Class–Domain Discriminability Guided Representation Enhancement for Domain Generalization

Bin Zheng[1], **Junbiao Cui**[1], **Jiao Zhao**[1], **Jiye Liang**[1]*
[1]Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology,
Shanxi University, Taiyuan, China
{zhengbin1, cjb, zhaojiao, ljy}@sxu.edu.cn

Abstract

Domain Generalization (DG) aims to leverage multiple source domains to train models that can generalize to unseen target domains, thereby mitigating the performance degradation caused by domain shifts. Most existing approaches impose unified alignment or suppression constraints in the global feature space, overlooking the heterogeneity of feature channels in terms of class discriminativeness and domain stability. Such coarse-grained constraints may lead to the loss of discriminative information or the retention of domain-sensitive features. To address this issue, we propose a novel DG framework consisting of two core modules: Channel Sensitivity Decomposition (CSD), which employs channel-wise response variances along the class and domain dimensions as empirical proxy signals to partition feature channels into four functional subspaces via a continuous and differentiable soft decomposition mechanism; and Subspace-Specific Feature Augmentation (SSA), which implements tailored augmentation and regularization strategies for different subspaces to suppress environment-related spurious correlations while preserving cross-domain stable discriminative features. Additionally, a KL-divergence-based prediction consistency constraint is introduced to stabilize the training process and enhance model robustness. Extensive experiments on multiple standard DG benchmarks demonstrate that our framework achieves strong average performance across four benchmark datasets.

1 Introduction

In recent years, deep learning has achieved remarkable progress in computer vision tasks, particularly demonstrating strong modeling capabilities in image classification [12, 7], object detection [29, 24]. However, these successes are built upon the assumption that the training data and the test data follow an independent and identically distributed setting. In real-world scenarios, however, training data and test data often originate from different acquisition environments or data generation mechanisms. Such distribution discrepancies caused by environmental variations are commonly referred to as domain shift, which in turn leads to significant performance degradation when models are evaluated on data from new environments. This issue severely limits the cross-domain applicability of deep learning models, especially in domains with complex data acquisition processes, such as autonomous driving.

Domain generalization is proposed against this background, aiming to train models that can robustly generalize to unseen target domains using only multiple source domains, without access to any target-domain data. A substantial body of prior work has approached this problem from the perspectives of representation learning [22, 25, 23] and data augmentation [33, 5], seeking to improve robustness through domain alignment [9], suppression of domain-discriminative information [11], or mixing

*Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

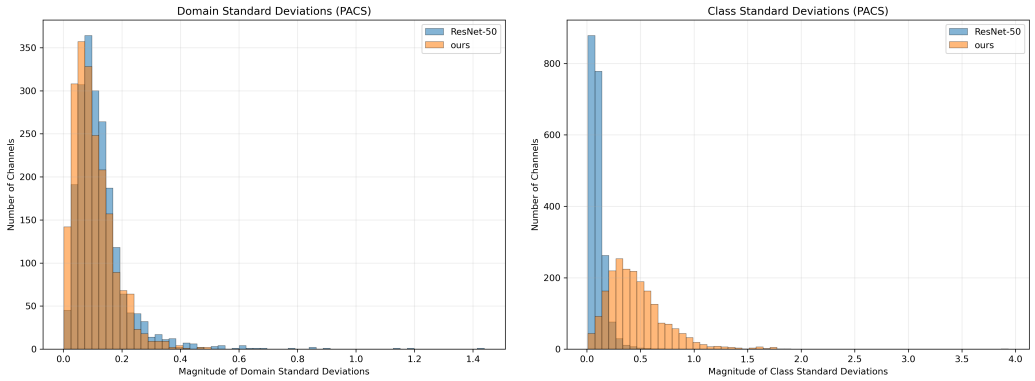


Figure 1: We quantify channel sensitivity to domain and class variations by computing the standard deviation of ResNet-50 features across the domains and classes of the PACS dataset.

cross-domain statistics [42, 14]. Although these methods have achieved encouraging progress to some extent, a considerable portion of them implicitly relies on a strong assumption: domain-related information can be uniformly aligned or globally suppressed within a shared feature space [43]. However, in deep neural networks, different feature channels often exhibit markedly heterogeneous responses to class semantics and environmental variations. Imposing uniform constraints at the global feature level may therefore lead to over-alignment that damages discriminative information, or allow residual domain-sensitive channels to continuously introduce environmental bias.

To more intuitively analyze this channel-level heterogeneity, we conduct a statistical analysis of intermediate features extracted by ResNet-50 [12] on the PACS dataset (Fig.1). The results show that different channels exhibit markedly different response variations along the class and domain dimensions: some channels fluctuate substantially across domains, indicating pronounced domain sensitivity, while others vary only marginally across classes and thus contribute limited discriminative power. This phenomenon suggests that feature channels possess significant heterogeneity in terms of class discriminability and environmental stability, a finding that has also been corroborated by recent studies [38] on channel-wise modeling.

Based on the above observations, we propose a novel domain generalization framework. This framework integrates a Channel Sensitivity Decomposition (CSD) module and a Subspace-Specific Feature Augmentation (SSA) module. Our main contributions are summarized as follows:

- We design a Channel Sensitivity Decomposition (CSD) module that jointly models the class variance and domain variance of feature channels, enabling a quantitative evaluation and soft decomposition of channel discriminativeness and environmental stability. This facilitates the identification and structured separation of cross-domain stable components, environment-sensitive components, and redundant components within the learned representations.

- We propose a Subspace-Specific Feature Augmentation (SSA) module that applies differentiated augmentation and regularization strategies to the four channel subspaces obtained from the decomposition. Specifically, this module weakens the domain dependence of useful but unstable features through conditional cross-domain augmentation, reduces the influence of purely noisy channels via random suppression, and stabilizes redundant representations through consistency constraints, thereby suppressing spurious correlations while maximally preserving discriminative information.

- We conduct comprehensive evaluations on multiple standard domain generalization benchmarks. Experimental results demonstrate that our method achieves strong average performance across four benchmark datasets, supporting the effectiveness of the proposed framework in improving model generalization to unseen domains.

2 Related Work

2.1 Domain Generalization

Domain generalization (DG) aims to learn models that can generalize to unseen environments using only data from multiple source domains, without access to any target-domain data. Existing studies have primarily explored this problem from the perspectives of representation learning [35, 9, 6] and feature augmentation [28, 32, 36]. For instance, [41] employed the Wasserstein distance to minimize the disparity in marginal distributions across different domains, thereby further advancing the development of explicit feature alignment methods. Mixup [39] synthesizes new data by performing linear interpolation between any two instances—as well as between their corresponding labels—using a mixing weight sampled from a Beta distribution. Although these methods have achieved promising progress to some extent, most of them operate at the level of the overall feature space and provide only coarse-grained characterization of the heterogeneity among different feature dimensions in terms of class discriminability and environmental stability, which may limit generalization performance under complex domain shifts.

2.2 Channel-level Modeling and Representation Decomposition.

Existing channel-level domain generalization methods have begun to leverage class-related or domain-related signals to select, weight, or suppress feature channels. For example, DomainDrop [11] weakens domain-related features by identifying and suppressing channels whose activations are unstable across different domains, while works such as CLIPCeil [38] model channel importance from the perspective of semantic alignment. However, such methods typically focus on characterizing channel properties along a single dimension, such as domain sensitivity or semantic relevance, and tend to integrate different statistical signals into a unified weighting or suppression strategy.

Different from existing channel-level methods, this work jointly models the class sensitivity and domain sensitivity of feature channels, enabling a simultaneous characterization of their discriminative relevance and cross-domain stability at the channel level. Instead of only identifying whether a channel is informative or domain-sensitive, our formulation further distinguishes whether a channel's discriminative behavior remains stable across domains, which is overlooked by prior approaches. This distinction allows channels with different properties to be explicitly separated and treated differently during training. Compared with methods that apply unified constraints in the global feature space or combine multiple statistics into a single channel selection criterion, our approach focuses on explicitly modeling channel heterogeneity and provides a more fine-grained basis for channel-level processing in domain generalization.

3 Method

3.1 Problem Setup

In domain generalization (DG), we are given M source domains $\{\mathcal{D}^1, \mathcal{D}^2, \ldots, \mathcal{D}^M\}$, where each domain follows a different distribution. Let $\mathcal{X} \in \mathbb{R}^d$ denote the input space and $\mathcal{Y} \in \mathbb{R}$ the label space. For the j-th source domain, we denote its samples as $\{(x_i^j, y_i^j)\}_{i=1}^{N_j}$, where $x_i^j \in \mathcal{X}$ and $y_i^j \in \mathcal{Y}$ represent the i-th sample and its label in domain j, and N_j is the number of samples in this domain. Our goal is to train a model on multiple source domains, consisting of a classifier $f_\phi(\cdot)$ and a feature extractor $g_\theta(\cdot)$, such that it maintains stable performance in unseen test environments. The feature extractor maps an input sample to a latent representation $Z = g_\theta(x) \in \mathbb{R}^d$, where $Z^{(m)}$ denotes the value of the m-th feature channel.

3.2 Channel-wise Modeling and Sensitivity Perspective

We model feature representations from a channel-wise perspective, treating each channel as a basic unit that carries specific statistical dependencies. Since it is difficult to directly characterize the statistical dependency between channels and either class labels or domains, we introduce *channel sensitivity* as a computable empirical proxy to capture channel response variations along two dimensions: (i) *class sensitivity*, which reflects how much a channel varies across classes, and (ii)

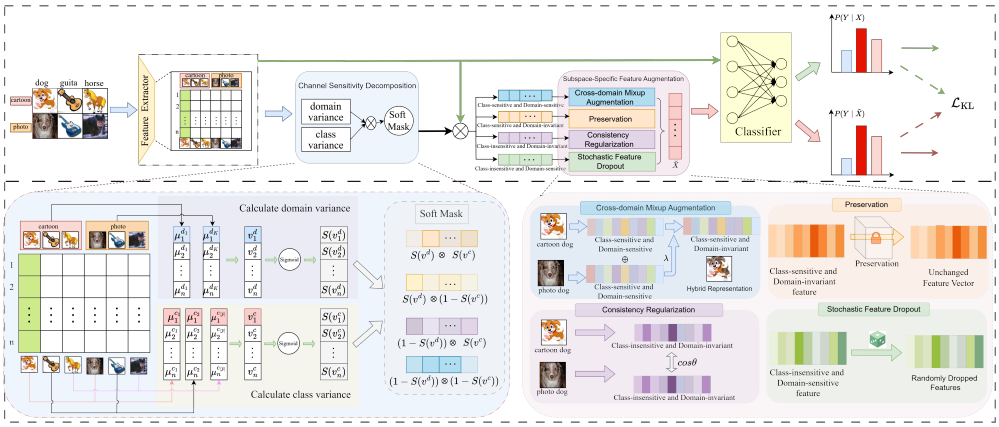


Figure 2: Overall framework of the proposed method.

domain sensitivity, which measures how much a channel changes across domains. Based on these two dimensions, channels can be conceptually partitioned into four subspaces: class-sensitive and domain-stable, class-sensitive but domain-unstable, class-insensitive but domain-sensitive, and insensitive to both class and domain. This structured decomposition provides a principled basis for subsequent channel-wise soft decomposition and subspace-specific modeling, enabling the method to explicitly align feature processing with generalization mechanisms rather than relying on unified constraints over the entire feature space. Figure 2 illustrates the overall framework of our method.

In the following, we detail how to empirically model channel sensitivity using class variance and domain variance (Section 3.3), and how to design feature augmentation strategies and training objectives based on the structured soft decomposition (Sections 3.4–3.5).

3.3 Channel Sensitivity Modeling

This section introduces an operational channel sensitivity modeling approach, which quantifies the roles of individual feature channels in generalization by characterizing their response statistics along the class dimension and the domain dimension.

We treat the class variance and domain variance as empirical proxy measures for characterizing the discriminative capacity and environmental sensitivity of feature channels. These statistics are used to provide relative, channel-level differentiation signals during training, rather than to model true statistical dependencies or to identify causal or strictly invariant discriminative features. Given that such statistics may be biased under class or domain imbalance, or when channel activations exhibit multimodal distributions, we adopt a continuously differentiable soft decomposition mechanism that allows each channel to participate in multiple subspace representations with different weights, thereby mitigating the risk of statistical noise being amplified into erroneous decisions. Overall, channel sensitivity modeling in this work primarily serves as a form of structured regularization and representation guidance.

Class Sensitivity via Class Variance. We first characterize channel sensitivity to class semantics. For the m-th channel, we compute its mean activation within each class, as illustrated in Figure 2.

Specifically, we group samples by class and compute the per-class mean of each channel:

$$\mu_c^{(m)} = \frac{1}{n_c} \sum_{i=1}^{n_c} Z_{x_i}^{c^{(m)}}, \tag{1}$$

where n_c denotes the number of samples belonging to class c, and $Z_{x_i}^{c^{(m)}}$ is the m-th channel value of sample i. We then compute the class variance of channel m as

$$v_{\text{cls}}^{(m)} = \frac{1}{|\mathcal{Y}|} \sum_{c \in \mathcal{Y}} \left(\mu_c^{(m)} - \bar{\mu}_c^{(m)} \right)^2, \tag{2}$$

where $|\mathcal{Y}|$ is the number of classes and $\bar{\mu}_c^{(m)} = \frac{1}{|\mathcal{Y}|}\sum_{c\in\mathcal{Y}} \mu_c^{(m)}$ is the mean of channel m over all classes. This statistic reflects how differently the channel responds across classes: a larger $v_{\text{cls}}^{(m)}$ indicates higher class sensitivity and thus a higher likelihood of contributing to class discrimination.

Domain Sensitivity via Domain Variance. Similarly, we characterize channel sensitivity to domain changes. If a channel exhibits significantly different activation distributions under different environments, it is more likely to encode environment-related factors and become a potential source of spurious correlations.

For the m-th channel, we compute its mean activation within each domain:

$$\mu_k^{(m)} = \frac{1}{n_k}\sum_{j=1}^{n_k} Z_{x_j}^{k^{(m)}}, \tag{3}$$

where n_k denotes the number of samples from domain k and $Z_{x_j}^{k(m)}$ is the m-th channel value of sample j in domain k. We then compute the domain variance as

$$v_{\text{dom}}^{(m)} = \frac{1}{K}\sum_{k=1}^{K}\left(\mu_k^{(m)} - \bar{\mu}_{\text{k}}^{(m)}\right)^2, \tag{4}$$

where $\bar{\mu}_{\text{k}}^{(m)} = \frac{1}{K}\sum_{k=1}^{K}\mu_k^{(m)}$ denotes the mean over all domains. A larger $v_{\text{dom}}^{(m)}$ indicates that the channel is highly sensitive to environmental changes and lacks stability across domains.

EMA Statistics for Robust Estimation. Since class/domain variances may be biased under class imbalance or multi-modality, we use them only as stable signals that provide relative separation and continuous weights. To reduce mini-batch noise, we maintain exponential moving average (EMA) [17] running statistics for the class-wise and domain-wise channel means, and compute variances across the class/domain dimension based on the running means.

Let $\mu_{c,t}^{(m)}$ and $\mu_{k,t}^{(m)}$ denote the class-wise and domain-wise mean value of channel m computed from the current batch at iteration t. We update the running means using EMA:

$$\hat{\mu}_{c,t}^{(m)} \leftarrow (1-\rho)\hat{\mu}_{c,t-1}^{(m)} + \rho\,\mu_{c,t}^{(m)}, \hat{\mu}_{k,t}^{(m)} \leftarrow (1-\rho)\hat{\mu}_{k,t-1}^{(m)} + \rho\,\mu_{k,t}^{(m)} \tag{5}$$

where ρ is a fixed momentum coefficient. If a class or domain is missing in the current batch, we skip updating its corresponding running mean. The class variance $v_{\text{cls}}^{(m)}$ and domain variance $v_{\text{dom}}^{(m)}$ are then computed across classes/domains using $\hat{\mu}_c^{(m)}$ and $\hat{\mu}_k^{(m)}$.

Continuous Soft Assignment to Four Subspaces. Based on the above class variance and domain variance, we further jointly model the roles of channels in terms of class discriminativeness and domain stability, and adopt a continuous and differentiable soft assignment mechanism to decompose channel features into four subspaces with different semantic properties.

Concretely, we feed class variance and domain variance into two independent gating functions to obtain sensitivity scores:

$$s_{\text{cls}}^{(m)} = \sigma\Big(\alpha\,(v_{\text{cls}}^{(m)} - \beta)\Big), s_{\text{dom}}^{(m)} = \sigma\Big(\alpha\,(v_{\text{dom}}^{(m)} - \gamma)\Big), \tag{6}$$

where $\sigma(\cdot)$ is the Sigmoid function,

$\alpha > 0$ is a learnable scaling factor, and β, γ are learnable thresholds. We enforce $\alpha > 0$ via softplus parameterization to ensure stable monotonic gating.

Given the sensitivity scores $(s_{\text{cls}}^{(m)}, s_{\text{dom}}^{(m)})$, we define the assignment weights of channel m to the four subspaces as:

$$Z_{\text{c,d}}^{(m)} = (s_{\text{cls}}^{(m)}\, s_{\text{dom}}^{(m)})Z^{(m)}, \qquad Z_{\text{c,nd}}^{(m)} = (s_{\text{cls}}^{(m)}\left(1 - s_{\text{dom}}^{(m)}\right))Z^{(m)}, \tag{7}$$

$$Z_{\text{nc,d}}^{(m)} = (\left(1 - s_{\text{cls}}^{(m)}\right)\, s_{\text{dom}}^{(m)})Z^{(m)}, \qquad Z_{\text{nc,nd}}^{(m)} = (\left(1 - s_{\text{cls}}^{(m)}\right)\left(1 - s_{\text{dom}}^{(m)}\right))Z^{(m)}, \tag{8}$$

where c/nc denote class-sensitive/insensitive and d/nd denote domain-sensitive/insensitive. Accordingly, $Z_{\text{c,d}}^{(m)}$ corresponds to *Class-sensitive and Domain-sensitive* channels, $Z_{\text{c,nd}}^{(m)}$ to *Class-sensitive*

and Domain-insensitive channels, $Z_{\text{nc,d}}^{(m)}$ corresponds to *Class-insensitive but Domain-sensitive* channels, and $Z_{\text{nc,nd}}^{(m)}$ corresponds to *Class-insensitive and Domain-insensitive* channels.

This soft partitioning mechanism allows a single channel to participate in the modeling of multiple subspaces with different weights, thereby avoiding hard semantic partitioning of channels and improving the robustness and generalization of feature decomposition.

3.4 Subspace-specific Feature Augmentation

Based on the class sensitivity and domain sensitivity scores $\left(s_{cls}^{(m)}, s_{dom}^{(m)}\right)$ obtained in Section 3.3, we partition the channel representations into four subspaces with distinct statistical properties, and design corresponding feature augmentation strategies for each subspace to suppress environment-related disturbances while preserving cross-domain stable discriminative information.

$Z_{\text{c,d}}$: This subspace simultaneously carries class-discriminative information and exhibits strong environmental dependence; thus, directly suppressing it would harm discriminative capacity. To address this issue, we adopt a class-conditional cross-domain feature augmentation strategy. Specifically, within a batch, we select samples belonging to the same class but originating from different domains, and perform random linear interpolation on their features within the class-sensitive and domain-sensitive channel subspace to generate cross-domain augmented subspace representations. This introduces cross-environment perturbations while preserving class semantic consistency, thereby weakening the dependence on environment-specific statistical patterns.

$$\tilde{Z}_{\text{c},\,d}^{\text{y},\,k} = \lambda Z_{c,d}^{y,k} + (1-\lambda) Z_{c,d}^{y,j}, \quad j \neq k \tag{9}$$

where $\tilde{Z}_{\text{c},\,d}^{\text{y},\,k}$ denotes the augmented feature representation, $Z_{c,d}^{y,j}$ denotes the class-sensitive and domain-sensitive features of samples from the same class but different domains, and λ is the interpolation weight sampled from a Beta distribution. For each sample, a mixing partner is selected from the same class but a different source domain within the mini-batch. If no eligible candidate is available, the original subspace feature is retained without mixing.

$Z_{\text{c,nd}}$: This subspace contains channels that are highly sensitive to class changes yet remain stable across domains, serving as the stable discriminative basis for cross-domain generalization. Thus, we do not apply additional perturbation to $Z_{\text{c,nd}}$ and keep it unchanged:

$$\tilde{Z}_{c,nd}^{y,k} = Z_{c,nd}^{y,k}, \tag{10}$$

$Z_{\text{nc,d}}$: Channels in this subspace contribute little to class discrimination while being highly sensitive to domain variations, making them potential carriers of spurious correlations. We therefore apply stochastic channel suppression during training:

$$\tilde{Z}_{nc,d}^{y,k} = \mathbf{m} \odot Z_{nc,d}^{y,k}, \tag{11}$$

where $\mathbf{m}$ denotes a stochastic channel-wise suppression mask.

$Z_{\text{nc,nd}}$: The dimensions in this subspace are insensitive to both class and domain variations and thus typically contribute little to prediction. To suppress redundant fluctuations, we impose a consistency regularization on this subspace by minimizing the cosine distance between the $Z_{\text{nc,nd}}^{(m)}$dimension features of samples from the same class, thereby encouraging consistency in their representation directions and improving overall stability.

$$\tilde{Z}_{nc,nd}^{y,k} = Z_{nc,nd}^{y,k} \tag{12}$$

Recombination and Inference. After applying subspace-specific processing, we recombine the four subspace features to obtain the enhanced representation:

$$\tilde{Z} = \tilde{Z}_{\text{c},\,d}^{\text{y},\,k} + \tilde{Z}_{\text{c},\,nd}^{\text{y},\,k} + \tilde{Z}_{\text{nc},\,d}^{\text{y},\,k} + \tilde{Z}_{\text{nc},\,nd}^{\text{y},\,k}. \tag{13}$$

The enhanced feature $\tilde{Z}$ is then fed into the classifier for prediction and is jointly optimized with the original representation. During inference, we do not perform subspace-specific augmentation or random suppression; we only use the learned feature representation for prediction.

3.5 Training Objective and Optimization

Our overall objective consists of multiple complementary loss terms that preserve discriminability while encouraging robustness to environmental changes. The total loss is:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda_1 \mathcal{L}_{\text{KL}} + \lambda_2 \mathcal{L}_{\text{reg}}, \tag{14}$$

where each component is defined as follows.

Classification Loss. To maintain discriminative ability, we apply cross-entropy loss on both the original representation and the enhanced representation:

$$\mathcal{L}_{\text{cls}} = \text{CE}\big(f_\phi(Z), y\big) + \text{CE}\big(f_\phi(\tilde{Z}), y\big). \tag{15}$$

Prediction Consistency Loss. For the enhanced representation, we enforce prediction consistency by minimizing the KL divergence between the original prediction and the enhanced prediction:

$$\mathcal{L}_{\text{KL}} = \frac{1}{2}(\text{KL}\Big(f_\phi(Z) \,\|\, f_\phi(\tilde{Z})\Big) + \text{KL}\Big(f_\phi(\tilde{Z}) \,\|\, f_\phi(Z)\Big)), \tag{16}$$

which stabilizes the predictive distribution under feature augmentation and prevents unnecessary shifts of the decision boundary.

Subspace Consistency Regularization. For the class-insensitive and domain-insensitive subspace, we further apply the consistency regularization described in Section 3.4, which is defined based on cosine similarity as follows:

$$\mathcal{L}_{\text{reg}} = 1 - (Z^{(i)}_{nc,nd} \cdot Z^{(j)}_{nc,nd})/(\|Z^{(i)}_{nc,nd}\|_2 \, \|Z^{(j)}_{nc,nd}\|_2) \tag{17}$$

where i and j denote two samples sharing the same class label. When available, the regularization is preferentially applied to sample pairs drawn from different source domains; otherwise, it degenerates to same-class pairs within the batch.

4 Experiments

4.1 Datasets

We evaluate the proposed method on four standard public domain generalization datasets, including PACS[20] , VLCS[8] , Office-Home[31] and TerraIncognita[2] . PACS contains 9,991 images from 7 categories and is divided into 4 domains: Photo, Art Painting, Cartoon and Sketch. VLCS contains 10,729 images from 5 categories and is divided into 4 domains: VOC2007, LabelMe, Caltech and SUN. Office-Home contains 15,588 images from 65 categories and is divided into 4 domains: Art, Clipart, Product and Real-World. TerraIncognita contains 24,788 images from 10 categories and is divided into 4 domains according to the shooting locations: L100, L38, L43 and L46.

4.2 Experimental Details

Our implementation is based on DomainBed. Unless otherwise specified, we use an ImageNet-pretrained ResNet-50 backbone and follow the standard leave-one-domain-out evaluation protocol. Models are trained using Adam, and all results are averaged over three random seeds. For cross-domain feature mixing, the interpolation coefficient is sampled from $\text{Beta}(2.0, 2.0)$, and the channel suppression probability is set to 0.6. Most baseline results are taken from the IAIE-Proxy [13] and I^3C [40] papers, which follow comparable DomainBed-style evaluation protocols. Additional implementation details are provided in Appendix A.

4.3 Experimental Results

On four standard domain generalization benchmarks, we conduct a systematic comparison between the proposed method and a variety of recent domain generalization approaches.

Table 1 reports the average accuracy over all target domains for each dataset under the leave-one-domain-out evaluation protocol, while the detailed accuracies for individual domains are provided in Appendix A.

Table 1: Comparison of ACC(%) with the SOTA methods on PACS, Office-Home, VLCS and TerraIncognita

Method	PACS	Office-Home	VLCS	TerraIncognita	Avg
CORAL [30]	86.2	68.7	78.8	47.7	70.35
MMD [22]	84.7	66.4	77.5	42.2	67.70
MLDG [21]	84.9	66.8	77.2	47.8	69.18
IRM [1]	83.5	64.3	78.6	47.6	68.50
RSC [16]	85.2	65.5	77.1	46.6	68.60
SupCon [18]	88.1	70.6	79.3	50.7	72.18
MTL [3]	84.6	66.4	77.2	45.6	68.45
ERM [10]	85.5	67.6	77.3	47.8	69.55
MixStyle [42]	85.2	60.4	77.9	44.0	66.88
SagNet [27]	86.3	68.1	77.8	48.6	70.20
SWAD [4]	88.1	70.6	79.1	50.0	71.95
PCL [37]	88.7	71.6	78.0	52.1	72.60
SAGM [34]	86.6	70.1	80.0	48.8	71.38
iDAG [15]	88.8	71.8	76.9	46.1	70.90
IAIE-Pair [13]	88.8	70.9	79.6	52.2	72.88
IAIE-Proxy [13]	89.2	71.8	78.5	52.6	73.03
I^3C [40]	88.2	72.1	79.9	51.5	72.93
Ours	89.5	72.3	80.2	53.8	73.95

Furthermore, we analyze the results from the perspective of different categories of methods. Compared with traditional domain generalization approaches (e.g., ERM[10], IRM[1], and MMD[22]), our method demonstrates consistently competitive overall performance. Compared with classical data augmentation methods (e.g., MixStyle[42]), it achieves superior or comparable results on most datasets. Meanwhile, when compared with contrastive learning–based methods (e.g., PCL[37], and IAIE-Proxy[13]), our approach also exhibits strong competitiveness. This trend is consistently observed across multiple benchmark datasets. For instance, on the PACS dataset, our method achieves consistent improvements across all four target domains, reaching an average accuracy of 89.5%, which outperforms PCL and IAIE-Proxy by 0.8% and 0.3%, respectively. On the TerraIncognita dataset, which exhibits substantial environmental variations, our approach attains an average accuracy of 53.8%, surpassing SWAD and PCL by 3.8% and 1.7%, respectively, demonstrating a clear performance margin.

Table 2: Results on four datasets with ViT-S/16.

Method	PACS	Office-Home	VLCS	TerraIncognita	Avg
PIC(GVB) [44]	87.8	72.9	80.6	46.9	72.1
GMOE [19]	88.1	74.2	80.2	48.51	72.8
ePIC(GVB) [44]	89.4	74.8	82.3	50.9	74.4
Ours	**89.6**	**75.2**	**82.1**	**51.6**	**74.6**

Furthermore, Table 2 reports the results of different methods using ViT-S as the backbone network. Our method achieves competitive performance across the four benchmarks, supporting its effectiveness across different backbone architectures.

Overall, the proposed method exhibits robust performance across domain generalization benchmarks with varying levels of difficulty and characteristics. In the following section, we further investigate the specific contributions of each module and the channel subspace strategy to the performance gains through ablation studies.

4.4 Ablation Experiments

To better analyze the contributions of the CSD and SSA modules, as well as the KL-divergence–based components, we conduct ablation studies to evaluate their respective effects.

Component Ablation. We conduct ablation studies on the PACS dataset to analyze the impact of the CSD and SSA modules, as well as their associated regularization design, on model performance (see Table 3). Starting from a baseline model trained with only the classification loss, introducing the CSD and SSA modules leads to a significant performance improvement, validating the effectiveness of channel-wise structured decomposition and subspace-specific augmentation. Further incorporating the KL-divergence regularization consistently improves performance, indicating that this regularization helps stabilize the feature variations induced by CSD and SSA, thereby enabling more robust cross-environment generalization.

Table 3: PACS ablation experiments based on ResNet-50 pretrained on ImageNet.

Model	Art	Cartoon	Photo	Sketch	Avg
CE	90.7	82.3	96.1	81.6	87.7
CE + CSD + SSA	90.5	83.8	97.6	84.5	89.1
CE + CSD + SSA + KL	90.8	84.2	98.1	84.9	89.5

Table 5: Results on PACS with two-source domain setting.

Model	Art + cartoon	Art + Photo	Art + Sketch	Cartoon + Photo	Cartoon + Sketch	Photo + Sketch	Avg
PCL	78.35	87.32	78.90	89.07	66.80	86.54	81.16
ours	80.09	90.95	83.55	91.04	69.70	84.72	83.34

Subspace strategy ablation analysis. To verify the necessity of different channel subspace strategies, we remove the processing strategy corresponding to each subspace from the full model and evaluate the resulting performance changes on PACS and TerraIncognita (Table 4). The results show that removing any subspace strategy leads to a performance drop, indicating that each subspace plays a positive role in the overall modeling. Among them, removing the cross-domain class-consistent mixing augmentation for the "category-sensitive and environment-sensitive" subspace causes the most pronounced performance degradation, suggesting that this subspace is particularly critical for mitigating environmental interference and extracting cross-domain stable discriminative information. In addition, simplified settings that retain only a single subspace strategy are also clearly inferior to the full model, further demonstrating the complementarity among different subspace strategies and the necessity of their joint modeling for improving cross-domain generalization performance.

Table 4: Subspace-strategy ablation analysis.

Model	PACS	TerraIncognita
Full Model	89.5	53.8
Without Cross-domain Mixing	88.9	53.0
Without Random Suppression	89.2	53.2
Without Consistency Regularization	89.3	53.2
Only Consistency Regularization	88.2	52.7

Channel ablation analysis. Figure 3 shows a test-time channel ablation analysis on the PACS dataset, where feature channels are grouped according to their class variance and domain variance estimated by the proposed proxies. Channels with high class variance and low domain variance, high class variance and high domain variance, and low class variance are masked at inference time without retraining. As shown, removing channels with high class variance and low domain variance leads to the largest performance degradation, whereas masking low class variance channels causes only a marginal accuracy drop. In contrast, ablating discriminative but domain-sensitive channels results in a smaller accuracy decrease. These results indicate that the proposed variance-based proxies provide a meaningful partition of feature channels and effectively distinguish cross-domain stable discriminative components from less informative or unstable ones in terms of their functional contributions.

4.5 Robustness to Limited Source Domains

We further evaluate our method on PACS under a two-source domain setting, where only two out of four domains are used for training (six combinations). As shown in Table 5, our method achieves an average accuracy of 83.34%, outperforming PCL (81.16%) by a large margin. This result indicates that our approach remains effective even with limited domain diversity, suggesting that it relies on relative channel differences rather than precise variance estimation.

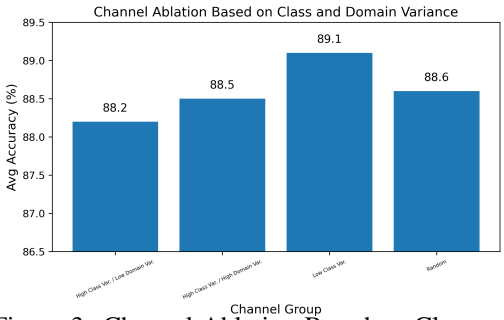


Figure 3: Channel Ablation Based on Class and Domain Variance (PACS)

5 Conclusion

This paper proposes a channel-sensitivity-based feature augmentation framework for domain generalization, aiming to address the limitations of existing methods that impose uniform invariance constraints over the entire feature space while neglecting the heterogeneity of feature channels in terms of discriminability and environmental stability. The proposed framework consists of two key components. The channel sensitivity decomposition module jointly models the class discriminability and domain stability of feature channels, enabling a structured partitioning of the representation space. The subspace-specific feature augmentation module then designs differentiated augmentation and regularization strategies for different channel subspaces, guiding the model to learn discriminative representations that remain stable across environments,

thereby effectively mitigating the influence of environment-related spurious correlations on model decisions. Extensive experiments on multiple DG benchmarks demonstrate the robust performance of our method.

Despite its practical robustness, the performance of the framework still depends to some extent on the diversity of training samples and environments, and certain augmentation and regularization strategies require empirical tuning. Future work will explore more adaptive channel scheduling and stability constraint mechanisms, and extend the proposed framework to more challenging generalization settings.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (62376141, 62676227, 62541608) and the Shanxi Provincial Natural Science Foundation for Youth (202503021212096). We extend our heartfelt gratitude to the anonymous reviewers for their invaluable and constructive comments.

References

[1] Kartik Ahuja, Karthikeyan Shanmugam, Kush R. Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 145–155, 2020.

[2] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 472–489, 2018.

[3] Gilles Blanchard, Aniket Anand Deshmukh, Ürün Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *Journal of Machine Learning Research*, 22:2:1–2:55, 2021.

[4] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 22405–22418, 2021.

[5] Chaoqi Chen, Luyao Tang, Feng Liu, Gangming Zhao, Yue Huang, and Yizhou Yu. Mix and reason: Reasoning over semantic topology with data mixing for domain generalization. *Advances in Neural Information Processing Systems*, 35:33302–33315, 2022.

[6] De Cheng, Zhipeng Xu, Xinyang Jiang, Dongsheng Li, Nannan Wang, and Xinbo Gao. Prompt disentanglement via language guidance and representation alignment for domain generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(6):6799–6816, 2026.

[7] Zhongying Deng, Kaiyang Zhou, Da Li, Junjun He, Yi-Zhe Song, and Tao Xiang. Dynamic instance domain adaptation. *IEEE Transactions on Image Processing*, 31:4585–4597, 2022.

[8] Chen Fang, Ye Xu, and Daniel N. Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1657–1664, 2013.

[9] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17:59:1–59:35, 2016.

[10] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[11] Jintao Guo, Lei Qi, and Yinghuan Shi. Domaindrop: Suppressing domain-sensitive channels for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19057–19067, 2023.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[13] Jiajun Hu, Lei Qi, Jian Zhang, and Yinghuan Shi. Domain generalization via inter-domain alignment and intra-domain expansion. *Pattern Recognition*, 146:110029, 2024.

[14] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

[15] Zenan Huang, Haobo Wang, Junbo Zhao, and Nenggan Zheng. idag: Invariant dag searching for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19112–19122, 2023.

[16] Zeyi Huang, Haohan Wang, Eric P. Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 124–140, 2020.

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 448–456, 2015.

[18] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[19] Bo Li, Yifei Shen, Jingkang Yang, Yezhen Wang, Jiawei Ren, Tong Che, Jun Zhang, and Ziwei Liu. Sparse mixture-of-experts are domain generalizable learners. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[20] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5543–5551, 2017.

[21] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 3490–3497, 2018.

[22] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5400–5409, 2018.

[23] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 647–663, 2018.

[24] JunYi Lim, Vishnu Monn Baskaran, Joanne Mun-Yee Lim, KokSheik Wong, John See, and Massimo Tistarelli. ERNet: An efficient and reliable human-object interaction detection network. *IEEE Transactions on Image Processing*, 32:964–979, 2023. doi: 10.1109/TIP.2022.3231528.

[25] Shaocong Long, Qianyu Zhou, Chenhao Ying, Lizhuang Ma, and Yuan Luo. Rethinking domain generalization: Discriminability and generalizability. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(11):11783–11797, 2024. doi: 10.1109/TCSVT.2024.3422887.

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

[27] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8690–8699, 2021.

[28] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12553–12562, 2020.

[29] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[30] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Proceedings of the European Conference on Computer Vision Workshops (ECCV Workshops)*, pages 443–450, 2016.

[31] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5385–5394, 2017.

[32] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31, 2018.

[33] Mengzhu Wang, Yuehua Liu, Jianlong Yuan, Shanshan Wang, Zhibin Wang, and Wei Wang. Inter-class and inter-domain semantic augmentation for domain generalization. *IEEE Transactions on Image Processing*, 33:1338–1347, 2024.

[34] Pengfei Wang, Zhaoxiang Zhang, Zhen Lei, and Lei Zhang. Sharpness-aware gradient matching for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3769–3778, 2023.

[35] Yunqi Wang, Furui Liu, Zhitang Chen, Yik-Chung Wu, Jianye Hao, Guangyong Chen, and Pheng-Ann Heng. Contrastive-ace: Domain generalization through alignment of causal mechanisms. *IEEE Transactions on Image Processing*, 32:235–250, 2022.

[36] Zhipeng Xu, De Cheng, Xinyang Jiang, Nannan Wang, Dongsheng Li, and Xinbo Gao. Adversarial domain prompt tuning and generation for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18584–18595, 2025.

[37] Xufeng Yao, Yang Bai, Xinyun Zhang, Yuechen Zhang, Qi Sun, Ran Chen, Ruiyu Li, and Bei Yu. Pcl: Proxy-based contrastive learning for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7087–7097, 2022.

[38] Xi Yu, Shinjae Yoo, and Yuewei Lin. Clipceil: Domain generalization through clip via channel refinement and image-text alignment. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[39] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[40] Chaoyang Zhou, Zengmao Wang, and Bo Du. Learning intrinsic invariance within intra-class for domain generalization. *IEEE Transactions on Multimedia*, 27:3807–3820, 2025.

[41] Fan Zhou, Zhuqing Jiang, Changjian Shui, Boyu Wang, and Brahim Chaib-draa. Domain generalization via optimal transport with metric similarity learning. *Neurocomputing*, 456:469–480, 2021.

[42] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[43] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4396–4415, 2023.

[44] Jilong Zhu, Junbao Zhuo, and Shuhui Wang. PIC: Domain generalization by path information constraint. *Pattern Recognition*, 168:111769, 2025.

A Technical appendices and supplementary material

Feature representation visualization. We employ t-SNE [26] to visualize the learned feature representations, as shown in Figure 4. Compared with ERM, which exhibits relatively limited inter-class separation, our method yields more clearly separated class-wise clusters. This observation indicates that the proposed CSD and SSA modules facilitate the learning of more discriminative and domain-consistent representations.

Batch size stability analysis. We evaluate the proposed method under different batch sizes (16, 32, and 64) while keeping other training settings unchanged. The performance remains stable across all datasets, with accuracy variations within 0.2% on PACS, 0.5% on Office-Home, and 0.3% on VLCS. These results indicate that the proposed method is robust to batch size choices and maintains stable training behavior even under small-batch settings.

Experimental Details Our method is implemented based on the DomainBed framework and built upon the publicly available code of SWAD[4] and PCL[37]. In experiments, an ImageNet pre-trained ResNet-50[12] is adopted as the backbone network, and the Adam optimizer is used for training.

In all experiments, the batch size is set to 32, the learning rate is set to 5×10^{-5}, and the EMA momentum coefficients for class and domain statistics are fixed at 0.9. The weighting coefficients for

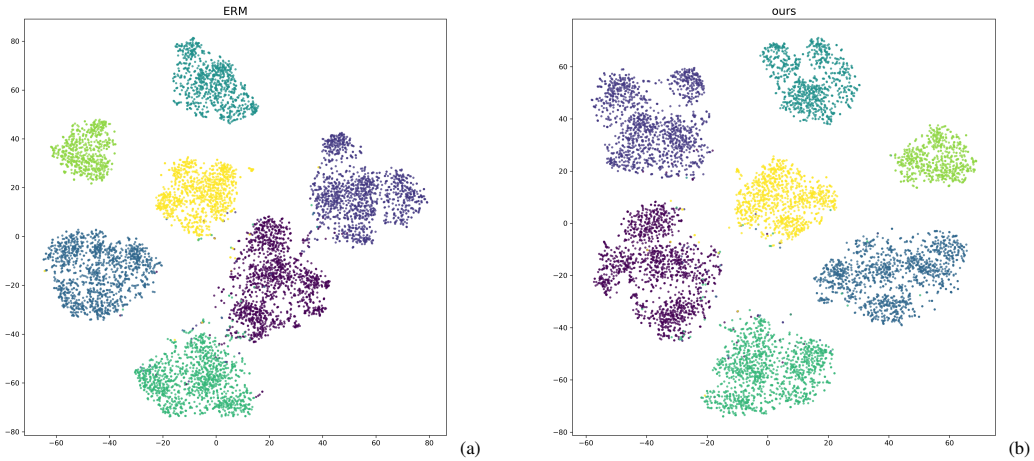


Figure 4: t-SNE visualization.

the prediction consistency term and the subspace consistency regularization term in the loss function are set to 0.1 , and remain unchanged across all datasets. Additionally, the gating function parameters in the channel sensitivity decomposition module are all learnable and are automatically optimized by the network through backpropagation.

We follow the standard leave-one-domain-out evaluation protocol, randomly selecting 20% of the samples from each source domain as the validation set. Except for the Office-Home dataset, which is trained for 3000 iterations, all other datasets are trained for 5000 iterations. To ensure the stability of the results, all experiments are conducted with three different random seeds, and the average accuracy along with the standard deviation are reported.

Experimental results in each domain In the main text, we presented the average accuracy for each dataset; here, we present the accuracy for each domain within each dataset.

Table 6: Comparison of ACC(%) with the SOTA methods on PACS.

Method	A	C	P	S	Avg
CORAL [30]	88.3±0.2	80.0±0.5	97.5±0.3	78.8±1.3	86.2
MMD [22]	86.1±1.4	79.4±0.9	96.6±0.2	76.5±0.5	84.7
MLDG [21]	85.5±1.4	80.1±0.7	97.4±0.3	76.6±1.1	84.9
IRM [1]	84.8±1.3	76.4±1.1	96.7±0.6	76.1±1.0	83.5
RSC [16]	85.4±0.8	79.7±1.8	97.6±0.3	78.2±1.2	85.2
SupCon [18]	89.4±0.5	83.5±0.9	97.4±0.2	82.3±0.4	88.1
MTL [3]	87.5±0.8	77.1±0.5	96.4±0.8	77.3±1.8	84.6
ERM [10]	84.7±0.4	80.8±0.6	97.2±0.3	79.3±1.0	85.5
MixStyle [42]	86.8±0.5	79.0±1.4	96.6±0.1	78.5±2.3	85.2
SagNet [27]	87.4±1.0	80.7±0.6	97.1±0.1	80.0±0.4	86.3
SWAD [4]	89.3±0.2	83.4±0.6	97.3±0.3	82.5±0.5	88.1
PCL [37]	90.2±0.2	83.9±0.6	98.1±0.1	82.6±1.2	88.7
iDAG [15]	90.8±0.4	83.7±0.5	98.0±0.3	82.7±0.9	88.8
IAIE-Pair [13]	90.1±0.4	83.9±1.0	97.2±0.3	83.8±0.1	88.8
IAIE-Proxy [13]	90.6±0.3	84.0±1.0	97.8±0.1	84.1±0.4	89.2
I^3C [40]	-	-	-	-	88.2
Ours	**90.8±0.3**	**84.2±0.4**	**98.1±0.6**	**84.9±0.8**	**89.5**

Parameter sensitivity analysis. We further investigated the effect of different weights of the loss term on performance. Although the optimal balance between them varies slightly across different datasets, the overall impact on accuracy remains small, as shown in the figure. For consistency and simplicity across all experiments, we therefore fixed both weights to 0.1.

Table 7: Comparison of ACC(%) with the SOTA methods on Office-Home.

Method	A	C	P	R	Avg
CORAL [30]	65.3±0.4	54.4±0.5	76.5±0.1	78.4±0.5	68.7
MMD [22]	60.4±0.2	53.3±0.3	74.3±0.1	77.4±0.6	66.4
MLDG [21]	61.5±0.9	53.2±0.6	75.0±1.2	77.5±0.4	66.8
IRM [1]	58.9±2.3	52.2±1.6	72.1±2.9	74.0±2.5	64.3
RSC [16]	60.7±1.4	51.4±0.3	74.8±1.1	75.1±1.3	65.5
SupCon [18]	66.0±0.2	57.0±0.3	78.8±0.3	80.6±0.2	70.6
MTL [3]	61.5±0.7	52.4±0.6	74.9±0.4	76.8±0.4	66.4
ERM [10]	63.1±0.3	51.9±0.4	77.2±0.5	78.1±0.2	67.6
MixStyle [42]	51.1±0.3	53.2±0.4	68.2±0.7	69.2±0.6	60.4
SagNet [27]	63.4±0.2	54.8±0.4	75.8±0.4	78.3±0.3	68.1
SWAD [4]	66.1±0.4	57.7±0.4	78.4±0.1	80.2±0.2	70.6
PCL [37]	67.3±0.2	**59.9±0.1**	78.7±0.2	80.7±0.1	71.6
iDAG [15]	68.2±0.4	57.9±0.3	**79.7±0.2**	81.4±0.4	71.8
IAIE-Pair [13]	65.8±0.4	57.9±0.2	79.1±0.3	80.8±0.1	70.9
IAIE-Proxy [13]	67.5±0.2	58.5±0.1	**79.7±0.3**	81.3±0.2	71.8
I^3C [40]	-	-	-	-	72.1
Ours	**67.9±0.3**	**59.9±0.1**	79.6±0.2	**81.6±0.2**	**72.3**

Table 8: Comparison of ACC(%) with the SOTA methods on VLCS.

Method	C	L	S	V	Avg
CORAL [30]	98.3±0.1	66.1±1.2	73.4±0.3	77.5±1.2	78.8
MMD [22]	97.7±0.1	64.0±1.1	72.8±0.2	75.3±3.3	77.5
MLDG [21]	97.4±0.2	65.2±0.7	71.0±1.4	75.3±1.0	77.2
IRM [1]	98.6±0.1	64.9±0.9	73.4±0.6	77.3±0.9	78.6
RSC [16]	97.9±0.1	62.5±0.7	72.3±1.2	75.6±0.8	77.1
SupCon [18]	98.7±0.1	63.7±0.2	75.6±0.5	79.0±0.2	79.3
MTL [3]	97.8±0.4	64.3±0.3	71.5±0.7	75.3±1.7	77.2
ERM [10]	98.0±0.3	64.7±1.2	71.4±1.2	75.2±1.6	77.3
MixStyle [42]	98.6±0.3	64.5±1.1	72.6±0.5	75.7±1.7	77.9
SagNet [27]	97.9±0.4	64.5±0.5	71.4±1.3	77.5±0.5	77.8
SWAD [4]	98.8±0.1	63.3±0.3	75.3±0.5	79.2±0.6	79.1
PCL [37]	**99.0±0.1**	63.4±0.6	73.8±0.3	75.6±1.8	78.0
IAIE-Pair [13]	98.9±0.1	64.0±0.2	**76.1±0.1**	79.5±0.3	79.6
IAIE-Proxy [13]	98.9±0.1	63.0±0.7	72.8±1.4	79.2±0.4	78.5
I^3C [40]	-	-	-	-	79.9
Ours	98.9±0.2	**67.9±0.4**	74.3±0.7	**79.7±0.9**	**80.2**

Table 9: Comparison of ACC(%) with the SOTA methods on TerraIncognita.

Method	L100	L38	L43	L46	Avg
CORAL [30]	51.6±2.4	42.2±1.0	57.0±1.0	39.8±2.9	47.7
MMD [22]	41.9±3.0	34.8±1.0	57.0±1.9	35.2±1.8	42.2
MLDG [21]	54.2±3.0	44.3±1.1	55.6±0.3	36.9±2.2	47.8
IRM [1]	54.6±1.3	39.8±1.9	56.2±1.8	39.6±0.8	47.6
RSC [16]	50.2±2.2	39.2±1.4	56.3±1.4	40.8±0.6	46.6
SupCon [18]	61.4±1.0	46.0±2.2	57.9±0.7	37.5±0.9	50.7
MTL [3]	49.3±1.2	39.6±6.3	55.6±1.1	37.8±0.8	45.6
ERM [10]	54.3±0.4	42.5±0.7	55.6±0.3	38.8±2.5	47.8
MixStyle [42]	54.3±1.1	34.1±1.1	55.9±1.1	31.7±2.1	44.0
SagNet [27]	53.0±2.9	43.0±2.5	57.9±0.6	40.4±1.3	48.6
SWAD [4]	55.4±0.0	44.9±1.1	59.7±0.4	39.9±0.2	50.0
PCL [37]	58.7±0.7	46.3±1.5	**60.0±0.7**	**43.6±0.6**	52.1
IAIE-Pair [13]	61.3±1.0	50.0±1.8	57.6±0.6	40.3±1.1	52.2
IAIE-Proxy [13]	59.5±1.4	**50.9±1.0**	58.5±0.3	41.6±0.6	52.6
I^3C [40]	-	-	-	-	51.5
Ours	**62.5±0.9**	50.6±1.4	59.7±1.4	42.3±1.6	**53.8**

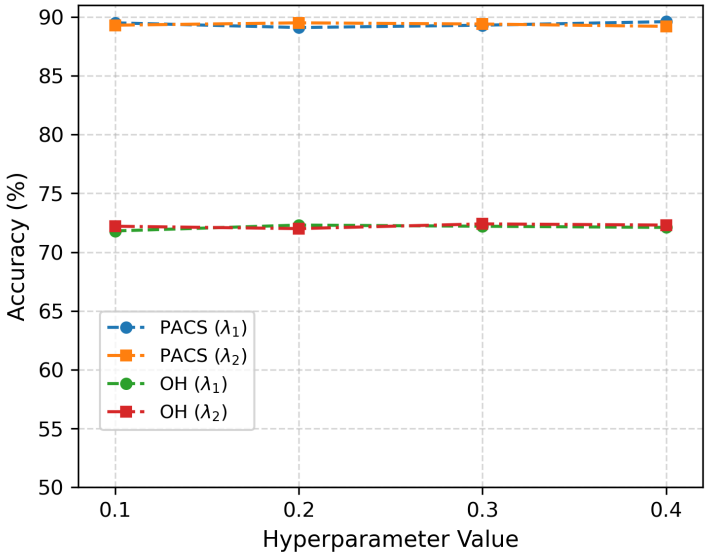


Figure 5: Hyperparameter analysis on PACS and Office-Home datasets

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The two core claims (CSD and SSA) stated in the abstract and introduction are fully aligned with the methodology in Sections 3–4 and the experimental results, accurately reflecting the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations (e.g., dependence on training diversity and empirical tuning of strategies) in the final paragraph of Section 5, so the answer is Yes.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors

should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not contain any theorems, lemmas, or formal proofs; all contributions are empirical methodology and experimental validation, so the answer is NA.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides most key details (backbone, optimizer, hyperparameters, data splits, random seeds), though some implementation specifics (e.g., image preprocessing) rely on DomainBed defaults without explicit mention; thus the answer is Yes with minor caveats.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.

- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper does not provide any open access to code or data, nor does it include instructions or supplemental material for reproducibility.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The paper specifies key experimental details (backbone, optimizer, learning rate, batch size, EMA momentum, loss weights, data splits, training iterations) in Section 4.2 and the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper explicitly reports mean ± standard deviation over three random seeds in the appendix tables (e.g., 90.8±0.3), which is appropriate for understanding statistical variability.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [No]

Justification: The paper provides the main training configurations but does not report detailed computational resource requirements.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper uses public benchmark datasets and does not involve human subjects, privacy, or discrimination issues, thus conforming to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper does not discuss any societal impacts; however, as foundational research on domain generalization methods, it has no direct path to specific societal applications or harms.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

Justification: The paper uses standard public benchmark datasets and an ImageNet-pretrained ResNet-50, without involving generative models or high-risk scraped data, thus posing no such risks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We provide credits to all works utilized in this study

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not release any new assets (e.g., code, datasets, or models), so this question is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve research with human subjects, so no IRB approval or related disclosure is applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core methodology of the paper does not involve LLMs; any potential LLM usage would be limited to writing assistance and does not affect the research contributions.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.