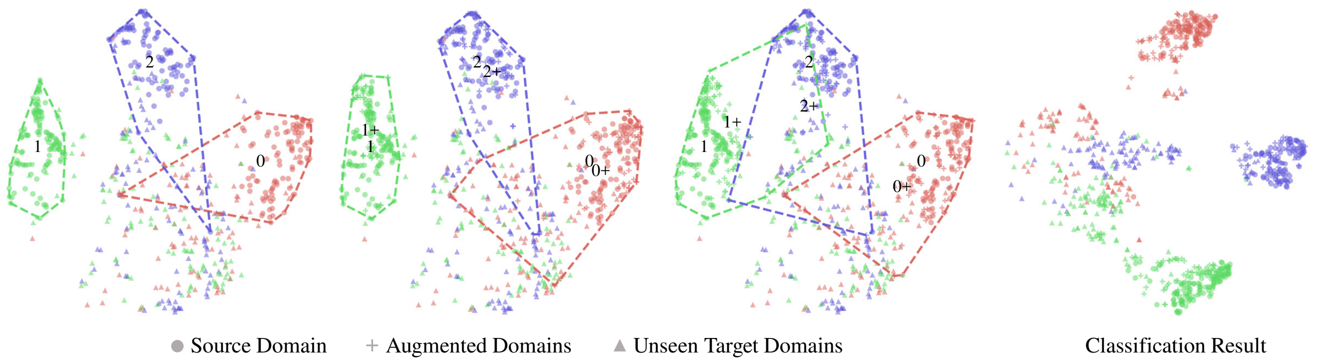


Figure 4. Visualization of domains and convex hulls in the embedding space (the first three figures) and classification space (the last figure). **From left to right**: (a) source domain $\mathcal{S}$ and unseen target domains $\mathcal{T}$; (b) augmented domains $\mathcal{S}^+$ w/o $\mathcal{L}_{\text{relax}}$; (c) $\mathcal{S}^+$ w/ $\mathcal{L}_{\text{relax}}$; (d) the classification result of M-ADA. Different colors denote different categories. The numbers mark the corresponding cluster centers. Note that **1**: cluster center of $\mathcal{S}$; **1+**: cluster center of $\mathcal{S}^+$. Best viewed in color and zoom in for details.

加强腐蚀和扰动的鲁棒性

and we have $W_{c,d}(\mathcal{S},\mathcal{S}^+) = \alpha W_c(\mathcal{S},\mathcal{S}^+) - \beta W_d(\mathcal{S},\mathcal{S}^+)$, $\phi_{\alpha,\beta}(\theta,\psi;\mathbf{x}) = \sup_{\mathbf{x}^+}\{\mathcal{L}_{\text{task}}(\theta;\mathbf{x}^+) - \mathcal{L}_{c,d}\}$, and $\mathcal{L}_{c,d} = \alpha c(\mathbf{z},\mathbf{z}^+) - \beta d(\mathbf{x},\mathbf{x}^+)$. Thus the problem in Eq. (10) is transformed to minimize the robust surrogate $\phi_{\alpha,\beta}$.

According to [44], $\phi_\alpha$ is smooth w.r.t. $\theta$ if $\alpha$ is large enough and the assumption of Lipschitzian smoothness holds. Since $\psi$ and $\theta$ are independent with each other, $\phi_{\alpha,\beta}$ is still smooth w.r.t. $\theta$. The gradient can be computed as:

$$\nabla_\theta \phi_{\alpha,\beta}(\theta,\psi;\mathbf{x}) = \nabla_\theta \mathcal{L}_{\text{task}}(\theta;\mathbf{x}^\star(\mathbf{x},\theta,\psi)),$$

where $\mathbf{x}^\star(\mathbf{x},\theta,\psi) = \arg\max_{\mathbf{x}^+}[\mathcal{L}_{\text{task}}(\theta;\mathbf{x}^+) - \mathcal{L}_{c,d}] = \arg\max_{\mathbf{x}^+}\mathcal{L}_{\text{ADA}}(\theta,\psi;\mathbf{x}^+,\mathbf{z}^+)$, which is exactly the adversarial perturbation defined in Eq. (3).

## 5. Experiments

We begin by introducing the experimental setups and implementation details in Secs. 5.1 and 5.2, respectively. In Sec. 5.3, we carry out detailed ablation study to validate the strength of the proposed relaxation, the efficiency of meta-learning scheme, and the selection and trade-off of key hyperparameters. In Sec. 5.4, we compare M-ADA with state of the arts on benchmark datasets. In Sec. 5.5, we further evaluate M-ADA in *few-shot domain adaptation*.

### 5.1. Datasets and Settings

**Datasets and settings:** (1) *Digits* consists of five subdatasets: MNIST [19], MNIST-M [8], SVHN [34], SYN [8], and USPS [5], and each of them can be viewed as a different domain. Each image in these datasets contains one single digit with different styles. This dataset is mainly employed for ablation studies. We use the first 10,000 samples in the training set of MNIST for training, and evaluate models on all other domains. (2) *CIFAR-10-C* [14] is a robustness benchmark consisting of 19 corruptions types with five levels of severities applied to the test set of CIFAR-10. The corruptions come from four main categories: noise, blur, weather and digital. Each corruption has five-level severities and "5" indicates the most corrupted one. All the models are trained on CIFAR-10 and evaluated on CIFAR-10-C. (3) *SYTHIA* [37] is a dataset synthesized for semantic segmentation in the context of driving scenarios. This dataset consists of the same traffic situation but under different locations (Highway, New York-like City and Old European Town are selected) and different weather/illumination/season conditions (Dawn, Fog, Night, Spring and Winter are selected). Following the protocol in [56], we only use the images from the left front camera and 900 images are randomly sample from each source domain.

**Evaluation metrics:** For Digits and CIFAR-10-C, we compute the mean accuracy on each unseen domain. For CIFAR-10-C, accuracy may not be sufficient to comprehensively evaluate the performance of models without measuring relative gain over baseline models (ERM [16]) and relative error evaluated on the *clean* dataset, *i.e.*, the test set of CIFAR-10 without any corruption. Inspired by the robustness metrics proposed in [14], two metrics are formulated to evaluate the robustness against image corruptions in the context of domain generalization: mean Corruption Error (mCE) and Relative mCE (RmCE). They are defined as: $\text{mCE} = \frac{1}{N}\sum_{i=1}^{N} E_i^f / E_i^{\text{ERM}}$, $\text{RmCE} = \frac{1}{N}\sum_{i=1}^{N}(E_i^f - E_{\text{clean}}^f)/(E_i^{\text{ERM}} - E_{\text{clean}}^{\text{ERM}})$, where $N$ is the number of corruptions. mCE is used for evaluating the robustness of the classifier $f$ compared with ERM [16]. RmCE measures the relative robustness compared with the *clean* data. For SYTHIA, we compute the standard mean Intersection Over Union (mIoU) on each unseen domain.

### 5.2. Implementation Details

**Task models:** We design specific task models and employ different training strategies accordingly for the three datasets. Please refer to the supplementary material for

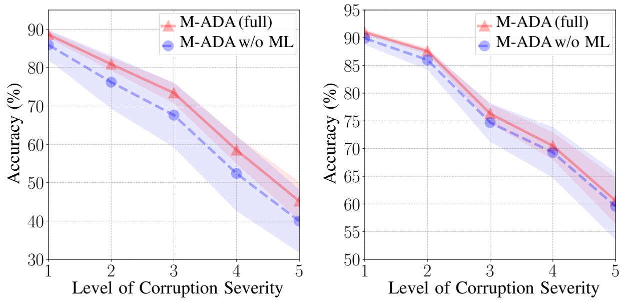


Figure 5. Validation of meta-learning scheme. Five levels of severity on *Impulse Noise* (**left**) and *Shot Noise* (**right**) are evaluated.

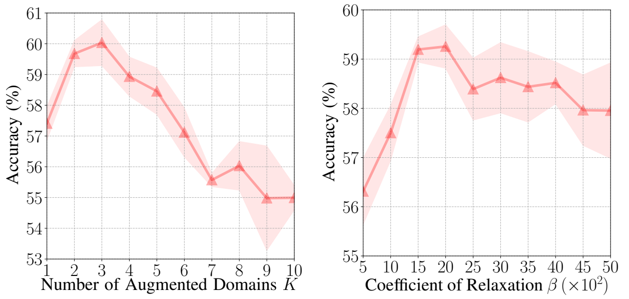


Figure 6. Hyper-parameter tuning of $K$ and $\beta$. We set $K = 3$ and $\beta = 2.0 \times 10^3$ according to the best classification accuracy.

| Method | # of params. | Inference time | Accuracy |
|---|---|---|---|
| GUD [56] | 31.9M | 22.1ms | 55.8% |
| **M-ADA** (full) | 4.54M | 3.07ms | 59.5% |

Table 1. Efficiency comparison in single domain generalization. GUD has to learn a series of ensemble models. M-ADA leverages meta-learning scheme to achieve a single model. M-ADA outperforms GUD marginally in terms of memory, speed, and accuracy.

| Method | Level 1 | Level 2 | Level 3 | Level 4 | Level 5 |
|---|---|---|---|---|---|
| ERM [16] | 87.8±0.1 | 81.5±0.2 | 75.5±0.4 | 68.2±0.6 | 56.1±0.8 |
| GUD [56] | 88.3±0.6 | 83.5±2.0 | 77.6±2.2 | 70.6±2.3 | 58.3±2.5 |
| **M-ADA** (full) | **90.5±0.3** | **86.8±0.4** | **82.5±0.6** | **76.4±0.9** | **65.6±1.2** |
| ↑ to ERM | 3.08% | 6.50% | 9.27% | 12.0% | 16.9% |
| ↑ to GUD | 2.49% | 3.95% | 6.31% | 8.22% | 12.5% |

Table 2. Accuracy comparison (%) on *CIFAR-10-C*. Boosts (↑) become more significant as corruption severity level (**1-5**) increases.

more details. For *Digits* dataset, we use a ConvNet [18] with architecture *conv-pool-conv-pool-fc-fc-softmax*. All images are resized to 32×32, and the channels of MNIST and USPS are duplicated to make them as RGB images. We use Adam with the learning rate $\eta = 0.0001$. The batch size is 32 and the total number of iterations is 10,000. For *CIFAR-10-C*, we use Wide Residual Network (WRN) [58] with 16 layers and the width is 4. Following the training procedure in [58], we use SGD with Nesterov momentum and set the batch size to 128. The initial learning rate is 0.1 with a linear decay and the number of epochs is 200. For *SYTHIA*, we use FCN-32s [25] with the backbone of ResNet-50 [13]. We use Adam with the learning rate $\alpha = 0.0001$. We set the batch size to 8 and the number of epochs to 50.

**Wasserstein Auto-Encodes:** We follow [52] to implement WAEs but slightly modify architectures for dataset adaptation. The encoder and decoder are built with Fully-Connected (FC) layers for Digits dataset. We utilize two convolutional neural networks to implement the auto-encoders for CIFAR-10-C and SYTHIA. When training WAEs, we use WAE-GAN [52] to minimize the JS divergence between $P(\mathbf{e})$ and $Q(\mathbf{e}|\mathbf{x})$ in the latent space. An additional discriminator implemented by FC layers is used for distinguishing the *true* points from $P(\mathbf{e})$ and *fake* points from $Q(\mathbf{e}|\mathbf{x})$. Due to the space limitation, we suggest readers refer to the supplementary material for detailed setups.

### 5.3. Ablation Study

**Validation of $\mathcal{L}_{\text{relax}}$:** To give an intuitive understanding of how $\mathcal{L}_{\text{relax}}$ affects the distribution of augmented domains $\mathcal{S}^+$, we use t-SNE [26] to visualize $\mathcal{S}^+$ with and without $\mathcal{L}_{\text{relax}}$ in the embedding space. Their results are shown in Fig. 4 (b) and (c), respectively. We observe that the convex hull of $\mathcal{S} \cup$ $\mathcal{S}^+$ with $\mathcal{L}_{\text{relax}}$ covers an enlarged region than that of $\mathcal{S} \cup \mathcal{S}^+$ without $\mathcal{L}_{\text{relax}}$. This indicates that $\mathcal{S}^+$ contains more distributional variance and better overlaps with unseen domains. Further, we compute Wasserstein distance to quantitatively measure the difference between $\mathcal{S}$ and $\mathcal{S}^+$. The distance between $\mathcal{S}$ and $\mathcal{S}^+$ with $\mathcal{L}_{\text{relax}}$ is 0.078, while if $\mathcal{L}_{\text{relax}}$ is not employed, the distance decreases to 0.032, indicating an improvement of 58.9% by introducing $\mathcal{L}_{\text{relax}}$. These results demonstrate that $\mathcal{L}_{\text{relax}}$ is capable of pushing $\mathcal{S}^+$ away from $\mathcal{S}$, which guarantees significant domain transportation in the input space.

**Validation of meta-learning scheme:** The comparisons of M-ADA with and without meta-learning (ML) scheme are presented in Tabs. 3 and 4. We observe that with the help of this meta-learning scheme, the results on average accuracy of Digits and CIFAR-10-C are improved by 0.94% and 1.37%, respectively. Specially, the results of two kinds of unseen corruptions are shown in Fig. 5. As seen, M-ADA can significantly reduce variance and yield better performance across all levels of severity. The experimental results prove that the meta-learning scheme plays a key role to improve the training stability and classification accuracy. This is extremely important when performing adversarial domain augmentation in challenging conditions.

**Hyper-parameter tuning of $K$ and $\beta$:** We study the effect of two important hyper-parameters of M-ADA: the number of augmented domains ($K$) and the deviation between the source and augmented domain ($\beta$). We plot

| Method | SVHN | MNIST-M | SYN | USPS | Avg. |
|---|---|---|---|---|---|
| ERM [16] | 27.83 | 52.72 | 39.65 | 76.94 | 49.29 |
| CCSA [31] | 25.89 | 49.29 | 37.31 | 83.72 | 49.05 |
| d-SNE [57] | 26.22 | 50.98 | 37.83 | **93.16** | 52.05 |
| JiGen [4] | 33.80 | 57.80 | 43.79 | 77.15 | 53.14 |
| GUD [56] | 35.51 | 60.41 | 45.32 | 77.26 | 54.62 |
| M-ADA w/o $\mathcal{L}_{\text{relax}}$ | 37.33 | 61.43 | 45.58 | 77.37 | 55.43 |
| M-ADA w/o $\mathcal{L}_{\text{const}}$ | 41.36 | 67.28 | 47.94 | 78.22 | 58.70 |
| M-ADA w/o ML | 41.45 | 67.86 | 48.76 | 76.12 | 58.55 |
| **M-ADA** (full) | **42.55** | **67.94** | **48.95** | 78.53 | **59.49** |

Table 3. Single domain generalization comparison (%) on *Digits*. Models are trained on MNIST. The variant (w/o $\mathcal{L}_{\text{relax}}$) has the most significant performance decrease, indicating it is crucial to perform Wasserstein relaxation for single domain generalization.

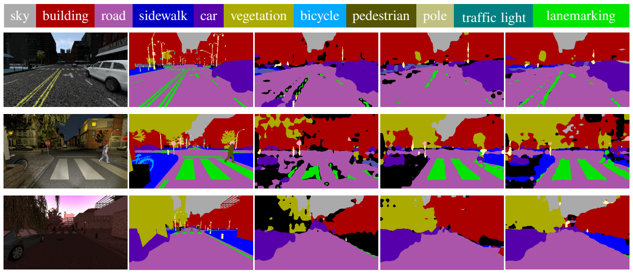


Figure 7. Examples of semantic segmentation on *SYNTHIA* [37]. **From left to right:** (a) images from unseen domains; (b) ground truth; (c) results of ERM [16]; (d) results of GUD [56]; and (e) results of M-ADA. Best viewed in color and zoom in for details.

the accuracy curve under different $K$ and $\beta$ in Fig. 6. In Fig. 6 (left), we find that the accuracy reaches the summit when $K = 3$ and keeps falling with $K$ increasing. This is due to the fact that excessive adversarial samples above a certain threshold will increase the instability and degrade the robustness of the model. In Fig. 6 (right), we observe that the accuracy reaches the summit when $\beta = 2.0 \times 10^3$ and drops slightly when $\beta$ increases. This is because large $\beta$ will produce domains too far way from the source $\mathcal{S}$ and even reach out of the manifold in the embedding space.

## 5.4. Evaluation of Single Domain Generalization

We compare our method with the following five state-of-the-art methods. (1) Empirical Risk Minimization (ERM) [53, 16] are models trained with cross-entropy loss, without any auxiliary loss and data augmentation scheme. (2) CCSA [31] uses semantic alignment to regularize the learned feature subspace for domain generalization. (3) d-SNE [57] minimizes the largest distance between the samples from the same class and maximizes the smallest distance between the samples from different classes. (4) GUD [56] proposes an adversarial data augmentation method for single domain generalization, which is the related work to M-ADA. (5) JiGen [4] learns to classify and predict the order of shuffled image patches at the same time for domain generalization.

**Comparison on Digits:** We train all models on MNIST and test them on unseen domains, *i.e.*, MNIST-M, SVHN, SYN, and USPS. We report the results in Tab. 3. We observe that M-ADA outperforms GUD with a large margin on SVHN, MNIST-M and SYN. The improvement on USPS is not as significant as those on other domains, mainly due to its great similarity with MNIST. On the contrary, CCSA and d-SNE obtain large improvements on USPS but perform poorly on other ones. We also compare M-ADA with an ensemble model of GUD, which aggregates prediction results of several models under different semantic constraints. Results are shown in Tab. 1. As seen, M-ADA outperforms GUD ensemble models in terms of generalization accuracy but with much less model parameters and even faster inference speed. The strong results, once again, testify the efficiency of the proposed learning to learn framework.

**Comparison on CIFAR-10-C:** We train all models on the clean data, *i.e.*, CIFAR-10, and test them on the corruption data, *i.e., CIFAR-10-C*. In this case, there are totally 19 unseen testing domains. Results on CIFAR-10-C across five levels of corruption severity are shown in Tab. 2. As seen, The gap between GUD and M-ADA gets larger with the level of severity increasing, and M-ADA can significantly reduce standard deviations across all levels. In addition, we present the result of each corruption with the highest severity in Tab. 4. We observe that M-ADA substantially outperforms other methods on most corruptions. Specially, in several corruptions such as *Snow*, *Glass blur*, *Pixelate* and corruptions related with *Noise*, M-ADA outperforms ERM [16] with more than 10%. More importantly, M-ADA has the lowest values on mCE and RmCE, indicating its strong robustness against image corruptions.

**Comparison on SYTHIA:** In this experiment, Highway is the source domain, and New York-like City together with Old European Town are unseen target domains. We report semantic segmentation results in Tab. 5 and show some examples in Fig. 7. Unseen domains are from different locations and other conditions. We observe that M-ADA obtains the highest values on average mIoUs across three source domains, suggesting its capability of coping with changes of locations, weather and time. Improvements over ERM [16] and GUD [56] are not significant compared with the other two datasets, mainly owing to the limited number of training images and high reliance of unseen domains.

## 5.5. Evaluation of Few-Shot Domain Adaptation

**Settings:** Although M-ADA is designed for single domain generalization, as mentioned in Sec. 3.3, we also show that M-ADA can be easily applied for few-shot domain adaptation [30]. In few-shot learning, models are usually

|  | Weather |  |  | Blur |  |  | Noise |  |  | Digital |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | Fog | Snow | Frost | Zoom | Defocus | Glass | Speckle | Shot | Impulse | Jpeg | Pixelate | Spatter | Avg. | mCE | RmCE |
| ERM [16] | 65.92 | 74.36 | 61.57 | 59.97 | 53.71 | 49.44 | 41.31 | 35.41 | 25.65 | 69.90 | 41.07 | 75.36 | 56.15 | 1.00 | 1.00 |
| CCSA [31] | 66.94 | 74.55 | 61.49 | 61.96 | 56.11 | 48.46 | 40.12 | 33.79 | 24.56 | 69.68 | 40.94 | 77.91 | 56.31 | 0.99 | 0.99 |
| d-SNE [57] | 65.99 | 75.46 | 62.25 | 58.47 | 53.71 | 50.48 | 45.30 | 39.93 | 27.95 | 70.20 | 38.46 | 73.40 | 56.96 | 0.99 | 1.00 |
| GUD [56] | 68.29 | 76.75 | 69.94 | 62.95 | 56.41 | 53.45 | 38.45 | 36.87 | 22.26 | 74.22 | **53.34** | 80.27 | 58.26 | 0.97 | 0.95 |
| M-ADA w/o $\mathcal{L}_{\text{relax}}$ | 66.99 | 80.09 | 74.93 | 54.15 | 44.67 | 60.57 | 59.88 | 59.18 | 43.46 | 76.45 | 53.13 | 80.75 | 61.92 | 0.90 | 0.86 |
| M-ADA w/o ML | 67.68 | **80.91** | 76.20 | 65.70 | 56.87 | **62.14** | 60.01 | 59.63 | 40.04 | **77.62** | 52.49 | **81.02** | 64.22 | 0.85 | 0.80 |
| **M-ADA** (full) | **69.36** | 80.59 | **76.66** | **68.04** | **61.18** | 61.59 | **60.88** | **60.58** | **45.18** | 77.14 | 52.25 | 80.62 | **65.59** | **0.82** | **0.77** |

Table 4. Robustness comparison on *CIFAR-10-C* [14]. The models are generalized from the clean data to different corruptions. We report the classification accuracy (%) of 19 corruptions (only 12 are shown) under the corruption level of "5" (the severest). We also report the mean Corruption Error (mCE) and relative mCE (RmCE) in the last two columns. The lower the better for mCE and RmCE.

|  |  | New York-like City |  |  |  |  | Old European Town |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Source Domain | Method | Dawn | Fog | Night | Spring | Winter | Dawn | Fog | Night | Spring | Winter | Avg. |
| Highway/Dawn | ERM [16] | 27.80 | 2.73 | 0.93 | 6.80 | 1.65 | 52.78 | 31.37 | 15.86 | 33.78 | 13.35 | 18.70 |
|  | GUD [56] | 27.14 | 4.05 | 1.63 | 7.22 | 2.83 | 52.80 | 34.43 | 18.19 | 33.58 | 14.68 | 19.66 |
|  | **M-ADA** | **29.10** | **4.43** | **4.75** | **14.13** | **4.97** | **54.28** | **36.04** | **23.19** | **37.53** | **14.87** | **22.33** |
| Highway/Fog | ERM [16] | 17.24 | 34.80 | 12.36 | 26.38 | 11.81 | 33.73 | 55.03 | 26.19 | 41.74 | 12.32 | 27.16 |
|  | GUD [56] | 18.75 | **35.58** | **12.77** | 26.02 | 13.05 | 37.27 | **56.69** | 28.06 | **43.57** | **13.59** | 28.53 |
|  | **M-ADA** | **21.74** | 32.00 | 9.74 | **26.40** | **13.28** | **42.79** | 56.60 | **31.79** | 42.77 | 12.85 | **29.00** |
| Highway/Spring | ERM [16] | 26.75 | 26.41 | 18.22 | 32.89 | 24.60 | 51.72 | 51.85 | 35.65 | 54.00 | 28.13 | 35.02 |
|  | GUD [56] | 28.84 | 29.67 | 20.85 | 35.32 | 27.87 | 52.21 | **52.87** | 35.99 | 55.30 | **29.58** | 36.85 |
|  | **M-ADA** | **29.70** | **31.03** | **22.22** | **38.19** | **28.29** | **53.57** | 51.83 | **38.98** | **55.63** | 25.29 | **37.47** |

Table 5. Semantic segmentation comparison on *SYNTHIA* [37]. The models are generalized from one source domain to many unseen environment settings. We report the standard mean Intersection Over Unions (mIoUs) and demonstrate visual results in Fig. 7.

| Method | $\|\mathcal{T}\|$ | U $\rightarrow$ M | M $\rightarrow$ S | S $\rightarrow$ M | Avg. |
|---|---|---|---|---|---|
| I2I [33] | All | 92.20 | - | 92.10 | - |
| DIRT-T [43] |  | - | 54.50 | **99.40** | - |
| SE [7] |  | **98.07** | 13.96 | 99.18 | 70.40 |
| SBADA [38] |  | 97.60 | **61.08** | 76.14 | 78.27 |
| G2A [39] |  | 90.80 | - | 92.40 | - |
| FADA [30] | 7 | 91.50 | 47.00 | 87.20 | 75.23 |
| CCSA [31] | 10 | 95.71 | 37.63 | 94.57 | 75.97 |
| **M-ADA** | 0 | 71.19 | 36.61 | 60.14 | 55.98 |
|  | 7 | 92.33 | 56.33 | 89.90 | 79.52 |
|  | 10 | 93.67 | 57.16 | 91.81 | **80.88** |

Table 6. Few-shot domain adaptation comparison on *MNIST(M), USPS(U), and SVHN(S)* in terms of accuracy (%). $|\mathcal{T}|$ denotes the number of target samples (per class) used during model training.

first pre-trained on the source domain $\mathcal{S}$ and then fine-tuned on the target domain $\mathcal{T}$. More specifically, we first train M-ADA on $\mathcal{S}$ using all training images. Then we randomly pick out 7 or 10 images per class from $\mathcal{T}$. These images are used to fine-tune the pre-trained model with a learning rate of 0.0001 and a batch size of 16.

**Discussions:** We compare our method with the state-of-the-art methods for few-shot domain adaptation. We also report the results of some unsupervised methods which use images in the target domain for training. Results on MNIST, USPS, and SVHN are shown in Tab. 6. We observe that M-ADA obtains competitive results compared with FADA [30] and CCSA [31]. And M-ADA also outperforms several unsupervised methods which take advantage of unlabeled images from the target domain. More importantly, we note that both FADA [30] and CCSA [31] are trained in a manner where samples from $\mathcal{S}$ and $\mathcal{T}$ are strongly coupled. This means that when the target domain changes, an entirely new model has to be trained. On the other hand, for a new target domain, M-ADA only needs to fine-tune the pre-trained model with a few samples within a small number of iterations. This demonstrates the high flexibility of M-ADA.

## 6. Conclusion

In this paper, we present Meta-Learning based Adversarial Domain Augmentation (M-ADA) to address the problem of single domain generalization. The core idea is to use a meta-learning based scheme for efficiently organizing the training of augmented "fictitious" domains, which are OOD from source domain and created by adversarial training. In the future, we expect to further extend our work to address regression problems, or knowledge transferring in multimodal learning.

## References

[1] Marcin Andrychowicz, Misha Denil, Sergio Gómez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. Learning to Learn by

Gradient Descent by Gradient Descent. In *NeurIPS*, pages 3981–3989, 2016.

[2] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *NeurIPS*, pages 998–1008, 2018.

[3] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor Sampedro, Kurt Konolige, Sergey Levine, and Vincent Vanhoucke. Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping. In *ICRA*, pages 4243–4250, 2018.

[4] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, pages 2229–2238, 2019.

[5] John S Denker, WR Gardner, Hans Peter Graf, Donnie Henderson, Richard E Howard, W Hubbard, Lawrence D Jackel, Henry S Baird, and Isabelle Guyon. Neural network recognizer for hand-written zip code digits. In *NeurIPS*, pages 323–331, 1989.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135, 2017.

[7] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *ICLR*, 2018.

[8] Yaroslav Ganin and Victor Lempitsky. Unsupervised Domain Adaptation by Backpropagation. In *ICML*, pages 1180–1189, 2015.

[9] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *ICCV*, pages 2551–2559, 2015.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

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

[12] Thomas Grubinger, Adriana Birlutiu, Holger Schöner, Thomas Natschläger, and Tom Heskes. Multi-domain transfer component analysis for domain generalization. *Neural Processing Letters (NPL)*, 46(3):845–855, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, pages 770–778, 2016.

[14] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019.

[15] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, 2015.

[16] 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.

[17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature Cell Biology (NCB)*, 521(7553):436–444, 5 2015.

[18] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation (NC)*, 1(4):541–551, 1989.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[20] Jaeho Lee and Maxim Raginsky. Minimax statistical learning with wasserstein distances. In *NeurIPS*, pages 2687–2696, 2018.

[21] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, Broader and Artier Domain Generalization. In *ICCV*, pages 5542–5550, 2017.

[22] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to Generalize: Meta-Learning for Domain Generalization. In *AAAI*, 2018.

[23] Ke Li and Jitendra Malik. Learning to optimize. In *ICLR*, 2017.

[24] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *ICML*, pages 4013–4022, 2019.

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[26] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)*, 9(Nov):2579–2605, 2008.

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[28] Massimiliano Mancini, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Best sources forward: domain generalization through source-specific nets. In *ICIP*, pages 1353–1357, 2018.

[29] Massimiliano Mancini, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Robust place categorization with deep domain generalization. *IEEE Robotics and Automation Letters (RAL)*, 3(3):2093–2100, 2018.

[30] Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. In *NeurIPS*, pages 6670–6680, 2017.

[31] Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto. Unified Deep Supervised Domain Adaptation and Generalization. In *ICCV*, pages 5715–5725, 2017.

[32] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain Generalization via Invariant Feature Representation. In *ICML*, pages 10–18, 2013.

[33] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *CVPR*, pages 4500–4509, 2018.

[34] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural

images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[35] Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio S Feris, and Dimitris Metaxas. Jointly Optimize Data Augmentation and Network Training: Adversarial Data Augmentation in Human Pose Estimation. In *CVPR*, pages 2226–2234, 2018.

[36] Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to Compose Domain-Specific Transformations for Data Augmentation. In *NeurIPS*, pages 3236–3246, 2017.

[37] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, pages 3234–3243, 2016.

[38] Paolo Russo, Fabio M Carlucci, Tatiana Tommasi, and Barbara Caputo. From source to target and back: symmetric bi-directional adaptive gan. In *CVPR*, pages 8099–8108, 2018.

[39] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, pages 8503–8512, 2018.

[40] Jürgen Schmidhuber. *Evolutionary principles in self-referential learning*. PhD thesis, Technische Universität München, 1987.

[41] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing Across Domains via Cross-Gradient Training. In *ICLR*, 2018.

[42] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russell Webb. Learning from Simulated and Unsupervised Images through Adversarial Training. In *CVPR*, pages 2242–2251, 2017.

[43] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. In *ICLR*, 2018.

[44] Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying distributional robustness with principled adversarial training. In *ICLR*, 2018.

[45] Jake Snell, Kevin Swersky, and Richard Zemel. Auto-encoding variational bayes. In *ICLR*, 2014.

[46] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, pages 4077–4087, 2017.

[47] David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *CVPR*, pages 6976–6987, 2019.

[48] Masashi Sugiyama and Amos J Storkey. Mixture regression for covariate shift. In *NeurIPS*, pages 1337–1344, 2007.

[49] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[50] Zhiqiang Tang, Xi Peng, Tingfeng Li, Yizhe Zhu, and Dimitris N Metaxas. Adaptive Data Transformation. In *ICCV*, pages 2998–3006, 2019.

[51] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

[52] I Tolstikhin, O Bousquet, S Gelly, and B Schölkopf. Wasserstein auto-encoders. In *ICLR*, 2018.

[53] Vladimir Vapnik and Vlamimir Vapnik. Statistical learning theory, 1998.

[54] Cédric Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

[55] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, and Daan Wierstra. Matching networks for one shot learning. In *NeurIPS*, pages 3630–3638, 2016.

[56] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *NeurIPS*, pages 5334–5344, 2018.

[57] Xiang Xu, Xiong Zhou, Ragav Venkatesan, Gurumurthy Swaminathan, and Orchid Majumder. d-sne: Domain adaptation using stochastic neighborhood embedding. In *CVPR*, pages 2497–2506, 2019.

[58] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

[59] Long Zhao, Xi Peng, Yuxiao Chen, Mubbasir Kapadi, and Dimitris Metaxas. Knowledge as priors: Cross-modal knowledge generalization for datasets without superior knowledge. In *CVPR*, 2020.

[60] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N. Metaxas. Semantic graph convolutional networks for 3D human pose regression. In *CVPR*, pages 3425–3435, 2019.

# Appendix A. Experimental Details

**Task models:** We design specific task models and employ different training strategies for the three datasets according to their characteristics.

In Digits dataset, the model architecture is *conv-pool-conv-pool-fc-fc-softmax*. There are two 5 × 5 convolutional layers with 64 and 128 channels respectively. Each convolutional layer is followed by a max pooling layer with the size of 2 × 2. The size of the two Fully-Connected (FC) layers is 1024 and the size of the softmax layer is 10.

In CIFAR-10-C [14], we use Wide Residual Network (WRN) [58] with 16 layers and the width is 4. The first layer is a 3×3 convolutional layer. It converts the original image with 3 channels to feature maps of 16 channels. Then the features go through three groups of 3×3 convolutional layers. Each group consists of two blocks and each block is composed of two convolutional layers with the same number of channels. And their channels are {64, 128, 256} respectively. Each convolutional layer is followed by batch normalization (BN). An average pooling layer with the size of 8 × 8 is appended to the output of the third group. Finally, a softmax layer with the size of 10 predicts the distribution over classes.

In SYTHIA [37], we use FCN-32s [25] with a backbone of ResNet-50 [13]. The model begins with ResNet-50. 1×1 convolutional layer with 14 channels is appended to predict scores for each class at each of the coarse output locations. A deconvolution layer is followed to up-sample the coarse outputs to the original size through bilinear interpolation.

**Wasserstein Auto-Encodes:** We follow [52] to implement WAEs but slightly modifying architectures for the three datasets according to their characteristics.

In Digits dataset, the encoder and decoder are built with FC layers. The encoder consists of two FC layers with the size of 400 and 20 respectively. Accordingly, the decoder consists of two FC layers with the size of 400 and 3072 respectively. The discriminator consists of two FC layers with the size of 128 and 1 respectively. The architecture of is shown in Fig. 8 (a).

In CIFAR-10-C [14], the encoder begins with four convolutional layers with the channels of {16, 32, 32, 32}. And two FC layers with the size of 1024 and 512 are followed. Accordingly, the decoder begins with two FC layers with the size of 512 and 1024 respectively. And four deconvolution layers with the channels of {32, 32, 16, 3} are followed. Each layer is followed by BN except for the final layer of the decoder. The discriminator consists of two FC layers with the size of 128 and 1 respectively. The architecture is shown in Fig. 8 (b).

In SYTHIA [37], the encoder begins with three convolutional layers with the channels of {32, 64, 128}. And two FC layers with the size of {3840, 512} are followed. Accordingly, the decoder begins with two FC layers with the size of {512, 3840}. And three deconvolution layers with the channels of {64, 32, 3} are followed. Each layer is followed by BN except for the final layer of the decoder. The discriminator consists of three FC layers with the size of {512, 512, 1}. The architecture is shown in Fig. 8 (c).

We apply the Adam optimizer in training WAEs. The learning rate is 0.001 for Digits and 0.0001 for both CIFAR-10-C and SYTHIA. The training epoches is 20 for Digits, 100 for CIFAR-10-C [14], and 200 for SYTHIA [37].

# Appendix B. Additional Experimental Results

## B.1. Ablation Study

**Validation of meta-learning scheme:** The results of four kinds of unseen corruptions are shown in Fig. 9. As seen, M-ADA can significantly reduce variance and yield better performance across all levels of severity. The experimental results prove that the meta-learning scheme plays a key role to improve the training stability and classification accuracy. This is extremely important when performing adversarial domain augmentation in challenging conditions.

**Hyper-parameter tuning of $K$, $\alpha$, and $\beta$:** We study the effect of three important hyper-parameters of M-ADA: the number of augmented domains ($K$), the distance between the source and augmented domain in the embedding space ($\alpha$), and the deviation between the source and augmented domain ($\beta$). We plot the accuracy curve under different $K$, $\alpha$, and $\beta$ in Fig. 10. In Fig. 10 (left), we find that the accuracy reaches the summit when $K = 3$ and keeps falling with $K$ increasing. This is due to the fact that excessive adversarial samples above a certain threshold will increase the instability and degrade the robustness of the model. Since the distance between the augmented and source domain increases as $K$ increases, a large $K$ may break down the constraint of semantic consistency yielding inferior model training. In Fig. 10 (middle), we find that the accuracy reaches the summit when $\alpha = 1.0$ and keeps falling with $\alpha$ increasing. This is because large $\alpha$ will make the source and augmented domain too close in the embedding space, yielding limited domain transportation. In Fig. 10 (right), we observe that the accuracy reaches the summit when $\beta = 2.0 \times 10^3$ and drops slightly when $\beta$ increases. This is because large $\beta$ will produce domains too far way from the source $\mathcal{S}$ and even reach out of the manifold in embedding space.

## B.2. Comparison of Different $\mathcal{L}_{\text{relax}}$

WAEs employ Wasserstein metric to measure the distribution distance between the input and reconstruction, which is desirable for domain augmentation. So the reconstruction error $\mathcal{L}_{\text{relax}} = \|\mathbf{x}^+ - V(\mathbf{x}^+)\|^2$ indicates if $\mathbf{x}^+$ lie in the same distribution as $\mathbf{x}$. Using WAE instead of vanilla AE is the key design to achieve this goal (Tab. 7). Additionally,

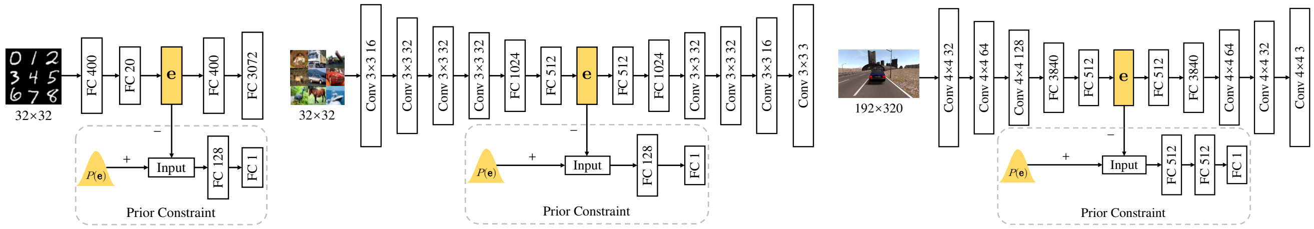


Figure 8. Architectures of WAEs. **From left to right:** (a) WAE for *Digits* ; (b) WAE for *CIFAR-10-C* [14]; and (c) WAE for *SYTHIA* [37]. Note that "**+**": positive samples for discriminator; "**-**": negative samples for discriminator.

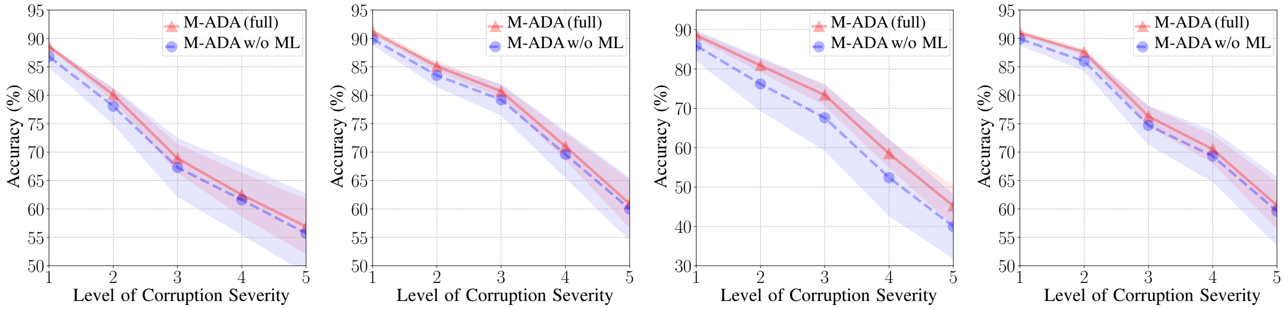


Figure 9. Validation of meta-learning scheme. Five levels of severity are evaluated on each unseen corruption. **From left to right:** (a) *Gaussian Noise*; (b) *Speckle Noise* ; (c) *Impulse Noise*; and (d) *Shot Noise*.

our experiments indicate that $\|V(\mathbf{x}) - V(\mathbf{x}^+)\|^2$ has better relaxation effect and yields improved accuracy. The distribution distance is more reliable in the reconstruction space where Wasserstein prior has been applied.

| | $\|\mathbf{x} - \mathbf{x}^+\|^2$ | Vanilla AE | WAE |
|---|---|---|---|
| Digits | 55.71% | 58.67% | 59.49% |
| CIFAR-10-C | 62.03% | 63.34% | 65.59% |

Table 7. Accuracy comparison using different relaxation terms.

### B.3. Comparison on CIFAR-10-C

We train all models on clean data, *i.e.*, CIFAR-10, and test them on corruption data, *i.e., CIFAR-10-C*. In this case, there are totally 19 unseen testing domains. We present the result of each corruption with the highest severity in Tab. 8. We observe that M-ADA substantially outperforms other methods on most corruptions. Specially, in several corruptions such as *Frost*, *Glass blur*, *Gaussian blur*, *Pixelate*, and corruptions related with *Noise*, M-ADA outperforms ERM [16] with more than 10%. More importantly, M-ADA has the lowest values on mCE and relative mCE, indicating its strong robustness against image corruptions.

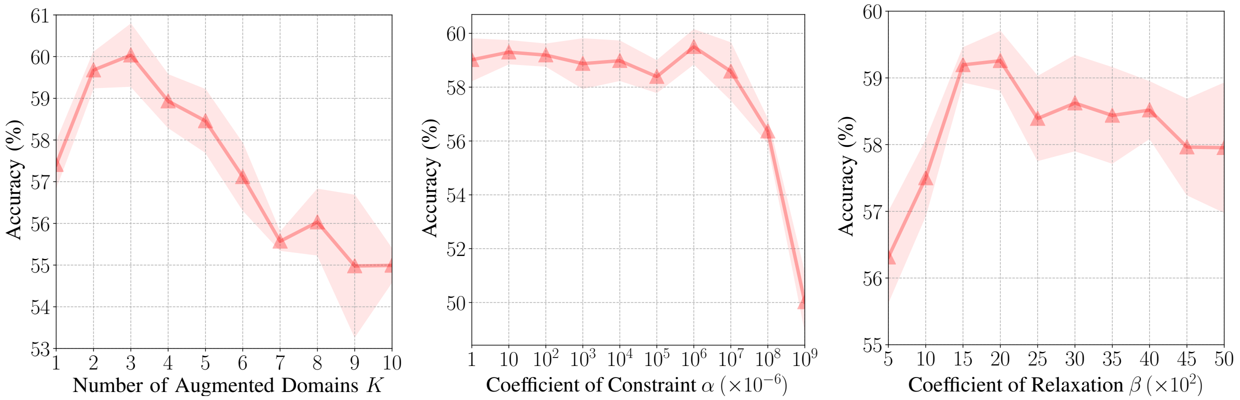


Figure 10. Hyper-parameter tuning of $K$, $\alpha$, and $\beta$. We set $K = 3$, $\alpha = 1.0$, and $\beta = 2.0 \times 10^3$ according to the best accuracy.

| | Weather | | | Blur | | | | | Noise | | | | Digital | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fog | Snow | Frost | Zoom | Defocus | Glass | Gaussian | Motion | Speckle | Shot | Impulse | Gaussian | Jpeg | Pixelate | Spatter | Elastic | Brightness | Saturate | Contrast | Avg. | mCE | RmCE |
| ERM [16] | 65.92 | 74.36 | 61.57 | 59.97 | 53.71 | 49.44 | 30.74 | 63.81 | 41.31 | 35.41 | 25.65 | 29.01 | 69.90 | 41.07 | 75.36 | 72.40 | **91.25** | 89.09 | **36.87** | 56.15 | 1.00 | 1.00 |
| CCSA [31] | 66.94 | 74.55 | 61.49 | 61.96 | 56.11 | 48.46 | 32.22 | **64.73** | 40.12 | 33.79 | 24.56 | 27.85 | 69.68 | 40.94 | 77.91 | 72.36 | 91.00 | **89.42** | 35.83 | 56.31 | 0.99 | 0.99 |
| d-SNE [57] | 65.99 | 75.46 | 62.25 | 58.47 | 53.71 | 50.48 | 33.06 | 63.70 | 45.30 | 39.93 | 27.95 | 34.02 | 70.20 | 38.46 | 73.40 | 73.33 | 90.90 | 89.27 | 36.28 | 56.96 | 0.99 | 1.00 |
| GUD [56] | 68.29 | 76.75 | 69.94 | 62.95 | 56.41 | 53.45 | 38.33 | 63.93 | 38.45 | 36.87 | 22.26 | 32.43 | 74.22 | **53.34** | 80.27 | 74.64 | 89.91 | 82.91 | 31.55 | 58.26 | 0.97 | 0.95 |
| M-ADA w/o $\mathcal{L}_{\text{relax}}$ | 66.99 | 80.09 | 74.93 | 54.15 | 44.67 | 60.57 | 30.53 | 57.06 | 59.88 | 59.18 | 43.46 | 55.07 | 76.45 | 53.13 | 80.75 | 73.85 | 90.86 | 87.01 | 27.83 | 61.92 | 0.90 | 0.86 |
| M-ADA w/o ML | 67.68 | **80.91** | 76.20 | 65.70 | 56.87 | **62.14** | 41.20 | 63.86 | 60.01 | 59.63 | 40.04 | 55.70 | **77.62** | 52.49 | **81.02** | 75.54 | 90.69 | 86.58 | 26.30 | 64.22 | 0.85 | 0.80 |
| **M-ADA** (full) | **69.36** | 80.59 | **76.66** | **68.04** | **61.18** | 61.59 | **47.34** | 64.23 | **60.88** | **60.58** | **45.18** | **56.88** | 77.14 | 52.25 | 80.62 | **75.61** | 90.78 | 87.62 | 29.71 | **65.59** | **0.82** | **0.77** |

Table 8. Full version of Tab. 4 in main paper. The models are generalized from clean data to different corruptions. We report the classification accuracy (%) of 19 corruptions under the corruption level of "5" (severest). We also report the mean Corruption Error (mCE) and relative mCE (RmCE) in the last two columns. The lower the better for mCE and RmCE.