GRADIENT PRINCIPAL COMPONENT ANALYSIS FOR DOMAIN GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain generalization aims to learn a robust model from multiple source domains that can perform well on target domain. Existing methods usually regularize model learning by reshaping forward feature distributions, thereby learning representations that are both domain-invariant and class-discriminative. These methods ultimately achieve the desired objective through regularization losses. However, during the optimization process, the parameter update directions are inevitably entangled with domain-specific directions. To address this issue, we propose a new framework that disentangles shared principal directions from domain-specific directions in the gradient space, and suppresses the latter to promote cross-domain generalization. Our method first estimates a cross-domain shared principal subspace from aggregated uncentered second-order feature statistics. Under shared-structure assumption, this subspace approximates principal task-relevant structures that recur across source domains. During backpropagation, raw gradients are projected along the input feature dimension onto this feature derived subspace. Comprehensive experiments across multiple benchmarks show that our method achieves superior generalization performance compared to state-of-the-art methods. Moreover, we provide theoretical analyses of the validity and effectiveness of principal-subspace-based gradient surgery, offering theoretical foundations for multiple domain generalization settings.

1 INTRODUCTION

Deep neural networks have achieved remarkable success in visual recognition tasks such as image classification. However, their strong performance usually relies on the assumption that training and test data are drawn from the same distribution (Ben-David et al., 2006; Peng et al., 2019; Langford & Schapire, 2005). In real-world applications, this assumption often fails due to variations in acquisition devices, imaging conditions, background, illumination, visual styles, and data sources. Such distribution shifts between training and test domains can lead to significant performance degradation when models are deployed in unseen target domains. **D**omain **G**eneralization (DG) aims to address this problem by learning a model from one or multiple source domains, without accessing any target domain during training, such that the learned model can generalize well to the target domain.

Existing multi-source domain generalization methods can be broadly categorized into two groups: data augmentation and domain alignment. Data augmentation methods attempt to mitigate distribution shifts by enlarging the source distribution. These methods synthesize potential domain variations through style transfer, feature perturbation, and domain randomization, such as MixStyle (Zhou et al., 2021), thereby reducing the model's reliance on domain-specific styles. Domain alignment methods usually encourage different source domains to have similar feature distributions through maximum mean discrepancy (MMD) (Li et al., 2018b), correlation alignment (CORAL) (Sun & Saenko, 2016), adversarial domain classifiers (DANN) (Li et al., 2019), invariant risk minimization (IRM) (Ahuja et al., 2020), or other inter-domain consistency constraints.

Rather than analyzing existing methods from the perspective of representation learning, we revisit them from an optimization perspective. The above two categories mainly influence model learning indirectly through loss function regularization. Data augmentation methods mainly alter input samples or intermediate features, supervising model learning through the forward process. Domain alignment methods typically impose additional losses on features obtained from the forward

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

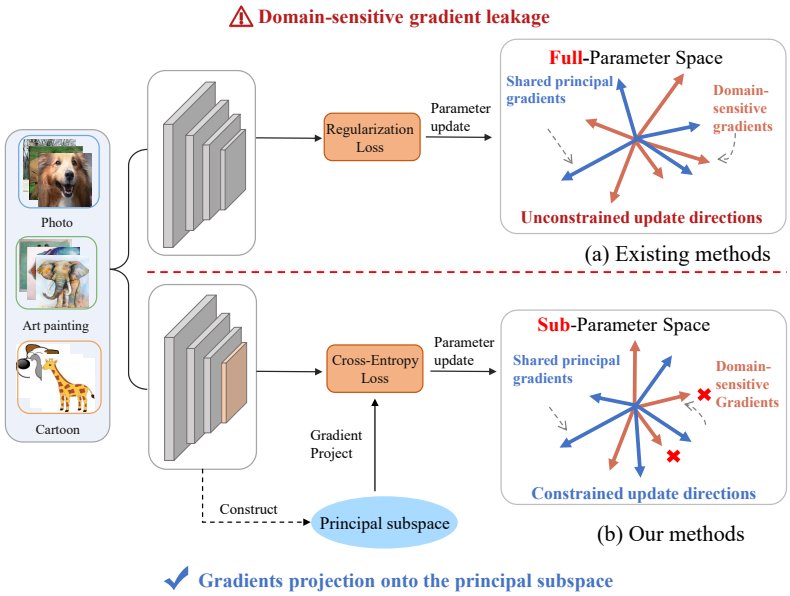


Figure 1: An illustration of our idea. (a) Existing methods suffer from domain-sensitive gradient leakage during loss optimization. (b) Our method constrains the directions of parameter updates to lie within the cross-domain shared principal subspace.

pass, forcing representations from different domains to become closer. However, they still suffer from a key limitation. During loss optimization process, existing methods do not explicitly constrain parameter update directions, leaving class-discriminative and domain-specific directions entangled. Consequently, the learned class-discriminative representations may still be contaminated by domain-specific components. See Figure 1. To address this issue, we formulate the problem from the perspective of gradient disentanglement and explicitly constrain parameter updates to follow directions supported by the cross-domain shared principal subspace. This encourages the model to update consistently along shared principal directions, thereby learning representations that are both domain-invariant and task-relevant.

Based on this observation, we propose **G**radient **P**rincipal **C**omponent **A**nalysis (**GPCA**) for Domain Generalization, a principal subspace gradient surgery framework. Our key idea is that the second-order statistics of source-domain features contain cross-domain shared structures, whereas domain-specific variations tend to manifest as residual directions that are not consistently aligned across domains. Our method is based on the assumption that principal second-order structures that consistently recur across source domains are more likely to encode shared task-relevant information, whereas domain-specific variations contribute primarily to residual structures whose aggregate spectral magnitude is relatively limited. Based on this assumption, we construct a cross-domain shared principal subspace from the aggregated uncentered second-order feature statistics and use it to guide gradient correction during backward optimization. In this way, parameter updates are encouraged to follow shared principal directions, while the influence of domain-specific directions on the optimization trajectory is suppressed.

Moreover, our method can naturally extend to different domain generalization settings. In conventional multi-source DG, source-domain labels can be used to estimate covariance structures for individual domains and construct a shared principal subspace. In domain-label-free multi-source DG, training can directly maintain a global second-moment matrix over all source samples, without explicitly separating samples by their domain labels. This follows from the equivalence between the weighted aggregation of domain-wise second moments and the second moment computed from concatenated features. Therefore, the proposed method serves as a unified framework for multi-source and domain-label-free multi-source DG settings. The main contributions are summarized as follows:

- We revisit domain generalization from the perspective of the optimization process and propose a new framework termed Gradient Principal Component Analysis.
- We introduce a principal-subspace-guided gradient correction strategy.
- We theoretically justify the proposed gradient principal component update mechanism by analyzing the validity and effectiveness of its gradient correction strategy.
- We conduct extensive experiments on multiple domain generalization benchmarks to validate the effectiveness of GPCA. The results demonstrate that GPCA consistently improves generalization performance on target domain.

In summary, our method directly regularizes parameter update directions during backward optimization, encouraging the model to optimize within the principal subspace.

2 Related work

Optimization-based strategy. Several studies have started to investigate domain generalization from the perspective of loss optimization and gradient updates. Gradient matching methods encourage consistent update directions across domains by reducing the discrepancy between domain-specific gradients, as in Fish (Shi et al., 2022). Gradient variance alignment methods further match domain-level gradient statistics from the perspective of Fisher information or local loss landscapes, as in Fishr (Rame et al., 2022). In addition, sharpness-aware methods prefer flatter minima to improve robustness against distribution shifts, such as SAGM (Wang et al., 2023). Mansilla et al. (2021) computes domain-specific gradients and constructs a consensus gradient by checking the element-wise sign agreement across domains. These studies suggest that, compared with only manipulating forward pass, directly regularizing the backward optimization process can more explicitly influence the solution learned by the model. Compared with existing gradient manipulation methods, our method has a fundamental difference. These methods usually compare gradients generated by different source domains and determine update directions based on inter-domain gradient consistency. In contrast, our method does not simply force gradients from different domains to become similar. Instead, it uses the second-order geometry of source-domain features to identify which directions are more likely to be shared and then performs gradient surgery based on the resulting principal subspace.

3 Problem formulation

Let $\mathcal{X}$ and $\mathcal{Y} = \{1, 2, \cdots, C\}$ be the input and the class label spaces, respectively. A domain can be represented by a joint probability distribution on $\mathcal{X} \times \mathcal{Y}$. For simplicity, let $\mathbb{P}^k(X, Y)$ be the joint probability distribution of the k-th source domain, where $k = 1, 2, \cdots, K$. And let $\mathbb{P}^{\mathrm{t}}(X, Y)$ be the joint probability distribution of the target domain.

In DG, the target domain is related but different from any of the source domains, i.e., $\forall k = 1, 2, \cdots, K$, $\mathbb{P}^{\mathrm{t}}(X, Y) \neq \mathbb{P}^k(X, Y)$. The k-th source domain is associated with a training sample set $D_k = \{x_i^k, y_i^k\}_{i=1}^{n^k}$, where $x_i^k \in \mathcal{X}$, $y_i^k \in \mathcal{Y}$, and $(x_i^k, y_i^k) \sim \mathbb{P}^k(X, Y)$. The goal of DG is to learn a classification function $f : \mathcal{X} \to \mathcal{Y}$ from source domain datasets $\{D_k\}_{k=1}^K$ that generalizes well on the target domain $\mathbb{P}^{\mathrm{t}}(X, Y)$.

4 Method

We propose a domain generalization method based on **G**radient **P**rincipal **C**omponent **A**nalysis. The overall framework consists of three consecutive steps. First, we maintain online uncentered covariance statistics of source-domain features, and employ exponential moving average to obtain cross-batch second-order feature structures. Next, we aggregate the covariance matrices from multiple source domains and perform eigenvalue decomposition to extract a principal feature subspace jointly supported by different domains. Finally, during standard supervised training, we project the original gradients onto the constructed principal subspace. The overall framework is illustrated in Figure 2. Algorithm pseudocode 1 is provided in the Appendix.

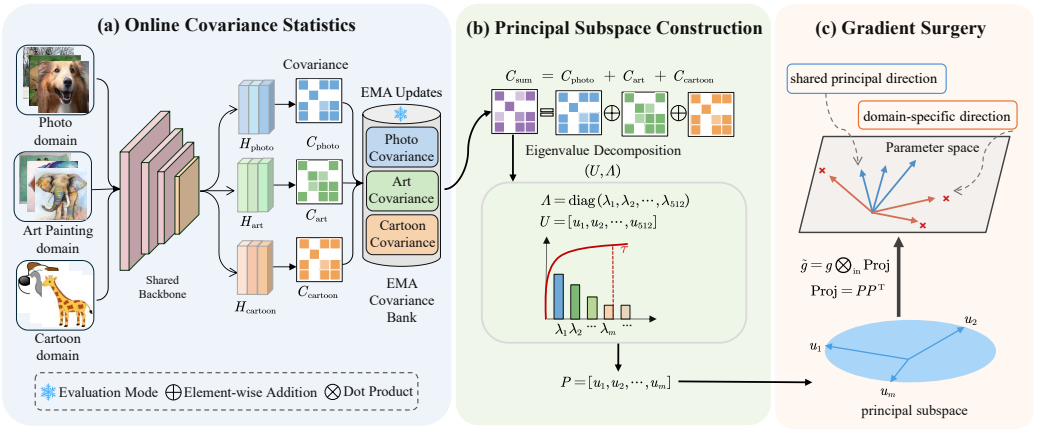


Figure 2: Overview of the proposed method. Our method consists of three consecutive steps. We first estimate online covariance matrices, then construct the principal subspace, and finally perform gradient correction within this subspace.

4.1 Online Covariance Statistics

Driven by the classification objective, we assume that high-energy second-order structures consistently shared across source domains are more likely to capture class-relevant information, while domain-specific variations mainly appear as residual components. Since second-order statistics characterize both feature magnitudes and co-activation patterns, we maintain layer-wise uncentered feature covariance matrices across source domains to identify a shared principal subspace. Please refer to Theorem 1 for the theoretical analysis.

Moreover, the number of sampled features is often much smaller than the feature dimensionality. For $\boldsymbol{Z}_{l,k}^{(t)} \in \mathbb{R}^{N_{l,k} \times d_l}$, we typically have $N_{l,k} \ll d_l$, making the minibatch-based covariance estimate $\boldsymbol{C}_{l,k}^{(t)}$ prone to rank deficiency and unstable eigenspaces. This issue is further aggravated by the continuously evolving intermediate representations during training, which can cause substantial fluctuations in the estimated principal directions. Leveraging the additivity of uncentered feature covariance matrices, we adopt an online estimation strategy with **E**xponential **M**oving **A**verage (**EMA**). This design preserves information from the current representation while smoothly accumulating historical statistics, thereby yielding stable and dynamically updated cross-batch feature structures. For the t-th training step, we receive one minibatch from each source domain: $(X_k, Y_k)_{k=1}^{K}$. To estimate feature statistics without interfering with gradient computation or stochastic training behavior, we first switch the network to evaluation mode and disable gradient tracking. For the k-th source domain, the intermediate feature extracted at the l-th target layer is denoted by $\boldsymbol{H}_{l,k}^{(t)}$. For convolutional layers, $\boldsymbol{H}_{l,k}^{(t)}$ is typically a four-dimensional tensor. We pool its spatial locations and reorganize the resulting activations into a set of feature vectors: $\boldsymbol{Z}_{l,k}^{(t)} \in \mathbb{R}^{N_{l,k} \times d_l}$, For fully connected layers, $\boldsymbol{Z}_{l,k}^{(t)}$ can be directly obtained from the input activations of the corresponding layer.

For clarity, we refer to the uncentered second-moment matrix as the uncentered covariance matrix throughout this paper. For the k-th source domain at step t, we compute

$$\boldsymbol{C}_{l,k}^{(t)} = \frac{1}{N_{l,k}} \left(\boldsymbol{Z}_{l,k}^{(t)}\right)^{\top} \boldsymbol{Z}_{l,k}^{(t)} \in \mathbb{R}^{d_l \times d_l}. \tag{1}$$

We maintain EMA as

$$\bar{\boldsymbol{C}}_{l,k}^{(t)} = \beta \bar{\boldsymbol{C}}_{l,k}^{(t-1)} + (1-\beta)\boldsymbol{C}_{l,k}^{(t)}, \tag{2}$$

where $\beta \in [0, 1)$ denotes the EMA decay rate. To account for potentially different numbers of feature vectors across source domains, we aggregate the domain-wise statistics using sample-

proportional weights:

$$\bar{\boldsymbol{C}}_l^{(t)} = \sum_{k=1}^{K} w_{l,k}^{(t)} \bar{\boldsymbol{C}}_{l,k}^{(t)}, \tag{3}$$

where $w_{l,k}^{(t)} = \frac{N_{l,k}}{\sum_{j=1}^{K} N_{l,j}}$. When all source domains contribute the same number of feature vectors, this aggregation reduces to uniform averaging.

4.2 Principal Subspace Construction

The principal directions of the aggregated covariance can serve as an approximation of cross-domain stable, high-energy feature structures and provide a basis for subsequently constraining parameter update directions. This subspace is expected to contain class-discriminative components, while domain-specific components are relatively suppressed. We then perform eigende value composition:

$$\bar{\boldsymbol{C}}_l^{(t)} = \boldsymbol{U}_l^{(t)} \boldsymbol{\Lambda}_l^{(t)} \left(\boldsymbol{U}_l^{(t)}\right)^{\top}, \tag{4}$$

where

$$\boldsymbol{\Lambda}_l^{(t)} = \operatorname{diag}\left(\lambda_{l,1}^{(t)}, \ldots, \lambda_{l,d_l}^{(t)}\right), \tag{5}$$

and $\lambda_{l,1}^{(t)} \geq \cdots \geq \lambda_{l,d_l}^{(t)} \geq 0$. Given an energy threshold $\tau \in (0, 1]$, we select

$$m_l^{(t)} = \min\left\{ m : \frac{\sum_{i=1}^{m} \lambda_{l,i}^{(t)}}{\sum_{i=1}^{d_l} \lambda_{l,i}^{(t)}} \geq \tau \right\}. \tag{6}$$

The principal basis is defined as

$$\boldsymbol{P}_l^{(t)} = \boldsymbol{U}_{l[:,1:m_l^{(t)}]}^{(t)}, \tag{7}$$

and the corresponding orthogonal projection matrix is $\boldsymbol{P}_l^{(t)}(\boldsymbol{P}_l^{(t)})^{\top}$.

Since network representations evolve rapidly during the early stage of training, reconstructing the subspace at a fixed frequency may fail to capture fast changes in the feature geometry. Conversely, repeatedly performing eigen decomposition in later training stages introduces unnecessary computational overhead. We therefore adopt a stage-wise dynamic update frequency:

$$F_t = \begin{cases} F_{\text{warmup}}, & t \leq T_{\text{warmup}} \\ F_{\text{normal}}, & t > T_{\text{warmup}}. \end{cases}$$

The covariance aggregation, eigen value decomposition, and subspace update are performed only when $t \bmod F_t = 0$. At the remaining iterations, the most recently constructed subspace is reused. In practice, we set $F_{\text{warmup}} < F_{\text{normal}}$, such that the model can rapidly track representation changes during early training and maintain a stable cross-domain subspace estimate with a lower update frequency in later stages. The resulting $\boldsymbol{P}_l^{(t)}$ is subsequently used for gradient projection, guiding the model to optimize along principal gradient components.

4.3 Gradient Correction

The proposed gradient correction constrains parameter updates induced by the gradients to lie within the constructed principal subspace, thereby suppressing domain-specific gradient components that are orthogonal to this subspace.

Specifically, for each selected layer $l \in L_{\text{lay}}$, the original gradient is obtained by standard backpropagation: $\boldsymbol{G}_l = \nabla_{\boldsymbol{W}_l} \mathcal{L}_{\text{ce}}$. $\mathcal{L}_{\text{ce}} = \text{CrossEntropy}(\hat{Y}, Y)$. This gradient specifies how the parameter matrix $\boldsymbol{W}_l$ should be updated in the full parameter space to reduce the current classification loss $\mathcal{L}_{\text{ce}}$. For a fully connected layer, $\boldsymbol{G}_l \in \mathbb{R}^{d_l^{\text{out}} \times d_l}$. We project the gradient along its input-feature dimension:

$$\widetilde{\boldsymbol{G}}_l = \boldsymbol{G}_l \boldsymbol{P}_l \boldsymbol{P}_l^{\top}. \tag{8}$$

Here, $\boldsymbol{P}_l \boldsymbol{P}_l^{\top}$ is the orthogonal projection matrix onto the cross-domain principal subspace $\mathcal{S}_l = \operatorname{span}(\boldsymbol{P}_l)$. Namely, each row of the original gradient is projected into the principal subspace spanned

by $\boldsymbol{P}_l$. See Definition 1 in the Appendix. For a convolutional layer, the same operation is applied to the input-channel dimension:

$$\widetilde{\boldsymbol{G}}_l[o,i,u,v] = \sum_{j=1}^{d_l} \boldsymbol{G}_l[o,j,u,v] \left[\boldsymbol{P}_l \boldsymbol{P}_l^\top\right]_{j,i}. \tag{9}$$

Here, o denotes the output channel, i and j denote the input channels, and (u, v) denotes the spatial location within the convolution kernel. This formulation indicates that, for each fixed output channel o and kernel position (u, v), only the gradient vector along the input-channel dimension is projected onto the principal subspace. Thus, the projection preserves the output-channel and spatial-kernel structure while filtering gradient components along the input-feature dimension. The gradient correction is only applied to the selected layers L_{lay}. Parameters outside L_{lay}, including the classifier parameters unless explicitly selected, retain their original gradients.

Theorem 3 establishes the descent validity and subspace-feasibility of the proposed gradient correction, showing that the corrected gradient remains locally descent-compatible while the actual parameter update is constrained to the principal subspace.

4.4 Computational Cost

The main overhead arises from covariance estimation and periodic eigenvalue decomposition. For a target layer of feature dimension d_l, covariance computation requires $\mathcal{O}(N_{l,k} d_l^2)$ operations for each source-domain mini-batch, while eigenvalue decomposition requires $\mathcal{O}(d_l^3)$ operations but is performed only once every F_t steps. During gradient correction, the product $\boldsymbol{G}_l \boldsymbol{P}_l \boldsymbol{P}_l^\top$ can be computed associatively, the cost of calculating the projection gradient is $\mathcal{O}(d_l^{\text{out}} d_l m_l)$,where d_l^{out} denotes the output dimension of the l-th layer, d_l denotes the input feature dimension, and m_l denotes the retained principal subspace dimension.

4.5 Theoretical analysis

Assumption 1. ***Shared Principal Structure***. *For the k-th source domain at layer l, the second-order feature structure is assumed to admit the decomposition*

$$\boldsymbol{C}_{l,k} = \boldsymbol{C}_{l,s} + \boldsymbol{R}_{l,k}, \tag{10}$$

where $\boldsymbol{C}_{l,s} \in \mathbb{R}^{d_l \times d_l}$ denotes the cross-domain shared principal second-order structure, and $\boldsymbol{R}_{l,k} \in \mathbb{R}^{d_l \times d_l}$ denotes the domain-specific residual variation.

Let the eigenvalues of $\boldsymbol{C}_{l,s}$ be arranged in descending order as

$$\lambda_{l,1}^s \geq \lambda_{l,2}^s \geq \cdots \geq \lambda_{l,d_l}^s, \tag{11}$$

and let m_l denote the dimension of the retained principal subspace. We assume that the shared component has a non-zero eigengap

$$\delta_l = \lambda_{l,m_l}^s - \lambda_{l,m_l+1}^s > 0, \tag{12}$$

and that the aggregated domain-specific residual satisfies

$$\left\| \sum_{k=1}^{K} \boldsymbol{R}_{l,k} \right\|_2 < \frac{K\delta_l}{2}. \tag{13}$$

Under this assumption, the shared principal eigenspace remains identifiable after aggregating the second-order statistics from multiple source domains.

Theorem 1. ***Multi-Source Aggregation Estimates Shared Principal Structures***.

Under Assumption 1, the aggregated second-order matrix over all K source domains is

$$\boldsymbol{C}_{l,\text{sum}} = \sum_{k=1}^{K} \boldsymbol{C}_{l,k} = K\boldsymbol{C}_{l,s} + \sum_{k=1}^{K} \boldsymbol{R}_{l,k}. \tag{14}$$

Let $\mathcal{S}_{l,s}$ denote the eigenspace associated with the top m_l eigenvalues of $\boldsymbol{C}_{l,s}$, and let $\widehat{\mathcal{S}}_l$ denote the eigenspace associated with the top m_l eigenvalues of $\boldsymbol{C}_{l,\mathrm{sum}}$. Then,

$$\left\|\sin\Theta\left(\widehat{\mathcal{S}}_l, \mathcal{S}_{l,s}\right)\right\|_2 \leq \frac{2\left\|\sum_{k=1}^{K} \boldsymbol{R}_{l,k}\right\|_2}{K\delta_l}. \tag{15}$$

Theorem 1 shows that multi-source aggregation provides a stable estimate of the principal subspace associated with the cross-domain shared second-order structure. Specifically, the discrepancy between the estimated and shared principal subspaces is controlled by the spectral magnitude of the aggregated domain-specific residual relative to the eigengap of the shared component. Therefore, when the domain-specific residuals remain sufficiently limited after aggregation, the leading eigenspace of $\boldsymbol{C}_{l,\mathrm{sum}}$ consistently captures the cross-domain shared feature structure. Please refer to Theorem 1 in the Appendix for the proof.

Theorem 2. ***Feature-Induced Gradient Subspace Characterization***. *Consider a linear layer*

$$\boldsymbol{O}_l = \boldsymbol{Z}_l \boldsymbol{W}_l^\top, \tag{16}$$

where $\boldsymbol{Z}_l \in \mathbb{R}^{N\times d_l}$ denotes the input feature matrix and $\boldsymbol{W}_l \in \mathbb{R}^{d_l^{\mathrm{out}}\times d_l}$ denotes the weight matrix. Let

$$\boldsymbol{\Delta}_l = \frac{\partial \mathcal{L}}{\partial \boldsymbol{O}_l} \tag{17}$$

denote the back-propagated gradient with respect to the layer output. Then, the gradient with respect to $\boldsymbol{W}_l$ satisfies

$$\boldsymbol{G}_l = \frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_l} = \boldsymbol{\Delta}_l^\top \boldsymbol{Z}_l. \tag{18}$$

Consequently,

$$\mathrm{Row}(\boldsymbol{G}_l) \subseteq \mathrm{Row}(\boldsymbol{Z}_l) \subseteq \mathbb{R}^{d_l}. \tag{19}$$

For any orthogonal projector

$$\boldsymbol{\Pi}_l = \boldsymbol{P}_l \boldsymbol{P}_l^\top \tag{20}$$

defined in the input-feature space, the projected weight gradient satisfies

$$\boldsymbol{G}_l \boldsymbol{\Pi}_l = \boldsymbol{\Delta}_l^\top \left(\boldsymbol{Z}_l \boldsymbol{\Pi}_l\right). \tag{21}$$

Theorem 2 establishes the feature-to-gradient subspace correspondence, showing that the weight-gradient space is induced by the input features and that feature-space projection directly determines the corresponding projected gradient components. This provides the theoretical basis for the proposed feature-guided gradient correction. Please refer to Theorem 2 in the Appendix for the proof.

5 Experiments

We conduct experiments on four domain generalization benchmark datasets, including PACS (Li et al., 2017), VLCS (Fang et al., 2013), OfficeHome (Venkateswara et al., 2017) and TerraIncognita (Beery et al., 2018). Please refer to subsection A.2 of the Appendix for more experimental details.

5.1 Baselines

For multi-source domain generalization, we conduct a comprehensive comparison between the proposed method and state-of-the-art domain generalization methods on four standard datasets with ResNet50. The experimental results are presented in Table 1. We conduct a comparative analysis between earlier methods (such as CORAL (Sun & Saenko, 2016), DANN (Li et al., 2019), CDANN (Li et al., 2018c), MMD (Li et al., 2018b), MLDG (Li et al., 2018a), IRM (Ahuja et al., 2020), RSC (Huang et al., 2020), SupCon (Khosla et al., 2020), MTL (Blanchard et al., 2021), ERM (Vapnik, 1999), Mixstyle (Zhou et al., 2021), Sagnet (Nam et al., 2021), SWAD (Cha et al., 2021), PCL (Yao et al., 2022), I3C (Zhou et al., 2025)) and our proposed method **GPCA**.

For domain-label free multi-source DG, we conduct a comparison between the proposed method and state-of-the-art domain generalization methods on the PACS datasets with ResNet50. We conduct

Table 1: Comparison of ACC (%) with the SOTA methods on four benchmarks for DG.

Method	Datasets				Avg.
	VLCS	PACS	OfficeHome	TerraInc.	
CORAL (Sun & Saenko, 2016)	78.8	86.2	68.7	47.7	70.350
DANN (Li et al., 2019)	78.6	83.7	65.9	46.7	68.725
CDANN (Li et al., 2018c)	77.5	82.6	65.7	45.8	67.900
MMD (Li et al., 2018b)	77.5	84.7	66.4	42.2	67.700
MLDG (Li et al., 2018a)	77.2	84.9	66.8	47.8	69.175
IRM (Ahuja et al., 2020)	78.6	83.5	64.3	47.6	68.500
RSC (Huang et al., 2020)	77.1	85.2	65.5	46.6	68.600
SupCon (Khosla et al., 2020)	79.3	88.1	70.6	50.7	72.175
MTL (Blanchard et al., 2021)	77.2	84.6	66.4	45.6	68.450
ERM (Vapnik, 1999)	77.3	85.5	67.6	47.8	69.550
Mixstyle (Zhou et al., 2021)	77.9	85.2	60.4	44.0	66.875
Sagnet (Nam et al., 2021)	77.8	86.3	68.1	48.6	70.200
SWAD (Cha et al., 2021)	79.1	88.1	70.6	50.0	71.950
PCL (Yao et al., 2022)	78.0	88.7	71.6	52.1	72.600
I3C (Zhou et al., 2025)	79.6	87.1	70.2	49.6	71.625
GGA (Ballas & Diou, 2025)	79.9	87.3	68.5	50.6	71.575
GPCA (Ours)	**80.2**	**88.9**	**72.1**	**52.5**	**73.450**

Table 2: Leave-one-domain-out results on the PACS dataset for domain-label free DG.

Method	Target domain				Avg.
	A	C	P	S	
ERM (Vapnik, 1999)	84.7 ± 0.4	80.8 ± 0.6	97.2 ± 0.3	79.3 ± 1.0	85.5
EFDMix (Zhang et al., 2022)	90.6 ± 0.3	82.5 ± 0.7	$\mathbf{98.1} \pm 0.2$	76.4 ± 1.2	86.9
Mixstyle (Zhou et al., 2021)	86.8 ± 0.5	79.0 ± 1.4	96.6 ± 0.1	78.5 ± 2.3	85.2
Sagnet (Nam et al., 2021)	87.4 ± 1.0	80.7 ± 0.6	97.1 ± 0.1	80.0 ± 0.4	86.3
SWAD (Cha et al., 2021)	89.3 ± 0.2	83.4 ± 0.6	97.3 ± 0.3	82.5 ± 0.5	88.1
SAGM (Wang et al., 2023)	—	—	—	—	86.6
GPCA (Ours)	$\mathbf{89.4} \pm 0.3$	$\mathbf{84.7} \pm 0.4$	98.0 ± 0.2	$\mathbf{83.5} \pm 0.4$	**88.9**

a comparative analysis between existing methods (such as ERM (Vapnik, 1999), EFDMix (Zhang et al., 2022), Mixstyle (Zhou et al., 2021), Sagnet (Nam et al., 2021), SWAD (Cha et al., 2021), SAGM (Wang et al., 2023)) and our proposed method **GPCA**. See Table 2.

In the domain-label-aware setting across four benchmarks and the domain-label-free setting on PACS, our method achieves improved performance.

5.2 Feature Visualization

To evaluate the learned representations, we visualize the feature distributions using t-SNE (Van der Maaten & Hinton, 2008), as illustrated in Figure 3. The visualization reveals that features extracted via the ERM baseline lack intra-class compactness. In contrast, our proposed method yields more compact intra-class clusters and a more distinct feature layout. As shown in the feature visualization, our method produces a more compact intra-class distribution for the "person" class, indicating stronger class consistency.

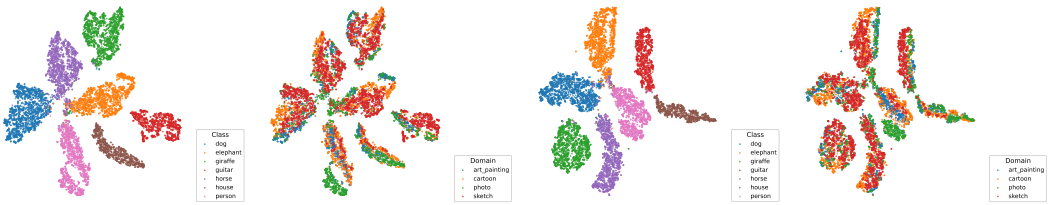
5.3 Parameter Studies

We conduct a sensitivity analysis on the choice of the feature layer for gradient projection, the energy ratio, the EMA decay coefficient, the number of warm-up steps, and the covariance matrix update frequency. See Table 3.

Specifically, varying the projected feature layer only leads to moderate performance fluctuations, indicating that the proposed gradient correction is not restricted to a specific network layer. The per-

Table 3: Parameter sensitivity analysis on the PACS dataset with ResNet50 for Multi-source DG.

Settings		A	C	P	S	Avg.
Selected layers	`layer4.conv3`	89.44	84.74	97.95	83.50	88.91
	`layer4_all`	89.69	83.85	97.23	81.30	88.02
	`layer3.conv6+layer4.conv3`	88.10	83.64	97.68	83.40	88.20
	`layer3_all+layer4_all`	88.23	84.12	97.23	82.09	87.92
Energy Ratio	0.85	88.77	84.44	96.78	83.27	88.32
	0.90	89.08	84.06	97.46	82.79	88.35
	0.95	89.02	85.02	97.68	82.67	88.60
	0.97	89.44	84.74	97.95	83.50	88.91
	0.99	88.96	83.58	96.86	83.05	88.11
EMA Decay	0.90	89.44	84.74	97.95	83.50	88.91
	0.95	88.23	84.38	96.71	83.87	88.30
	0.99	88.53	85.24	96.63	82.16	88.14
Warmup Steps	500	88.53	83.53	97.16	82.70	87.98
	800	88.84	83.32	97.23	81.97	87.84
	1000	89.44	84.74	97.95	83.50	88.91
	2000	89.51	84.44	97.16	82.98	88.52
Cov Frequency	100	88.35	82.25	97.46	84.35	88.10
	200	89.26	83.58	97.16	83.56	88.39
	300	89.44	84.74	97.95	83.50	88.91



(a) ERM-class (b) ERM-domain (c) Ours-class (d) Ours-domain

Figure 3: Comparison of t-SNE visualizations for the baseline and our proposed method on the PACS dataset. Different colors denote different classes or domains.

formance also remains stable across different energy ratios, suggesting that the principal subspace construction is robust to the retained spectral energy. In addition, changes in the EMA decay coefficient and covariance update frequency do not cause significant degradation, which demonstrates that the online covariance estimation is insensitive to the exact smoothing and updating schedule. Finally, our method performs consistently under different warm-up steps, showing that it does not rely on a carefully tuned warm-up stage. These results demonstrate that our method is robust to hyperparameter variations, maintaining stable performance across different hyperparameter settings.

6 Conclusion

We propose a method based on Gradient Principal Component Analysis (GPCA). First, we maintain online uncentered feature covariance statistics to construct shared principal subspace. Then, gradients are projected to principal subspace during supervised training, encouraging updates along shared principal directions while suppressing domain-specific directions. Extensive experiments on multiple public benchmarks demonstrate that our method achieves superior performance compared to state-of-the-art methods. Under an explicit shared structure assumption, our analysis characterizes when the aggregated principal eigenspace approximates the latent shared structure and establishes the descent property of the corresponding projected gradient.

As a unified framework, our method can be applied to multi-source and domain-label-free domain generalization, and can be extended to different backbone networks. Our method highlights the importance of the backward optimization process in domain generalization and suggest that optimization direction correction is a promising path toward improving cross-domain generalization.

AI USE STATEMENT

We used generative AI tools solely to assist with translation and language editing to improve the clarity and readability of the manuscript. We have reviewed all AI-assisted text to ensure that it accurately reflects our intended meaning. We have not used generative AI tools for any other research tasks. We take full responsibility for the final content of this paper.

REFERENCES

Kartik Ahuja, Karthikeyan Shanmugam, Kush R. Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *Proceedings of the International Conference on Machine Learning*, pp. 145–155, 2020.

Aristotelis Ballas and Christos Diou. Gradient-guided annealing for domain generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20558–20568, 2025.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European Conference on Computer Vision*, pp. 472–489, 2018.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Proceedings of the Advances in Neural Information Processing Systems*, pp. 137–144, 2006.

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.

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*, pp. 22405–22418, 2021.

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

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

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*, pp. 124–140, 2020.

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*, 2020.

John Langford and Robert Schapire. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, 6(3):273–306, 2005.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. In *IEEE International Conference on Computer Vision*, pp. 5543–5551, 2017.

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*, pp. 3490–3497, 2018a.

Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M. Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1446–1455, 2019.

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*, pp. 5400–5409, 2018b.

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*, pp. 647–663, 2018c.

Lucas Mansilla, Rodrigo Echeveste, Diego H Milone, and Enzo Ferrante. Domain generalization via gradient surgery. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6630–6638, 2021.

Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8690–8699, 2021.

Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. In *Proceedings of the International Conference on Machine Learning*, pp. 5102–5112, 2019.

Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 18347–18377, 2022.

Yuge Shi, Jeffrey Seely, Philip H. S. Torr, Siddharth Narayanaswamy, Awni Y. Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. In *International Conference on Learning Representations*, 2022.

Baochen Sun and Kate Saenko. Deep CORAL: correlation alignment for deep domain adaptation. In *Proceedings of the European Conference on Computer Vision Workshops*, pp. 443–450, 2016.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11):2579–2605, 2008.

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

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5385–5394, 2017.

Pengfei Wang, Zhaoxiang Zhang, Zhen Lei, and Lei Zhang. Sharpness-aware gradient matching for domain generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition,*, pp. 3769–3778, 2023.

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*, pp. 7087–7097, 2022.

Yabin Zhang, Minghan Li, Ruihuang Li, Kui Jia, and Lei Zhang. Exact feature distribution matching for arbitrary style transfer and domain generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition,*, pp. 8025–8035, 2022.

Chaoyang Zhou, Zengmao Wang, and Bo Du. Learning intrinsic invariance within intra-class for domain generalization. *IEEE Trans. Multim.*, 27:3807–3820, 2025.

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

A Appendix

A.1 Theoretical Details

A.1.1 Definition

Definition 1. ***Feature Subspace***. *Let $\boldsymbol{P}_l \in \mathbb{R}^{d_l \times m_l}$ satisfy $\boldsymbol{P}_l^\top \boldsymbol{P}_l = \boldsymbol{I}$. The principal feature subspace is defined as*

$$\mathcal{S}_l = \operatorname{range}(\boldsymbol{P}_l) = \operatorname{span}\left\{\boldsymbol{p}_{l,1}, \ldots, \boldsymbol{p}_{l,m_l}\right\}. \tag{22}$$

For a matrix $\boldsymbol{G}_l$ whose rows lie in $\mathbb{R}^{d_l}$, its orthogonal projection onto $\mathcal{S}_l$ is

$$\mathrm{Proj}_{\mathcal{S}_l}(\boldsymbol{G}_l) = \boldsymbol{G}_l \boldsymbol{P}_l \boldsymbol{P}_l^\top. \tag{23}$$

A.1.2 Proof of theorem

Theorem 1. Multi-Source Aggregation Estimates Shared Principal Structures.

Proof. Let

$$\boldsymbol{A} = K\boldsymbol{C}_{l,s}, \tag{24}$$

and define the aggregated residual perturbation as

$$\boldsymbol{E} = \sum_{k=1}^{K} \boldsymbol{R}_{l,k}. \tag{25}$$

Then the aggregated second-order matrix can be written as

$$\boldsymbol{C}_{l,\mathrm{sum}} = \boldsymbol{A} + \boldsymbol{E}. \tag{26}$$

Since multiplying a matrix by a positive scalar does not change its eigenvectors, $\boldsymbol{A}$ and $\boldsymbol{C}_{l,s}$ share the same principal eigenspace $\mathcal{S}_{l,s}$. Moreover, the eigengap of $\boldsymbol{A}$ between its m_l-th and (m_l+1)-th eigenvalues is

$$\lambda_{m_l}(\boldsymbol{A}) - \lambda_{m_l+1}(\boldsymbol{A}) = K\delta_l. \tag{27}$$

By assumption,

$$\|\boldsymbol{E}\|_2 < \frac{K\delta_l}{2}. \tag{28}$$

Therefore, the perturbation is smaller than half of the corresponding eigengap, ensuring separation between the leading m_l-dimensional eigenspace and the remaining eigenspace.

Applying the Davis–Kahan sinΘ theorem to $\boldsymbol{A}$ and $\boldsymbol{A}+\boldsymbol{E}$ gives

$$\left\|\sin\Theta\left(\widehat{\mathcal{S}}_l, \mathcal{S}_{l,s}\right)\right\|_2 \leq \frac{2\|\boldsymbol{E}\|_2}{K\delta_l}. \tag{29}$$

Substituting $\boldsymbol{E} = \sum_{k=1}^{K} \boldsymbol{R}_{l,k}$ yields

$$\left\|\sin\Theta\left(\widehat{\mathcal{S}}_l, \mathcal{S}_{l,s}\right)\right\|_2 \leq \frac{2\left\|\sum_{k=1}^{K} \boldsymbol{R}_{l,k}\right\|_2}{K\delta_l}, \tag{30}$$

which completes the proof. □

Theorem 2. Feature-Induced Gradient Subspace Characterization.

Proof. For the linear transformation

$$\boldsymbol{O}_l = \boldsymbol{Z}_l \boldsymbol{W}_l^\top, \tag{31}$$

the differential of the layer output with respect to the weight matrix is

$$d\boldsymbol{O}_l = \boldsymbol{Z}_l d\boldsymbol{W}_l^\top. \tag{32}$$

Let $\boldsymbol{\Delta}_l = \partial\mathcal{L}/\partial\boldsymbol{O}_l$. Using the differential form of the loss, we have

$$d\mathcal{L} = \mathrm{tr}\left(\boldsymbol{\Delta}_l^\top d\boldsymbol{O}_l\right). \tag{33}$$

Substituting $d\boldsymbol{O}_l = \boldsymbol{Z}_l d\boldsymbol{W}_l^\top$ gives

$$d\mathcal{L} = \mathrm{tr}\left(\boldsymbol{\Delta}_l^\top \boldsymbol{Z}_l d\boldsymbol{W}_l^\top\right). \tag{34}$$

Therefore,

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{W}_l} = \boldsymbol{\Delta}_l^\top \boldsymbol{Z}_l, \tag{35}$$

which yields

$$\boldsymbol{G}_l = \boldsymbol{\Delta}_l^\top \boldsymbol{Z}_l. \tag{36}$$

Each row of $\boldsymbol{G}_l$ is thus a linear combination of the rows of $\boldsymbol{Z}_l$. Hence,

$$\mathrm{Row}(\boldsymbol{G}_l) \subseteq \mathrm{Row}(\boldsymbol{Z}_l). \tag{37}$$

Furthermore, for the orthogonal projector $\boldsymbol{\Pi}_l = \boldsymbol{P}_l \boldsymbol{P}_l^\top$, we have

$$\boldsymbol{G}_l \boldsymbol{\Pi}_l = \left(\boldsymbol{\Delta}_l^\top \boldsymbol{Z}_l\right) \boldsymbol{\Pi}_l \tag{38}$$

$$= \boldsymbol{\Delta}_l^\top \left(\boldsymbol{Z}_l \boldsymbol{\Pi}_l\right), \tag{39}$$

where the second equality follows from the associativity of matrix multiplication.

Therefore, projecting the weight gradient along its input-feature dimension retains exactly the gradient components associated with the projected feature directions, which completes the proof. □

Theorem 3. ***Descent and Subspace-Feasibility Properties of Principal-Subspace Gradient Correction***. *Let $\mathcal{L}(\boldsymbol{W})$ denote the classification objective, and let*

$$\boldsymbol{G} = \nabla_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}) \tag{40}$$

be the raw gradient at the current iterate. Let $\boldsymbol{P} \in \mathbb{R}^{d \times m}$ be an orthonormal basis of the principal feature subspace satisfying

$$\boldsymbol{P}^\top \boldsymbol{P} = \boldsymbol{I}, \tag{41}$$

and define the corresponding orthogonal projector as

$$\boldsymbol{\Pi} = \boldsymbol{P}\boldsymbol{P}^\top. \tag{42}$$

The principal-subspace corrected raw gradient is defined as

$$\overline{\boldsymbol{G}} = \boldsymbol{G}\boldsymbol{\Pi} = \boldsymbol{G}\boldsymbol{P}\boldsymbol{P}^\top. \tag{43}$$

Then,

$$\left\langle \boldsymbol{G}, \overline{\boldsymbol{G}} \right\rangle_F = \|\boldsymbol{G}\boldsymbol{P}\|_F^2 = \left\|\overline{\boldsymbol{G}}\right\|_F^2 \geq 0. \tag{44}$$

Assume that there exists a constant $L_s > 0$ such that $\mathcal{L}(\boldsymbol{W})$ has an L_s-Lipschitz continuous gradient in a neighborhood of the current iterate. For the auxiliary projected-gradient update

$$\boldsymbol{W}^+ = \boldsymbol{W} - \eta \overline{\boldsymbol{G}}, \tag{45}$$

we have

$$\mathcal{L}(\boldsymbol{W}^+) \leq \mathcal{L}(\boldsymbol{W}) - \eta \left(1 - \frac{L_s \eta}{2}\right) \left\|\overline{\boldsymbol{G}}\right\|_F^2. \tag{46}$$

Therefore, if

$$0 < \eta < \frac{2}{L_s} \tag{47}$$

and $\overline{\boldsymbol{G}} \neq \boldsymbol{0}$, the corrected raw gradient defines a strict local descent update for the classification objective.

Furthermore, let $\boldsymbol{U}_t$ denote the candidate parameter-update direction generated by Adam from the corrected gradients up to iteration t. Define the final re-projected update as

$$\widetilde{\boldsymbol{U}}_t = \boldsymbol{U}_t \boldsymbol{P}_t \boldsymbol{P}_t^\top. \tag{48}$$

Then,

$$\widetilde{\boldsymbol{U}}_t \boldsymbol{P}_t \boldsymbol{P}_t^\top = \widetilde{\boldsymbol{U}}_t. \tag{49}$$

Hence, the rows of $\widetilde{\boldsymbol{U}}_t$ lie in the current principal subspace $\mathcal{S}_t = \mathrm{range}(\boldsymbol{P}_t)$. Consequently, the actual parameter displacement

$$\boldsymbol{W}_{t+1} - \boldsymbol{W}_t = -\eta_t \widetilde{\boldsymbol{U}}_t \tag{50}$$

is constrained to the current principal subspace along the input-feature dimension.

Theorem 3 characterizes two complementary properties of the proposed gradient correction strategy. First, before adaptive optimization, projecting the raw gradient onto the principal feature subspace preserves non-negative alignment with the original gradient. Under the local smoothness condition, the corrected raw gradient further admits an explicit descent bound, establishing the local optimization validity of the feature-guided gradient correction.

Second, adaptive optimization may change the corrected gradient direction through momentum accumulation and coordinate-wise preconditioning. We therefore re-project the Adam-generated candidate update onto the current principal subspace before applying it to the model parameters. This final projection ensures that the actual parameter displacement satisfies the intended principal-subspace constraint despite the intermediate directional changes introduced by Adam.

The descent bound characterizes the corrected raw-gradient update, whereas the final re-projection characterizes the geometric feasibility of the actual Adam parameter update. Together, these two properties establish the optimization and geometric roles of the proposed gradient correction strategy.

Proof. Since $\boldsymbol{\Pi} = \boldsymbol{P}\boldsymbol{P}^\top$ and $\boldsymbol{P}^\top\boldsymbol{P} = \boldsymbol{I}$, $\boldsymbol{\Pi}$ is a symmetric and idempotent orthogonal projection matrix. For $\overline{\boldsymbol{G}} = \boldsymbol{G}\boldsymbol{\Pi}$, we have

$$\begin{aligned}\left\langle \boldsymbol{G}, \overline{\boldsymbol{G}} \right\rangle_F &= \operatorname{tr}\left(\boldsymbol{G}^\top \boldsymbol{G}\boldsymbol{P}\boldsymbol{P}^\top\right) && (51)\\ &= \operatorname{tr}\left(\boldsymbol{P}^\top \boldsymbol{G}^\top \boldsymbol{G}\boldsymbol{P}\right) && (52)\\ &= \|\boldsymbol{G}\boldsymbol{P}\|_F^2 . && (53)\end{aligned}$$

Moreover,

$$\begin{aligned}\left\|\overline{\boldsymbol{G}}\right\|_F^2 &= \operatorname{tr}\left(\boldsymbol{\Pi}\boldsymbol{G}^\top \boldsymbol{G}\boldsymbol{\Pi}\right) && (54)\\ &= \operatorname{tr}\left(\boldsymbol{P}^\top \boldsymbol{G}^\top \boldsymbol{G}\boldsymbol{P}\right) && (55)\\ &= \|\boldsymbol{G}\boldsymbol{P}\|_F^2 . && (56)\end{aligned}$$

Therefore,

$$\left\langle \boldsymbol{G}, \overline{\boldsymbol{G}} \right\rangle_F = \left\|\overline{\boldsymbol{G}}\right\|_F^2 \geq 0. \tag{57}$$

Since $\mathcal{L}$ has an L_s-Lipschitz continuous gradient in a neighborhood of the current iterate, the descent lemma gives

$$\mathcal{L}(\boldsymbol{W} + \boldsymbol{D}) \leq \mathcal{L}(\boldsymbol{W}) + \langle \boldsymbol{G}, \boldsymbol{D} \rangle_F + \frac{L_s}{2} \|\boldsymbol{D}\|_F^2 . \tag{58}$$

Setting

$$\boldsymbol{D} = -\eta \overline{\boldsymbol{G}}, \tag{59}$$

yields

$$\begin{aligned}\mathcal{L}(\boldsymbol{W}^+) &\leq \mathcal{L}(\boldsymbol{W}) - \eta \left\langle \boldsymbol{G}, \overline{\boldsymbol{G}} \right\rangle_F + \frac{L_s \eta^2}{2} \left\|\overline{\boldsymbol{G}}\right\|_F^2 && (60)\\ &= \mathcal{L}(\boldsymbol{W}) - \eta \left(1 - \frac{L_s \eta}{2}\right) \left\|\overline{\boldsymbol{G}}\right\|_F^2 . && (61)\end{aligned}$$

Thus, if $0 < \eta < 2/L_s$ and $\overline{\boldsymbol{G}} \neq \boldsymbol{0}$, the right-hand side is strictly smaller than $\mathcal{L}(\boldsymbol{W})$.

We next consider the final Adam update. By definition,

$$\widetilde{\boldsymbol{U}}_t = \boldsymbol{U}_t \boldsymbol{P}_t \boldsymbol{P}_t^\top . \tag{62}$$

Using $\boldsymbol{P}_t^\top \boldsymbol{P}_t = \boldsymbol{I}$, we obtain

$$\begin{aligned}\widetilde{\boldsymbol{U}}_t \boldsymbol{P}_t \boldsymbol{P}_t^\top &= \boldsymbol{U}_t \boldsymbol{P}_t \boldsymbol{P}_t^\top \boldsymbol{P}_t \boldsymbol{P}_t^\top && (63)\\ &= \boldsymbol{U}_t \boldsymbol{P}_t \boldsymbol{P}_t^\top && (64)\\ &= \widetilde{\boldsymbol{U}}_t . && (65)\end{aligned}$$

Therefore, $\widetilde{\boldsymbol{U}}_t$ is invariant under projection onto the current principal subspace, implying that each of its rows lies in $\mathrm{range}(\boldsymbol{P}_t)$. Since

$$\boldsymbol{W}_{t+1} - \boldsymbol{W}_t = -\eta_t \widetilde{\boldsymbol{U}}_t, \tag{66}$$

the actual parameter displacement satisfies the same principal-subspace constraint along the input-feature dimension. This completes the proof. □

Theorem 4. ***First-Order Perturbation under Principal-Subspace-Constrained Updates.***

Consider the l-th linear layer

$$\boldsymbol{o}_l = \boldsymbol{W}_l \boldsymbol{z}_l, \tag{67}$$

where $\boldsymbol{z}_l \in \mathbb{R}^{d_l}$ denotes the input feature and $\boldsymbol{W}_l$ denotes the layer weight matrix.

Let $\boldsymbol{P}_l \in \mathbb{R}^{d_l \times m_l}$ be an orthonormal basis of the principal feature subspace, and define

$$\boldsymbol{\Pi}_l = \boldsymbol{P}_l \boldsymbol{P}_l^\top. \tag{68}$$

The feature representation can be decomposed as

$$\boldsymbol{z}_l = \boldsymbol{z}_{l,\parallel} + \boldsymbol{z}_{l,\perp}, \tag{69}$$

where

$$\boldsymbol{z}_{l,\parallel} = \boldsymbol{\Pi}_l \boldsymbol{z}_l, \tag{70}$$

and

$$\boldsymbol{z}_{l,\perp} = (\boldsymbol{I} - \boldsymbol{\Pi}_l)\boldsymbol{z}_l. \tag{71}$$

Let $\boldsymbol{U}_l$ denote the candidate parameter-update direction generated by the optimizer, and let the final re-projected update be

$$\widetilde{\boldsymbol{U}}_l = \boldsymbol{U}_l \boldsymbol{\Pi}_l. \tag{72}$$

For the parameter update

$$\boldsymbol{W}_l^+ = \boldsymbol{W}_l - \eta \widetilde{\boldsymbol{U}}_l, \tag{73}$$

the induced output perturbation satisfies

$$\Delta \boldsymbol{o}_l = -\eta \widetilde{\boldsymbol{U}}_l \boldsymbol{z}_{l,\parallel}, \tag{74}$$

while

$$\widetilde{\boldsymbol{U}}_l \boldsymbol{z}_{l,\perp} = \boldsymbol{0}. \tag{75}$$

Therefore, the principal-subspace-constrained update responds only to feature components contained in the selected principal subspace and is insensitive, at the layer output, to feature components lying in its orthogonal complement.

Theorem 4 characterizes the functional effect of the proposed principal-subspace constraint. Combined with Theorem 1, which shows that multi-source aggregation provides a stable estimate of the shared principal structure, this result indicates that the parameter update mainly responds to feature components supported by the estimated cross-domain shared subspace, while suppressing the direct influence of orthogonal residual components.

Proof. Since

$$\boldsymbol{\Pi}_l = \boldsymbol{P}_l \boldsymbol{P}_l^\top \tag{76}$$

is an orthogonal projection matrix, it satisfies

$$\boldsymbol{\Pi}_l^2 = \boldsymbol{\Pi}_l \tag{77}$$

and

$$\boldsymbol{\Pi}_l(\boldsymbol{I} - \boldsymbol{\Pi}_l) = \boldsymbol{0}. \tag{78}$$

The feature representation is decomposed as

$$\boldsymbol{z}_l = \boldsymbol{z}_{l,\parallel} + \boldsymbol{z}_{l,\perp}, \tag{79}$$

where

$$\boldsymbol{z}_{l,\|} = \boldsymbol{\Pi}_l \boldsymbol{z}_l \tag{80}$$

and

$$\boldsymbol{z}_{l,\perp} = (\boldsymbol{I} - \boldsymbol{\Pi}_l)\boldsymbol{z}_l. \tag{81}$$

Because the final optimizer update is re-projected as

$$\widetilde{\boldsymbol{U}}_l = \boldsymbol{U}_l \boldsymbol{\Pi}_l, \tag{82}$$

its response to the orthogonal feature component is

$$\widetilde{\boldsymbol{U}}_l \boldsymbol{z}_{l,\perp} = \boldsymbol{U}_l \boldsymbol{\Pi}_l (\boldsymbol{I} - \boldsymbol{\Pi}_l)\boldsymbol{z}_l \tag{83}$$

$$= \boldsymbol{0}. \tag{84}$$

For the updated parameter

$$\boldsymbol{W}_l^+ = \boldsymbol{W}_l - \eta \widetilde{\boldsymbol{U}}_l, \tag{85}$$

the corresponding layer output becomes

$$\boldsymbol{o}_l^+ = \boldsymbol{W}_l^+ \boldsymbol{z}_l \tag{86}$$

$$= \left(\boldsymbol{W}_l - \eta \widetilde{\boldsymbol{U}}_l\right) \boldsymbol{z}_l. \tag{87}$$

Therefore, the output perturbation is

$$\Delta \boldsymbol{o}_l = \boldsymbol{o}_l^+ - \boldsymbol{o}_l \tag{88}$$

$$= -\eta \widetilde{\boldsymbol{U}}_l \boldsymbol{z}_l \tag{89}$$

$$= -\eta \widetilde{\boldsymbol{U}}_l \left(\boldsymbol{z}_{l,\|} + \boldsymbol{z}_{l,\perp}\right) \tag{90}$$

$$= -\eta \widetilde{\boldsymbol{U}}_l \boldsymbol{z}_{l,\|}, \tag{91}$$

where the last equality follows from $\widetilde{\boldsymbol{U}}_l \boldsymbol{z}_{l,\perp} = \boldsymbol{0}$.

Thus, the parameter update affects the layer output only through feature components contained in the selected principal subspace, while components in its orthogonal complement have no direct contribution to the output perturbation. This completes the proof. □

Theorem 5. ***Additivity of Uncentered Feature Covariance under Feature Concatenation***. *Let $\boldsymbol{Z}_k \in \mathbb{R}^{n_k \times d}$ denote the feature matrix of the k-th source domain, where n_k is the number of feature vectors and d is the feature dimension. The uncentered second-order feature matrix of the k-th domain is defined as*

$$\boldsymbol{C}_k = \boldsymbol{Z}_k^\top \boldsymbol{Z}_k. \tag{92}$$

By concatenating the feature matrices from all K source domains along the sample dimension, we obtain

$$\boldsymbol{Z} = [\boldsymbol{Z}_1; \boldsymbol{Z}_2; \cdots; \boldsymbol{Z}_K] \in \mathbb{R}^{(\sum_{k=1}^K n_k) \times d}. \tag{93}$$

Then, the uncentered second-order feature matrix computed from the concatenated features is equivalent to the summation of the domain-wise uncentered second-order feature matrices:

$$\boldsymbol{Z}^\top \boldsymbol{Z} = \sum_{k=1}^K \boldsymbol{C}_k. \tag{94}$$

Proof. According to the definition of feature concatenation, we have

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{Z}_1 \\ \boldsymbol{Z}_2 \\ \vdots \\ \boldsymbol{Z}_K \end{bmatrix}. \tag{95}$$

Therefore,

$$\boldsymbol{Z}^\top = \left[\boldsymbol{Z}_1^\top, \boldsymbol{Z}_2^\top, \ldots, \boldsymbol{Z}_K^\top\right]. \tag{96}$$

Then, we have

$$\begin{aligned} \boldsymbol{Z}^\top \boldsymbol{Z} &= \left[\boldsymbol{Z}_1^\top, \boldsymbol{Z}_2^\top, \ldots, \boldsymbol{Z}_K^\top\right] \begin{bmatrix} \boldsymbol{Z}_1 \\ \boldsymbol{Z}_2 \\ \vdots \\ \boldsymbol{Z}_K \end{bmatrix} \\ &= \boldsymbol{Z}_1^\top \boldsymbol{Z}_1 + \boldsymbol{Z}_2^\top \boldsymbol{Z}_2 + \cdots + \boldsymbol{Z}_K^\top \boldsymbol{Z}_K \\ &= \sum_{k=1}^{K} \boldsymbol{Z}_k^\top \boldsymbol{Z}_k \\ &= \sum_{k=1}^{K} \boldsymbol{C}_k. \end{aligned} \tag{97}$$

□

A.2 Implementation Details

A.2.1 Datasets

- **PACS**. PACS (Li et al., 2017) contains 9,991 images, with 4 domains, i.e., Photo (P), Art Painting (A), Cartoon (C), and Sketch (S), and 7 categories: dog, elephant, giraffe, guitar, house, horse, and person.
- **VLCS**. VLCS (Fang et al., 2013) contains 10,729 images, with 4 domains: PASCAL VOC2007 (V), LabelMe (L), Caltech-101 (C), SUN09 (S), and 5 shared classes (bird, car, chair, dog and person) .
- **OfficeHome**. OfficeHome (Venkateswara et al., 2017) contains 15,588 images, with 4 domains (Art, Clipart, Product and Real-World) and 65 shared classes.
- **TerraIncognita**. TerraIncognita (Beery et al., 2018) contains 24,788 images, with 4 domains (L38, L43, L46 and L100) and 10 shared classes.

A.2.2 Experimental Settings

For multi-source domain generalization, we evaluate our method using the DomainBed (Gulrajani & Lopez-Paz, 2021) benchmark across 4 datasets. A leave-one-domain-out strategy is adopted, where one domain serves as the target domain for test, and the remaining domains are used as source domains for training. For model validation and selection, 20% of the samples from each source domain are set as the test set. To ensure result stability, we run experiments with three different random seeds and report the average accuracy. For the number of iterations, Office-Home dataset is trained for $3,000$ steps, while the other datasets are trained for $5,000$ steps. The checkpoint frequency is configured at 500 steps. We employ the Adam optimizer with a learning rate of $7e-5$. Following the standard training and evaluation procedure of SWAD (Cha et al., 2021), we use a batch size of 32 for each domain. The experimental setting for domain-label free multi-source DG is the same as that for multi-source DG.

A.2.3 Algorithm Description

The algorithm 1 of our proposed method is described as follows.

A.2.4 Experiments Computing Resources

We execute our code on a computer with NVIDIA GeForce RTX 3090 531 (GPU) and Intel Xeon Gold 6254 (CPU).

Algorithm 1 Training Procedure of **GPCA**

Require: Target layers L_{lay}, parameters $\{\boldsymbol{W}_l\}$, learning rate η, energy threshold τ, EMA decay β, warmup steps T_{warmup}, frequencies $F_{\text{warmup}}, F_{\text{normal}}$.
Ensure: Optimized parameters θ, ϕ.

Initialize EMA covariance $\bar{\boldsymbol{C}}_{l,k} = 0$ and projection basis $\boldsymbol{P}_l = \text{None}$.
for each training step t **do**
  Obtain intermediate features $\boldsymbol{H}_{l,k}$ from source domains.
  for each layer $l \in L_{\text{lay}}$ and domain k **do**
    Compute covariance: $\boldsymbol{C}_{l,k} = \frac{1}{N} \boldsymbol{Z}_{l,k}^T \boldsymbol{Z}_{l,k}$.
    Update EMA: $\bar{\boldsymbol{C}}_{l,k} \leftarrow \beta \bar{\boldsymbol{C}}_{l,k} + (1-\beta) \boldsymbol{C}_{l,k}$.
  end for
  Set construction frequency: $F_t = F_{\text{warmup}}$ if $t \leq T_{\text{warmup}}$, otherwise $F_t = F_{\text{normal}}$.
  if $t \bmod F_t = 0$ **then**
    for each layer $l \in L_{\text{lay}}$ **do**
      Compute aggregated covariance: $\boldsymbol{C}_{\text{sum}} = \sum_k \bar{\boldsymbol{C}}_{l,k}$.
      Perform EVD: $(\boldsymbol{U}, \boldsymbol{\Lambda}) = \text{EVD}(\boldsymbol{C}_{\text{sum}})$.
      Select principal basis $\boldsymbol{P}_l$ according to energy threshold τ.
    end for
  end if
  Compute classification loss $\mathcal{L}_{ce}$ and gradients $\boldsymbol{g}_l = \nabla_{\boldsymbol{W}_l} \mathcal{L}_{ce}$.
  for each layer $l \in L_{\text{lay}}$ with $\boldsymbol{P}_l$ **do**
    Rectify gradients:

$$\tilde{\boldsymbol{g}}_l = \boldsymbol{g}_l (\boldsymbol{P}_l \boldsymbol{P}_l^T).$$

  end for
  Update network parameters using $\tilde{\boldsymbol{g}}_l$ and learning rate η.
end for
return θ, ϕ.