IMPROVING GENERALIZATION WITH DOMAIN CONVEX GAME

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain generalization (DG) tends to alleviate the poor generalization capability of deep neural networks by learning model with multiple source domains. A classical solution to DG is domain augmentation, the common belief of which is that diversifying source domains will be conducive to the out-of-distribution generalization. However, these claims are understood intuitively, rather than mathematically, and the relation between the diversity of source domains and model generalization still remains unclear. We thus made some explorations and found that the model generalization does not strictly improve with the increase of domain diversity, limiting the effectiveness of domain augmentation. In view of this observation, we propose a new perspective on DG that recast it as a convex game between domains. We formulate a regularization term based on the supermodularity property of convex game which rigorously demonstrates that the growth of domain diversity will enhance model generalization monotonically. This enables model to best utilize the rich information within input data so that each diversified domain contributes to model generalization. Furthermore, we construct a sample filter to eliminate the bad samples which contain unprofitable or even harmful information to generalization performance, such as noisy or redundant samples. Our framework presents a new avenue for the formal analysis of DG, the rationality and effectiveness of which have been demonstrated on extensive benchmark datasets.

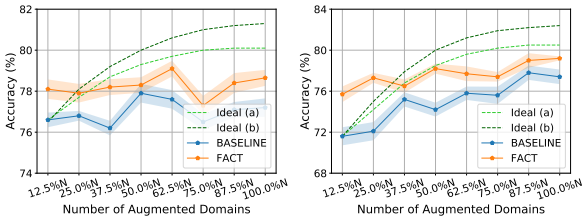
1 INTRODUCTION

Owning extraordinary representation learning ability, deep neural networks (DNNs) have achieved remarkable advancement on a variety of tasks when the training and test data are drawn from the same distribution (Goodfellow et al., 2016; He et al., 2016; LeCun et al., 2015). Whereas for out-of-distribution data, DNNs have demonstrated poor generalization capability since the i.i.d. assumption is violated, which is common in real-world conditions (Long et al., 2016; Ma et al., 2019; Taori et al., 2020). To tackle this issue, domain generalization (DG) has become a propulsion technology, aiming to learn a robust model from multiple source domains so that can generalize well to any unseen target domains with different statistics (Balaji et al., 2018; Muandet et al., 2013; Li et al., 2018b; 2017a).

Among extensive solutions to improve generalization, domain augmentation (Volpi et al., 2018; Shankar et al., 2018; Zhou et al., 2021a; Xu et al., 2021b) has been a classical and prevalent strategy, which focuses on exposing the model with more diverse domains via some augmentation techniques. A common belief is that generalizable models would become easier to learn when the training distributions become more diverse, which has been also emphasized by a recent work (Xu et al., 2021a). Notwithstanding the promising results shown by this strand of approaches, the claims above are vague and lack of theoretical justification, formal analyses of the relation between domain diversity and model generalization are sparse. Further, the transfer of knowledge may even hurt the performance on target domains in some cases, which is referred to as negative transfer (Pan & Yang, 2010; Tan et al., 2017). Thus the relation of domain diversity and model generalization remains unclear. In light of these points, we begin by considering the question: **The stronger the domain diversity, will it certainly help to improve the model generalization capability?**

We conduct a brief experiment to explore the relation between model generalization and domain diversity. The results presented in Fig 1 empirically reveal that with the increase of domain diversity,

i.e., the number of augmented domains, the model generalization (measured by the accuracy on unseen target domain) does not necessarily increase, but sometimes decreases instead, as the solid lines show. On the one hand, this may be because the model does not best utilize the rich information of diversified domains; on the other hand, it may be due to the existence of low-quality samples which contain redundant or noisy information that is unprofitable to generalization (Lee et al., 2018). This indicates that there is still room for improvement of the effectiveness of domain augmentation if we enable each training domain to be certainly conducive to the model generalization as the dash lines in Fig 1.



(a) Cartoon. (b) Sketch.

Figure 1: The relation between model generalization and domain diversity. N is the maximum number of augmented domains. Note that the *solid lines* denote the actual results of a BASELINE method that combines DeepAll with an augmentation strategy and a SOTA domain augmentation method FACT, while the *dash lines* represent the ideal relation in this work.

In this work, we therefore aim to ensure the monotonic increasing relation of model generalization with domain diversity to guarantee and further enhance the effectiveness of domain augmentation. To do this, we take inspiration from the literature of game theory and ask what can be achieved in a game where each player is required to bring profit to the coalition, which is referred to as convex game (Shapley, 1971; Ichiishi, 1981; Brânzei et al., 2003). In this view, we propose a novel framework named Domain Convex Game (DCG) which recasts DG as a convex game between domains since our key insight is to make each training domain bring benefit to model generalization that happens to fit the supermodularity property of convex game. On the one hand, we construct a novel regularization term based on the supermodularity property via meta-learning (Finn et al., 2017; Li et al., 2018a), which is direct to our insight and allows for a heuristic analysis of the rationality. This regularization encourages each diversified domain to contribute to improving model generalization, thus enables the model to better exploit the diverse information. On the other hand, considering that there may exist samples with unprofitable or even harmful information to generalization, we further design a sample filter to eliminate the bad samples such as noisy or redundant ones, so that their deterioration to the model generalization can be avoided. We also provide some heuristic insights in Appendix C to analyse the mechanisms behind and demonstrate the rationality, which indicates the proposed regularization will enforce domain consistency on discriminability to help the model improve when the model is not trained so well. Whereas when the model falls into local optima, it will further squeeze out the information within hard samples to help the model jump out. And the sample filter can filter out different types of low-quality samples during different training stages.

Nevertheless, it is well known that the supermodularity property also indicates increasing marginal contribution, which may not hold intuitively in DG, where the marginal contribution of domains is generally decreasing. One may wonder why our algorithm should work. Actually, to mitigate the gap between theory and practice, we impose a constraint on the naive supermodularity property to construct our regularization term, which enforces the regularization to work only under the case that the supermodularity is violated, i.e., when the marginal contribution of domains decreases. Thus, the limit of our regularization optimization is actually to keep a *constant marginal contribution*, rather than achieving an unsuitable *increasing marginal contribution*. Hence, our regularization can additionally regularize the decreasing speed of the marginal contribution as slow as possible by optimizing towards the limit situation, i.e., *constant marginal contribution*, just like changing the line *Ideal (a)* in Fig 1 into line *Ideal (b)*. Generally, the role of the proposed supermodularity is to encourage the contribution of each augmented domain, and further relieve the *decreasing marginal contribution* of domains to a certain extent, so as to better utilize the diversified information.

To summarize, we provide new insights into the relation of model generalization and source domain diversity, and cast DG as a convex game between domains. The proposed framework enables model to better utilize the information within diversified domains while avoiding the negative impact of bad samples, enabling each domain to contribute to improving generalization, so as to guarantee and further enhance the validity of domain augmentation. Furthermore, this new perspective provides theoretical support and formalizes several claims that until now only have been stated intuitively. The effectiveness and superiority are verified empirically across extensive real-world datasets.

2 RELATED WORK

Domain Generalization researches out-of-distribution generalization with knowledge only extracted from multiple source domains. A promising direction is to diversify training domains so as to improve generalization, referring as to domain augmentation (Shankar et al., 2018; Volpi et al., 2018; Zhou et al., 2021a; Xu et al., 2021b; Zhou et al., 2020b). L2A-OT (Zhou et al., 2020b) creates pseudo-novel domains from source data by maximizing an optimal transport-based divergence measure. CrossGrad (Shankar et al., 2018) generates samples from fictitious domains via gradient-based domain perturbation while AdvAug (Volpi et al., 2018) achieves so via adversarially perturbing images. MixStyle (Zhou et al., 2021a) and FACT (Xu et al., 2021b) mix style information extracted from different instances to synthetic novel domains. Instead of enriching domain diversity, another popular solution that learning domain-invariant representations by distribution alignment via kernel-based optimization (Muandet et al., 2013; Ghifary et al., 2017), adversarial learning (Li et al., 2018b; Motiian et al., 2017), or using uncertainty modeling (Li et al., 2022) demonstrate effectiveness for model generalization. Other recent DG works also explore low-rank decomposition (Piratla et al., 2020), self-supervised signals (Carlucci et al., 2019), gradient-guided dropout (Huang et al., 2020), etc. Our proposed framework builds on the domain augmentation group, where we provide a convex game perspective to guarantee and further enhance the efficacy of domain augmentation.

Convex Game is a highly interesting class of cooperative games introduced by Shapley (Shapley, 1971). A game is called convex when it satisfies the condition that the profit obtained by the cooperation of two coalitions plus the profit obtained by their intersection will not be less than the sum of profit obtained by the two respectively (a.k.a. supermodularity) (Shapley, 1971; Ichiishi, 1981; Brânzei et al., 2003). Co-Mixup(Kim et al., 2021) formulates the optimal construction of mixup augmentation data while encouraging diversity among them collectively by introducing supermodularity. Nevertheless, this method is applied to supervised learning which aims to construct salience mixed samples. Recently, (Rosenfeld et al., 2021) rethinks the single-round minmax setting of DG and recasts it as a repeated online game between a player minimizing risk and an adversary presenting test distributions in light of online convex optimization (Hazan, 2016). We note that the definition of convex game exploited in our work follows (Shapley, 1971), distinct from that in (Rosenfeld et al., 2021; Hazan, 2016). To the best of our knowledge, this work is the first to introduce convex game into DG to enhance generalization capability.

Meta Learning (Thrun & Pratt, 2012) is a long-term research exploring to learn how to train a particular model through the training of a meta-model, which has been successfully applied to few-shot learning (Li & Malik, 2017; Finn et al., 2017; Ravi & Larochelle, 2017). Recently, meta-learning has drawn increasing attention from DG community (Balaji et al., 2018; Dou et al., 2019; Li et al., 2019; 2018a). The main idea is to simulate domain shift during training by drawing virtual-train/test domains from the original source domains. MLDG (Li et al., 2018a) originates the episode training paradigm from (Finn et al., 2017), back-propagating the second-order gradients from an ordinary task loss on a random meta-test domain split from the source domains at each iteration. Subsequent meta learning-based DG methods utilize a similar strategy to meta-learn a regularizer (Balaji et al., 2018), feature-critic network (Li et al., 2019), or semantic relationships (Dou et al., 2019). Different from the former paradigm that purely leveraging gradient of task objective, which may cause sub-optimal, our proposed DCG, instead, constructs a novel regularization term with the ordinary task losses on meta-test domains, aiming to enforce each training domain to contribute to model generalization.

3 DOMAIN CONVEX GAME

Motivated by such an observation in Section 1, we propose Domain Convex Game (DCG) framework to train models that can best utilize domain diversity, as illustrated in Fig. 2. First, we cast DG as a convex game between domains and design a novel regularization term employing the supermodularity, which can encourage each domain to benefit model generalization. Further, we construct a sample filter based on the regularization term to eliminate bad samples which may cause negative effect on generalization. In this section, we define the problem setup and present the general form of DCG.

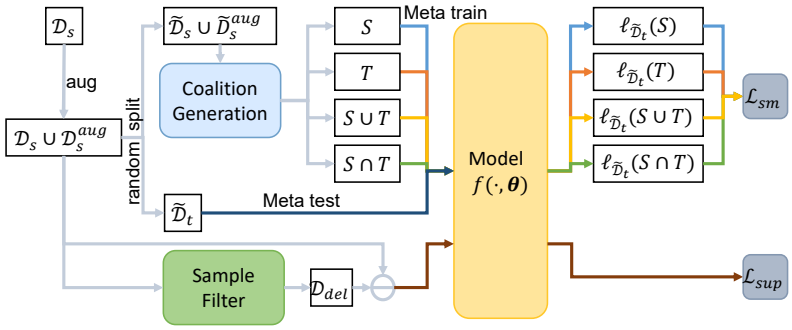


Figure 2: The pipeline of DCG. We first randomly split the diversified domains into meta-train and meta-test domains, and generate four coalitions from the former according to the definition of convex game. Then we conduct meta learning on the four coalitions respectively and construct our regularization loss utilizing the meta-test losses of them based on the supermodularity. Meanwhile, we eliminate the bad samples by a sample filter and calculate supervision loss on the retained samples.

3.1 Preliminary

Assuming that there are P source domains of data $\mathcal{D}_s = \cup_{k=1}^{P} D_k$ with n_k labelled samples $\{(\boldsymbol{x}_i^k, y_i^k)\}_{i=1}^{n_k}$ in the k-th domain D_k, where $\boldsymbol{x}_i^k$ and $y_i^k \in \{1, 2, \cdots, C\}$ denote the samples and corresponding labels. DG aims to train a domain-agnostic model $f(\cdot, \boldsymbol{\theta})$ parametrized with $\boldsymbol{\theta}$ on source domains that can generalize well on unseen target domain(s) $\mathcal{D}_t$. As an effective solution for DG, domain augmentation aims to enrich the diversity of source domains generally by synthesizing novel domains via mixing domain-related information, hence boosting model generalization (Zhou et al., 2020b; 2021a; Xu et al., 2021b). In this work, we exploit a Fourier-based augmentation technique inspired by (Xu et al., 2021b; Yang & Soatto, 2020) to prepare our diversified source domains. Owing to the property that the phase component of Fourier spectrum preserves high-level semantics of the original signal, while the amplitude component contains low-level statistics (Oppenheim & Lim, 1981; Piotrowski & Campbell, 1982), we augment the source data by distorting the amplitude information while keeping the phase information unchanged. Specifically, we mix the amplitude spectrum of an instance with that of another arbitrary instance by a linear interpolation strategy to synthesize augmented instances from novel domains. We refer readers to (Yang & Soatto, 2020; Xu et al., 2021b) for implementation details. Since each augmented sample is generated by mixing domain information of sample pairs from random source domains in a random proportion, it has statistics distinct from the others so that can be regarded as drawn from a novel augmented domain. Thus, we possess another Q augmented source domains of data $\mathcal{D}_s^{aug} = \cup_{k=1}^{Q} D_{P+k}$ with only one sample $\{(\boldsymbol{x}_i^{P+k}, y_i^{P+k})\}_{i=1}^{1}$ in the $(P+k)$-th domain D_{P+k}, where $\boldsymbol{x}_i^{P+k}$ and y_i^{P+k} denote the augmented samples and corresponding labels. Note that the number of augmented domains generated this way is equivalent to the total number of all the original samples since each original sample pair will generate a pair of augmented samples. The goal of DCG is to train a generalizable model $f(\cdot, \boldsymbol{\theta})$ for unseen target domain(s) $\mathcal{D}_t$ with the aid of all $P + Q$ diversified source domains $\mathcal{D}_s \cup \mathcal{D}_s^{aug}$.

3.2 Supermodularity Regularization Term

Let $M = \{1, 2, \cdots, m\}$ be a finite set of players and 2^M is the family of $2^{|M|}$ crisp subsets of M. A (crisp) cooperative game with player set M is a map $v : 2^M \to \mathbb{R}$. For $S \in 2^M$, $v(S)$ is called the worth of coalition S and it is interpreted as the amount of profit the coalition can obtain, when the players in S work together. Here a game v is called convex if it satisfies the *supermodularity property* (Brânzei et al., 2003; Shapley, 1971; Ichiishi, 1981), i.e., for each $S, T \in 2^M$:

$$v(S \cup T) + v(S \cap T) \geq v(S) + v(T). \tag{1}$$

The practical purpose of convex game is that the profit obtained by the cooperation of two coalitions plus the profit obtained by their intersection will not be less than the sum of profit obtained by the two respectively. Moreover, consider coalition $S \in 2^M \backslash \{\varnothing\}$ and two players i, j which are not in S, we can deduce *increasing marginal contribution property* for players according to Eq. 1:

$$v(S \cup \{j\}) - v(S) \leq v(S \cup \{i\} \cup \{j\}) - v(S \cup \{i\}). \tag{2}$$

We can see that convex game requires that each player contributes to the coalition, which happens to meet our expectation for DG, where each training domain is required to benefit model generalization. Besides, convex game also possesses *increasing marginal contribution property* for players, which may not hold in DG, however, this property does not hinder our goal, but can further regularize the model and alleviate the *decreasing marginal contribution* for domains, as discussed in Section 1.

Thus, we first cast DG as a convex game between domains. To achieve this, at each training iteration, we randomly split the original source data $\mathcal{D}_s$ into $P - V$ meta-train domains of data $\tilde{\mathcal{D}}_s$ and V meta-test domains of data $\tilde{\mathcal{D}}_t$, where $\tilde{\mathcal{D}}_s$ and $\tilde{\mathcal{D}}_t$ share no domain. Then we pick out the augmented domains generated by data in $\tilde{\mathcal{D}}_s$, denoted as $\tilde{\mathcal{D}}_s^{aug}$, and incorporate them into the meta-train domains. This strategy to conduct meta-train/test domains is to mimic the real train-test domain shift in domain augmentation strand, which is discussed in Section 4.4. Then, since one domain may contain multiple samples, we specifically consider involving a specific convex game: *convex fuzzy game* (Brânzei et al., 2003) where each player (i.e., each domain) can be partitioned into multiple parts. And each part represents a sample in DG. Thus, we have a finite set of partitioned players $\tilde{M} = \tilde{\mathcal{D}}_s \cup \tilde{\mathcal{D}}_s^{aug}$. We can obtain coalitions $S, T \in 2^{\tilde{M}}$ by randomly sampling two sets of data from meta-train data $\tilde{\mathcal{D}}_s \cup \tilde{\mathcal{D}}_s^{aug}$, respectively. And $S \cup T, S \cap T$ can be naturally constructed by the union and intersection of S and T. As for the profit $v(O), O \in \{S, T, S \cup T, S \cap T\}$, we take the generalization performance in meta-test evaluated on virtual-test domains $\tilde{\mathcal{D}}_t$ after the meta-training on each coalition O as the value of profit $v(O)$. We quantify the generalization performance by the negative empirical risk loss on meta-test domains, then according to the supermodularity property in Eq. 1 we can have:

$$\ell_{\tilde{\mathcal{D}}_t}(S \cup T) + \ell_{\tilde{\mathcal{D}}_t}(S \cap T) \leq \ell_{\tilde{\mathcal{D}}_t}(S) + \ell_{\tilde{\mathcal{D}}_t}(T), \tag{3}$$

$\ell_{\tilde{\mathcal{D}}_t}(O)$ denotes the loss evaluated on $\tilde{\mathcal{D}}_t$ by the model trained on coalition O and $v(O) = -\ell_{\tilde{\mathcal{D}}_t}(O)$.

Specifically, assuming a loss function $\ell(f(\boldsymbol{x}, \boldsymbol{\theta}), y)$ for a sample between its output and label, e.g., cross-entropy loss for classification task, we conduct virtual training on the four coalitions $\{S, T, S \cup T, S \cap T\}$, respectively, with the optimization objective:

$$\mathcal{F}(O) := \sum_{x \in O} \ell(f(\boldsymbol{x}, \boldsymbol{\theta}), y)), O \in \{S, T, S \cup T, S \cap T\}. \tag{4}$$

Then the updated virtual parameters $\boldsymbol{\theta}'$ can be computed using one step of gradient descent:

$$\boldsymbol{\theta}' = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{F}(O), \tag{5}$$

where α is the step size. Thus, we can have the corresponding meta-test loss evaluated on the virtual-test domains $\tilde{\mathcal{D}}_t$ as below:

$$\mathcal{G}(\boldsymbol{\theta}') := \mathbb{E}_{\boldsymbol{x} \in \tilde{\mathcal{D}}_t} \ell(f(\boldsymbol{x}, \boldsymbol{\theta}'), y). \tag{6}$$

This objective simulates test on new domains, which can measure the model generalization obtained by training with one specific coalition. Hence, the supermodularity regularization term can be constructed naturally utilizing the four meta-test losses of the four coalitions, based on Eq. equation 3:

$$\begin{aligned} \mathcal{L}_{sm} = \max\{0, &\mathcal{G}(\boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{F}(S \cup T)) + \mathcal{G}(\boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{F}(S \cap T)) \\ &- \mathcal{G}(\boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{F}(S)) - \mathcal{G}(\boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{F}(T))\}. \end{aligned} \tag{7}$$

Note that we exploit a $max(0, \cdot)$ function combined with the pure supermodularity property to construct our regularization. In this way, $\mathcal{L}_{sm} > 0$ only when the inequality in Eq. 3 is violated, i.e., the domain marginal contribution is decreasing. Thus, the limit of our regularization optimization corresponds to constant marginal contribution, not the inappropriate increasing marginal contribution. Therefore, this regularization term can not only encourage each training domain to contribute to model generalization, but also alleviate the decrease of marginal contributions to some extent, enabling the model to fully leverage the rich information in diversified domains.

3.3 Sample Filter

Through the optimization of the regularization term, the model will be trained to better utilize the rich information of diversified source domains. However, what we cannot avoid is that there may exist some bad samples with low quality, which indicates damage to model generalization. For instance,

noisy samples will disturb model to learn generalizable knowledge; while redundant samples may lead to overfitting that hinder the model from learning more diverse patterns.

In this view, we further conduct a sample filter based on the regularization term to eliminate bad samples. Considering that the proposed regularization aims to penalize the decreasing marginal contribution of domains and then better utilize the diverse information, the samples that contribute more to the regularization loss (i.e., cause larger increase) are more unfavorable to our goal, hindering the improvement of model generalization. Thus, we try to measure the contribution of each input to our regularization loss and define the contribution as its score. Specifically, we apply a simple but powerful interpretation technique inspired by (Bach et al., 2015), in which they define the contribution of each input to the prediction by introducing layer-wise relevance propagation. Essentially it is equivalent to an elementwise product between the saliency maps of (Simonyan et al., 2014) and the input, i.e., Gradient $\times$ Input. Accordingly, the score of each input can be formulated as follows:

$$score = \boldsymbol{x}^T \nabla_{\boldsymbol{x}} \mathcal{L}_{sm}, x \in \tilde{\mathcal{D}}_s \cup \tilde{\mathcal{D}}_s^{aug}. \tag{8}$$

The higher the score of the sample, the greater the regularization loss will be increased caused by it, and the more it will hinder model from benefiting from diversified domains. Therefore, we pick out the samples with the top-k score, denoted as $\mathcal{D}_{del}$, and cast them away when calculating the supervision loss for diversified source domains to eliminate the negative effect of low quality samples:

$$\mathcal{L}_{sup} = \mathbb{E}_{\boldsymbol{x} \in \mathcal{D}_s \cup \mathcal{D}_s^{aug} \setminus \mathcal{D}_{del}} \ell(f(\boldsymbol{x}, \boldsymbol{\theta}), y). \tag{9}$$

Thus, we optimize the regularization loss to enable model to better utilize the rich information within diversified domains. In the meanwhile, we eliminate the bad samples (e.g, noisy samples, redundant samples, etc) by the sample filter to avoid their negative effects. Moreover, it is found that different types of low-quality samples are more likely to be discarded in different training stages, as discussed in Appendix C. And we have explored out that low quality sample filtering is necessary for both original and augmented samples in Section 4.4.

The overall optimization objective is:

$$\arg\min_{\boldsymbol{\theta}} \mathcal{L}_{sup} + \omega \mathcal{L}_{sm}, \tag{10}$$

where ω weights the supervision loss and the regularization term. The overall methodological flow is illustrated schematically in Fig. 2 and summarized in Algorithm 1. Besides, to further validate the rationality of DCG, we make some mathematical derivations and then provide some high-level insights and heuristic analysis about the mechanisms behind the effectiveness in Appendix C.

4 EXPERIMENTS

4.1 DATASET AND IMPLEMENTATION DETAILS

To evaluate our method, we conduct extensive experiments on three popular benchmarks for domain generalization: **PACS**(Li et al., 2017b) is an object recognition benchmark that covers 9991 images of 7 categories from four different domains, i.e., Art, Cartoon, Photo and Sketch, which with large discrepancy in image styles. **Office-Home**(Venkateswara et al., 2017) is a commonly-used benchmark including four domains (Art, Clipart, Product, RealWorld). It contains 15,500 images of 65 classes in total. **mini-DomainNet**(Zhou et al., 2021b) is a very large-scale domain generalization benchmark consists of about 140k images with 126 classes from four different domains (Clipart, Painting, Real, Sketch). For all benchmarks, we conduct the commonly used leave-one-domain-out experiments (Li et al., 2017a). We adopt ResNet-18 pre-trained on ImageNet (He et al., 2016) as backbone for all datasets. We train the network using mini-batch SGD with batch size 16, momentum 0.9 and weight decay 5e-4. The initial learning rate is 0.001 and decayed by 0.1 at 80% of the total epochs. The meta step size α is set to be the same as the learning rate. For hyper-parameters, we set $\omega = 0.1$ and $k = 5$ for all experiments, which are selected on validation set following standard protocol. All results are reported based on the average accuracy over three independent runs. More details and results with error bars are provided in Appendix D, E.

4.2 EXPERIMENTAL RESULTS

Results on PACS are summarized in Table 1. It is clear that our DCG achieves the best performance among all the competitors. Our method is the first to reach 86% average accuracy on PACS dataset, which exceeds the DeepAll baseline by 6.4%. We notice that DCG surpasses the domain augmentation based methods L2A-OT, MixStyle and FACT by a large margin of $3.5\%, 2.6\%$ and 1.8% respectively, which indicates the importance of encouraging each domain to contribute to model generalization. Specifically, on the harder target domains Cartoon and Sketch, our method still outperforms the SOTA with a large margin of 0.5% and 1.2% respectively. All the comparisons reveal the effectiveness of DCG and further demonstrate that the convex game between domains improves model generalization.

Table 1: Leave-one-domain-out results on PACS.

Methods	Art	Cartoon	Photo	Sketch	Avg.
DeepAll(Xu et al., 2021b)	77.63	76.77	95.85	69.50	79.94
MLDG (Li et al., 2018a)	78.70	73.30	94.00	65.10	80.70
MASF (Dou et al., 2019)	80.29	77.17	94.99	71.69	81.04
L2A-OT (Zhou et al., 2020b)	83.30	78.20	<u>96.20</u>	73.60	82.80
DDAIG (Zhou et al., 2020a)	84.20	78.10	95.30	74.70	83.10
RSC (Huang et al., 2020)	83.43	<u>80.31</u>	95.99	<u>80.85</u>	<u>85.15</u>
MixStyle (Zhou et al., 2021a)	84.10	78.80	96.10	75.90	83.70
FACT (Xu et al., 2021b)	<u>85.37</u>	78.38	95.15	79.15	84.51
DSU (Li et al., 2022)	83.60	79.60	95.80	77.60	84.10
ITL-Net(Gao et al., 2022)	83.90	78.90	94.80	80.10	84.40
DCG (*ours*)	**85.94**	**80.76**	**96.41**	**82.08**	**86.30**

Results on Mini-DomainNet are shown in Table 2. The much larger number of categories and images makes DomainNet a much more challenging benchmark than PACS. DCG still achieves the state-of-the-art performance of 65.18%, surpassing the SOTA by a large margin of 2.31%, further validating the efficacy of DCG.

Table 2: Leave-one-domain-out results on mini-DomainNet.

Methods	Clipart	Painting	Real	Sketch	Avg.
DeepAll	65.30	58.40	64.70	59.00	61.86
ERM (Vapnik, 1999)	65.50	57.10	62.30	57.10	60.50
MLDG (Li et al., 2018a)	65.70	57.00	63.70	58.10	61.12
Mixup (Zhang et al., 2018)	<u>67.10</u>	59.10	64.30	59.20	62.42
MMD (Lee et al., 2019)	65.00	58.00	63.80	58.40	61.30
SagNet (Nam et al., 2019)	65.00	58.10	64.20	58.10	61.35
CORAL (Peng et al., 2019)	66.50	<u>59.50</u>	<u>66.00</u>	<u>59.50</u>	<u>62.87</u>
MTL (Blanchard et al., 2021)	65.30	59.00	65.60	58.50	62.10
DCG (*ours*)	**69.38**	**61.79**	**66.34**	**63.21**	**65.18**

Results on Office-Home are presented in Table 3, where we beat all the compared baselines in terms of the average accuracy. Due to the similarity to the pre-trained dataset ImageNet, DeepAll acts as a strong baseline on Office-Home. Many previous DG methods, such as MLDG, SagNet, and RSC, can not improve over the simple DeepAll baseline. Nevertheless, our DCG achieves a consistent improvement over DeepAll on all the held-out domains. Moreover, DCG surpasses the latest domain augmentation methods L2A-OT and FACT. The incremental advantages may be due to the relatively smaller domain shift, where the decreasing marginal contribution of domains is more severe.

Table 3: Leave-one-domain-out results on Office-Home.

Methods	Art	Clipart	Product	Real	Avg.
DeepAll	57.88	52.72	73.50	74.80	64.72
MLDG (Li et al., 2018a)	52.88	45.72	69.90	72.68	60.30
SagNet Nam et al. (2019)	60.20	45.38	70.42	73.38	62.34
RSC (Huang et al., 2020)	58.42	47.90	71.63	74.54	63.12
DDAIG (Zhou et al., 2020a)	59.20	52.30	74.60	76.00	65.50
L2A-OT (Zhou et al., 2020b)	<u>60.60</u>	50.10	<u>74.80</u>	**77.00**	65.60
MixStyle (Zhou et al., 2021a)	58.70	53.40	74.20	75.90	65.50
FACT (Xu et al., 2021b)	60.34	<u>54.85</u>	74.48	76.55	<u>66.56</u>
STNP (Kang et al., 2022)	59.55	55.01	73.57	75.52	65.89
DSU (Li et al., 2022)	60.20	54.80	74.10	75.10	66.10
DCG (*ours*)	**60.67**	**55.46**	**75.26**	<u>76.82</u>	**67.05**

4.3 Analysis

Ablation Study. In Table 4, we investigate the role of each component in DCG, which consists of Fourier augmentation (Aug.), supermodularity regularization (Reg. ($\mathcal{L}_{sm}$)) and sample filter (Filter. ($\mathcal{F}_{sm}$)). The Baseline is trained only with the supervision loss of all the original source data. Based on it, we diversify source domains by Fourier augmentation to obtain Model 1, which improves much over Baseline, showing the success of domain augmentation. Further incorporating our supermodularity regularization $\mathcal{L}_{sm}$ results in Model 3, which is superior to Model 1, demonstrating the effectiveness of encouraging each diversified domain to contribute to generalization. Besides, we replace our $\mathcal{L}_{sm}$ by a regularization $\mathcal{L}_{maml}$ which sums the meta-test losses of all the tasks as MAML (Finn et al., 2017), the improvement of Model 3 over Model 2 indicates conducting

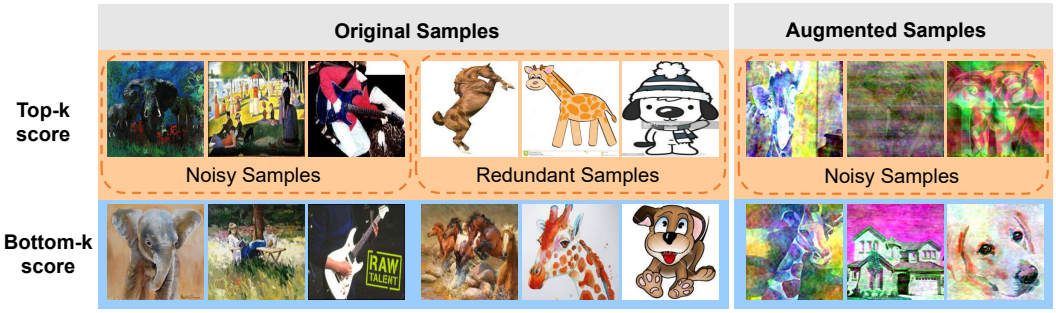
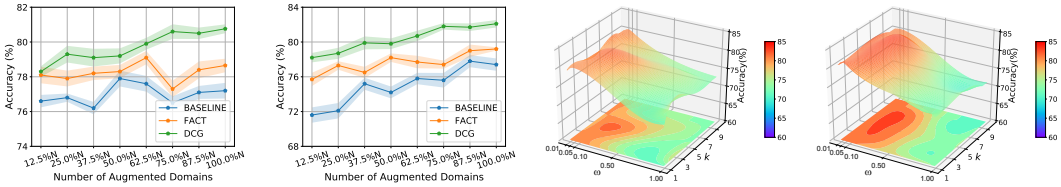


Figure 3: The visualization of samples with top-k and bottom-k score respectively.



(a) Cartoon. (b) Sketch. (c) Cartoon. (d) Sketch.

Figure 4: (a)(b): relation between model generalization and domain diversity; (c)(d): sensitivity to hyper-parameters ω and k; with Cartoon and Sketch on PACS dataset as the unseen target domain.

convex game between domains is more helpful to generalization than simply applying the meta loss. Comparing Model 5 with Model 1, we can observe that the proposed sample filter is also conducive to generalization, suggesting the importance of eliminating nonprofitable information. And the improvement of Model 5 over Model 4 can further prove our superiority. Finally, DCG performs best in all variants, indicating that the two proposed components complement and benefit each other.

Table 4: Ablation study of DCG on PACS dataset.

Method	Aug.	Reg.	Filter.	Art	Cartoon	Photo	Sketch	Avg.
Baseline	-	-	-	77.6	76.8	95.9	69.5	79.9
Model 1	✓	-	-	83.9	77.0	95.6	77.4	83.4
Model 2	✓	$\mathcal{L}_{maml}$	-	84.7	79.0	95.7	80.1	84.9
Model 3	✓	$\mathcal{L}_{sm}$	-	85.1	80.1	95.9	81.4	85.6
Model 4	✓	-	$\mathcal{F}_{maml}$	84.1	77.7	95.5	78.2	83.9
Model 5	✓	-	$\mathcal{F}_{sm}$	84.4	78.2	95.8	79.3	84.4
Model 6	✓	$\mathcal{L}_{maml}$	$\mathcal{F}_{maml}$	85.3	79.9	96.0	81.5	85.7
DCG	✓	$\mathcal{L}_{sm}$	$\mathcal{F}_{sm}$	**85.9**	**80.8**	**96.4**	**82.1**	**86.3**

Generalization with Domain Diversity. Figure 4a, 4b show the model generalization with the increase of domain diversity. We use the classification accuracy on the held-out target domain as the metric of model generalization across domains, and the number of augmented domains to measure the domain diversity. It is clear that on both Cartoon and Sketch tasks, the model generalization capability of the baseline method BASELINE and the SOTA method FACT do not necessarily improve with the increase of domain diversity, but sometimes decrease instead. While in our DCG, the model generalization increases monotonically with the domain diversity on the whole and the decrease of marginal contribution of domains is alleviated. Meanwhile, in a few cases, the generalization of our model drops a little when domain diversity increases. This is reasonable since the additional augmented domains may be low-quality or nonprofitable to generalization. These results demonstrate that our method indeed encourages each diversified domain to contribute to model generalization, hence further improving the performance of domain augmentation based methods.

Visualization of Filtered Samples. To visually verify the claim that our sample filter can effectively eliminate the low-quality samples by discarding the samples with top-k score, we provide the samples that have been discarded the most times in the whole training process, as well as the samples that always possess the bottom-k score (a.k.a., high-quality samples). The results on the task with Cartoon as unseen target domain are shown in Figure 3. We can see that the discarded original samples in the first row either be noisy images that have messy background and fuzzy objects, or images containing naive or classical information which may be redundant. While the high-quality original images in the bottom row are all vivid and rich in information. As for the augmented samples, the discarded ones are almost distinguishable while the retained high-quality ones are limpid. These comparisons demonstrate the effectiveness of our sample filter, further validate the superiority of DCG.

Sensitivity of Hyper-parameters. Figure 4c, 4d show the sensitivity of DCG to hyper-parameters ω and k. Specifically, the value of ω varies from $\{0.01, 0.05, 0.1, 0.5, 1, 0\}$, while k changes from $\{1, 3, 5, 7, 9\}$. It can be observed that DCG achieves competitive performances robustly under a wide range of hyper-parameter values, i.e., $0.05 \leq \omega \leq 0.3$ and $3 \leq k \leq 7$, in either task Cartoon or Sketch, which further verifies the stability of our method.

4.4 DISCUSSION

Can the augmented samples be regarded as independent novel domains? Since DCG considers all the diversified source domains $\mathcal{D}_s \cup \mathcal{D}_s^{aug}$ into training, how to conduct the meta training and meta testing domains is turned out to be a problem. In Section 3.2, we randomly split the original source domains $\mathcal{D}_s$ into meta-train and meta-test domains first, next pick out the domains in $\mathcal{D}_s^{aug}$ that are augmented by the current meta-train domains and then merge them into together. In this way, there is no domain augmented by the meta-test domains in the meta-train domains, and vice versa. However, why don't we just randomly split all the diversified source domains, i.e., $\mathcal{D}_s \cup \mathcal{D}_s^{aug}$, into two parts, since each diversified domain can be regarded as a novel domain? We conduct experiments of this variant and the results is shown in the first line in Table 5. We can see that the variant achieves inferior performance to DCG. This may be because the synthetic novel domains still contain part of the domain-related information of the original ones, thus they can not be considered completely independent with each other. In this view, the strategy to conduct meta train/ test domains in Section 3.2 can guarantee the meta-test domains are completely unseen, which better simulates the domain shift between the diversified source domains and the held-out unseen target domain.

Table 5: Leave-one-domain-out results on PACS.

Methods	Art	Cartoon	Photo	Sketch	Avg.
Random_meta_split	85.6	80.2	96.0	81.8	85.9
Filter_only_on_aug	85.4	80.6	96.7	81.8	86.1
Filter_only_on_ori	85.2	80.0	96.5	82.3	86.0
DCG	**85.9**	**80.8**	**96.4**	**82.1**	**86.3**

Is low-quality sample filtering necessary for both original and augmented samples? To explore whether the sample filter is necessary for both original and augmented samples, we conduct experiments that apply the proposed sample filter only on the original samples or augmented samples and the results are shown in Table 5. It can be seen that only filtering the original samples or the augmented ones both suffer from a performance drop, which indicates that there exist bad samples among both original and augmented samples. Limiting the filtering range will make some low-quality samples be retained to participate in the training process, which will damage the model generalization. Besides, the performance of only filtering the original samples is slightly lower than that of only filtering the augmented ones, which should be due to the augmented samples being less natural.

5 CONCLUSION & OUTLOOK

This work introduces a novel perspective based on game theory into domain generalization and casts DG as a convex game between domains. We explore the relation of model generalization and domain diversity, the main idea is to make each diversified domain contribute to the improvement on generalization, validating and further enhancing the effectiveness of domain augmentation strand. We then propose a framework consists of a regularization term based on the supermodularity of convex game, which encourages model to best utilize the rich information within diversified domains, and a sample filter to eliminate the bad samples that will hamper generalization. Heuristic analysis demonstrate the mechanisms and rationales behind our method, and extensive experiments on real-world applications empirically show the effectiveness of DCG.

Considering this work is a pioneering exploration of domain diversity and model generalization, there are two limitations could be explored in future research. First, the use of meta-learning inevitably leads to a decrease in training efficiency. Second, our approach can be valuable to other domain augmentation methods since the domain convex game we perform can be easily generalized to the methods which construct augmented domains explicitly. However, there exist some methods that diversify source domains implicitly and our method cannot be directly applied. It remains an open problem to generalize our mechanism to all the domain augmentation methods in a more efficient and general way for future exploration. Nevertheless, we believe our work can inspire the future work of enriching domain diversity with improved generalization capability.

REFERENCES

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7), 2015.

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *NeurIPS*, pp. 1006–1016, 2018.

Gilles Blanchard, Aniket Anand Deshmukh, Ürün Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *J. Mach. Learn. Res.*, pp. 2:1–2:55, 2021.

Rodica Brânzei, Dinko Dimitrov, and Stef Tijs. Convex fuzzy games and participation monotonic allocation schemes. *Fuzzy sets and systems*, 139(2):267–281, 2003.

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

Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *NeurIPS*, pp. 6447–6458, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, volume 70, pp. 1126–1135, 2017.

Boyan Gao, Henry Gouk, Yongxin Yang, and Timothy M. Hospedales. Loss function learning for domain generalization by implicit gradient. In *ICML*, volume 162, pp. 7002–7016, 2022.

Muhammad Ghifary, David Balduzzi, W. Bastiaan Kleijn, and Mengjie Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *TPAMI*, 39(7): 1414–1430, 2017.

Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016.

Elad Hazan. Introduction to online convex optimization. *Found. Trends Optim.*, 2(3-4):157–325, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Zeyi Huang, Haohan Wang, Eric P. Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *ECCV*, pp. 124–140, 2020.

Tatsuro Ichiishi. Super-modularity: applications to convex games and to the greedy algorithm for lp. *Journal of Economic Theory*, 25(2):283–286, 1981.

Juwon Kang, Sohyun Lee, Namyup Kim, and Suha Kwak. Style neophile: Constantly seeking novel styles for domain generalization. In *CVPR*, pp. 7130–7140, June 2022.

Jang-Hyun Kim, Wonho Choo, Hosan Jeong, and Hyun Oh Song. Co-mixup: Saliency guided joint mixup with supermodular diversity. In *ICLR*, 2021.

Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nat.*, 521(7553):436–444, 2015.

Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*, pp. 10285–10295, 2019.

Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *CVPR*, pp. 5447–5456, 2018.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, pp. 5543–5551, 2017a.

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

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, pp. 3490–3497, 2018a.

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

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

Xiaotong Li, Yongxing Dai, Yixiao Ge, Jun Liu, Ying Shan, and Lingyu Duan. Uncertainty modeling for out-of-distribution generalization. In *ICLR*, 2022.

Yiying Li, Yongxin Yang, Wei Zhou, and Timothy M. Hospedales. Feature-critic networks for heterogeneous domain generalization. In *ICML*, pp. 3915–3924, 2019.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pp. 2999–3007. IEEE Computer Society, 2017.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *NeurIPS*, pp. 136–144, 2016.

Xinhong Ma, Tianzhu Zhang, and Changsheng Xu. Deep multi-modality adversarial networks for unsupervised domain adaptation. *IEEE Trans. Multim.*, 21(9):2419–2431, 2019.

Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, pp. 5716–5726, 2017.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, pp. 10–18, 2013.

Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap via style-agnostic networks. *CoRR*, abs/1910.11645, 2019.

A. V. Oppenheim and J. S. Lim. The importance of phase in signals. *Proc IEEE*, 69(5):529–541, 1981.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22 (10):1345–1359, 2010.

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

L. N. Piotrowski and F. W. Campbell. A demonstration of the visual importance and flexibility of spatial-frequency amplitude and phase. *Perception*, 11(3):337–46, 1982.

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

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. An online learning approach to interpolation and extrapolation in domain generalization. *CoRR*, abs/2102.13128, 2021.

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

Lloyd S Shapley. Cores of convex games. *International journal of game theory*, 1(1):11–26, 1971.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR*, 2014.

Ben Tan, Yu Zhang, Sinno Jialin Pan, and Qiang Yang. Distant domain transfer learning. In *AAAI*, pp. 2604–2610, 2017.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. In *NeurIPS*, 2020.

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

Vladimir Vapnik. An overview of statistical learning theory. *IEEE Trans. Neural Networks*, 10(5): 988–999, 1999.

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

Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C. Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *NeurIPS*, pp. 5339–5349, 2018.

Keyulu Xu, Mozhi Zhang, Jingling Li, Simon Shaolei Du, and Stefanie Jegelka. How neural networks extrapolate: From feedforward to graph neural networks. In *ICLR*, 2021a.

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

Yanchao Yang and Stefano Soatto. FDA: fourier domain adaptation for semantic segmentation. In *CVPR*, pp. 4084–4094, 2020.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

Kaiyang Zhou, Yongxin Yang, Timothy M. Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI*, pp. 13025–13032, 2020a.

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

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021a.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain adaptive ensemble learning. *IEEE Transactions on Image Processing*, 30:8008–8018, 2021b.

A SOCIAL IMPACT & LIMITATION

Our work focuses on domain generalization and attempts to make each training domain contribute to model generalization, which validates and further enhances the effectiveness of domain augmentation strand. This method produces a positive impact on the society and community, saves the cost and time of data annotation, boosts the reusability of knowledge across domains, and greatly improves the efficiency. Nevertheless, this work suffers from some negative influences, which is worthy of further research and exploration. Specifically, more jobs of classification or target detection for rare or variable conditions may be cancelled. Moerover, we should be cautious about the result of the failure of the system, which could render people believe that classification was unbiased. Still, it might be not, which might be misleading, e.g., when using the system in a highly variable unseen target domain.

B ALGORITHM

The algorithm of our proposed Domain Comvex Game is summarized as follows:

Algorithm 1 The Algorithm of Domain Convex Game.

Input: $P+Q$ diversified source domains $\mathcal{D}_s \cup \mathcal{D}_s^{aug}$; Hyper-parameters: α, ω, k.

randomly initialize model parameters $\boldsymbol{\theta}$.
for iter in iterations **do**
  Randomly sample a mini-batch of $\mathcal{D}_s$ as B and a mini-batch of $\mathcal{D}_s^{aug}$ as B^{aug}.
  Split: $\tilde{\mathcal{D}}_s$ and $\tilde{\mathcal{D}}_t \leftarrow B$, Pick out: $\tilde{\mathcal{D}}_s^{aug}$ from B^{aug}.
  Construct coalitions S, T by randomly sampling from $\tilde{\mathcal{D}}_s \cup \tilde{\mathcal{D}}_s^{aug}$; construct coalitions $S \cup T, S \cap T$.
  Calculate supermodularity regularization loss $\mathcal{L}_{sm}$ as Eq. equation 7.
  Pick out bad samples $\mathcal{D}_{del}$ with the top-k score calculated by Eq. equation 8.
  Calculate supervision loss $\mathcal{L}_{sup}$ as Eq. equation 9
  Update $\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{sup} + \omega \mathcal{L}_{sm}$.
end for

C HEURISTIC INSIGHT

In this section we provide some heuristic explanations about the mechanisms behind our proposed method to further demonstrate the rationality. For brevity, we take two coalitions $S = \{(\boldsymbol{x}_i, y_i)\}$ and $T = \{(\boldsymbol{x}_j, y_j)\}$ as example, where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are from different domains. The optimization goal of the proposed regularization loss is to make the model generalization loss obtained by $S \cup T$ plus the model generalization loss obtained by $S \cap T$ be not greater than the sum of generalization losses obtained by S and T separately, i.e., $\ell_{\tilde{\mathcal{D}}_t}(\{\boldsymbol{x}_i\} \cup \{\boldsymbol{x}_j\}) + \ell_{\tilde{\mathcal{D}}_t}(\{\boldsymbol{x}_i\} \cap \{\boldsymbol{x}_j\}) \leq \ell_{\tilde{\mathcal{D}}_t}(\{\boldsymbol{x}_i\}) + \ell_{\tilde{\mathcal{D}}_t}(\{\boldsymbol{x}_j\})$. According to Eq. equation 7, we have:

$$\begin{aligned} &\mathcal{G}(\boldsymbol{\theta} - \nabla_{\boldsymbol{\theta}} \ell(f(\boldsymbol{x}_i, \boldsymbol{\theta}), y_i) - \nabla_{\boldsymbol{\theta}} \ell(f(\boldsymbol{x}_j, \boldsymbol{\theta}), y_j)) + \mathcal{G}(\boldsymbol{\theta}) \\ &- \mathcal{G}(\boldsymbol{\theta} - \nabla_{\boldsymbol{\theta}} \ell(f(\boldsymbol{x}_i, \boldsymbol{\theta}), y_i)) - \mathcal{G}(\boldsymbol{\theta} - \nabla_{\boldsymbol{\theta}} \ell(f(\boldsymbol{x}_j, \boldsymbol{\theta}), y_j)) \leq 0. \end{aligned} \tag{11}$$

We then carry out the second-order Taylor expansion on the terms in Eq. equation 11 and get:

$$\begin{aligned} &(\nabla_i + \nabla_j)^T H (\nabla_i + \nabla_j) - \nabla_i^T H \nabla_i - \nabla_j^T H \nabla_j \\ &= \nabla_i^T H \nabla_j + \nabla_j^T H \nabla_i \leq 0, \end{aligned} \tag{12}$$

∇_i, ∇_j denote $\nabla_{\boldsymbol{\theta}} \ell(f(\boldsymbol{x}_i, \boldsymbol{\theta}), y_i)$, $\nabla_{\boldsymbol{\theta}} \ell(f(\boldsymbol{x}_j, \boldsymbol{\theta}), y_j)$ respectively, $H = \frac{\partial^2 \mathcal{G}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T}$ is the Hessian matrix of $\mathcal{G}(\boldsymbol{\theta})$. We can see that all the zero- and first-order terms of the Taylor-expansion have been dissolved and only the second-order terms are left. Then two main cases can be analysed respectively.

Since Hessian matrix H is a real symmetric matrix, for the case where H is positive definite, we can perform Cholesky decomposition on it: $H = L^T L$, where L is an upper triangular matrix with real and positive diagonal elements. Thus, we can then deduce Eq. 12 as follows:

$$\nabla_i^T H \nabla_j + \nabla_j^T H \nabla_i = (L \nabla_i)^T (L \nabla_j) + (L \nabla_j)^T (L \nabla_i) \leq 0. \tag{13}$$

Similarly, for the case where H is negative definite, we can still perform Cholesky decomposition on positive definite matrix $-H$, thus we can have $H = -L^T L$ and

$$\nabla_i^T H \nabla_j + \nabla_j^T H \nabla_i = -((L\nabla_i)^T (L\nabla_j) + (L\nabla_j)^T (L\nabla_i)) \leq 0. \tag{14}$$

Denote $L\nabla_i, L\nabla_j$ as $\tilde{\nabla}_i, \tilde{\nabla}_j$ respectively, which can be regarded as a mapping transformation of the original gradient vector. Intuitively, ∇_i, ∇_j are the sample gradients generated in the original "training space" during the meta training procedure, while $\tilde{\nabla}_i, \tilde{\nabla}_j$ are the sample gradients transformed by matrix L, which is derivated from the regularization loss calculated on meta test data that can indicate the model generalization. Hence, we can regard the transformed $\tilde{\nabla}_i, \tilde{\nabla}_j$ as sample gradients mapped to a "generalization space", and constrain sample gradients in this mapped space will generalize better on the real test set compared to constrain the sample gradients in the naive "training space".

Then two main cases can be analysed respectively.

Case 1. *For Hessian matrix $H \prec 0$ (a.k.a. negative definite), Eq. 12 holds when $\tilde{\nabla}_i^{\ T} \tilde{\nabla}_j \geq 0$.*

mechanism. When $H \prec 0$, i.e., achieving local maxima, which suggests inferior model generalization, the proposed regularization would help the model improve by enforcing domain consistency on discriminability, that is, pulling the samples from different classes apart and bringing the ones from the same class closer in the "generalization space". As for sample filtering, samples that possess inconsistent gradients, e.g., noisy samples, are more prone to be discarded.

analysis. As Eq. 14 shows, when H is negative definite, the goal of our proposed regularization is to make the inner product of transformed sample gradients positive, i.e., make the sample gradients consistent in the "generalization space". Assuming samples $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ belong to the same class, then their transformed gradients will be inconsistent when they are apart, and be consistent when they are close. Thus, the optimization of our regularization will draw the samples from the same class closer. Similarly, if $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ are from different classes, their transformed gradients would certainly be inconsistent if the samples are close in the "generalization space", since they share the same model while possessing different labels. Thus, the model will pull the samples from different classes apart to make their gradients consistent. In conclusion, when the model is not well optimized, our regularization will help the model improve by enforcing domain consistency on discriminability considering the samples are from different domains. As for sample filtering, the samples that have very inconsistent gradients are contrary to our goal most, and are more likely to obtain larger scores and be discarded. Generally speaking, samples with inconsistent gradients are often noise samples, since they are often far from the center of the class, i.e., being located at outliers. Therefore, the noise samples are more prone to be filtered in this case.

Case 2. *For Hessian matrix $H \succ 0$ (a.k.a. positive definite), Eq. 12 holds when $\tilde{\nabla}_i^{\ T} \tilde{\nabla}_j \leq 0$.*

mechanism. When $H \succ 0$, i.e., achieving local optima, the proposed regularization would help the model jump out by further squeezing out the information within hard samples, that is, detecting the hard samples and then assigning them larger weights implicitly. As for sample filtering, samples that possess very consistent gradients, e.g., redundant samples, are more prone to be discarded.

analysis. As Eq. 13 shows, when $H \succ 0$, our proposed regularization aims to make the inner product of the transformed sample gradients negative, i.e., make the transformed gradients inconsistent in the "generalization space". This objective is contrary to the goal of our main supervision loss which aims to make all the samples clustered, so that it can be regarded as an adversarial optimization that generates adversarial samples first and then trains with these adversarial samples. Therefore, this objective enables the model to generate and detect hard samples that are difficult to classify since the hard samples are far away from the class center and are more likely to possess inconsistent gradients. Besides, due to these hard samples contributing more to the main supervision loss, they can be considered as being assigned larger weights implicitly during the optimization, just like the mechanism of focal loss (Lin et al., 2017). Thus, our regularization can help model jump out of the local optima by squeezing out more information within hard samples, avoiding the model depending on easy patterns or even overfitting on redundant ones. For sample filtering, the samples that produce very consistent gradients, which also means they are redundant ones to a certain, are more likely to be detrimental to our regularization loss and be discarded.

For the general case that H is not fully positive or negative definite, we can take SVD decomposition and regard the model as combined by positive or negative definite sub-matrices. Then our conclusion

holds for each subspace represented by each submatrix. As a conclusion, the optimization of our proposed regularization term is directly aimed at our goal. In practice, it constrains domain consistency on discriminability when the model is trained not that well, while when the model falls into local optimal, it will further squeeze out the information within hard samples that helps the model jump out, which demonstrates the rationality and effectiveness of our method.

D IMPLEMENTATION DETAILS

For all benchmarks, we conduct the commonly used leave-one-domain-out experiments (Li et al., 2017a), where we choose one domain as the unseen target domain for evaluation, and train the model on all remaining domains. We adopt the standard augmentation protocol as in (Carlucci et al., 2019), all images are resized to 224 × 224, following with random resized cropping, horizontal flipping and color jittering. And the Fourier domain augmentation strategy utilized to diversify source domains closely follows the implementations in (Xu et al., 2021b). The network backbone is set to ResNet-18 pre-trained on ImageNet (He et al., 2016) for all datasets following other related works. We train the network using mini-batch SGD with batch size 16, momentum 0.9 and weight decay 5e-4 for 50 epochs. The initial learning rate is 0.001 and decayed by 0.1 at 80% of the total epochs. The meta step size α is set to be the same as the learning rate. For the hyper-parameters, i.e., the weight of regularization loss ω and the number of discarded bad samples in each iteration k, their values are selected on validation data following standard practice, where we use 90% of available data as training data and 10% as validation data. Specifically, we set $\omega = 0.1$ and $k = 5$ for all experiments. Our framework is implemented with PyTorch on NVIDIA GeForce RTX 3090 GPUs. All results are reported based on the average accuracy over three independent runs for a fair comparison.

E EXPERIMENTAL RESULTS WITH ERROR BARS

For the sake of objective, we run all the experiments multiple times with random seed. We report the average results in the main body of paper for elegant, and show the complete results with error bars in the form of mean±std below (Table. 6, 7, 8).

Table 6: Leave-one-domain-out results on PACS. The best and second-best results are bolded and underlined respectively.

Methods	Art	Cartoon	Photo	Sketch	Avg.
MLDG (Li et al., 2018a)	78.70	73.30	94.00	65.10	80.70
L2A-OT (Zhou et al., 2020b)	83.30	78.20	96.20	73.60	82.80
RSC (Huang et al., 2020)	83.43	80.31	95.99	80.85	85.15
DSU (Li et al., 2022)	83.60	79.60	95.80	77.60	84.10
DeepAll(Zhou et al., 2020a)	77.63±0.84	76.77±0.33	95.85±0.20	69.50±1.26	79.94
MASF (Dou et al., 2019)	80.29±0.18	77.17±0.08	94.99±0.09	71.69±0.22	81.04
DDAIG (Zhou et al., 2020a)	84.20±0.30	78.10±0.60	95.30±0.40	74.70±0.80	83.10
MixStyle (Zhou et al., 2021a)	84.10±0.4	78.80±0.4	96.10±0.3	75.90±0.9	83.70
FACT (Xu et al., 2021b)	85.37±0.29	78.38±0.29	95.15±0.26	79.15±0.69	84.51
ITL-Net (Gao et al., 2022)	83.90±0.4	78.90±0.6	94.80±0.2	80.10±0.6	84.40
DCG (*ours*)	**85.94±0.21**	**80.76±0.36**	**96.41±0.17**	**82.08±0.44**	**86.30**

Table 7: Leave-one-domain-out results on Mini-DomainNet.

Methods	Clipart	Painting	Real	Sketch	Avg.
DeepAll (Zhou et al., 2020a)	65.30	58.40	64.70	59.00	61.86
ERM (Vapnik, 1999)	65.50 ± 0.3	57.10 ± 0.5	62.30 ± 0.2	57.10 ± 0.1	60.50
MLDG (Li et al., 2018a)	65.70 ± 0.2	57.00 ± 0.2	63.70 ± 0.3	58.10 ± 0.1	61.12
Mixup (Zhang et al., 2018)	67.10 ± 0.2	59.10 ± 0.5	64.30 ± 0.3	59.20 ± 0.3	62.42
MMD (Lee et al., 2019)	65.00 ± 0.5	58.00 ± 0.2	63.80 ± 0.2	58.40 ± 0.7	61.30
SagNet (Nam et al., 2019)	65.00 ± 0.4	58.10 ± 0.2	64.20 ± 0.3	58.10 ± 0.4	61.35
CORAL (Peng et al., 2019)	66.50 ± 0.2	59.50 ± 0.4	66.00 ± 0.6	59.50 ± 0.1	62.87
MTL (Blanchard et al., 2021)	65.30 ± 0.5	59.00 ± 0.4	65.60 ± 0.4	58.50 ± 0.2	62.10
DCG (*ours*)	**69.38±0.19**	**61.79±0.22**	**66.34±0.27**	**63.21±0.09**	**65.18**

Table 8: Leave-one-domain-out results on Office-Home.

Methods	Art	Clipart	Product	Real	Avg.
MLDG (Li et al., 2018a)	52.88	45.72	69.90	72.68	60.30
SagNet (Nam et al., 2019)	60.20	45.38	70.42	73.38	62.34
RSC (Huang et al., 2020)	58.42	47.90	71.63	74.54	63.12
L2A-OT (Zhou et al., 2020b)	60.60	50.10	74.80	**77.00**	65.60
DSU (Li et al., 2022)	60.20	54.80	74.10	75.10	66.10
DeepAll (Zhou et al., 2020a)	57.88±0.20	52.72±0.50	73.50±0.30	74.80±0.10	64.72
DDAIG (Zhou et al., 2020a)	59.20±0.10	52.30±0.30	74.60±0.30	76.00±0.10	65.50
MixStyle (Zhou et al., 2021a)	58.70±0.3	53.40±0.2	74.20±0.1	75.90±0.1	65.50
FACT (Xu et al., 2021b)	60.34±0.11	54.85±0.37	74.48±0.13	76.55±0.10	66.56
STNP (Kang et al., 2022)	59.55±0.21	55.01±0.29	73.57±0.28	75.52±0.21	65.89
DCG (*ours*)	**60.67±0.14**	**55.46±0.32**	**75.26±0.18**	76.82±0.09	**67.05**