Moderately Distributional Exploration for Domain Generalization

Rui Dai [1] **Yonggang Zhang** [2 †] **Zhen Fang** [3] **Bo Han** [2] **Xinmei Tian** [1 4 †]

Abstract

Domain generalization (DG) aims to tackle the distribution shift between training domains and unknown target domains. Generating new domains is one of the most effective approaches, yet its performance gain depends on the distribution discrepancy between the generated and target domains. Distributionally robust optimization is promising to tackle distribution discrepancy by exploring domains in an uncertainty set. However, the uncertainty set may be overwhelmingly large, leading to low-confidence prediction in DG. It is because a large uncertainty set could introduce domains containing semantically different factors from training domains. To address this issue, we propose to perform a *mo*derately *d*istributional *e*xploration (MODE) for domain generalization. Specifically, MODE performs distribution exploration in an uncertainty *subset* that shares the same semantic factors with the training domains. We show that MODE can endow models with provable generalization performance on unknown target domains. The experimental results show that MODE achieves competitive performance compared to state-of-the-art baselines.

1. Introduction

Deep neural networks (DNNs) have achieved exciting performance on various tasks. The successes of DNNs heavily depend on an underlying assumption that the training domains and target domain share the same distribution. However, this assumption may not hold in some practical scenarios, which leads to the failure of DNNs. To release this assumption, researchers have studied a more practical learning setting called *Domain Generalization* (DG) (Muandet et al., 2013; Ye et al., 2021; Shen et al., 2021). The goal of DG is to train models using training domains such that these models can generalize well in the unknown target domain which shares the same semantics with the training domains.

To generalize well on the unknown target domains, previous works introduce a domain generation strategy, enhancing the performance of DNNs by generating new domains (Zhou et al., 2020b;a; Wang et al., 2021; Xu et al., 2021). The underlying intuition of this approach is that learning with many generated domains could make DNNs robust against domain shifts. However, it remains challenging how to construct new domains to achieve a provable generalization performance on target domains. Namely, it is challenging to guarantee a mitigated distribution discrepancy between the generated domains and target domains. Accordingly, the generated domains may fail to promote generalizability or even cause performance degradation of DNNs. The reason lies in the fact that target domains are unknown in the training process, leading to an uncontrollable distribution discrepancy between the generated and the target domains.

Distributionally Robust Optimization (DRO) is a possible strategy to tackle the distribution discrepancy between training and target domains (Csiszar, 1967; Namkoong & Duchi, 2016; Staib & Jegelka, 2019). The intuition of DRO is to extend one distribution to a distribution space, i.e., uncertainty set, and uses the worst-case distribution in the uncertainty set for model training (Sinha et al., 2018; Michel et al., 2021; Mehra et al., 2022). By ensuring uniformly well performance inside the uncertainty set around the training domains, DRO can enlarge the influence of the training domains and thus shrink the distribution discrepancy between training and test domains. Unfortunately, directly employing DRO to DG has shown limited performance improvement in practice (Shen et al., 2021). The failure of DRO may be related to the overwhelmingly large property of the employed uncertainty set. Such a large uncertainty set may introduce some unrelated domains containing semantics inconsistently with training domains. Consequently, models trained over the uncertainty set make decisions with fairly low confidence, known as the low confidence issue (Hu et al., 2018; Frogner et al., 2021; Shen et al., 2021).

[1]University of Science and Technology of China, Hefei, China [2]Department of Computer Science, Hong Kong Baptist University, HongKong, China [3]Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, Australia [4]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China. Correspondence to: Xinmei Tian <xinmei@ustc.edu.cn>, Yonggang Zhang <csygzhang@comp.hkbu.edu.hk>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

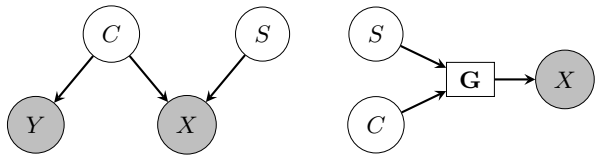

Figure 1. The left shows the causal relationship of data X, label Y, semantic factor C and non-semantic factor S. The right shows that data X are generated by an causal mechanism $\mathbf{G}$ with two causes: semantic factor C and non-semantic factor S.

To fully unleash the potential of DRO in DG, we propose to perform distribution exploration in an uncertainty *subset*, which shares the same semantic factors with the training domains, avoiding the exploration of semantically unrelated domains. The insight lies in that merely exploring the semantically related subset could shrink the space of the uncertainty set, mitigating the low confidence issue above.

Specifically, following prior works (Suter et al., 2019; Zhang et al., 2020a; Mitrovic et al., 2021; Mahajan et al., 2021; Zhang et al., 2022b; Veitch et al., 2021; Lv et al., 2022), we assume that observed data X are generated by an causal mechanism $\mathbf{G}$ with two causes: semantic factor C and non-semantic factor S, i.e., $X = \mathbf{G}(S, C)$, where the label Y is the effect of the semantic factor C. Built upon this assumption, we can perform DRO on the subset of non-semantic factor S, rather than on the original uncertainty set containing both semantic and non-semantic factors. Motivated by this insight, we propose a novel approach *moderately distributional exploration* (MODE) for domain generalization.

To support our approach, we develop a theoretical framework that provides the generalization estimation of our learning principle and gives the risk estimation for the unknown target domain. Empirically, we conduct extensive experiments to verify the effectiveness of our approach. The experimental results show that MODE achieves competitive performance compared with the state-of-the-art baselines.

2. Related Work

2.1. Domain Generalization

Domain generalization aims to learn more generalized knowledge from existing multiple source domains and finally test on the unknown target domain. Over the years, great efforts have been made in many directions, such as Invariant Representation (Chuang et al., 2020; Nguyen et al., 2021; Xiao et al., 2021; Shi et al., 2022), Causal (Mahajan et al., 2021; Mouli & Ribeiro, 2021; Lv et al., 2022), and Optimization (Krueger et al., 2021; Zhang et al., 2021; Lei et al., 2021; Gulrajani & Lopez-Paz, 2021). To generalize well on the unknown target domains, previous works introduce a domain generation strategy, enhancing the performance of DNNs by generating new domains. Shankar et al. (2018) perturbs the input samples along the direction of the most significant domain change while maintaining semantics. Zhou et al. (2020a) trains a domain transformation model to transform images to unseen domains by fooling a domain classifier. Somavarapu et al. (2020); Borlino et al. (2020) simply use a style transfer like AdaIN (Huang & Belongie, 2017) to argument data in style aspects to optimize the model. Zhou et al. (2020b) train a data generator to generate new domains using optimal transport to measure the distribution divergence. Zhou et al. (2021a;b) achieves style augmentation in the feature level by mixing the CNN feature map's mean and std between instances of different domains. Li et al. (2022) focuses on addressing the uncertain nature of domain shifts by modeling feature statistics as uncertain distributions, which is also achieved through the use of AdaIN, where non-semantic factors are replaced with randomly chosen values from the modeled distributions. Tang et al. (2021) address the problem of domain shift by developing two simple and efficient normalization methods that can reduce the non-semantic domain shift between different distributions, while Zhang et al. (2022a) jointly learns semantic and variation encoders to disentangle the semantic and non-semantic factors. Our approach explores the non-semantic factor to create augmented samples, which to some extent, is similar to approaches of data augmentation.

2.2. Distributionally Robust Optimization

Distributionally robust optimization is a promising approach to tackle distribution discrepancy by exploring unknown domains in a fixed uncertainty set (Sagawa et al., 2020). DRO has developed plenty of approaches with different methods to measuring distribution discrepancy, such as Wasserstein distance (Sinha et al., 2018; Mehra et al., 2022), f-divergence (Csiszar, 1967; Ben-Tal et al., 2013; Namkoong & Duchi, 2016; Michel et al., 2021) and maximum mean discrepancy (MMD) (Staib & Jegelka, 2019). Unfortunately, employing DRO to DG has shown limited performance improvement in practice (Shen et al., 2021). (Hu et al., 2018; Frogner et al., 2021; Shen et al., 2021) have pointed out that in order to capture the unknown target domain, the uncertainty set is often overwhelmingly large, leading the learned model to make decisions with fairly low confidence in DRO. Liu et al. (2021; 2022a) focuses on the low confidence problem, and use a Wasserstein distance is employed to determine the uncertainty set. Liu et al. (2022b) uses data geometry to construct more reasonable and effective uncertainty sets, while Qiao & Peng (2023) constructs the uncertainty using the data topology. Our approach MODE tackles the low confidence problem by performing distribution exploration in a specific uncertainty subset (non-semantic factor) and uses Wasserstein distance (Sinha et al., 2018; Mehra et al., 2022) to measure the distribution discrepancy in DG.

3. Learning Setups

Let $\mathcal{X}$ denote the feature space and $\mathcal{Y} = \{1, \ldots, \mathrm{K}_{\mathcal{Y}}\}$ denote the label space. We consider the training domains $D_{X_l Y_l}$ ($l = 1, ..., N$), N joint distributions defined over $\mathcal{X} \times \mathcal{Y}$, where X_l and Y_l are random variables whose outputs are from $\mathcal{X}$ and $\mathcal{Y}$, respectively. We also have a target domain $D_{X_t Y_t}$, a joint distribution defined over $\mathcal{X} \times \mathcal{Y}$, shares the same semantics with training domains $D_{X_l Y_l}$.

In this paper, we focus on domain generalization. The formal definition of domain generalization is given as follows.

Problem 1. (Domain Generalization). Let $D_{X_l Y_l}$ ($l = 1, ..., N$) and $D_{X_t Y_t}$ be training domains and unseen target domain, respectively. Given sets of samples called the training data: for any $l = 1, ..., N$,

$$TR_l = \left\{ \left(\mathbf{x}_l^1, y_l^1 \right), \ldots, \left(\mathbf{x}_l^{n_l}, y_l^{n_l} \right) \right\} \sim D_{X_l Y_l}^{n_l}, \text{ i.i.d.},$$

the aim of domain generalization is to train a classifier f by using the training data $TR_l, l = 1, ..., N$ such that, for any test data $\mathbf{x} \sim D_{X_t}$, f can classify $\mathbf{x}$ into the correct class.

Causal Assumption. Following prior works (Mitrovic et al., 2021; Zhang et al., 2020a; Suter et al., 2019; Mahajan et al., 2021; Zhang et al., 2022b; Lv et al., 2022; Nguyen et al., 2022; Chen et al., 2022), we *assume* that the feature random variables are generated by the following causal mechanism. Let $\mathcal{S}$ and $\mathcal{C}$ be the non-semantic factor space and semantic factor space, respectively. There exists an causal mechanism $\mathbf{G} : \mathcal{S} \times \mathcal{C} \to \mathcal{X}$ such that,

$$X_t = \mathbf{G}(S_t, C) \text{ and } X_l = \mathbf{G}(S_l, C), \ \forall\, l = 1, ..., N \quad (1)$$

where S_l and S_t are random variables defined over the non-semantic factor space $\mathcal{S}$, and C is the random variable defined over the semantic factor space $\mathcal{C}$. In summary, Eq. (1) means that the feature randoms X share the same semantic C, but don't share the non-semantic factors S.

Generally, one hopes that the non-semantic random variable cannot affect the label random variable Y, which can be determined only by the semantic C. Therefore, following Mitrovic et al. (2021); Zhang et al. (2020a); Suter et al. (2019); Mahajan et al. (2021); Lv et al. (2022); Nguyen et al. (2022), we further assume that for any $l = 1, ..., N$,

$$Y \leftarrow C \text{ and } Y_l = Y_t = Y. \quad (2)$$

Uncertainty Set and Non-semantic Space. To enhance the diversity of training domains and preserve the semantics among domains, the uncertainty set where we perform DRO is defined in the following:

$$\Omega = \{S_{\boldsymbol{\alpha}} = \Psi(\boldsymbol{\alpha}, S_1, \ldots, S_N) : \boldsymbol{\alpha} \in \mathcal{A}\}, \quad (3)$$

where $\mathcal{A}$ is a parametric space, Ψ is a function that could generate random variables, Ω is a set of random variables defined over $\mathcal{S}$.

The non-semantic space w.r.t. Ω is defined in following:

$$\mathcal{S}_\Omega = \bigcup_{S_{\boldsymbol{\alpha}} \in \Omega} \mathrm{supp} D_{S_{\boldsymbol{\alpha}}}, \quad (4)$$

where $D_{S_{\boldsymbol{\alpha}}}$ is the distribution w.r.t. random variable $S_{\boldsymbol{\alpha}}$.

Model and Risks. Here we introduce some necessary concepts about models and risks. Denote $\mathrm{f}_{\mathbf{w}} : \mathcal{X} \to \mathbb{R}^K$ by the model depending on the parameters $\mathbf{w} \in \mathcal{W}$, where $\mathcal{W}$ is the parameter space. Given a loss ℓ w.r.t. training domain $D_{X_l Y_l}$, the training domain risk w.r.t. the model $\mathbf{f}_{\mathbf{w}}$ is

$$R_l(\mathbf{w}) = \mathbb{E}\, \ell\left(\mathbf{f}_{\mathbf{w}}; X_l, Y_l\right) = \mathbb{E}\, \ell\left(\mathbf{f}_{\mathbf{w}} \circ \mathbf{G}; S_l, C, Y_l\right), \quad (5)$$

and the corresponding empirical risk w.r.t. $\mathbf{f}_{\mathbf{w}}$ is

$$\widehat{R}_l(\mathbf{w}) = \frac{1}{n_l} \sum_{i=1}^{n_l} \ell\left(\mathbf{f}_{\mathbf{w}}; \mathbf{x}_l^i, y_l^i\right). \quad (6)$$

Lastly, the target domain risk w.r.t. $\mathbf{f}_{\mathbf{w}}$ is defined as follows:

$$R_t(\mathbf{w}) = \mathbb{E}\ell\left(\mathbf{f}_{\mathbf{w}}; X_t, Y_t\right) = \mathbb{E}\, \ell\left(\mathbf{f}_{\mathbf{w}} \circ \mathbf{G}; S_t, C, Y_t\right). \quad (7)$$

4. Learning Strategy

In this section, we introduce our main motivation and develop a theoretical framework to support our insight and guide the algorithm design.

4.1. Motivation

To generalize well on the unknown target domain, it is one of the most effective strategies to generate new domains to enhance the performance of DNNs (Zhou et al., 2020b;a; Wang et al., 2021). There is an underlying intuition that learning with many generated domains could make DNNs robust against domain shifts. However, it remains challenging how to mitigate the distribution discrepancy between the generated domains and target domains. Accordingly, the generated domains may fail to promote generalizability or even cause performance degradation of DNNs. The reason is the invisibility of the target domain, which leads to an unmeasurable distribution discrepancy between the generated and the target domain (Liang et al., 2021).

Distributionally Robust Optimization (DRO) is a possible strategy to tackle distribution discrepancy (Sinha et al., 2018; Mehra et al., 2022). This is because DRO extends one distribution to a distribution space, i.e., uncertainty set, and trains models with the worst-case distribution in the uncertainty set. By ensuring uniformly well performance inside the uncertainty set around the training domains, DRO can

enlarge the influence of the training domains and thus shrink the distribution discrepancy between training and test domains. However, employing DRO to DG has shown limited performance improvement in practice (Shen et al., 2021). The failure of DRO may be related to the overwhelmingly large property of the employed uncertainty set. Such a large uncertainty set may introduce unrelated domains containing semantics inconsistently with training domains. Consequently, models trained over the set can make decisions with fairly low confidence, known as the low confidence issue (Hu et al., 2018; Frogner et al., 2021; Shen et al., 2021).

4.2. Moderately Distributional Exploration

To fully unleash the potential of DRO in DG, we propose moderately distributional exploration to perform distribution exploration MODE in an uncertainty *subset*, which shares the same semantic factors with the training domains, avoiding the exploration in the direction of semantics. The insight lies in that merely exploring the semantically related subset can shrink the space of the uncertainty set, mitigating the low confidence issue.

The considered uncertainty subset is used for exploiting the worst-case distribution, i.e., performing DRO on the subset of non-semantic factor S, which can be captured as follows:

$$\min_{\mathbf{w}\in\mathcal{W}} R_{\Omega}(\mathbf{w}) = \min_{\mathbf{w}\in\mathcal{W}} \max_{S_{\boldsymbol{\alpha}}\in\Omega} \mathbb{E}\, \ell\left(\mathbf{f}_{\mathbf{w}} \circ \mathbf{G}; S_{\boldsymbol{\alpha}}, C, Y\right). \quad (8)$$

In practical scenarios, it is challenging to exactly estimate a distribution under DG scenarios, resulting in a restricted searching capacity (more discussions are shown in Appendix C). Therefore, we propose to explore the non-semantic factor for each sample rather than for each domain:

$$\min_{\mathbf{w}\in\mathcal{W}} R_{\mathcal{S}_{\Omega}}(\mathbf{w}) = \min_{\mathbf{w}\in\mathcal{W}} \mathbb{E} \max_{\mathbf{s}\in\mathcal{S}_{\Omega}} \ell\left(\mathbf{f}_{\mathbf{w}} \circ \mathbf{G}; \mathbf{s}, C, Y\right), \quad (9)$$

where $\mathcal{S}_{\Omega}$ stands for the non-semantic space introduced in Eq. (4) and C represents the semantic random variable. The corresponding empirical risk $\widehat{R}_{\mathcal{S}_{\Omega}}(\mathbf{w})$ w.r.t. $R_{\mathcal{S}_{\Omega}}(\mathbf{w})$ is:

$$\frac{1}{\sum_{l=1}^{N} n_l} \sum_{l=1}^{N} \sum_{i=1}^{n_l} \max_{\mathbf{s}\in\mathcal{S}_{\Omega}} \ell\left(\mathbf{f}_{\mathbf{w}} \circ \mathbf{G}; \mathbf{s}, \mathbf{c}_l^i, y_l^i\right), \quad (10)$$

where $\mathbf{c}_l^i$ is the element of the semantic part of $\mathbf{G}^{-1}(\mathbf{x}_l^i)$. Besides the worst-case optimization, following previous works (Zhou et al., 2020a;b; Xu et al., 2021), we further introduce the empirical risk in our optimization. Namely, both the exploited and original data are used for model training with a parameter β used for trading off the risks:

$$\min_{\mathbf{w}\in\mathcal{W}} \widehat{R}_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w}) = (1-\beta) \sum_{l=1}^{N} \lambda_l \widehat{R}_l(\mathbf{w}) + \beta \widehat{R}_{\mathcal{S}_{\Omega}}(\mathbf{w}), \quad (11)$$

where $\boldsymbol{\lambda} = [\lambda_1, ..., \lambda_N] \in \Delta_N$ are fixed weights.

4.3. Theoretical Insights of MODE

Here, we give a learning theory to provide theoretical support for our proposed learning strategy. The main conclusions are summarized as follows:

• Theorem 1 shows that the empirical model given by Eq. (11) can achieve consistent learning performance.

• Theorem 2 shows the risk estimation for the unknown target domain w.r.t. the empirical model given by Eq. (11).

Before giving detailed theoretical results, we introduce several necessary concepts. Specifically, we use notation $R_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w})$ to represent the ideal form of $\widehat{R}_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w})$ in Eq. (11):

$$R_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w}) = (1-\beta) \sum_{l=1}^{N} \lambda_l R_l(\mathbf{w}) + \beta R_{\mathcal{S}_{\Omega}}(\mathbf{w}). \quad (12)$$

To measure the distribution discrepancy between the two domains, we use Optimal Transport Cost (Sinha et al., 2018; Mehra et al., 2022) defined as follows:

Definition 1. (Optimal Transport Cost and Wasserstein-1 Distance (Villani, 2009; 2021)). Given a cost function $c : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}_+$, the *Optimal Transport Cost* w.r.t. c between two probability distances D and D' is defined as:

$$\mathrm{W}_c\left(D, D'\right) = \inf_{\pi\in\Pi(D,D')} \mathbb{E}_{(\mathbf{x},\mathbf{x}')\sim\pi} c\left(\mathbf{x}, \mathbf{x}'\right),$$

where $\Pi\left(D, D'\right)$ is the space of all couplings for D and D'. Furthermore, if the cost c is a *metric*, then the *Optimal Transport Cost* is also called the *Wasserstein-1* distance.

Similar to Sinha et al. (2018), our results rely on the usual covering number (Vershynin, 2018) for the model classes $\mathcal{F} = \{\ell\left(\mathbf{f}_{\mathbf{w}}; \cdot\right) : \mathbf{w} \in \mathcal{W}\}$ to represent the complexity. Intuitively the covering numbers $\mathcal{N}\left(\mathcal{F}, \epsilon, L^{\infty}\right)$ is the minimal number of L^{∞} balls of radius $\epsilon > 0$ needed to cover the model classes $\mathcal{F}$, respectively. The rigorous definition on covering number is given in the Appendix A.1.

We first show that our approach can achieve consistent learning performance under mild assumptions.

Theorem 1. *(Excess Generalization Bound). Assume that*

• $0 \le \ell(\mathbf{f}_{\mathbf{w}}; \mathbf{x}, y) \le M_{\ell} < +\infty$,

• $S_1, S_2, ..., S_l$ *are mutually independent,*

• $S_l \perp\!\!\!\perp C$ *and* $Y_l = Y_t = Y, \forall l = 1, ..., N$.

Let $\widehat{\mathbf{w}}$ *be the solution of Eq.* (11)*, i.e.,*

$$\widehat{\mathbf{w}} \in \arg\min_{\mathbf{w}\in\mathcal{W}} \widehat{R}_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w}).$$

With the probability at least $1 - 4e^{-t} > 0$,

$$R_{\boldsymbol{\lambda}}^{\beta}(\widehat{\mathbf{w}}) - \min_{\mathbf{w}\in\mathcal{W}} R_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w}) \le \epsilon_{\boldsymbol{\lambda}}^{\beta}(n_1, ..., n_N; t), \quad (13)$$

where $\epsilon_{\boldsymbol{\lambda}}^{\beta}(n_1, ..., n_N; t)$ *is equal to*

$$(1-\beta)\sum_{l=1}^{N}\frac{b_0 M_\ell \lambda_l}{\sqrt{n_l}}\int_0^1\sqrt{\log\mathcal{N}(\mathcal{F}, M_\ell\epsilon, L^\infty)}d\epsilon$$
$$+\beta\frac{b_1 M_\ell}{\sqrt{\sum_{l=1}^{N} n_l}}\int_0^1\sqrt{\log\mathcal{N}(\mathcal{F}, M_\ell\epsilon, L^\infty)}\mathrm{d}\epsilon$$
$$+2(1-\beta)\sum_{l=1}^{N}\lambda_l M_\ell\sqrt{\frac{2t}{n_l}}+2\beta M_\ell\sqrt{\frac{2t}{\sum_{l=1}^{N} n_l}},$$

here b_0 *and* b_1 *are uniform constants.*

Under proper conditions, one can show that the bound (Eq. (13)) can attained $\tilde{\mathcal{O}}(\sum_{l=1}^{N}\frac{\lambda_l}{\sqrt{n_l}})+\tilde{\mathcal{O}}(\frac{1}{\sqrt{\sum_{l=1}^{N} n_l}})$, i.e.,

$$R_{\boldsymbol{\lambda}}^{\beta}(\widehat{\mathbf{w}})-\min_{\mathbf{w}\in\mathcal{W}}R_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w})\leq\tilde{\mathcal{O}}(\sum_{l=1}^{N}\frac{\lambda_l}{\sqrt{n_l}})+\tilde{\mathcal{O}}(\frac{1}{\sqrt{\sum_{l=1}^{N} n_l}}).$$

Corollary 1 in Appendix A.3 gives an example supporting this claim. Next, the following theorem gives a learning bound to estimate the unknown target domain risk.

Theorem 2. *(Risk Estimation). Given the same conditions in Theorem 1 and let* $\widehat{\mathbf{w}}$ *be the solution of Eq.* (11)*, i.e.,*

$$\widehat{\mathbf{w}}\in\arg\min_{\mathbf{w}\in\mathcal{W}}\widehat{R}_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w}).$$

If the cost function $c(\cdot,\cdot):\mathcal{S}\times\mathcal{S}\to\mathbb{R}_+$ *is a continuous metric and* $\ell(\mathbf{f_w}\circ\mathbf{G};\mathbf{s},\mathbf{c},y)$ *is* L_c*-Lipschitz w.r.t. c, i.e.,*

$$|\ell(\mathbf{f_w}\circ\mathbf{G};\mathbf{s},\mathbf{c},y)-\ell(\mathbf{f_w}\circ\mathbf{G};\mathbf{s}',\mathbf{c},y)|\leq L_c c(\mathbf{s},\mathbf{s}'),$$

then with the probability at least $1-4e^{-t}>0$ *,*

$$\begin{aligned}&R_t(\widehat{\mathbf{w}})-\min_{\mathbf{w}\in\mathcal{W}}R_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w})\\&\leq(1-\beta)L_c\sum_{l=1}^{N}\lambda_l\mathrm{W}_c(D_{S_t},D_{S_l})\\&+\beta L_c\min_{S_{\boldsymbol{\alpha}}\in\Omega}\mathrm{W}_c(D_{S_t},D_{S_{\boldsymbol{\alpha}}})+\epsilon_{\boldsymbol{\lambda}}^{\beta}(n_1,...,n_N;t),\end{aligned}\tag{14}$$

where $\epsilon_{\boldsymbol{\lambda}}^{\beta}(n_1, ..., n_N; t)$ *is introduced in Theorem 1.*

The Trade-off of Ω. From Theorem 2, we can see that the distribution discrepancy between the target distribution and distributions in distribution searching space can hurt the network's generalization ability due to the term $\beta L_c\min_{S_{\boldsymbol{\alpha}}\in\Omega}\mathrm{W}_c(D_{S_t},D_{S_{\boldsymbol{\alpha}}})$. When Ω is large enough to include D_{S_t}, this term becomes 0. Although a larger Ω will decrease this term, the approximate risk $\min_{\mathbf{w}\in\mathcal{W}}R_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w})$ will be increased, which means that there is a trade-off between these two terms about the choice of Ω.

The Trade-off of β. It can be observed that the increase of β leads to the decrease of $(1-\beta)L_c\sum_{l=1}^{N}\lambda_l\mathrm{W}_c(D_{S_t},D_{S_l})+\beta L_c\min_{S_{\boldsymbol{\alpha}}\in\Omega}\mathrm{W}_c(D_{S_t},D_{S_{\boldsymbol{\alpha}}})$. But β also determines the value of $\min_{\mathbf{w}\in\mathcal{W}}R_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w})$ and $\epsilon_{\boldsymbol{\lambda}}^{\beta}(n_1,...,n_N;t)$, leading to the trade-off of the choice of β in practice.

5. Realization of MODE

Motivated by our theoretical insights, we propose a realization of MODE by using some existing style transfer approaches, which will be introduced below [1].

• **Loss Functions.** Following Li et al. (2017); Xu et al. (2021), we use the cross entropy loss as ℓ.

• **Algorithm Design.** The key in algorithm design is the implementation of the causal mechanism $\mathbf{G}$. In practice, we use Fourier-based transfer and AdaIN-based transfer to construct $\mathbf{G}$ in our algorithm.

Other style transfer methods not introduced in this paper can also be applied to our approach in the same way.

MODE-F: Fourier-based MODE. The Fourier-based transfer (Xu et al., 2021) is considered able to separate the stylistic information from the semantic information by using the discrete Fourier transform to decompose the data $\mathbf{x}$ into its amplitude $\mathcal{A}(\mathbf{x})$ and phase $\mathcal{P}(\mathbf{x})$. It is believed that the phase information contains more high-level semantics and is not easily affected by domain shifts (Oppenheim et al., 1979; Oppenheim & Lim, 1981), which makes it possible to create samples of different styles by mixing amplitudes.

In practice, we treat the amplitude $\mathcal{A}(\mathbf{x})$ as the non-semantic factor S and treat the phase $\mathcal{P}(\mathbf{x})$ as the semantic factor C. Following our approach, we explore $\mathcal{A}(\mathbf{x})$ corresponding to the worst-case generated data fixing $\mathcal{P}(\mathbf{x})$.

Since Fourier-based transfer creates a new sample by mixing amplitudes and maintaining the original phase, we have:

$$\begin{aligned}\hat{\mathcal{A}}_\gamma(\boldsymbol{\alpha},\mathbf{x})=\gamma[\alpha_0\mathcal{A}(\mathbf{x})+\sum_{l=1}^{M}\alpha_l\mathcal{A}(\mathbf{x}_l)]\\+(1-\gamma)\mathcal{A}(\mathbf{x}),\end{aligned}\tag{15}$$

$$\mathbf{G}_\gamma(\boldsymbol{\alpha},\mathbf{x})=\mathrm{iFFT}\left[\hat{\mathcal{A}}_\gamma(\boldsymbol{\alpha},\mathbf{x})*e^{-j*\mathcal{P}(\mathbf{x})}\right],\tag{16}$$

where $\mathcal{A}(\mathbf{x}_l), l=1,\cdots,M$ are the amplitudes of M other images and $\boldsymbol{\alpha}\in\Delta_{M+1}$. So that we could control the direction of stylization by changing $\boldsymbol{\alpha}=[\alpha_0,\alpha_1,\cdots,\alpha_M]$ and control the degree of stylization by changing γ. More details of Fourier-based MODE are shown in Appendix B.1.

MODE-A: AdaIN-based MODE. AdaIN (Huang & Belongie, 2017) is one of the representative methods of neural style transfer. It uses the mean $\boldsymbol{\mu}$ and std $\boldsymbol{\sigma}$ of feature map output by the fixed encoder E to control style information and trains a decoder D to restore the stylized image from the feature map whose mean and std had been changed.

In practice, we treat these mean $\boldsymbol{\mu}$ and std $\boldsymbol{\sigma}$ as the non-semantic factor S and treat the normalized feature map as the semantic factor C. Following our approach, by fixing

[1] Code: `github.com/Rxsw/MODE`.

the normalized feature map, we explore $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ which corresponds to the worst-case generated data.

Since AdaIN-based transfer creates a new sample by mixing mean-std and maintaining the original normalized feature map, we first calculate mixed mean and mixed std:

$$\tilde{\boldsymbol{\mu}}(\boldsymbol{\alpha}, \mathbf{x}) = \alpha_0 \boldsymbol{\mu}(\mathrm{E}(\mathbf{x})) + \sum_{l=1}^{M} \alpha_l \boldsymbol{\mu}(\mathrm{E}(\mathbf{x}_l)), \tag{17}$$

$$\tilde{\boldsymbol{\sigma}}(\boldsymbol{\alpha}, \mathbf{x}) = \alpha_0 \boldsymbol{\sigma}(\mathrm{E}(\mathbf{x})) + \sum_{l=1}^{M} \alpha_l \boldsymbol{\sigma}(\mathrm{E}(\mathbf{x}_l)), \tag{18}$$

and then apply mixed mean and mixed std to the normalized feature map and restore image by decoder D:

$$\tilde{\mathbf{Z}}(\boldsymbol{\alpha}, \mathbf{x}) = \tilde{\boldsymbol{\mu}}(\boldsymbol{\alpha}, \mathbf{x}) + \tilde{\boldsymbol{\sigma}}(\boldsymbol{\alpha}, \mathbf{x}) \frac{\mathrm{E}(\mathbf{x}) - \boldsymbol{\mu}(\mathrm{E}(\mathbf{x}))}{\boldsymbol{\sigma}(\mathrm{E}(\mathbf{x}))}, \tag{19}$$

$$\mathbf{G}_\gamma(\boldsymbol{\alpha}, \mathbf{x}) = \mathrm{D}(\gamma \tilde{\mathbf{Z}}(\boldsymbol{\alpha}, \mathbf{x}) + (1-\gamma)\, \mathrm{E}(\mathbf{x})), \tag{20}$$

where $\mathrm{E}(\mathbf{x}_l), l = 1, \cdots, M$ are the feature map of M other images and $\boldsymbol{\alpha} \in \Delta_{M+1}$. So that we could control the direction of stylization by changing $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \cdots, \alpha_M]$ and control the degree of stylization by changing γ. More details of AdaIN-based MODE are shown in Appendix B.2.

The Non-semantic Space $\mathcal{S}_\Omega$. In each iteration, the non-semantic space $\mathcal{S}_\Omega$ in Eq. (10) is defined as:

$$\mathcal{S}_\Omega = \{ s_{\boldsymbol{\alpha}} = \sum_{l=0}^{M} \alpha_l s_l : \boldsymbol{\alpha} = [\alpha_0, ..., \alpha_M] \in \Delta_{M+1} \}, \tag{21}$$

where s_l is $\mathcal{A}(\mathbf{x}_l)$ in MODE-F and $(\boldsymbol{\mu}, \boldsymbol{\sigma})_l$ in MODE-A.

Update $\boldsymbol{\alpha}$. In each iteration, exploring the optimal $\boldsymbol{\alpha}$ requires multiple inner steps, leading the training time to increase exponentially. To speed up the exploring process, inspired by methods of adversarial attacks (Szegedy et al., 2014; Goodfellow et al., 2015; Madry et al., 2018; Zhang et al., 2020b), we use the gradient's direction and fixed step size μ to update $\boldsymbol{\alpha}$ every time after generating augmented data $\hat{\mathbf{x}}$ using $\mathbf{G}_\gamma$:

$$\boldsymbol{\alpha}^k = \mathrm{Normalize}(\boldsymbol{\alpha}^{k-1} + \mu\, \mathrm{sign}(\nabla_{\boldsymbol{\alpha}} \ell(\mathbf{f}_{\mathbf{w}}; \hat{\mathbf{x}}^{k-1}, y))), \tag{22}$$

where $\hat{\mathbf{x}}^{k-1} = \mathbf{G}_\gamma(\boldsymbol{\alpha}^{k-1}, \mathbf{x})$ and $\boldsymbol{\alpha} \in \Delta_{M+1}$.

• **Stochastic Realization**[2]**.** Algorithm 1 gives a stochastic realization for MODE, where minibatch is randomly sampled in each iteration. We first explore the worst-case generated data by *Maximization*. Specifically, the value of $\boldsymbol{\alpha}$ is uniformly initialized and updated by Eq.(22) for K steps. After getting the final augmented data, we update the model parameters $\mathbf{w}$ by *Minimization* which is one step of minibatch gradient descent with $\widehat{R}_l(\mathbf{w})$, $\widehat{R}_s(\mathbf{w})$ and β.

[2]In practice, we set $\boldsymbol{\lambda} : \{\lambda_l = n_l / \sum_{i=1}^{N} n_i, l = 1, \cdots, N\}$ aforementioned in Eq. (11), which means that each sample in different domains is given the same weight.

Algorithm 1 MODE

Input: training set, batch size n, number of inner steps K, number of style provider M, step size μ, model architecture parametrized by $\mathbf{w}$, hyperparameter β and γ, the causal mechanism $\mathbf{G}_\gamma$
Output: Robust model $\mathbf{f}_{\mathbf{w}}$
Randomly initialize model $\mathbf{f}_{\mathbf{w}}$, or initialize model with pre-trained configuration
repeat
  Read a mini-batch $\boldsymbol{x} = [\mathbf{x}_1, ..., \mathbf{x}_n]$, $\boldsymbol{y} = [y_1, ..., y_n]$ from the training set
  # Maximization: Exploration
  for $i = 1$ **to** n (*in parallel*) **do**
    Initialize $\alpha_0, \alpha_1, \cdots, \alpha_M$ as $\boldsymbol{\alpha}_i^0$
    Initialize $\hat{\mathbf{x}}_i^0 \leftarrow \mathbf{G}_\gamma(\boldsymbol{\alpha}_i^0, \mathbf{x}_i)$
    for $k = 1$ **to** K **do**
      # Inner Step
      $\tilde{\boldsymbol{\alpha}}_i^k \leftarrow$ Eq.(22)
      $\hat{\mathbf{x}}_i^k \leftarrow \mathbf{G}_\gamma(\boldsymbol{\alpha}_i^k, \mathbf{x}_i)$
    end for
  end for
  # Minimization: Update Model
  $\widehat{R}_l(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{f}_{\mathbf{w}}; \mathbf{x}_i, y_i)$
  $\widehat{R}_s(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{f}_{\mathbf{w}}; \hat{\mathbf{x}}_i^K, y_i)$
  $\mathbf{w} \leftarrow \mathbf{w} - \mathrm{lr}\, \nabla_{\mathbf{w}} \left[(1-\beta)\widehat{R}_l(\mathbf{w}) + \beta \widehat{R}_s(\mathbf{w}) \right]$
until convergence

6. Experiments

In this section, we demonstrate the superiority of our approach on several DG benchmarks.

6.1. Datasets

Following previous works (Zhou et al., 2020a; Huang et al., 2020; Xu et al., 2021; Lv et al., 2022), we evaluate our approach on three standard DG benchmark datasets described below. More results include VLCS (Torralba & Efros, 2011), DomainNet (Peng et al., 2019) and Mini-DomainNet (Zhou et al., 2021a) are given in the Appendix D.

Digits-DG (Zhou et al., 2020a) consists of 4 digit datasets: MNIST (M) (LeCun et al., 1998), MNIST-M (M-M) (Ganin & Lempitsky, 2015), SVHN (SV) (Netzer et al., 2011) and SYN (SY) (Ganin & Lempitsky, 2015) which differ in font style, color, and background. Following (Zhou et al., 2020a), we randomly select 600 images of each class from each domain, where 80% of the selected images are used for training, and 20% are used for validation.

Table 1. Leave-one-domain-out classification accuracies (in %) on PACS and Office-Home in ResNet18. The best and second-best results are highlighted in bold and underlined, respectively. DRO† is the result of directly applying Group DRO (Sagawa et al., 2020) to DG. CIRL† (Lv et al., 2022) is the result of reproducing using the authors' official codes and following the same settings in the original papers.

Dataset	PACS					Office-Home				
Methods	A	C	P	S	Avg.	A	C	P	R	Avg.
DeepAll (Zhou et al., 2020a)	77.6	76.8	95.9	69.5	79.9	57.9	52.7	73.5	74.8	64.7
Jigen (Carlucci et al., 2019)	79.4	75.3	96.0	71.4	80.5	53.0	47.5	71.5	72.8	61.2
MMD-AAE (Li et al., 2018b)	75.2	72.7	96.0	64.2	77.0	56.5	47.3	72.1	74.8	62.7
CrossGrad (Shankar et al., 2018)	79.8	76.8	96.0	70.2	80.7	58.4	49.4	73.9	75.8	64.4
DDAIG (Zhou et al., 2020a)	84.2	78.1	95.3	74.7	83.1	59.2	52.3	74.6	76.0	65.5
L2A-OT (Zhou et al., 2020b)	83.3	78.2	96.2	73.6	82.8	**60.6**	50.1	**74.8**	77.0	65.6
MixStyle (Zhou et al., 2021a)	83.0	78.6	96.3	71.2	82.3	58.7	53.4	74.2	75.9	65.5
MatchDG (Mahajan et al., 2021)	81.3	80.7	96.5	79.7	84.6	-	-	-	-	-
CICF (Li et al., 2021)	80.7	76.9	95.6	74.5	81.9	57.1	52.0	74.1	75.6	64.7
RSC (Huang et al., 2020)	83.4	80.3	96.0	80.9	85.2	58.4	47.9	71.6	74.5	63.1
FACT (Xu et al., 2021)	**85.9**	79.4	**96.6**	80.8	85.7	60.3	54.9	74.5	**76.6**	66.6
CIRL† (Lv et al., 2022)	85.5±0.2	79.6±0.3	96.1±0.5	82.7±0.3	86.0	58.6±0.2	55.4±0.1	73.8±0.3	75.1±0.1	65.7
DRO† (Sagawa et al., 2020)	82.5±1.3	79.1±1.0	95.1±0.2	78.5±1.2	83.2	52.8±0.3	49.2±0.6	67.6±0.4	70.8±0.5	60.1
MODE-F (ours)	84.5±0.6	80.4±0.8	95.5±0.2	82.2±0.7	85.7	57.7±0.1	54.0±0.4	73.9±0.2	76.1±0.3	65.4
MODE-A (ours)	84.4±0.9	**81.9±0.9**	95.2±0.3	**85.8±0.3**	**86.9**	60.1±0.8	**57.3±0.6**	74.2±0.5	76.0±0.2	**66.9**

Table 2. Leave-one-domain-out classification accuracies (in %) on Digit-DG. The best and second-best results are highlighted in bold and underlined, respectively.

Methods	M	M-M	SV	SY	Avg.
DeepAll (Zhou et al., 2020a)	95.8	58.8	61.7	78.6	73.7
Jigen (Carlucci et al., 2019)	96.5	61.4	63.7	74.0	73.9
MMD-AAE (Li et al., 2018b)	96.5	58.4	65.0	78.4	74.6
CrossGrad (Shankar et al., 2018)	96.7	61.1	65.3	80.2	75.8
DDAIG (Zhou et al., 2020a)	96.6	64.1	68.6	81.0	77.6
L2A-OT (Zhou et al., 2020b)	96.7	63.9	68.6	83.2	78.1
MixStyle (Zhou et al., 2021a)	96.5	63.5	64.7	81.2	76.5
CICF (Li et al., 2021)	95.8	63.7	65.8	80.7	76.5
FACT (Xu et al., 2021)	97.9	65.6	72.4	90.3	81.6
CIRL (Lv et al., 2022)	96.1	69.8	**76.2**	87.7	82.5
MODE-F (ours)	**98.5**±0.1	**72.7**±0.1	73.2 ±0.6	**91.1**±0.4	**83.9**

PACS (Li et al., 2017) is an object recognition benchmark designed for DG which consists of 9,991 images from 4 domains namely Photo (P), Art-painting (A), Cartoon (C), Sketch (S) with large style discrepancy and has seven categories in each domain. For a fair comparison, we use the training-validation-test split provided by (Li et al., 2017).

Office-Home (Venkateswara et al., 2017) consists of 15,500 images of 65 classes from four domains: Art (A), Clipart (C), Product (P), Real-World (R), which differ in image style and viewpoint. For a fair comparison, we use the training-validation-test split same as Xiao et al. (2021).

6.2. Implementation Details

Following the commonly used leave-one-domain-out strategy (Li et al., 2017; Xu et al., 2021), DG models are evaluated using one domain, after training on the other domains.

Basic Details. For Digits-DG, we use the backbone introduced by (Zhou et al., 2020b; Xu et al., 2021). All images are resized to 32×32. Following Xu et al. (2021), we train the network using an SGD optimizer with a learning rate of 0.05, batch size of 128, a momentum of 0.9, and weight decay 5e-4 for 50 epochs. The learning rate is decayed by 0.1 every 20 epochs. We use random cropping in data augmentation. For PACS and Office-Home, following Li et al. (2017); Xu et al. (2021), we use a pre-trained ResNet-18 backbone (He et al., 2016), all images are resized to 224×224. We train the network using SGD optimizer with learning rate 5e-4, momentum 0.9, and weight decay 5e-4. We train the model for 80 epochs with batch size 16 and 50 epochs with batch size 32, respectively. The learning rate is decayed by 0.1 every 40 epochs. We use the standard augmentation protocol in Li et al. (2017); Xu et al. (2021).

Method-specific Details. The hyperparameters of our approach: The number of inner steps K, inner step size μ, β, γ, and The number of style providers M. The settings of these hyperparameters are shown in Appendix D.4.

6.3. Experimental Results

Results on Digit-DG[3]**.** We show the Leave-one-domain-out classification accuracies on Digit-DG in Table 2. It can be observed that our approach achieves the highest accuracy in most domains and the second-highest accuracy in the remaining domain.4 In particular, in MNIST-M (M-M), which has complex backgrounds and rich colors, our approach exceeds the previous approach of state-of-the-art by 2.7%, which proves the capability of our approach.

Results on PACS. We show the Leave-one-domain-out classification accuracies on PACS on Table 1. It can be observed that our approach achieves the highest average accu-

[3]We only use the Fourier-based method MODE-F for Digit-DG, since AdaIN could not process low-resolution images originally.

racy. Our approach also exceeds Group DRO (Sagawa et al., 2020) by 3% on average. In particular, in the most challenging domain Sketch (S), where the image is only composed of simple lines without background, our approach exceeds the previous approach of state-of-the-art by 3%.

Results on Office-Home. We show the Leave-one-domain-out classification accuracies on Office-Home in Table 1. It can be observed that our approach achieves the highest average accuracy. Our approach also exceeds Group DRO (Sagawa et al., 2020) by 6% on average. In particular, in the Clipart (C), which is very similar to the domain Sketch in PACS, we still get 1.9% better than the previous best result, which proves that our approach has consistent performance.

6.4. Analytical Experiments

Number of Inner Steps K. Figure 2 is the average loss of samples in different inner steps to the current model changes with the number of epochs in a training process, and the effect of the number of inner steps K. It shows that our approach indeed finds samples with higher empirical risk and with the increase in the number of inner steps, the final accuracy also maintains the trend of generally increasing, which proves the effectiveness of our approach.

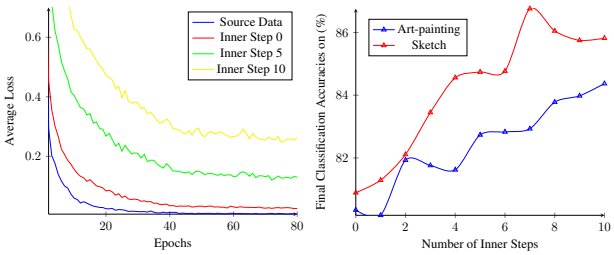

Figure 2. The left shows the average loss of augmented samples in different inner steps changes with the number of epochs in a training process. The right shows the effect of the number of inner steps. All results are conducted on the PACS dataset with Sketch (S) or Art-painting (A) as the unknown target domain.

Hyperparameter β and γ. Figure 3 show the final accuracies of different β and γ. Since γ controls the degree of stylization, the change of γ can be viewed as a change of Ω. It can be observed that there is a trade-off across the choices of β. The best choice for β is between 0.2 and 0.4. We can also observe that choosing a larger Ω (corresponds to larger γ) would not always be better, which is consistent with our theory Eq.(14). Some visualization results with too large Ω are shown in Figure 6 and Figure 12.

Number of Style Provider M and Inner Steps K. Figure 4 shows the final accuracies of different β, number of style provider M, and number of inner steps K. M determines the number of styles that can be used in the exploration, which may affect $\mathcal{S}_\Omega$. It can be observed that there is also a trade-off across the choices of M. And in most cases, the larger K results in better performance. More discussion is in Appendix C.

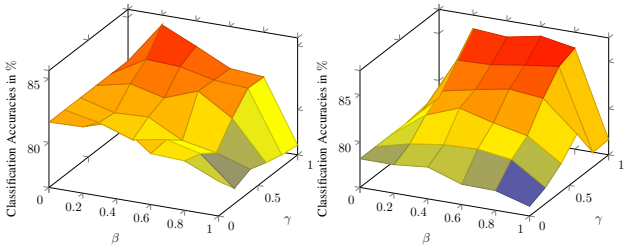

Figure 3. Classification accuracies of different β and γ. The left results are conducted on the PACS dataset with Art-painting (A) as the unknown target domain; the right results are conducted with Sketch (S) as the unknown target domain.

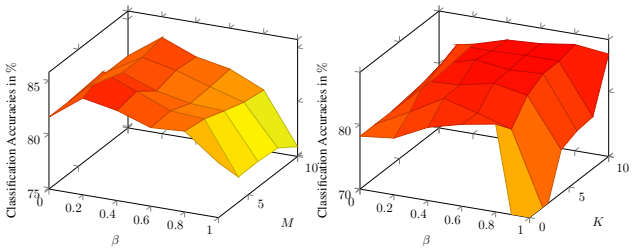

Figure 4. Classification accuracies of different β, number of style provider M and number of iterations K in inner optimization. The left results are conducted on the PACS dataset with Art-painting (A) as the unknown target domain; the right results are conducted with Sketch (S) as the unknown target domain.

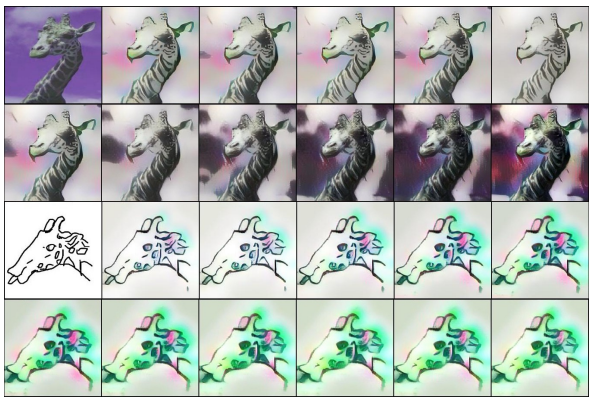

Figure 5. Visualization results of the normal exploration process, where the semantic information is preserved when exploring non-semantic factors. More results are shown in Appendix D.5.

Limitation. In our work, different methods to construct the mechanism $\mathbf{G}$ have a great impact on the results. How to find a better way to construct $\mathbf{G}$ remains to be explored.

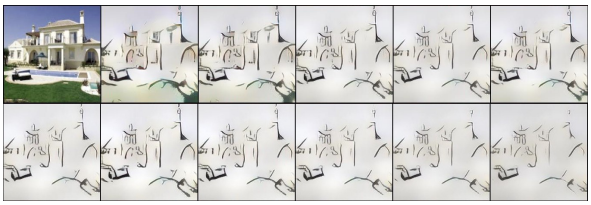

Figure 6. Visualization results of the exploration process with too large Ω. The semantic information is lost during the exploration.

7. Conclusion

Generating new domains using these accessible training domains is one of the most effective approaches in domain generalization, yet their performance gain depends on the distribution discrepancy between the generated domain and the unknown target domain. The low-confidence issue hinders the application of Distributionally robust optimization to DG. To address this issue, we propose an approach called MODE, which performs distribution exploration in an uncertainty subset that shares the same semantic factors with the training domains, and theoretically shows the convergence guarantee toward the generalization performance on the unknown target domain. Empirically, we conduct extensive experiments to verify the effectiveness of our approach. We hope that our work can inspire more ideas in the future.

8. Acknowledgment

This work was supported by NSFC No.62222117 and the Fundamental Research Funds for the Central Universities under contract WK3490000005. YGZ and BH were supported by NSFC Young Scientists Fund No.62006202 and Guangdong Basic and Applied Basic Research Foundation No.2022A1515011652, and HKBU CSD Departmental Incentive Grant.

References

Ben-Tal, A., Den Hertog, D., De Waegenaere, A., Melenberg, B., and Rennen, G. Robust solutions of optimization problems affected by uncertain probabilities. In *Management Science*, volume 59, pp. 341–357. INFORMS, 2013.

Borlino, F. C., D'Innocente, A., and Tommasi, T. Rethinking domain generalization baselines. In *ICPR*, 2020.

Carlucci, F. M., D'Innocente, A., Bucci, S., Caputo, B., and Tommasi, T. Domain generalization by solving jigsaw puzzles. In *CVPR*, 2019.

Chen, Y., Zhang, Y., Bian, Y., Yang, H., Kaili, M., Xie, B., Liu, T., Han, B., and Cheng, J. Learning causally invariant representations for out-of-distribution generalization on graphs. In *NeurIPS*, 2022.

Chuang, C.-Y., Torralba, A., and Jegelka, S. Estimating generalization under distribution shifts via domain-invariant representations. In *ICML*, 2020.

Csiszar, I. Information-type measures of difference of probability distributions and indirect observations. In *Studia . Math. Hungar*, volume 2, 1967.

Dou, Q., Coelho de Castro, D., Kamnitsas, K., and Glocker, B. Domain generalization via model-agnostic learning of semantic features. In *NeurIPS*, 2019.

Fang, Z., Lu, J., Liu, F., Xuan, J., and Zhang, G. Open set domain adaptation: Theoretical bound and algorithm. In *IEEE Transactions on Neural Networks and Learning Systems*, volume 32, pp. 4309–4322. IEEE, 2020.

Fang, Z., Li, Y., Lu, J., Dong, J., Han, B., and Liu, F. Is out-of-distribution detection learnable? In *NeurIPS*, 2022.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

Frogner, C., Claici, S., Chien, E., and Solomon, J. Incorporating unlabeled data into distributionally robust learning. In *J. Mach. Learn. Res.*, volume 22, pp. 56:1–56:46, 2021.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. In *ICLR*, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hu, W., Niu, G., Sato, I., and Sugiyama, M. Does distributionally robust supervised learning give robust classifiers? In *ICML*, 2018.

Huang, X. and Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

Huang, Z., Wang, H., Xing, E. P., and Huang, D. Self-challenging improves cross-domain generalization. In *ECCV*, 2020.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., and Courville, A. Out-of-distribution generalization via risk extrapolation (rex). In *ICML*, 2021.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pp. 2278–2324. IEEE, 1998.

Lei, Q., Hu, W., and Lee, J. Near-optimal linear regression under distribution shift. In *ICML*, 2021.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. Deeper, broader and artier domain generalization. In *ICCV*, 2017.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018a.

Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y.-Z., and Hospedales, T. M. Episodic training for domain generalization. In *ICCV*, 2019.

Li, H., Pan, S. J., Wang, S., and Kot, A. C. Domain generalization with adversarial feature learning. In *CVPR*, 2018b.

Li, X., Zhang, Z., Wei, G., Lan, C., Zeng, W., Jin, X., and Chen, Z. Confounder identification-free causal visual feature learning. In *arXiv preprint arXiv:2111.13420*, 2021.

Li, X., Dai, Y., Ge, Y., Liu, J., Shan, Y., and Duan, L.-Y. Uncertainty modeling for out-of-distribution generalization. In *ICLR*, 2022.

Liang, J., Gong, K., Li, S., Liu, C. H., Li, H., Liu, D., Wang, G., et al. Pareto domain adaptation. In *NeurIPS*, 2021.

Liu, J., Shen, Z., Cui, P., Zhou, L., Kuang, K., Li, B., and Lin, Y. Stable adversarial learning under distributional shifts. In *AAAI*, 2021.

Liu, J., Shen, Z., Cui, P., Zhou, L., Kuang, K., and Li, B. Distributionally robust learning with stable adversarial training. In *IEEE Transactions on Knowledge and Data Engineering*, 2022a.

Liu, J., Wu, J., Li, B., and Cui, P. Distributionally robust optimization with data geometry. In *NeurIPS*, 2022b.

Lv, F., Liang, J., Li, S., Zang, B., Liu, C. H., Wang, Z., and Liu, D. Causality inspired representation learning for domain generalization. In *CVPR*, 2022.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Mahajan, D., Tople, S., and Sharma, A. Domain generalization using causal matching. In *ICML*, 2021.

Matsuura, T. and Harada, T. Domain generalization using a mixture of multiple latent domains. In *AAAI*, 2020.

Mehra, A., Kailkhura, B., Chen, P.-Y., and Hamm, J. On certifying and improving generalization to unseen domains. In *arXiv preprint arXiv:2206.12364*, 2022.

Michel, P., Hashimoto, T., and Neubig, G. Modeling the second player in distributionally robust optimization. In *ICLR*, 2021.

Mitrovic, J., McWilliams, B., Walker, J., Buesing, L., and Blundell, C. Representation learning via invariant causal mechanisms. In *ICLR*, 2021.

Mouli, S. C. and Ribeiro, B. Asymmetry learning for counterfactually-invariant classification in ood tasks. In *ICLR*, 2021.

Muandet, K., Balduzzi, D., and Schölkopf, B. Domain generalization via invariant feature representation. In *ICML*, 2013.

Namkoong, H. and Duchi, J. C. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *NeurIPS*, 2016.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Nguyen, A. T., Tran, T., Gal, Y., and Baydin, A. G. Domain invariant representation learning with domain density transformations. In *NeurIPS*, 2021.

Nguyen, T., Do, K., Nguyen, D. T., Duong, B., and Nguyen, T. Front-door adjustment via style transfer for out-of-distribution generalisation. In *arXiv preprint arXiv:2212.03063*, 2022.

Oppenheim, A., Lim, J., Kopec, G., and Pohlig, S. Phase in speech and pictures. In *ICASSP*, 1979.

Oppenheim, A. V. and Lim, J. S. The importance of phase in signals. In *Proceedings of the IEEE*, volume 69, pp. 529–541. IEEE, 1981.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

Qiao, F. and Peng, X. Topology-aware robust optimization for out-of-distribution generalization. In *ICLR*, 2023.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks. In *ICLR*, 2020.

Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. Adversarial training for free! In *NeurIPS*, 2019.

Shankar, S., Piratla, V., Chakrabarti, S., Chaudhuri, S., Jyothi, P., and Sarawagi, S. Generalizing across domains via cross-gradient training. In *ICLR*, 2018.

Shen, Z., Liu, J., He, Y., Zhang, X., Xu, R., Yu, H., and Cui, P. Towards out-of-distribution generalization: A survey. In *arXiv preprint arXiv:2108.13624*, 2021.

Shi, Y., Seely, J., Torr, P., N, S., Hannun, A., Usunier, N., and Synnaeve, G. Gradient matching for domain generalization. In *ICLR*, 2022.

Sinha, A., Namkoong, H., Volpi, R., and Duchi, J. Certifying some distributional robustness with principled adversarial training. In *ICLR*, 2018.

Somavarapu, N., Ma, C.-Y., and Kira, Z. Frustratingly simple domain generalization via image stylization. In *arXiv preprint arXiv:2006.11207*, 2020.

Staib, M. and Jegelka, S. Distributionally robust optimization and generalization in kernel methods. In *NeurIPS*, 2019.

Su, J., Vargas, D. V., and Sakurai, K. One pixel attack for fooling deep neural networks. In *IEEE Transactions on Evolutionary Computation*, volume 23, pp. 828–841. IEEE, 2019.

Suter, R., Miladinovic, D., Schölkopf, B., and Bauer, S. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *ICML*, 2019.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *ICLR*, 2014.

Tang, Z., Gao, Y., Zhu, Y., Zhang, Z., Li, M., and Metaxas, D. N. Crossnorm and selfnorm for generalization under distribution shifts. In *ICCV*, 2021.

Torralba, A. and Efros, A. A. Unbiased look at dataset bias. In *CVPR*, 2011.

Veitch, V., D'Amour, A., Yadlowsky, S., and Eisenstein, J. Counterfactual invariance to spurious correlations in text classification. In *NeurIPS*, 2021.

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press, 2018.

Villani, C. *Optimal transport: old and new*, volume 338. Springer, 2009.

Villani, C. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.

Wang, Q., Liu, F., Zhang, Y., Zhang, J., Gong, C., Liu, T., and Han, B. Watermarking for out-of-distribution detection. In *NeurIPS*, 2022.

Wang, Z., Luo, Y., Qiu, R., Huang, Z., and Baktashmotlagh, M. Learning to diversify for single domain generalization. In *ICCV*, 2021.

Xiao, Z., Shen, J., Zhen, X., Shao, L., and Snoek, C. A bit more bayesian: Domain-invariant learning with uncertainty. In *ICML*, 2021.

Xu, Q., Zhang, R., Zhang, Y., Wang, Y., and Tian, Q. A fourier-based framework for domain generalization. In *CVPR*, 2021.

Ye, H., Xie, C., Cai, T., Li, R., Li, Z., and Wang, L. Towards a theoretical framework of out-of-distribution generalization. In *NeurIPS*, 2021.

Zhang, C., Zhang, K., and Li, Y. A causal view on robustness of neural networks. In *NeurIPS*, 2020a.

Zhang, D., Zhang, T., Lu, Y., Zhu, Z., and Dong, B. You only propagate once: Accelerating adversarial training via maximal principle. In *NeurIPS*, 2019.

Zhang, D., Ahuja, K., Xu, Y., Wang, Y., and Courville, A. Can subnetwork structure be the key to out-of-distribution generalization? In *ICML*, 2021.

Zhang, H., Zhang, Y.-F., Liu, W., Weller, A., Schölkopf, B., and Xing, E. P. Towards principled disentanglement for domain generalization. In *CVPR*, 2022a.

Zhang, Y., Tian, X., Li, Y., Wang, X., and Tao, D. Principal component adversarial example. In *IEEE Transactions on Image Processing*, volume 29, pp. 4804–4815. IEEE, 2020b.

Zhang, Y., Gong, M., Liu, T., Niu, G., Tian, X., Han, B., Schölkopf, B., and Zhang, K. Causaladv: Adversarial robustness through the lens of causality. In *ICLR*, 2022b.

Zhou, K., Yang, Y., Hospedales, T., and Xiang, T. Deep domain-adversarial image generation for domain generalisation. In *AAAI*, 2020a.

Zhou, K., Yang, Y., Hospedales, T., and Xiang, T. Learning to generate novel domains for domain generalization. In *ECCV*, 2020b.

Zhou, K., Yang, Y., Qiao, Y., and Xiang, T. Domain generalization with mixstyle. In *ICLR*, 2021a.

Zhou, K., Yang, Y., Qiao, Y., and Xiang, T. Mixstyle neural networks for domain generalization and adaptation. In *arXiv preprint arXiv:2107.02053*, 2021b.

A. Proofs

A.1. Covering Number

We use the covering number for the model classes in our derivation. Here we give the formal definition.

Definition 2. (ϵ-covering (Vershynin, 2018)). Let $(V, \|\cdot\|)$ be a normed space, $\Theta \in V$, and $B(\cdot, \epsilon)$ the ball of radius ϵ . Then $\{V_1, \ldots, V_N\}$ is an ϵ-covering of Θ if $\Theta \subset \bigcup_{i=1}^{N} B(V_i, \epsilon)$, or equivalently, $\forall \theta \in \Theta, \exists i$ such that $\|\theta - V_i\| \leq \epsilon$.

Upon our definition of ϵ -covering, covering number is the minimal number of ϵ-balls one needs to cover Θ .

Definition 3. (Covering Number (Vershynin, 2018)). $\mathcal{N}(\Theta, \|\cdot\|, \epsilon) = \min\{n : \exists \epsilon\text{-covering over } \Theta \text{ of size } n\}$.

A.2. Proof of Theorem 1

Proof. We first recall the notations as follows:

$$R_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w}) = (1-\beta)\sum_{l=1}^{N} \lambda_l R_l(\mathbf{w}) + \beta R_{\mathcal{S}_\Omega}(\mathbf{w}), \tag{23}$$

$$\widehat{R}_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w}) = (1-\beta)\sum_{l=1}^{N} \lambda_l \widehat{R}_l(\mathbf{w}) + \beta \widehat{R}_{\mathcal{S}_\Omega}(\mathbf{w}), \tag{24}$$

$$R_l(\mathbf{w}) = \mathbb{E}\, \ell\left(\mathbf{f}_{\mathbf{w}} \circ \mathbf{G}; S_l, C, Y\right), \tag{25}$$

$$\widehat{R}_l(\mathbf{w}) = \frac{1}{n_l}\sum_{i=1}^{n_l} \ell\left(\mathbf{f}_{\mathbf{w}}; \mathbf{x}_l^i, y_l^i\right), \tag{26}$$

$$R_{\mathcal{S}_\Omega}(\mathbf{w}) = \mathbb{E} \max_{\mathbf{s}\in\mathcal{S}_\Omega} \ell\left(\mathbf{f}_{\mathbf{w}} \circ \mathbf{G}; \mathbf{s}, C, Y\right), \tag{27}$$

$$\widehat{R}_{\mathcal{S}_\Omega}(\mathbf{w}) = \frac{1}{\sum_{l=1}^{N} n_l}\sum_{l=1}^{N}\sum_{i=1}^{n_l} \max_{\mathbf{s}\in\mathcal{S}_\Omega} \ell\left(\mathbf{f}_{\mathbf{w}} \circ \mathbf{G}; \mathbf{s}, \mathbf{c}_l^i, y_l^i\right). \tag{28}$$

Let $\mathbf{w}^*$ be the solution of $\min_{\mathbf{w}\in\mathcal{W}} R_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w})$. Then similar to Fang et al. (2020; 2022), we have

$$\begin{aligned}
R_{\boldsymbol{\lambda}}^{\beta}(\widehat{\mathbf{w}}) - R_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w}^*) &\leq R_{\boldsymbol{\lambda}}^{\beta}(\widehat{\mathbf{w}}) - \widehat{R}_{\boldsymbol{\lambda}}^{\beta}(\widehat{\mathbf{w}}) + \widehat{R}_{\boldsymbol{\lambda}}^{\beta}(\widehat{\mathbf{w}}) - R_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w}^*) + \widehat{R}_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w}^*) - \widehat{R}_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w}^*) \\
&\leq (1-\beta)\left[\sum_{l=1}^{N}\lambda_l R_l(\widehat{\mathbf{w}}) - \sum_{l=1}^{N}\lambda_l R_l(\mathbf{w}^*)\right] + \beta\left[R_{\mathcal{S}_\Omega}(\widehat{\mathbf{w}}) - R_{\mathcal{S}_\Omega}(\mathbf{w}^*)\right] \\
&\quad - (1-\beta)\left[\sum_{l=1}^{N}\lambda_l \widehat{R}_l(\widehat{\mathbf{w}}) - \sum_{l=1}^{N}\lambda_l \widehat{R}_l(\mathbf{w}^*)\right] - \beta\left[\widehat{R}_{\mathcal{S}_\Omega}(\widehat{\mathbf{w}}) - \widehat{R}_{\mathcal{S}_\Omega}(\mathbf{w}^*)\right] \\
&= (1-\beta)\sum_{l=1}^{N}\lambda_l\left[R_l(\widehat{\mathbf{w}}) - \widehat{R}_l(\widehat{\mathbf{w}})\right] + \beta\left[R_{\mathcal{S}_\Omega}(\widehat{\mathbf{w}}) - \widehat{R}_{\mathcal{S}_\Omega}(\widehat{\mathbf{w}})\right] \\
&\quad - (1-\beta)\sum_{l=1}^{N}\lambda_l\left[R_l(\mathbf{w}^*) - \widehat{R}_l(\mathbf{w}^*)\right] - \beta\left[R_{\mathcal{S}_\Omega}(\mathbf{w}^*) - \widehat{R}_{\mathcal{S}_\Omega}(\mathbf{w}^*)\right],
\end{aligned} \tag{29}$$

where $\widehat{R}_{\boldsymbol{\lambda}}^{\beta}(\widehat{\mathbf{w}}) - \widehat{R}_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w}^*) \leq 0$.

By Lemma 1 and Lemma 4, we have that with the probability at least $1-2e^{-t}>0$,

$$\begin{aligned}&(1-\beta)\sum_{l=1}^{N}\lambda_l\left[R_l(\widehat{\mathbf{w}})-\widehat{R}_l(\widehat{\mathbf{w}})\right]+\beta\left[R_{\mathcal{S}_\Omega}(\widehat{\mathbf{w}})-\widehat{R}_{\mathcal{S}_\Omega}(\widehat{\mathbf{w}})\right]\\ \leq&(1-\beta)\sum_{l=1}^{N}\frac{b_0M_\ell\lambda_l}{\sqrt{n_l}}\int_0^1\sqrt{\log\mathcal{N}\left(\mathcal{F},M_\ell\epsilon,L^\infty\right)}d\epsilon\\ &+\beta\frac{b_1M_\ell}{\sqrt{\sum_{l=1}^{N}n_l}}\int_0^1\sqrt{\log\mathcal{N}\left(\mathcal{F},M_\ell\epsilon,L^\infty\right)}\mathrm{d}\epsilon\\ &+(1-\beta)\sum_{l=1}^{N}\lambda_lM_\ell\sqrt{\frac{2t}{n_l}}+\beta M_\ell\sqrt{\frac{2t}{\sum_{l=1}^{N}n_l}},\end{aligned}\tag{30}$$

here b_0 and b_1 are uniform constants.

By Lemma 2 and Lemma 5, we have that with the probability at least $1-2e^{-t}>0$,

$$\begin{aligned}&(1-\beta)\sum_{l=1}^{N}\lambda_l\left[R_l\left(\mathbf{w}^*\right)-\widehat{R}_l\left(\mathbf{w}^*\right)\right]+\beta\left[R_{\mathcal{S}_\Omega}\left(\mathbf{w}^*\right)-\widehat{R}_{\mathcal{S}_\Omega}\left(\mathbf{w}^*\right)\right]\\ \leq&(1-\beta)\sum_{l=1}^{N}\lambda_lM_\ell\sqrt{\frac{2t}{n_l}}+\beta M_\ell\sqrt{\frac{2t}{\sum_{l=1}^{N}n_l}}.\end{aligned}\tag{31}$$

Combining Eqs. (29), (30) and (31), we have that with the probability at least $1-4e^{-t}>0$,

$$R_{\boldsymbol{\lambda}}^{\beta}(\widehat{\mathbf{w}})-\min_{\mathbf{w}\in\mathcal{W}}R_{\boldsymbol{\lambda}}^{\beta}(\mathbf{w})\leq\epsilon_{\boldsymbol{\lambda}}^{\beta}(n_1,...,n_N;t)\tag{32}$$

where

$$\begin{aligned}\epsilon_{\boldsymbol{\lambda}}^{\beta}(n_1,...,n_N;t)=&(1-\beta)\sum_{l=1}^{N}\frac{b_0M_\ell\lambda_l}{\sqrt{n_l}}\int_0^1\sqrt{\log\mathcal{N}\left(\mathcal{F},M_\ell\epsilon,L^\infty\right)}d\epsilon\\ &+\beta\frac{b_1M_\ell}{\sqrt{\sum_{l=1}^{N}n_l}}\int_0^1\sqrt{\log\mathcal{N}\left(\mathcal{F},M_\ell\epsilon,L^\infty\right)}\mathrm{d}\epsilon\\ &+2(1-\beta)\sum_{l=1}^{N}\lambda_lM_\ell\sqrt{\frac{2t}{n_l}}+2\beta M_\ell\sqrt{\frac{2t}{\sum_{l=1}^{N}n_l}},\end{aligned}\tag{33}$$

here b_0 and b_1 are uniform constants. □

A.3. Corollary 1

Corollary 1. *Given the same conditions in Theorem 1, if*

- $\ell(\cdot;\mathbf{x},y)$ *is L-Lipschitz w.r.t. norm* $\|\cdot\|$*, i.e., for any* $(\mathbf{x},y)\in\mathcal{X}\times\mathcal{Y}$ *, and* $\mathbf{w},\mathbf{w}'\in\mathcal{W}$*,*

$$\left|\ell(\mathbf{f}_{\mathbf{w}};\mathbf{x},y)-\ell\left(\mathbf{f}_{\mathbf{w}'};\mathbf{x},y\right)\right|\leq L\left\|\mathbf{w}-\mathbf{w}'\right\|,\tag{34}$$

- *the parameter space* $\mathcal{W}\subset\mathbb{R}^{d'}$ *satisfies that*

$$\operatorname{diam}(\mathcal{W})=\sup_{\mathbf{w},\mathbf{w}'\in\mathcal{W}}\|\mathbf{w}-\mathbf{w}'\|<+\infty,\tag{35}$$

With the probability at least $1-4e^{-t}>0$,

$$R^{\beta}_{\boldsymbol{\lambda}}(\widehat{\mathbf{w}})-\min_{\mathbf{w}\in\mathcal{W}}R^{\beta}_{\boldsymbol{\lambda}}(\mathbf{w})\leq\tilde{\epsilon}^{\beta}_{\boldsymbol{\lambda}}(n_1,...,n_N;t) \tag{36}$$

where

$$\begin{aligned}\tilde{\epsilon}^{\beta}_{\boldsymbol{\lambda}}(n_1,...,n_N;t)=&(1-\beta)\sum_{l=1}^{N}b_0\lambda_l\sqrt{\frac{M_\ell\operatorname{diam}(\mathcal{W})Ld'}{n_l}}\\&+\beta b_1\sqrt{\frac{M_\ell\operatorname{diam}(\mathcal{W})Ld'}{\sum_{l=1}^{N}n_l}}\\&+2(1-\beta)\sum_{l=1}^{N}\lambda_lM_\ell\sqrt{\frac{2t}{n_l}}+2\beta M_\ell\sqrt{\frac{2t}{\sum_{l=1}^{N}n_l}},\end{aligned} \tag{37}$$

here b_0 *and* b_1 *are uniform constants.*

Proof. Similar to the proof of Theorem 1.

By Lemma 3 and Lemma 6, we have that with the probability at least $1-2e^{-t}>0$,

$$\begin{aligned}&(1-\beta)\sum_{l=1}^{N}\lambda_l\left[R_l(\widehat{\mathbf{w}})-\widehat{R}_l(\widehat{\mathbf{w}})\right]+\beta\left[R_{\mathcal{S}_\Omega}(\widehat{\mathbf{w}})-\widehat{R}_{\mathcal{S}_\Omega}(\widehat{\mathbf{w}})\right]\\\leq&(1-\beta)\sum_{l=1}^{N}b_0\lambda_l\sqrt{\frac{M_\ell\operatorname{diam}(\mathcal{W})Ld'}{n_l}}\\&+\beta b_1\sqrt{\frac{M_\ell\operatorname{diam}(\mathcal{W})Ld'}{\sum_{l=1}^{N}n_l}}\\&+(1-\beta)\sum_{l=1}^{N}\lambda_lM_\ell\sqrt{\frac{2t}{n_l}}+\beta M_\ell\sqrt{\frac{2t}{\sum_{l=1}^{N}n_l}}.\end{aligned} \tag{38}$$

Combining Eqs. (29), (38) and (31), we have that with the probability at least $1-4e^{-t}>0$,

$$R^{\beta}_{\boldsymbol{\lambda}}(\widehat{\mathbf{w}})-\min_{\mathbf{w}\in\mathcal{W}}R^{\beta}_{\boldsymbol{\lambda}}(\mathbf{w})\leq\tilde{\epsilon}^{\beta}_{\boldsymbol{\lambda}}(n_1,...,n_N;t) \tag{39}$$

where

$$\begin{aligned}\tilde{\epsilon}^{\beta}_{\boldsymbol{\lambda}}(n_1,...,n_N;t)=&(1-\beta)\sum_{l=1}^{N}b_0\lambda_l\sqrt{\frac{M_\ell\operatorname{diam}(\mathcal{W})Ld'}{n_l}}\\&+\beta b_1\sqrt{\frac{M_\ell\operatorname{diam}(\mathcal{W})Ld'}{\sum_{l=1}^{N}n_l}}\\&+2(1-\beta)\sum_{l=1}^{N}\lambda_lM_\ell\sqrt{\frac{2t}{n_l}}+2\beta M_\ell\sqrt{\frac{2t}{\sum_{l=1}^{N}n_l}},\end{aligned} \tag{40}$$

here b_0 and b_1 are uniform constants. □

A.4. Proof of Theorem 2

Proof. We first recall the notations as follows:

$$R_t(\mathbf{w}) = \mathbb{E}\ell\left(\mathbf{f}_{\mathbf{w}}; X_t, Y_t\right) = \mathbb{E}\,\ell\left(\mathbf{f}_{\mathbf{w}} \circ \mathbf{G}; S_t, C, Y\right), \tag{41}$$

$$R_l(\mathbf{w}) = \mathbb{E}\,\ell\left(\mathbf{f}_{\mathbf{w}} \circ \mathbf{G}; S_l, C, Y\right), \tag{42}$$

$$R_{\mathcal{S}_\Omega}(\mathbf{w}) = \mathbb{E}\max_{\mathbf{s}\in\mathcal{S}_\Omega}\ell\left(\mathbf{f}_{\mathbf{w}} \circ \mathbf{G}; \mathbf{s}, C, Y\right), \tag{43}$$

$$R^{\beta}_{\boldsymbol{\lambda}}(\mathbf{w}) = (1-\beta)\sum_{l=1}^{N}\lambda_l R_l(\mathbf{w}) + \beta R_{\mathcal{S}_\Omega}(\mathbf{w}). \tag{44}$$

Consider

$$R_t(\widehat{\mathbf{w}}) - R^{\beta}_{\boldsymbol{\lambda}}(\widehat{\mathbf{w}}) = (1-\beta)\sum_{l=1}^{N}\lambda_l(R_t(\widehat{\mathbf{w}}) - R_l(\widehat{\mathbf{w}})) + \beta(R_t(\widehat{\mathbf{w}}) - R_{\mathcal{S}_\Omega}(\widehat{\mathbf{w}})). \tag{45}$$

We have

$$R_t(\widehat{\mathbf{w}}) - R_l(\widehat{\mathbf{w}}) = \mathbb{E}\,\ell\left(\mathbf{f}_{\widehat{\mathbf{w}}} \circ \mathbf{G}; S_t, C, Y\right) - \mathbb{E}\,\ell\left(\mathbf{f}_{\widehat{\mathbf{w}}} \circ \mathbf{G}; S_l, C, Y\right) \leq L_c \mathrm{W}_c\left(D_{S_t}, D_{S_l}\right). \tag{46}$$

We set the largest risk in Ω w.r.t. $\mathbf{f}_{\mathbf{w}}$ is

$$R_\Omega(\mathbf{w}) = \max_{S_{\boldsymbol{\alpha}}\in\Omega}\mathbb{E}\ell\left(\mathbf{f}_{\mathbf{w}} \circ \mathbf{G}; S_{\boldsymbol{\alpha}}, C, Y\right), \tag{47}$$

where Ω stands for the uncertainty set introduced in Eq. (3).

We have

$$R_\Omega(\mathbf{w}) \leq R_{\mathcal{S}_\Omega}(\mathbf{w}), \tag{48}$$

then

$$R_t(\widehat{\mathbf{w}}) - R_{\mathcal{S}_\Omega}(\widehat{\mathbf{w}}) \leq R_t(\widehat{\mathbf{w}}) - R_\Omega(\widehat{\mathbf{w}}) = \mathbb{E}\,\ell\left(\mathbf{f}_{\widehat{\mathbf{w}}} \circ \mathbf{G}; S_t, C, Y\right) - \max_{S_{\boldsymbol{\alpha}}\in\Omega}\mathbb{E}\ell\left(\mathbf{f}_{\widehat{\mathbf{w}}} \circ \mathbf{G}; S_{\boldsymbol{\alpha}}, C, Y\right). \tag{49}$$

We set

$$S_{\boldsymbol{\alpha}_M} = \arg\min_{S_{\boldsymbol{\alpha}}\in\Omega}\mathrm{W}_c\left(D_{S_t}, D_{S_{\boldsymbol{\alpha}}}\right), \tag{50}$$

then

$$\begin{aligned}\mathbb{E}\,\ell\left(\mathbf{f}_{\widehat{\mathbf{w}}} \circ \mathbf{G}; S_t, C, Y\right) - \max_{S_{\boldsymbol{\alpha}}\in\Omega}\mathbb{E}\ell\left(\mathbf{f}_{\widehat{\mathbf{w}}} \circ \mathbf{G}; S_{\boldsymbol{\alpha}}, C, Y\right) \leq& \mathbb{E}\,\ell\left(\mathbf{f}_{\widehat{\mathbf{w}}} \circ \mathbf{G}; S_t, C, Y\right) - \mathbb{E}\ell\left(\mathbf{f}_{\widehat{\mathbf{w}}} \circ \mathbf{G}; S_{\boldsymbol{\alpha}_M}, C, Y\right)\\ \leq& L_c\mathrm{W}_c\left(D_{S_t}, D_{S_{\boldsymbol{\alpha}_M}}\right) = \min_{S_{\boldsymbol{\alpha}}\in\Omega} L_c\mathrm{W}_c\left(D_{S_t}, D_{S_{\boldsymbol{\alpha}}}\right).\end{aligned} \tag{51}$$

Combining Eqs. (45), (46), (49) and (51), we have

$$R_t(\widehat{\mathbf{w}}) - R^{\beta}_{\boldsymbol{\lambda}}(\widehat{\mathbf{w}}) \leq (1-\beta)L_c\sum_{l=1}^{N}\lambda_l\mathrm{W}_c\left(D_{S_t}, D_{S_l}\right) + \beta L_c\min_{S_{\boldsymbol{\alpha}}\in\Omega}\mathrm{W}_c\left(D_{S_t}, D_{S_{\boldsymbol{\alpha}}}\right). \tag{52}$$

Then by Theorem 1, we complete this proof. □

A.5. Necessary Lammas

Lemma 1. *If* $0 \leq \ell(\mathbf{f_w}; \mathbf{x}, y) \leq M_\ell$ *, then with the probability at least* $1 - e^{-t} > 0$ *, we have that for any* $\mathbf{w} \in \mathcal{W}$

$$\begin{aligned} &\mathbb{E}_{(\mathbf{x},y)\sim D_{X_l Y_l}} \ell(\mathbf{f_w}; \mathbf{x}, y) - \frac{1}{n}\sum_{i=1}^{n} \ell\left(\mathbf{f_w}; \mathbf{x}_l^i, y_l^i\right) \\ \leq &\frac{b_0 M_\ell}{\sqrt{n}} \int_0^1 \sqrt{\log \mathcal{N}\left(\mathcal{F}, M_\ell \epsilon, L^\infty\right)} \mathrm{d}\epsilon + M_\ell \sqrt{\frac{2t}{n}} \end{aligned} \tag{53}$$

,where b_0 *is a uniform constant.*

Proof. Let

$$X_{\ell(\mathbf{f_w};\cdot)} = \mathbb{E}_{(\mathbf{x},y)\sim D_{X_1 Y_l}} \ell(\mathbf{f_w}; \mathbf{x}, y) - \frac{1}{n}\sum_{i=1}^{n} \ell\left(\mathbf{w}; \mathbf{x}_l^i, y_l^i\right). \tag{54}$$

Then it is clear that

$$\mathbb{E}_{S\sim D^n_{X_1 Y_1}} X_{\ell(\mathbf{f_w};\cdot)} = 0. \tag{55}$$

By Proposition 2.6.1 and Lemma 2.6.8 in Vershynin (2018),

$$\left\| X_{\ell(\mathbf{f_w};\cdot)} - X_{\ell(\mathbf{f_{w'}};\cdot)} \right\|_{\Phi_2} \leq \frac{c_0}{\sqrt{n}} \left\| \ell(\mathbf{f_w};\cdot) - \ell\left(\mathbf{f_{w'}};\cdot\right) \right\|_{L^\infty}, \tag{56}$$

where $\|\cdot\|_{\Phi_2}$ is the sub-gaussian norm and c_0 is a uniform constant. Therefore, by Dudley's entropy integral (Vershynin, 2018), we have

$$\begin{aligned} &\mathbb{E}_{S\sim D^n_{X_1 Y_1}} \sup_{\mathbf{w}\in\mathcal{W}} X_{\ell(\mathbf{f_w};\cdot)} \\ \leq &\frac{b_0}{\sqrt{n}} \int_0^{+\infty} \sqrt{\log \mathcal{N}\left(\mathcal{F}, \epsilon, L^\infty\right)} \mathrm{d}\epsilon, \end{aligned} \tag{57}$$

where b_0 is a uniform constant and

$$\mathcal{F} = \{\ell(\mathbf{f_w};\cdot) : \mathbf{w} \in \mathcal{W}\}. \tag{58}$$

Note that

$$\begin{aligned} \mathbb{E}_{S\sim D^n_{X_1 Y_1}} \sup_{\mathbf{w}\in\mathcal{W}} X_{\ell(\mathbf{f_w};\cdot)} &\leq \frac{b_0}{\sqrt{n}} \int_0^{+\infty} \sqrt{\log \mathcal{N}\left(\mathcal{F}, \epsilon, L^\infty\right)} \mathrm{d}\epsilon \\ &= \frac{b_0}{\sqrt{n}} \int_0^{M_\ell} \sqrt{\log \mathcal{N}\left(\mathcal{F}, \epsilon, L^\infty\right)} \mathrm{d}\epsilon \\ &= \frac{b_0}{\sqrt{n}} M_\ell \int_0^{1} \sqrt{\log \mathcal{N}\left(\mathcal{F}, M_\ell\epsilon, L^\infty\right)} \mathrm{d}\epsilon \end{aligned} \tag{59}$$

Then, similar to the proof of Lemma 2, we use the McDiarmid's Inequality, then with the probability at least $1 - e^{-t} > 0$, for any $\mathbf{w} \in \mathcal{W}$,

$$X_{\ell(\mathbf{f_w};))} \leq \frac{b_0}{\sqrt{n}} M_\ell \int_0^1 \sqrt{\log \mathcal{N}\left(\mathcal{F}, M_\ell\epsilon, L^\infty\right)} \mathrm{d}\epsilon + M_\ell \sqrt{\frac{2t}{n}}. \tag{60}$$

□

Lemma 2. *If* $0 \leq \ell(\mathbf{f_w}; \mathbf{x}, y) \leq M_\ell$ *, then for a fixed* $\mathbf{w}_0 \in \mathcal{W}$ *, with the probability at least* $1 - e^{-t} > 0$ *,*

$$\frac{1}{n}\sum_{i=1}^{n} \ell\left(\mathbf{w}_0; \mathbf{x}_l^i, y_l^i\right) - \mathbb{E}_{(\mathbf{x},y)\sim D_{X_l Y_l}} \ell\left(\mathbf{w}_0; \mathbf{x}, y\right) \leq M_\ell \sqrt{\frac{2t}{n}}. \tag{61}$$

Proof. By (Sinha et al., 2018), it is clear that

$$\sup_{\mathrm{w}_c\left(D_{X'},D_{X_{\mathrm{A}}}\right)\leq\rho}\mathbb{E}_{\mathbf{x}\sim D_{X'}}\ell\left(\mathbf{f}_{\mathbf{w}_0};\mathbf{x}\right)=\inf_{\gamma\geq 0}\left[\gamma\rho+\mathbb{E}_{\mathbf{x}\sim D_{X_{\mathrm{A}}}}\phi_{\gamma}\left(\mathbf{w}_0;\mathbf{x}\right)\right] \tag{62}$$

Therefore, for each $\epsilon>0$, there exists a constant $\gamma_\epsilon\geq 0$ such that

$$\gamma_\epsilon\rho+\mathbb{E}_{\mathbf{x}\sim D_{X_{\mathrm{A}}}}\phi_{\gamma_\epsilon}\left(\mathbf{w}_0;\mathbf{x}\right)\leq\sup_{\mathrm{w}_c\left(D_{X',D_{X_{\mathrm{A}}}}\right)\leq\rho}\mathbb{E}_{\mathbf{x}\sim D_{X'}}\ell\left(\mathbf{f}_{\mathbf{w}_0};\mathbf{x}\right)+\epsilon. \tag{63}$$

Combining the above inequality and McDiarmid's Inequality, then with the probability at least

$$1-\exp\left(\frac{-\epsilon_0^2 m}{2M_{\ell\mathrm{OE}}^2}\right)>0, \tag{64}$$

we have

$$\mathbb{E}_{\mathbf{x}\sim\widehat{D}_{X_{\mathrm{A}}}}\phi_{\gamma_\epsilon}\left(\mathbf{w}_0;\mathbf{x}\right)\leq\mathbb{E}_{\mathbf{x}\sim D_{X_{\mathrm{A}}}}\phi_{\gamma_\epsilon}\left(\mathbf{w}_0;\mathbf{x}\right)+\epsilon_0 \tag{65}$$

If we set $t=\epsilon_0^2 m/2M_\ell^2$, then

$$\epsilon_0=M_\ell\sqrt{\frac{2t}{m}} \tag{66}$$

Hence, with the probability at least $1-e^{-t}>0$, we have

$$\gamma_\epsilon\rho+\mathbb{E}_{\mathbf{x}\sim\widehat{D}_{X_{\mathrm{A}}}}\phi_{\gamma_\epsilon}\left(\mathbf{w}_0;\mathbf{x}\right)\leq\sup_{\mathrm{w}_c\left(D_{X'},D_{X_{\mathrm{A}}}\right)\leq\rho}\mathbb{E}_{\mathbf{x}\sim D_{X'}}\ell\left(\mathbf{f}_{\mathbf{w}_0};\mathbf{x}\right)+\epsilon+M_\ell\sqrt{\frac{2t}{m}}, \tag{67}$$

which implies that with the probability at least $1-e^{-t}>0$,

$$\sup_{\mathrm{W}_c\left(D_{X'},\widehat{D}_{X_{\mathrm{A}}}\right)\leq\rho}\mathbb{E}_{\mathbf{x}\sim D_{X'}}\ell\left(\mathbf{f}_{\mathbf{w}_0};\mathbf{x}\right)\leq\sup_{\mathrm{W}_c\left(D_{X'},D_{X_{\mathrm{A}}}\right)\leq\rho}\mathbb{E}_{\mathbf{x}\sim D_{X'}}\ell\left(\mathbf{f}_{\mathbf{w}_0};\mathbf{x}\right)+\epsilon+M_\ell\sqrt{\frac{2t}{m}}, \tag{68}$$

because

$$\gamma_\epsilon\rho+\mathbb{E}_{\mathbf{x}\sim\widehat{D}_{X_{\mathrm{A}}}}\phi_{\gamma_\epsilon}\left(\mathbf{w}_0;\mathbf{x}\right)\geq\sup_{\mathrm{W}_c\left(D_{X'},\widehat{D}_{X_{\mathrm{A}}}\right)\leq\rho}\mathbb{E}_{\mathbf{x}\sim D_{X'}}\ell\left(\mathbf{f}_{\mathbf{w}_0};\mathbf{x}\right) \tag{69}$$

By setting $\epsilon=M_\ell\sqrt{2t/m}$ and $\rho=0$, we complete this proof.

□

Lemma 3. *If*

- $0\leq\ell(\mathbf{f}_{\mathbf{w}};\mathbf{x},y)\leq M_\ell$ *;*
- $\ell(\cdot;\mathbf{x},y)$ *is L-Lipschitz w.r.t. norm* $\|\cdot\|$*, i.e., for any* $(\mathbf{x},y)\in\mathcal{X}\times\mathcal{Y}$ *, and* $\mathbf{w},\mathbf{w}'\in\mathcal{W}$*,*

$$\left|\ell(\mathbf{f}_{\mathbf{w}};\mathbf{x},y)-\ell\left(\mathbf{f}_{\mathbf{w}'};\mathbf{x},y\right)\right|\leq L\left\|\mathbf{w}-\mathbf{w}'\right\|, \tag{70}$$

- *the parameter space* $\mathcal{W}\subset\mathbb{R}^{d'}$ *satisfies that*

$$\mathrm{diam}(\mathcal{W})=\sup_{\mathbf{w},\mathbf{w}'\in\mathcal{W}}\|\mathbf{w}-\mathbf{w}'\|<+\infty, \tag{71}$$

then with the probability at least $1-e^{-t}>0$ *, we have that for any* $\mathbf{w}\in\mathcal{W}$ *,*

$$\begin{aligned}&\mathbb{E}_{(\mathbf{x},y)\sim D_{X_lY_l}}\ell(\mathbf{f_w};\mathbf{x},y)-\frac{1}{n}\sum_{i=1}^{n}\ell\left(\mathbf{f_w};\mathbf{x}_l^i,y_l^i\right)\\ \leq& b_0\sqrt{\frac{M_\ell\,\mathrm{diam}(\mathcal{W})Ld'}{n}}+M_\ell\sqrt{\frac{2t}{n}}\end{aligned}\tag{72}$$

where b_0 is a uniform constant.

Proof. The proof is similar to Corollary 1 in Sinha et al. (2018). Note that

$$\mathcal{F}=\{\ell(\mathbf{f_w};\mathbf{x},y):\mathbf{w}\in\mathcal{W}\},\tag{73}$$

and $\ell(\cdot;\mathbf{x},y)$ is L-Lipschitz w.r.t. norm $\|\cdot\|$, therefore,

$$\begin{aligned}\mathcal{N}\left(\mathcal{F},M_\ell\epsilon,L^\infty\right)&\leq\mathcal{N}\left(\mathcal{W},M_\ell\epsilon/L,\|\cdot\|\right)\\ &\leq\left(1+\frac{\mathrm{diam}(\mathcal{W})L}{M_\ell\epsilon}\right)^{d'},\end{aligned}\tag{74}$$

which implies that

$$\begin{aligned}&\int_0^1\sqrt{\log\left(\mathcal{N}\left(\mathcal{F},M_\ell\epsilon,L^\infty\right)\right)}\mathrm{d}\epsilon\\ \leq&\sqrt{d'}\int_0^1\sqrt{\log\left(1+\frac{\mathrm{diam}(\mathcal{W})L}{M_\ell\epsilon}\right)}\mathrm{d}\epsilon\\ \leq&\sqrt{d'}\int_0^1\sqrt{\frac{\mathrm{diam}(\mathcal{W})L}{M_\ell\epsilon}}\mathrm{d}\epsilon=2\sqrt{\frac{\mathrm{diam}(\mathcal{W})Ld'}{M_\ell}}.\end{aligned}\tag{75}$$

By Lemma 1, we obtain this result. □

Lemma 4. *If $0\leq\ell(\mathbf{f_w}\circ\mathbf{G};\mathbf{s},\mathbf{c},y)\leq M_\ell$, then with the probability at least $1-e^{-t}>0$, we have that for any $\mathbf{w}\in\mathcal{W}$*

$$\begin{aligned}&\mathbb{E}_{(\mathbf{x},y)\sim D_{X_lY_l}}\max_{\mathbf{s}\in\mathcal{S}_\Omega}\ell(\mathbf{f_w}\circ\mathbf{G};\mathbf{s},\mathbf{c},y)-\frac{1}{n}\sum_{i=1}^{n}\max_{\mathbf{s}\in\mathcal{S}_\Omega}\ell\left(\mathbf{f_w}\circ\mathbf{G};\mathbf{s},\mathbf{c}_l^i,y_l^i\right)\\ &\leq\frac{b_0M_\ell}{\sqrt{n}}\int_0^1\sqrt{\log\mathcal{N}\left(\mathcal{F},M_\ell\epsilon,L^\infty\right)}\mathrm{d}\epsilon+M_\ell\sqrt{\frac{2t}{n}}\end{aligned}\tag{76}$$

,where b_0 is a uniform constant.

Proof. The proof is similar to the proof of Lemma 1. □

Lemma 5. *If $0\leq\ell(\mathbf{f_w}\circ\mathbf{G};\mathbf{s},\mathbf{c},y)\leq M_\ell$, then for a fixed $\mathbf{w}_0\in\mathcal{W}$, with the probability at least $1-e^{-t}>0$,*

$$\frac{1}{n}\sum_{i=1}^{n}\max_{\mathbf{s}\in\mathcal{S}_\Omega}\ell\left(\mathbf{f_w}\circ\mathbf{G};\mathbf{s},\mathbf{c}_l^i,y_l^i\right)-\mathbb{E}_{(\mathbf{x},y)\sim D_{X_lY_l}}\max_{\mathbf{s}\in\mathcal{S}_\Omega}\ell(\mathbf{f_w}\circ\mathbf{G};\mathbf{s},\mathbf{c},y)\leq M_\ell\sqrt{\frac{2t}{n}}.\tag{77}$$

Proof. The proof is similar to the proof of Lemma 2. □

Lemma 6. *If*

- $0 \leq \ell(\mathbf{f}_{\mathbf{w}} \circ \mathbf{G}; \mathbf{s}, \mathbf{c}, y) \leq M_\ell$ *;*
- $\ell(\cdot; \mathbf{x}, y)$ *is L-Lipschitz w.r.t. norm* $\|\cdot\|$*, i.e., for any* $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ *, and* $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$*,*

$$|\ell(\mathbf{f}_{\mathbf{w}}; \mathbf{x}, y) - \ell(\mathbf{f}_{\mathbf{w}'}; \mathbf{x}, y)| \leq L \|\mathbf{w} - \mathbf{w}'\|, \tag{78}$$

- *the parameter space* $\mathcal{W} \subset \mathbb{R}^{d'}$ *satisfies that*

$$\operatorname{diam}(\mathcal{W}) = \sup_{\mathbf{w}, \mathbf{w}' \in \mathcal{W}} \|\mathbf{w} - \mathbf{w}'\| < +\infty, \tag{79}$$

then with the probability at least $1 - e^{-t} > 0$ *, we have that for any* $\mathbf{w} \in \mathcal{W}$ *,*

$$\begin{aligned} &\mathbb{E}_{(\mathbf{x},y)\sim D_{X_l Y_l}} \max_{\mathbf{s}\in\mathcal{S}_\Omega} \ell(\mathbf{f}_{\mathbf{w}} \circ \mathbf{G}; \mathbf{s}, \mathbf{c}, y) - \frac{1}{n} \sum_{i=1}^{n} \max_{\mathbf{s}\in\mathcal{S}_\Omega} \ell\left(\mathbf{f}_{\mathbf{w}} \circ \mathbf{G}; \mathbf{s}, \mathbf{c}_l^i, y_l^i\right) \\ \leq & b_0 \sqrt{\frac{M_\ell \operatorname{diam}(\mathcal{W}) L d'}{n}} + M_\ell \sqrt{\frac{2t}{n}} \end{aligned} \tag{80}$$

where b_0 *is a uniform constant.*

Proof. The condition (78) is equivalent to

- $\ell(\cdot; G(\mathbf{s}, \mathbf{c}), y)$ is L_{G}-Lipschitz w.r.t. norm $\|\cdot\|$, i.e., for any $(\mathbf{s}, \mathbf{c}, y) \in \mathcal{S} \times \mathcal{C} \times \mathcal{Y}$, and $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$,

$$|\ell(\mathbf{f}_{\mathbf{w}} \circ \mathbf{G}; \mathbf{s}, \mathbf{c}, y) - \ell(\mathbf{f}_{\mathbf{w}'} \circ \mathbf{G}; \mathbf{s}, \mathbf{c}, y)| \leq L_{\mathrm{G}} \|\mathbf{w} - \mathbf{w}'\|, \tag{81}$$

Then the proof is similar to the proof of Lemma 3. □

B. Details of Realization

B.1. MODE-F: Fourier-based MODE

Fourier-based Transfer. The Fourier-based transfer has been used in many domain generalization methods. This transfer method is considered able to separate the stylistic information from the semantic information by using the discrete Fourier transform to decompose the image into its amplitude and phase, and then create more samples of different styles using different mixing methods for amplitude.

For a image x, its discrete Fourier transformation $\mathcal{F}(\mathbf{x})$:

$$\mathcal{F}(\mathbf{x})(\mathbf{u}, \mathbf{v}) = \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} x(h, w) e^{-j2\pi\left(\frac{h}{H}u + \frac{w}{W}v\right)} \tag{82}$$

$\mathcal{F}^{-1}(x)$ is defined as the discrete inverse Fourier transformation. And both of these transformations can be implemented using FFT and do not require additional neural networks.

After discrete Fourier transformation, the amplitude and phase are defined as:

$$\mathcal{A}(\mathbf{x})(\mathbf{u}, \mathbf{v}) = \left[R^2(\mathbf{x})(\mathbf{u}, \mathbf{v}) + I^2(\mathbf{x})(\mathbf{u}, \mathbf{v})\right]^{1/2} \tag{83}$$

$$\mathcal{P}(\mathbf{x})(\mathbf{u}, \mathbf{v}) = \arctan\left[\frac{I(\mathbf{x})(\mathbf{u}, \mathbf{v})}{R(\mathbf{x})(\mathbf{u}, \mathbf{v})}\right] \tag{84}$$

where $R(x)$ and $I(x)$ represent the real and imaginary part of $\mathcal{F}(\mathbf{x})$, respectively.

In (Xu et al., 2021), Fourier-based data augmentation is implemented by:

$$\hat{\mathcal{A}}_\gamma(\mathbf{x}) = (1-\lambda)\mathcal{A}(\mathbf{x}) + \lambda\mathcal{A}(\mathbf{x}') \tag{85}$$

$$\hat{x} = \mathcal{F}^{-1}\left[\hat{\mathcal{A}}_\gamma(\mathbf{x})(\mathbf{u},\mathbf{v}) * e^{-j*\mathcal{P}(\mathbf{x})(\mathbf{u},\mathbf{v})}\right] \tag{86}$$

where x' represents randomly selected image, $\lambda \sim U(0,\eta)$, and the hyperparameter η controls the strength of the augmentation.$\hat{x}$ represents the augmented images.

Adapt Fourier-based Transfer. Although the previous Fourier-based transfer methods have achieved good results in many domain generalization methods (Xu et al., 2021; Lv et al., 2022), our approach requires that the transformation method used should be controllable. More specifically, It means that the direction of style change in the transformation process can be controlled by some parameters, and these parameters can be updated regularly in the process.

To meet the requirements, since a single image could not provide a sufficient amount of style, we randomly select M images $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M$ as style providers and calculate their amplitudes $\mathcal{A}(\mathbf{x}_1), \mathcal{A}(\mathbf{x}_2), \cdots, \mathcal{A}(\mathbf{x}_M)$, and then acquire their linear combination of amplitudes:

$$\hat{\mathcal{A}}_\gamma(\mathbf{x}) = \gamma[\alpha_0\mathcal{A}(\mathbf{x}) + \sum_{i=1}^{M}\alpha_i\mathcal{A}(\mathbf{x}_i)] + (1-\gamma)\mathcal{A}(\mathbf{x}) \tag{87}$$

$$\hat{x} = \mathcal{F}^{-1}\left[\hat{\mathcal{A}}_\gamma(\mathbf{x})(\mathbf{u},\mathbf{v}) * e^{-j*\mathcal{P}(\mathbf{x})(\mathbf{u},\mathbf{v})}\right] \tag{88}$$

where $\alpha_0, \alpha_1, \cdots, \alpha_M$ are the parameter of linear weighted which could be updated, γ is hyperparameter to determine what percentage of the amplitude is involved in the searching.

By controlling the parameter $\alpha_0, \alpha_1, \cdots, \alpha_M$, we can control the changes in the augmented output's style.

Adversarial-attacks-inspired Update Strategy. Motivated by these theoretical insights, in each iteration, the parameters $\alpha_0, \alpha_1, \cdots, \alpha_M$ will be updated by maximizing the empirical risk of the augmented sample, so that we could guide the augmented sample to be closer to the distribution with highest empirical risk.

Inspired by Adversarial attacks (Szegedy et al., 2014; Goodfellow et al., 2015; Madry et al., 2018; Zhang et al., 2020b), to speed up the searching process, after the gradient backpropagation, we update the parameters $\alpha_0, \alpha_1, \cdots, \alpha_M$ using the gradient's direction and fixed step size:

$$\tilde{\boldsymbol{\alpha}}^k = \boldsymbol{\alpha}^{k-1} + \mu\,\mathrm{sign}\left(\nabla_{\boldsymbol{\alpha}}\ell(\mathbf{f}_\mathbf{w}; \hat{\mathbf{x}}^{k-1}, y)\right) \tag{89}$$

$$\alpha_l^k = \tilde{\alpha}_l^k / \sum_{i=0}^{M}\tilde{\alpha}_i^k \tag{90}$$

where $\boldsymbol{\alpha}^k = [\alpha_0^k, \alpha_1^k, \cdots, \alpha_M^k]^\top$, μ is the step size, $\hat{x}^k$ comes from Equ Equ 87, Equ 88 with α^k, ℓ is standard Cross-entropy loss, $\mathbf{f}_\mathbf{w}$ is the network. Equ 90 means that since the sum is limited to 1, the parameters $\alpha_0, \alpha_1, \cdots, \alpha_M$ are normalized after each update.

After K iterations in inner optimization, we get $\boldsymbol{\alpha}^K$, and then we calculate $\hat{x}^{final}$ using Equ 87, Equ 88 and $\boldsymbol{\alpha}^K$.

Finally, to maintain the category label and thus enforce semantic consistency, we require that the generated sample $\hat{\mathbf{x}}^{final}$ is classified into the same category together with the original sample $\mathbf{x}$, and calculate the total loss to update network $\mathbf{f}_\mathbf{w}$:

$$\mathcal{L}_{MODE} = (1-\beta)\ell(\mathbf{f}_\mathbf{w}; \mathbf{x}, y) + \beta\ell(\mathbf{f}_\mathbf{w}; \hat{\mathbf{x}}^{final}, y) \tag{91}$$

where ℓ is standard Cross-entropy loss, β is hyperparameter to control the influence of augmented images.

The full Fourier-based training algorithm is shown in Algorithm 2 and Figure 7.

B.2. MODE-A: AdaIN-based MODE

Neural Style Transfer. Although this Fourier-based transfer method has been widely used in many domain generalization methods, it still has obvious disadvantages: most of the changes in the generated augmented image are reflected in adding

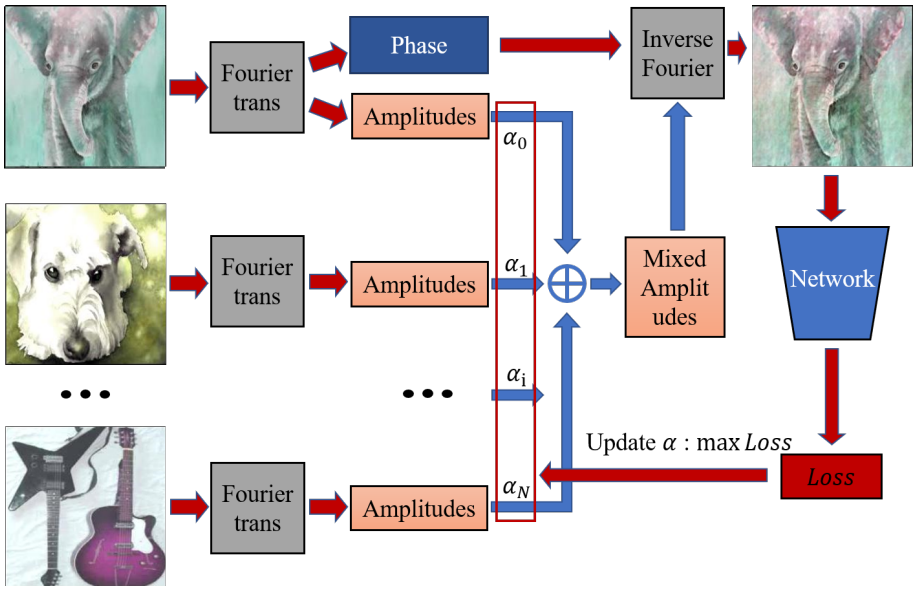

Figure 7. Fourier-based algorithm approach. The top left picture is the input picture, and the rest are the amplitude provider. In an iteration, the controlling parameters α are used to mix the amplitudes to obtain a new amplitude, together with the phase of the original input image to generate an augmented image. The augmented is input to the network to calculate the loss, and the controlling parameters α are updated by maximizing the loss.

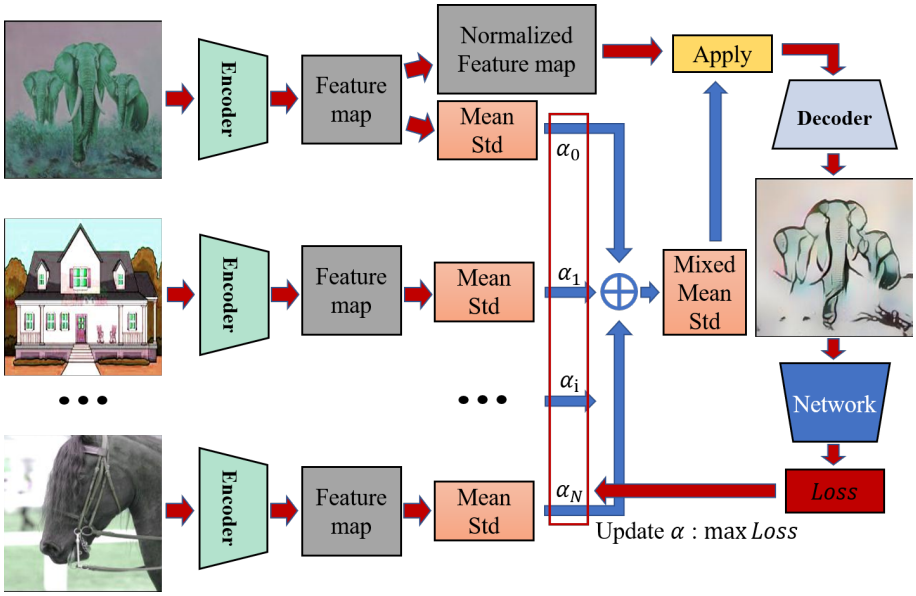

Figure 8. AdaIN-based approach. The top left image is the input content image, and the rest are the style provider. In an iteration, the controlling parameters α are used to mix the mean and std to obtain new mean and std, which is applied to the normalized feature map of the original input content image to generate an augmented image. The augmented is input to the network to calculate the loss, and the controlling parameters α is updated by maximizing the loss.

Algorithm 2 Fourier-based MODE

Input: data x_i, batch size n, number of iterations K in inner optimization, step size μ, number of amplitude providers M, network architecture parametrized by $\mathbf{w}$, hyperparameter β and γ
Output: Robust network $\mathbf{f_w}$
Randomly initialize network $\mathbf{f_w}$, or initialize network with pre-trained configuration
repeat
  Read mini-batch $\boldsymbol{x} = [\mathbf{x}_1, ..., \mathbf{x}_n]$ from training set
  Calculate their phase and amplitude as $[P_1, ..., P_n]$, $[A_1, ..., A_n]$, respectively
  # Exploration
  for $i = 1$ **to** n (in parallel) **do**
    Randomly select M other image's amplitudes in B as amplitude providers $[\hat{A}_1, ..., \hat{A}_M]$
    Initialize $\alpha_0, \alpha_1, \cdots, \alpha_M$ as $\boldsymbol{\alpha}^0$
    for $k = 1$ **to** K **do**
      Calculate $\hat{\mathbf{x}}_i^k$ using Equ 87, 88 and $\boldsymbol{\alpha}^{k-1}$
      Update $\boldsymbol{\alpha}^k$ using Equ 89, 90
    end for
    Calculate $\hat{\mathbf{x}}_i^{final}$ using Equ 87, 88 and $\boldsymbol{\alpha}^K$
  end for
  # Update Model
  Calculate $\mathcal{L}_{MODE}$ using Equ 91
  Update $\mathbf{w}$ by performing one step gradient update using $\nabla_{\mathbf{w}} \mathcal{L}_{MODE}$
until training converged

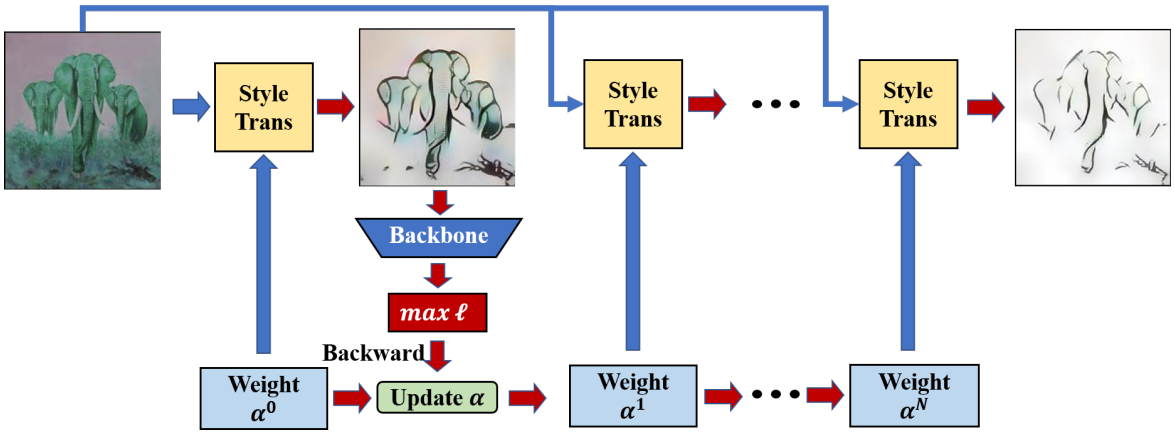

Figure 9. The overall approach. Using existing generation methods like Style trans, **MODE** could create more aggressive samples by updating the controlling parameters α through multiple steps, inspired by Adversarial attack.

irregular color blocks and textures to the original image, which is rarely seen in reality. At the same time, compared with the good results applied to some low-resolution datasets such as handwritten digit datasets, when applied to some real-world high-resolution datasets, the generated results lacking authenticity are also difficult to be satisfactory. It prompts us to consider other style transfer methods with better results.

Neural Style Transfer (Huang & Belongie, 2017) has developed rapidly in recent years. Compared with the Fourier-based method, using the pre-trained neural network model to process the image, the results generated by neural style transfer are usually more authentic. AdaIN (Huang & Belongie, 2017) is one of the representative methods of neural style transfer. It uses the mean and std of feature map output by the fixed encoder to represent style information and trains a decoder to restore stylized images from the feature map whose mean and std had been changed.

To apply AdaIN in our approach, we use the mean and std introduced above to represent the non-semantic factor S and use the normalized feature map to represent the semantic factor C.

AdaIN consists of an encoder E, a decoder D, and a mean-std processing module for the feature map. Encoder E will process input content image x and style image $\hat{\mathbf{x}}$ into feature map $\mathbf{z} = E(\mathbf{x})$, $\hat{\mathbf{z}} = E(\hat{\mathbf{x}})$, respectively. And then mean-std processing module will calculate the mean and std of the $\mathbf{z}$ and $\hat{\mathbf{z}}$ separately:

$$\boldsymbol{\mu}(\mathbf{z}) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}\mathbf{z}_{c,h,w} \tag{92}$$

$$\boldsymbol{\sigma}(\mathbf{z}) = \sqrt{\frac{1}{HW-1}\sum_{h=1}^{H}\sum_{w=1}^{W}\left(\mathbf{z}_{c,h,w} - \boldsymbol{\mu}(\mathbf{z})\right)^2} \tag{93}$$

where $\mathbf{z}$ should be a feature map of shape $C \times H \times W$ with C, H, W being the number of channels, height, and width.

After obtaining the $\boldsymbol{\mu}(\mathbf{z}), \boldsymbol{\sigma}(\mathbf{z}), \boldsymbol{\mu}(\hat{\mathbf{z}}), \boldsymbol{\sigma}(\hat{\mathbf{z}})$, module will control their mixing by hyperparameter λ :

$$\tilde{\boldsymbol{\mu}}(\mathbf{z}, \hat{\mathbf{z}}, \lambda) = \lambda\boldsymbol{\mu}(\mathbf{z}) + (1-\lambda)\boldsymbol{\mu}(\hat{\mathbf{z}}) \tag{94}$$

$$\tilde{\boldsymbol{\sigma}}(\mathbf{z}, \hat{\mathbf{z}}, \lambda) = \lambda\boldsymbol{\sigma}(\mathbf{z}) + (1-\lambda)\boldsymbol{\sigma}(\hat{\mathbf{z}}) \tag{95}$$

where $0 \leq \lambda \leq 1$ is the hyperparameter to control the level of stylization

And then mean-std processing module will apply the mixed mean and std to the normalized feature map of the content image :

$$\tilde{\mathbf{z}} = \tilde{\boldsymbol{\mu}}(\mathbf{z}, \hat{\mathbf{z}}, \lambda) + \tilde{\boldsymbol{\sigma}}(\mathbf{z}, \hat{\mathbf{z}}, \lambda)\frac{\mathbf{z} - \boldsymbol{\mu}(\mathbf{z})}{\boldsymbol{\sigma}(\mathbf{z})} \tag{96}$$

where $\tilde{\mathbf{z}}$ is the output feature map.

Finally, the decoder D will process the output feature map $\tilde{\mathbf{z}}$ to restore stylized images $\tilde{x} = D(\tilde{\mathbf{z}})$.

Adapt AdaIN Transfer. Original AdaIN style transfer can only control the degree of stylization but not the direction of stylization. We should make AdaIN style transfer controllable, i.e. the direction of style change in the transformation process can be controlled by some parameters, and these parameters can be updated regularly in the process.

To satisfy this requirement, since AdaIN use mean and std of feature map to represent style information, we randomly select M images $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M$ as style providers and calculate their feature map output using encoder $\mathbf{z}_1 = E(\mathbf{x}_1), \mathbf{z}_2 = E(\mathbf{x}_2), \cdots, \mathbf{z}_M = E(\mathbf{x}_M)$, and finally calculate their mean and std of the feature map $\boldsymbol{\mu}(\mathbf{z}_1), \boldsymbol{\mu}(\mathbf{z}_2), \cdots, \boldsymbol{\mu}(\mathbf{z}_M)$, $\boldsymbol{\sigma}(\mathbf{z}_1), \boldsymbol{\sigma}(\mathbf{z}_2), \cdots, \boldsymbol{\sigma}(\mathbf{z}_M)$, and then acquire their linear combination of mean and std:

$$\tilde{\boldsymbol{\mu}}(\mathbf{z}, \mathbf{z}_{1,\cdots,M}, \alpha_{0,\cdots,M}) = \alpha_0\boldsymbol{\mu}(\mathbf{z}) + \sum_{i=1}^{M}\alpha_i\boldsymbol{\mu}(\mathbf{z}_i) \tag{97}$$

$$\tilde{\boldsymbol{\sigma}}(\mathbf{z}, \mathbf{z}_{1,\cdots,M}, \alpha_{0,\cdots,M}) = \alpha_0\boldsymbol{\sigma}(\mathbf{z}) + \sum_{i=1}^{M}\alpha_i\boldsymbol{\sigma}(\mathbf{z}_i) \tag{98}$$

And then apply the linearly mixed mean and std to the normalized feature map of content image $\mathbf{z}$, and use the decoder D to restore stylized images $\tilde{x}$:

$$\begin{aligned}\tilde{\mathbf{z}} &= \tilde{\boldsymbol{\mu}}(\mathbf{z}, \mathbf{z}_{1,\cdots,M}, \alpha_{0,\cdots,M}) \\ &\quad + \tilde{\boldsymbol{\sigma}}(\mathbf{z}, \mathbf{z}_{1,\cdots,M}, \alpha_{0,\cdots,M})\frac{\mathbf{z} - \boldsymbol{\mu}(\mathbf{z})}{\boldsymbol{\sigma}(\mathbf{z})}\end{aligned} \tag{99}$$

$$\tilde{x} = D(\gamma\tilde{\mathbf{z}} + (1-\gamma)\mathbf{z}) \tag{100}$$

where γ is hyperparameter to determine what percentage of the amplitude is involved in the Exploration.

Adversarial-attacks-inspired Update Strategy. Motivated by these theoretical insights, in each iteration, the parameters $\alpha_0, \alpha_1, \cdots, \alpha_M$ will be updated by maximizing the empirical risk of the augmented data, so that we could guide the augmented data to be closer to the distribution with highest empirical risk.

Inspired by adversarial attacks (Madry et al., 2018; Szegedy et al., 2014; Goodfellow et al., 2015; Zhang et al., 2020b), to speed up the process, after the gradient backpropagation, we update the parameters $\alpha_0, \alpha_1, \cdots, \alpha_M$ using the gradient's direction and fixed step size:

$$\tilde{\boldsymbol{\alpha}}^k = \boldsymbol{\alpha}^{k-1} + \mu \operatorname{sign}\left(\nabla_{\boldsymbol{\alpha}} \ell(\mathbf{f_w}; \hat{\mathbf{x}}^{k-1}, y)\right) \tag{101}$$

$$\alpha_l^k = \tilde{\alpha}_l^k / \sum_{i=0}^{M} \tilde{\alpha}_i^k \tag{102}$$

where $\boldsymbol{\alpha}^k = [\alpha_0^k, \alpha_1^k, \cdots, \alpha_M^k]^\top$, μ is the step size, $\hat{x}^k$ comes from Equ 97, 98, 99, 100 with α^k, ℓ is standard Cross-entropy loss, $\mathbf{f_w}$ is the network. Equ 102 means that since the sum is limited to 1, the parameters $\alpha_0, \alpha_1, \cdots, \alpha_M$ are normalized after each update.

After K iterations in inner optimization, we get $\boldsymbol{\alpha}^K$, and then we calculate $\hat{x}^{final}$ using Equ 97, 98, 99, 100 and $\boldsymbol{\alpha}^K$.

Finally, we use the original image and augmented image to compute the total loss :

$$\mathcal{L}_{MODE} = (1-\beta)\ell(\mathbf{f_w}; \mathbf{x}, y) + \beta \ell(\mathbf{f_w}; \hat{\mathbf{x}}^{final}, y) \tag{103}$$

where ℓ is standard Cross-entropy loss, β is hyperparameter to control the influence of augmented images.

The full AdaIN-based training algorithm is shown in Algorithm 3 and Figure 8. The overall approach is shown in Figure 9.

Algorithm 3 AdaIN-based MODE

Input: data x_i, batch size n, number of iterations K in inner optimization, step size μ, number of style providers M, network architecture parametrized by $\mathbf{w}$, hyperparameter β and γ
Output: Robust network $\mathbf{f_w}$
Randomly initialize network $\mathbf{f_w}$, or initialize network with pre-trained configuration
repeat
  Read mini-batch $\boldsymbol{x} = [\mathbf{x}_1, ..., \mathbf{x}_n]$ from training set
  Calculate their feature map $[\mathbf{z}_1 = E(\mathbf{x}_1), \cdots, \mathbf{z}_n = E(\mathbf{x}_n)]$
  Calculate their mean and std of feature map $[\boldsymbol{\mu}(\mathbf{z}_1), \cdots, \boldsymbol{\mu}(\mathbf{z}_n)]$, $[\boldsymbol{\sigma}(\mathbf{z}_1), \cdots, \boldsymbol{\sigma}(\mathbf{z}_n)]$ using Equ 94, 95
  # Exploration
  for $i = 1$ **to** n (in parallel) **do**
    Randomly select M other image's mean and std as style providers $[\mu_1, ..., \mu_M]$, $[\sigma_1, ..., \sigma_M]$
    Initialize $\alpha_0, \alpha_1, \cdots, \alpha_M$ as $\boldsymbol{\alpha}^0$
    for $k = 1$ **to** K **do**
      Calculate $\hat{\mathbf{x}}_i^k$ using Equ 97, 98, 99, 100 and $\boldsymbol{\alpha}^{k-1}$
      Update $\boldsymbol{\alpha}^k$ using Equ 101, 102
    end for
    Calculate $\hat{\mathbf{x}}_i^{final}$ using Equ 97, 98, 99, 100 and $\boldsymbol{\alpha}^K$
  end for
  # Update Model
  Calculate $\mathcal{L}_{MODE}$ using Equ 103
  Update $\mathbf{w}$ by performing one step gradient update using $\nabla \mathbf{w} \mathcal{L}_{MODE}$
until training converged

C. Discussion

C.1. Why exploring the worst-case for each sample rather than data distribution is more suitable

MODE explores the worst case for each sample, namely, $\min \mathbb{E}[\mathbf{max} \ell(x, y)]$. In contrast, DRO performs exploration for the data distribution, namely, $\min \mathbf{max} \mathbb{E}[\ell(x, y)]$.

Restricted capacity for searching distribution may make it hard to find the worst-case distribution. In contrast, the restricted capacity for searching worst-case samples is relatively easy for deep models (Su et al., 2019).

Moreover, it is challenging to exactly estimate a distribution under DG scenarios. Specifically, if we use a batch of samples to estimate and update the distribution, the estimated distribution could be imprecise, leading to bad explorations. But if we use all samples for the estimation in each iteration, the computational complexity could be excessive, particularly for high-resolution and large-scale datasets (e.g., DomainNet with 0.6M 224x224 images). A possible solution to address the challenge is to use the dual method (dual theorems for optimization need to be developed) for simplification. However, it is beyond the scope, as we mainly propose to perform moderately distributional exploration for DG. Thus, we will leave it as our future work. In contrast, performing sample-level exploration makes the distribution estimation unnecessary, bypassing the above issues.

C.2. MODE does not require domain ID

MODE does not require domain ID, as demonstrated in our theoretical proof and Algorithms. However, Domain ID, in certain situations, may provide a marginal performance gain (see Table below).

Table 3. The Impact of the Domain ID on MODE-A OfficeHome

MODE-A OfficeHome	A	C	P	R	Avg.
with domain ID	60.0	**57.3**	74.0	75.7	66.7
without domain ID	**60.1**	57.0	**74.2**	**76.0**	**66.8**

C.3. Whether learnable λ can bring additional improvements

In the initial design, we merely thought of a simple approach, where λ stands for the ratio of each domain. Namely, $\lambda = \frac{n_i}{N}$, where n_i is the number of samples in the ith domain and N denotes the total number.

We conduct experiments to investigate whether learnable λ (such as the way in Group DRO) can bring additional improvements. The results are shown below. We can see that making λ learnable can bring a certain performance gain, compared with a fixed λ.

Table 4. Leave-one-domain-out classification accuracies (in %) on PACS with Learnable λ

MODE-F	A	C	P	S	Avg.
Fixed λ	84.5±0.6	80.4±0.8	95.5±0.2	82.2±0.7	85.7
Learnable λ	84.1±0.7	**81.2±0.9**	95.1±0.3	**83.6±0.4**	**86.0**
MODE-A	A	C	P	S	Avg.
Fixed λ	84.4±0.9	81.9±0.9	95.2±0.3	85.8±0.3	86.9
Learnable λ	**85.5±1.2**	81.7±0.6	95.2±0.4	85.5±0.9	**87.1**

C.4. Why randomly select the style providers

We choose to randomly select other images as style providers instead of using fixed images as providers. The motivation is straightforward. Randomly selecting samples can diversify the styles used for exploration. Theoretical results indicate a trade-off between the size of the search space in exploration, while exploring more styles can help improve performance. Therefore, we provide different search spaces for each exploration.

To further verify the perspective, we conducted experiments under various experimental settings. In our experiments, the only difference between the two settings is whether the style provider is fixed or not. The results are presented in the following tables. Built upon the results, we find that randomly selecting style providers can indeed enhance the model's performance, demonstrating the rationality of the random selection mechanism.

Inspired by the mentioned approach of fixed-style providers, it is interesting to explore whether there exist optimal style providers for a given sample. This is an interesting and challenging problem that we would like to explore in our future work.

Table 5. Leave-one-domain-out classification accuracies (in %) on PACS with Fixed Style Provider

MODE-F	A	C	P	S	Avg.
Randomly	84.5±0.6	80.4±0.8	95.5±0.2	82.2±0.7	**85.7**
Fixed	83.1±0.5	79.2±0.7	95.4±0.4	81.1±0.3	84.4
MODE-A	A	C	P	S	Avg.
Randomly	84.4±0.9	81.9±0.9	95.2±0.3	85.8±0.3	**86.9**
Fixed	82.4±0.7	80.6±0.2	95.2±0.2	83.2±1.0	85.3

C.5. Why MODE-A outperforms MODE-F

The mentioned two realizations are related to the semantic and non-semantic partition. Thus, a possible explanation for the difference in model performance is that these two approaches have different abilities in partitioning semantic and non-semantic factors.

Regarding MODE-A, we use AdaIN (Huang & Belongie, 2017; Li et al., 2022) as the partition mechanism. AdaIN separates the semantic and non-semantic factors by processing the feature maps output by the model. The statistical features of the feature maps, such as mean and variance, are used as a good representation of the non-semantic factors, while the normalized feature maps are used as a good representation of the semantic factors. AdaIN creates new samples with different styles by applying different mean and std to the normalized feature map. Regarding MODE-F, we use the Fourier-based method (Xu et al., 2021) as the partition mechanism. The Fourier-based method assumes that the amplitude spectrum contains more style information, and the phase spectrum contains more semantic information. The Fourier-based method creates new samples with different styles by adjusting the amplitude spectrum of the samples.

In the Fourier-based method, it is difficult to produce a reasonable stylized image by directly adjusting the amplitude spectrum. This often adds some chaotic and disordered color blocks to the generated image, which are unlikely to occur in the real world and may even affect semantic factors. On the other hand, AdaIN with the help of a pre-trained network can combine the low-level style features with the original semantics of the image in a reasonable way, in line with human intuition. These more reasonable images with different styles will enable the model to better learn the distinctions and connections between semantic and non-semantic factors.

Furthermore, we believe that the difference between domains is not limited to the style difference that Fourier and AdaIN target. For example, viewing angle and distance of objects in images are not something that Fourier and AdaIN can change, but these kinds of domain shifts often exist in reality. However, we also believe that selecting appropriate mechanisms to address various domain shifts will consistently yield favorable outcomes with our framework.

C.6. More comparisons of work related to low confidence issues

Liu et al. (2021; 2022a) focuses on the uncertainty set in the DRO problem, and a Wasserstein distance is employed to determine the uncertainty. In contrast, MODE addresses the challenge when applying DRO to DG problems, where the overly large uncertainty set is shrunk to a subset through a semantic and non-semantic partition. Our semantic and non-semantic strategy is a unique contribution that distinguishes MODE from existing DRO methods (Liu et al., 2021; 2022a).

Previous work (Liu et al., 2021; 2022a) uses adversarial attacks to generate new samples for exploitation. In contrast, MODE employs style transformation methods commonly used in DG to generate new samples.

GroupDRO (Sagawa et al., 2020) explores the worst-case by leveraging group information to re-weight groups. In contrast, MODE explores the worst-case by constructing new samples.

Geometric Wasserstein DRO (Liu et al., 2022b) uses data geometry to construct more reasonable and effective uncertainty sets. In contrast, MODE shrinks the uncertainty set by introducing a semantic and non-semantic partition.

Topology-aware robust optimization (TRO) (Qiao & Peng, 2023) constructs the uncertainty using the data topology. In contrast, MODE constructs the uncertainty subset by constraining the search space with the same semantic factors. Thus, the main difference lies in how to constrain the uncertainty set.

Besides the above difference, previous methods perform exploration for the data distribution, namely, $\min \mathbf{max} \mathbb{E}[\ell(x, y)]$.

In contrast, MODE explores the worst case for each sample, namely, $\min \mathbb{E}[\mathbf{max}\ell(x, y)]$.

C.7. More comparisons of work related to data augmentation for non-semantic information

DSU (Li et al., 2022) focuses on addressing the uncertain nature of domain shifts by modeling feature statistics as uncertain distributions. This is achieved through the use of AdaIN, where non-semantic factors (i.e., feature map's mean and std) are replaced with randomly chosen values from the modeled distributions. By effectively modeling domain shifts with uncertainty, DSU significantly enhances the network's generalization ability.

CrossNorm and SelfNorm (Tang et al., 2021) address the problem of domain shift by developing two simple and efficient normalization methods that can reduce the non-semantic domain shift between different distributions. It has been discovered that processing the mean and variance of channels for samples or feature maps can help improve generalization ability. These methods are complementary and can be applied to various fields.

In contrast, we take a different approach to solving the problem of domain shift. We aim to improve the model's overall generalization ability by exposing it to more difficult domains during training. This min-max game is common in DRO, but simply applying DRO does not always lead to good results. Instead, we focus on constraining the exploration of semantic and non-semantic factors and propose a theoretical framework to demonstrate the feasibility of our approach.

We theoretically prove that actively improving the model's performance on a range of data distributions can help enhance its overall generalization ability, even if the final test domain is not included in the range of distributions explored. To achieve this, we use Fourier and AdaIN and actively search for the most challenging domains before each update step.

By generating new samples, we enable the model to explore more difficult domains. In contrast, CrossNorm and SelfNorm focus on designing a new normalization method that can be embedded into the model, processing the mean and variance of channels of feature maps.

Although MODE and DSU both use AdaIN to generate samples, DSU models non-semantic factors as a multivariate Gaussian distribution and randomly samples the factors within this distribution. In contrast, following our theoretical results, we actively explore more challenging non-semantic factors in the space, resulting in more challenging samples each time.

Moreover, we highlight the difference between our method and previous works (Tang et al., 2021; Li et al., 2022) through an empirical perspective. Specifically, we compare our baselines in experiments. We have since reproduced these two methods:

Table 6. Additional experiments about DSU (Li et al., 2022) and CNSN (Tang et al., 2021).

ResNet18 PACS	A	C	P	S	Avg.
CNSN (Tang et al., 2021)	83.6±0.3	79.1±0.3	96.5±0.1	80.2±0.3	84.8
DSU (Li et al., 2022)	83.1±0.3	79.8±0.4	96.3±0.1	77.3±0.1	84.1
MODE-F	84.5±0.6	80.4±0.8	95.5±0.2	82.2±0.7	85.7
MODE-A	84.4±0.9	81.9±0.9	95.2±0.3	85.8±0.3	86.9

We can see that our method can outperform the baselines. Besides the performance gain, we realize that it is necessary to consider the running time of each method. Accordingly, we also compare our method with our baselines, taking running time and FLOPs into consideration.

Table 7. Running Time and FLOPs of ResNet18 PACS

ResNet18 PACS	Running Time	FLOPs
CNSN (Tang et al., 2021)	25min	1x
DSU (Li et al., 2022)	35min	1.2x
MODE-F	4h	∼8x
MODE-A	5h	∼9x

It can be observed that due to the presence of the inner step, our method takes much longer running time than our baselines as the cost of promoting model performance. Thus, it is interesting to explore a more efficient approach to reduce the time cost while improving model performance, like Shafahi et al. (2019); Zhang et al. (2019). We thank the reviewer for the insightful comments and we will explore the exciting direction in our future work.

Table 8. Performance Comparison on CNSN (Tang et al., 2021) with different number of epochs. All results are conducted on the PACS dataset with Sketch(S) as the unknown target domain.

CNSN (Tang et al., 2021) num of epoch	50 epoch	100 epoch
Running Time (min)	30	59
Acc (%)	80.2	80.1

Table 9. Performance Comparison on different number of Inner Steps K in MODE-A. All results are conducted on the PACS dataset with Sketch(S) as the unknown target domain.

K	0 (Random)	1	2	3	4	5	6	7	8	9	10
Running Time (min)	41	68	95	122	157	189	211	242	278	301	331
Acc (%)	80.49	81.29	82.11	83.45	84.56	84.74	84.77	86.76	86.05	85.75	85.82

Our method is unable to adequately augment out-of-distribution (OOD) samples which have different semantic factors with training domains (Wang et al., 2022).

D. More Result and Implementation Details

D.1. Datasets

VLCS (Torralba & Efros, 2011) consists of 10,729 images from four domains, namely Caltech (C), Labelme (L), Pascal(V), Sun (S). There are five classes in each domain.

DomainNet (Peng et al., 2019) is a large-scale dataset designed for domain generalization, which contains 6.3 million images from 345 categories covering a wide range of visual domains. It has 6 domains: ClipArt(C), Infograph (I), Painting (P), Quickdraw (Q), Real (R) and Sketch (S).

Mini-Domainnet (Zhou et al., 2021a) is a highly challenging subset of DomainNet with a lower resolution (96x96) and 0.1M images. It has about 140K images with 126 classes and 4 domains: ClipArt(C), Painting (P), Real (R) and Sketch (S).

D.2. Implementation Details

Following the commonly used leave-one-domain-out strategy (Li et al., 2017; Xu et al., 2021), the model will be tested on one domain after training on all other domains.

Basic Details For VLCS, we use a pre-trained AlexNet backbone. We train the network using SGD optimizer with learning rate 5e-4, momentum 0.9, and weight decay 5e-4. We train the model for 50 epochs with batch size 32. The learning rate is decayed by 0.1 every 40 epochs.

For DomainNet, we use a pre-trained ResNet50 backbone. We train the network using adam optimizer with learning rate 2e-4 and weight decay 1e-4. We train the model for 50 epochs with batch size 256. The learning rate is decayed by 0.1 every 30 epochs.

For Mini-DomainNet, we use a pre-trained ResNet18 backbone. We train the network using sgd optimizer with learning rate 5e-3 and weight decay 5e-4. We train the model for 60 epochs with batch size 256. Cosine learning rate scheduler is used.

Method-specific Details The method-specific details are shown in D.4.

D.3. Experimental Results

Results on VLCS We show the Leave-one-domain-out classification accuracies (in %) on VLCS on Tab $D.3$. It can be observed that our approach achieves the highest average accuracy, but our result is only a little better than other methods. We think that it is because VLCS is different from other datasets in domain shift. All the data in VLCS are real-world images having complex compositions and background, which can't be handled well by Fourier-based transfer and AdaIN transfer. There will be better results by choosing a more suitable generation method to apply to our framework.

Results on DoaminNet We show the Leave-one-domain-out classification accuracies (in %) on VLCS on Tab 11. It can be observed that our approach achieves the higher average accuracy.

Table 10. Leave-one-domain-out classification accuracies (in %) on VLCS in AlexNet. The best and second-best results are highlighted in bold and underlined, respectively.

Dataset	VLCS				
Methods	C	L	V	S	Avg.
DeepAll (Zhou et al., 2020a)	96.3	59.7	70.6	64.5	72.8
MLDG (Li et al., 2018a)	97.9	59.5	66.4	64.8	72.2
Epi-FCR (Li et al., 2019)	94.1	64.3	67.1	65.9	72.9
MAML (Finn et al., 2017)	97.8	58	67.1	64.1	71.8
Jigen (Carlucci et al., 2019)	96.9	60.9	70.6	64.3	73.2
MMLD (Matsuura & Harada, 2020)	96.6	58.7	**72.1**	66.8	73.5
CICF (Li et al., 2021)	97.8	60.1	69.7	67.3	73.7
MASF (Dou et al., 2019)	94.8	**64.9**	69.1	67.6	74.1
MODE-F (ours)	**97.87**	61.17	69.54	**68.73**	74.33
MODE-A (ours)	96.92	63.05	70.28	67.97	**74.55**

Table 11. Leave-one-domain-out classification accuracies (in %) on DomainNet in ResNet50.

Methods	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Avg.
Baseline	66.35	23.01	50.48	13.82	63.57	50.79	44.67
MODE-F (ours)	**68.50**	23.14	**53.04**	15.92	**63.72**	**54.99**	**46.55**
MODE-A (ours)	68.26	**23.39**	52.45	**16.78**	63.05	53.96	46.31

Results on Mini-DoaminNet We show the Leave-one-domain-out classification accuracies (in %) on VLCS on Tab 12. It can be observed that our approach achieves the higher average accuracy.

Table 12. Leave-one-domain-out classification accuracies (in %) on Mini-DomainNet in ResNet18.

Methods	Clipart	Painting	Real	Sketch	Avg.
Baseline	59.04	47.20	**56.18**	51.74	53.54
MODE-F (ours)	60.63	48.09	54.92	**55.39**	54.75
MODE-A (ours)	**63.56**	**48.25**	55.87	52.69	**55.09**

D.4. The Overall Method-specific Details

The method-specific details of our approach are shown in Table D.4 and Table D.4.

Table 13. The method-specific details of the Fourier-based approach.

Dataset	Domain	Number of inner steps K	Inner step size μ	β	γ	The number of style providers M
Digit-DG	All	10	0.05	0.3	1	3
DomainNet	All	7	0.05	0.3	1	12
Others	All	10	0.05	0.3	1	8

Table 14. The method-specific details of the AdaIN-based approach.

Dataset	Domain	Number of inner steps K	Inner step size μ	β	γ	The number of style providers M
PACS	All	10	0.05	0.4	1	3 (Each domain provide one)
OfficeHome	C					
	other				0.3	8
VLCS	All					
Mini-DomainNet	All				1	
DomainNet	All	5				5 (Each domain provide one)

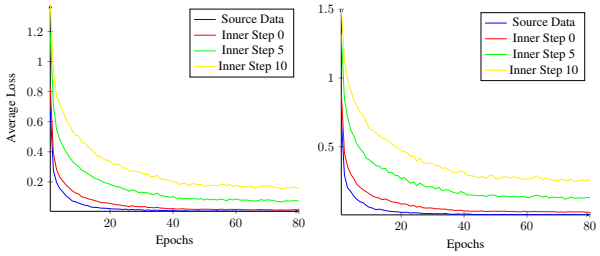

Figure 10. The average loss of augmented samples in different inner steps to the current model changes with the number of epochs in a training process. The left results are conducted on the PACS dataset with Art-painting (A) as the unknown target domain; the right results are conducted with Sketch (S) as the unknown target domain.

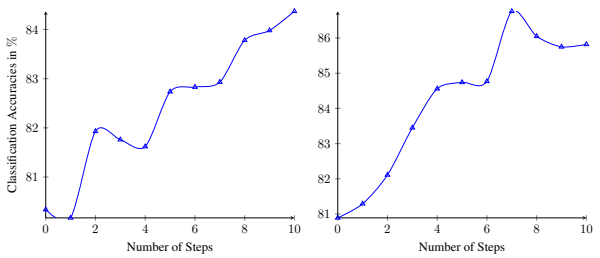

Figure 11. The effect of the number of inner steps. The left results are conducted on the PACS dataset with Art-painting (A) as the unknown target domain; the right results are conducted with Sketch (S) as the unknown target domain.

D.5. Visualization Results

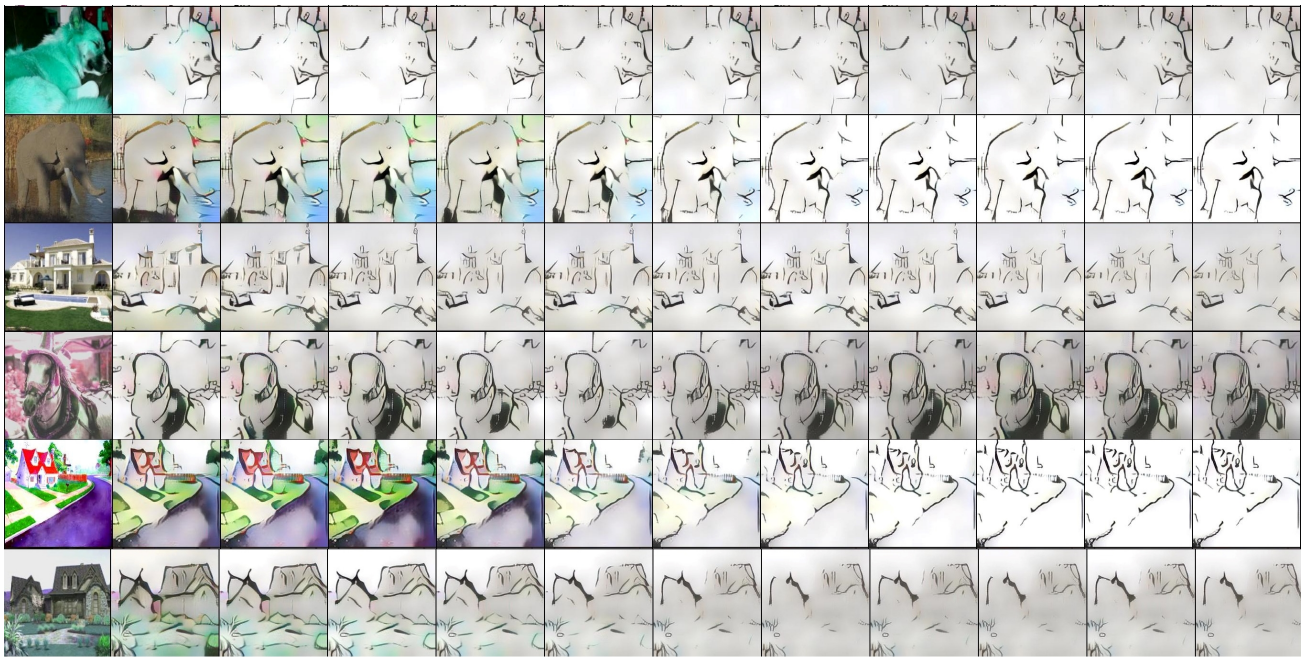

Figure 12. The changes of images during exploration when Ω is too large in our approach. In each row, the leftmost image is the original image, and from left to right is the result of each inner explore step of this image. The results are conducted on the PACS dataset with Art-painting (A) as the target with $\gamma = 1$ and $M = 10$.

Figure 13. The changes of images during exploration in our approach. In each row, the leftmost image is the original image, and from left to right is the result of each inner explore step of this image.