ANTICIPATING DOMAIN SHIFTS VIA GEOMETRY-GUIDED FUNCTIONAL PROBING

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain generalization (DG) aims to learn a model from multiple source domains that remains reliable on unseen target domains. Recent studies have sought flat minima in the loss landscape to enhance robustness to diverse input variations, thereby mitigating domain shift in classification tasks. However, these methods typically rely on generic parameter-space perturbations to explore flatness, which not only lacks explicit grounding in class-relevant predictive variations but also suffers from additional perturbation-searching overhead. This motivates us to leverage the learned geometry of the classifier's decision boundary to randomly construct diverse predictive variations in anticipation of unseen domain shifts. To this end, we propose Anticipating Shifts via Functional Probing (ASFP), a lightweight framework that promotes local flatness along the explored functional directions by jointly optimizing a family of perturbed predictive functions. Specifically, ASFP constructs perturbation directions that evolve with the classifier by randomly combining pairwise differences between its class-weight vectors, enabling structured functional exploration without costly perturbation search. *Theoretically*, with the SmoothMax objective, we characterize the induced curvature and sensitivity regularization and establish a local bound on loss increases under unseen-domain perturbations covered by this geometry. *Empirically*, we demonstrate through extensive experiments on five benchmarks that ASFP consistently outperforms strong baselines while maintaining low training overhead.

1 INTRODUCTION

Conventional machine learning typically assumes that training and test data are drawn from the same distribution Vapnik et al. (1998), following the independent and identically distributed (i.i.d.) assumption. However, this assumption is often violated in real-world applications, where distribution shifts can substantially degrade predictive performance Shi et al. (2022). To address this challenge, domain generalization (DG) Eastwood et al. (2022) aims to induce a model from multiple source domains that generalizes reliably to unseen target domains.

A wide range of methods have been proposed to improve DG, including invariant representation learning Zhou et al. (2025), data augmentation Zhou et al. (2024), and meta-learning Li et al. (2018a). Beyond these paradigms, recent studies have demonstrated the efficacy of pursuing flat minima, where the loss landscape remains relatively insensitive to local parameter perturbations, to mitigate domain shift in the classification tasks. SWAD Cha et al. (2021) achieves flatter solutions through weight averaging, while MeCAM Chen et al. (2024) introduces curvature-aware optimization to guide convergence toward flatter solutions. Building upon sharpness-aware minimization Foret et al. (2021), SAGM Wang et al. (2023a) jointly minimizes empirical risk, perturbed loss, and their gap through implicit gradient matching. SFT Li et al. (2025) pursues consistent flat minima across source domains by refining their loss landscapes. Nevertheless, single-direction perturbation strategies may overlook loss-landscape anisotropy, yielding solutions that remain sharp in other directions. To address this limitation, DA-SAM He et al. (2026) incorporates domain-adaptive perturbations and multi-directional exploration to address domain heterogeneity and anisotropic sharpness. Despite their methodological differences, these perturbation-based methods typically explore flatness through gradient-based perturbation search, incurring additional training overhead. Moreover, as illustrated in Fig. 1, their perturbation neighborhoods are primarily defined using generic parameter-space metrics, without explicitly exploiting the classifier's learned structure of the deci-

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.

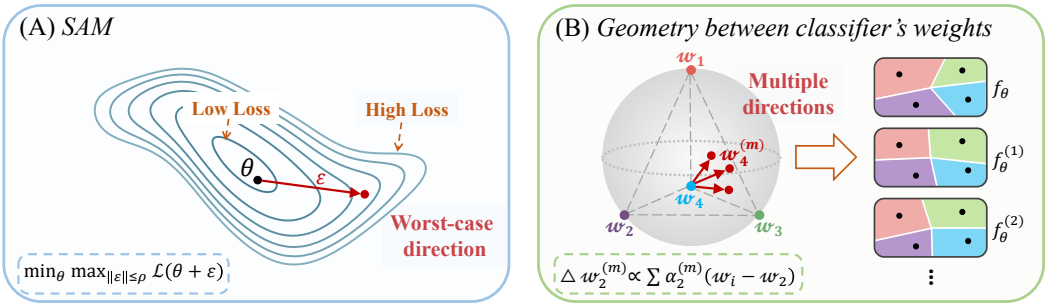


Figure 1: Comparison of perturbation strategies. (A) SAM searches for a worst-case perturbation within a bounded parameter neighborhood. (B) Multiple geometry-guided perturbation directions (red arrows →) are constructed from pairwise classifier weight differences (gray dashed lines ---), yielding diverse predictive functions without gradient-based perturbation search. Notably, the sphere in (B) is purely illustrative and does not imply the need for spherical constraints.

sion boundary. Consequently, the resulting perturbation exploration may be less targeted toward variations that affect classification decisions.

These aforementioned limitations highlight the requirement for a computationally efficient and decision-aware strategy for exploring diverse perturbation directions. Intuitively, in classification, domain shifts can potentially alter the class-relevant predictive scores, thereby compromising prediction reliability. Crucially, the sensitivity of these scores to representation variations is governed by the classifier's learned inter-class weight geometry Papyan et al. (2020). This insight inspires us to exploit such geometry to construct diverse functional perturbations, rehearsing a broad range of potential predictive variations in anticipation of unseen domain shifts, without costly gradient-based search. To this end, we propose Anticipating Shifts via Functional Probing (ASFP), a lightweight framework that promotes local flatness along the explored functional directions by jointly optimizing a family of perturbed predictive functions. Specifically, ASFP generates classifier-adaptive perturbation directions by combining pairwise differences between classifier weight vectors using random coefficients, allowing these directions to evolve with the classifier during training. The resulting perturbed functions share the same backbone and are jointly optimized using a temperature-controlled SmoothMax objective Li et al. (2023), which emphasizes high-loss functional variations. Moreover, we theoretically demonstrate that our ASFP induces curvature and sensitivity regularization and derive a local upper bound on loss increases under unseen-domain perturbations covered by the explored functional geometry. In summary, our main contributions can be highlighted as follows:

- We propose a lightweight framework, namely Anticipating Shifts via Functional Probing (ASFP), to promote local flatness along the explored diverse functional directions.
- We employ the SmoothMax objective to jointly optimize a family of predictive functions perturbed along directions that evolve with the classifier through random combinations of pairwise weight differences.
- We theoretically demonstrate the induced curvature and sensitivity regularization locally bounds loss increases under unseen-domain perturbations covered by this geometry.
- We empirically conduct extensive experiments on five benchmarks, demonstrating that ASFP consistently outperforms strong baselines while maintaining low training overhead.

2 Methodology

In this section, a more detailed and rigorous discussion will be presented, including both theoretical interpretation and practical methodology.

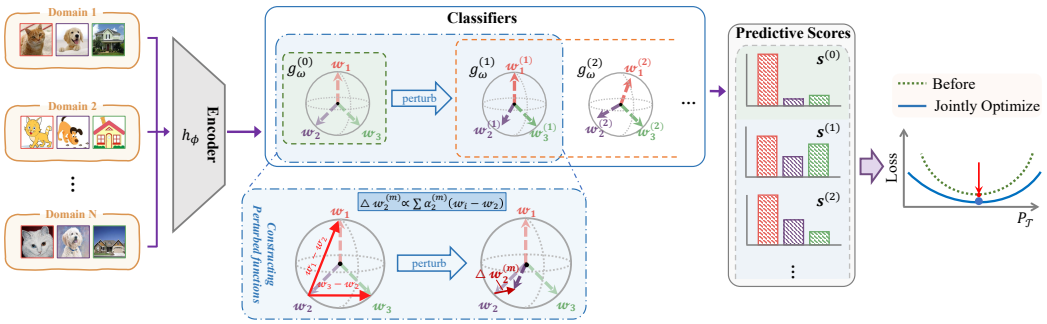


Figure 2: The framework of the proposed ASFP. Compared with the standard network $g_\omega^{(0)} \circ h_\phi$, ASFP additionally constructs a family of perturbed predictive functions guided by the classifier's weight geometry, as illustrated by the blue dashed box, and jointly optimizes them through the corresponding predictive scores $\boldsymbol{s}^{(m)}$. In this way, ASFP promotes local functional flatness along the explored perturbation directions, without requiring gradient-based perturbation search.

2.1 Problem Setup and Overview

DG aims to learn from N source domains and perform well on unseen target domains. The d-th source domain provides $\mathcal{D}_{\mathcal{S}}^d = \{(\boldsymbol{x}_i^d, y_i^d)\}_{i=1}^{N_d}$, sampled from a joint distribution $P_{\mathcal{S}}^d(\boldsymbol{X}, Y)$ Zhou et al. (2023). The target distribution $P_{\mathcal{T}}(\boldsymbol{X}, Y)$ is unavailable during training, and all domains share the label space $\mathcal{Y} = \{1, \ldots, C\}$. We consider a standard classification network f_θ, such as ResNet-50 He et al. (2016), which consists of a feature extractor $h_\phi : \mathcal{X} \to \mathbb{R}^r$ and an affine classification head $g_\omega : \mathbb{R}^r \to \mathbb{R}^C$. In this way, we can achieve

$$f_\theta(\boldsymbol{x}) = g_\omega(h_\phi(\boldsymbol{x})), \quad \boldsymbol{z} = h_\phi(\boldsymbol{x}), \quad \text{and} \quad g_\omega(\boldsymbol{z}) = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{b},$$

where $\boldsymbol{W} = [\boldsymbol{w_1}, \ldots, \boldsymbol{w_K}]^\top \in \mathbb{R}^{K \times r}$, $\omega = (\boldsymbol{W}, \boldsymbol{b})$, and $\theta = (\phi, \omega)$.

Since unseen domains may induce diverse changes in predictive behavior Cao & Chen (2025), relying on a single source-induced predictor provides only a limited view of such variations. To broaden the predictive variations explored during training, our proposed ASFP augments the original classifier g_ω with multiple perturbed heads, forming a family of neighboring predictive functions that can facilitate generalization to unseen domains.

2.2 ASFP

The overall framework of our proposed ASFP is illustrated in Fig. 2 and consists of two stages: 1) constructing perturbed functions from pairwise class-weight contrasts, and 2) jointly optimizing the generated function family with low training overhead.

Constructing perturbed functions from pairwise class-weight contrasts. For a multi-class classifier, generally, identifying class c inherently relies on comparing its predictive score with those of the remaining classes. Given the original classifier output $\boldsymbol{s}^{(0)} = g_\omega(\boldsymbol{z})$, the relative score between classes j and c can be written as

$$s_j^{(0)} - s_c^{(0)} = (\boldsymbol{w}_j - \boldsymbol{w}_c)^\top \boldsymbol{z} + (b_j - b_c),$$

where $s_j^{(0)}$ and $s_c^{(0)}$ denote the corresponding entries of $\boldsymbol{s}^{(0)}$. Thus, the pairwise difference $\boldsymbol{w}_j - \boldsymbol{w}_c$ directly governs how the relative score between classes j and c responds to variations in the representation. Accordingly, for each class c, we collect its pairwise class-weight contrasts as

$$\mathcal{G}_c = [\boldsymbol{w}_j - \boldsymbol{w}_c]_{j \neq c} \in \mathbb{R}^{r \times (C-1)}. \tag{1}$$

Subsequently, we utilize $\mathcal{G}c$ to transform simple random coefficients into class-conditioned perturbation directions. For each perturbed head $m \in \{1, \ldots, M\}$ and class c, we sample a random

coefficient vector $\boldsymbol{\alpha}_c^{(m)} = (\alpha_{c,j}^{(m)})_{j\neq c}$ at initialization, normalize it to satisfy $\|\boldsymbol{\alpha}_c^{(m)}\|_2 = 1$, and keep it fixed throughout training. In this way, the perturbation is constructed as

$$\boldsymbol{n}_c^{(m)} = \mathcal{G}_c \boldsymbol{\alpha}_c^{(m)}, \tag{2}$$

$$\Delta \boldsymbol{w}_c^{(m)} = \gamma \|\boldsymbol{w}_c\|_2 \frac{\boldsymbol{n}_c^{(m)}}{\|\boldsymbol{n}_c^{(m)}\|_2}, \tag{3}$$

where $\gamma > 0$ controls the relative perturbation magnitude, and the normalization isolates the perturbation direction from its scale, while the factor $\|\boldsymbol{w}_c\|_2$ ensures a consistent relative perturbation strength across classes. Notably, although $\boldsymbol{\alpha}_c^{(m)}$ remains fixed, both $\mathcal{G}_c$ and $\|\boldsymbol{w}_c\|_2$ evolve during training, allowing the realized perturbation to adapt continuously to the current classifier geometry.

Stacking the class-wise perturbations yields $\Delta \boldsymbol{W}^{(m)}$, giving the m-th perturbed head as

$$g_\omega^{(m)}(\boldsymbol{z}) = (\boldsymbol{W} + \Delta \boldsymbol{W}^{(m)})\boldsymbol{z} + \boldsymbol{b} = \boldsymbol{s}^{(0)} + \boldsymbol{e}^{(m)}(\boldsymbol{z}), \tag{4}$$

where $\boldsymbol{e}^{(m)}(\boldsymbol{z}) = \Delta \boldsymbol{W}^{(m)} \boldsymbol{z}$, and the bias $\boldsymbol{b}$ is shared by all heads.

The following lemma explicitly characterizes the structure of the induced functional variations. To this end, we define the class-contrast subspace as follows:

$$\mathcal{S}(\boldsymbol{W}) = \operatorname{span}\{\boldsymbol{w}_j - \boldsymbol{w}_c : j, c \in \mathcal{Y}\}, \tag{5}$$

and let $\boldsymbol{\Pi}_{\mathcal{S}}$ denote the corresponding orthogonal projector.

Lemma 1 (Geometry of the Induced Predictive Variations)**.** *For every perturbed head constructed by Eq. (2), there exists $\boldsymbol{A}^{(m)} \in \mathbb{R}^{C\times C}$ such that*

$$\boldsymbol{A}^{(m)}\mathbf{1} = \mathbf{0}, \quad \Delta \boldsymbol{W}^{(m)} = \boldsymbol{A}^{(m)} \boldsymbol{W}, \quad \boldsymbol{e}^{(m)}(\boldsymbol{z}) = \boldsymbol{A}^{(m)}(\boldsymbol{s}^{(0)} - \boldsymbol{b}).$$

Furthermore,

$$\boldsymbol{e}^{(m)}(\boldsymbol{z}) = \boldsymbol{e}^{(m)}(\boldsymbol{\Pi}_{\mathcal{S}}\boldsymbol{z}), \quad \|\boldsymbol{e}^{(m)}(\boldsymbol{z})\|_2 \leq \gamma \|\boldsymbol{W}\|_F \|\boldsymbol{\Pi}_{\mathcal{S}}\boldsymbol{z}\|_2.$$

In particular, for any $\delta\boldsymbol{z} \in \mathcal{S}(\boldsymbol{W})^\perp$,

$$\operatorname{Softmax}\Big(g_\omega^{(0)}(\boldsymbol{z} + \delta\boldsymbol{z})\Big) = \operatorname{Softmax}\Big(g_\omega^{(0)}(\boldsymbol{z})\Big), \quad \boldsymbol{e}^{(m)}(\boldsymbol{z} + \delta\boldsymbol{z}) = \boldsymbol{e}^{(m)}(\boldsymbol{z}).$$

Accordingly, the induced perturbations can be viewed as linear recombinations of the feature-dependent class scores and depend only on the representation components involved in pairwise discrimination. Variations along $\mathcal{S}(\boldsymbol{W})^\perp$ shift all class logits equally and therefore leave the softmax prediction unchanged. Hence, the proposed perturbations naturally focus on classifier-relevant variations that can alter relative class score, rather than arbitrary changes in the representation space. This provides an explicit structural characterization of the explored functions without assuming domain-invariant directions. The corresponding proofs have been provided in Appendix A.1.

Jointly optimizing the function family. Uniformly averaging the losses of all heads $\{g_\omega^{(m)}\}_{m=0}^M$ may be suboptimal, since different perturbed functions can exhibit varying degrees of prediction risk. We therefore adopt a temperature-controlled SmoothMax objective Li et al. (2023) as

$$\mathcal{L} = \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_i = \frac{1}{B}\sum_{i=1}^{B} \tau \log\left[\frac{1}{M+1}\sum_{m=0}^{M} \exp\left(\frac{\ell(\boldsymbol{s}_i^{(m)}, y_i)}{\tau}\right)\right], \tag{6}$$

where B denotes the batch size, $\ell(\cdot,\cdot)$ denotes the cross-entropy loss Vapnik et al. (1998), and $\tau > 0$ denotes the temperature controlling the degree of emphasis on high-loss functions. In this way, its sensitivity to to each constituent loss can be formulated as

$$p_i^{(m)} = \frac{\partial \mathcal{L}_i}{\partial \ell(\boldsymbol{s}_i^{(m)}, y_i)} = \frac{\exp(\ell(\boldsymbol{s}_i^{(m)}, y_i)/\tau)}{\sum_{k=0}^{M} \exp(\ell(\boldsymbol{s}_i^{(k)}, y_i)/\tau)}. \tag{7}$$

Therefore, compared with uniform averaging, Eq. (6) implicitly assigns greater emphasis to the high-loss functional variations specific to each example, allowing the optimization to focus more on locally challenging perturbations. As discussed in Appendix A.2, this behavior admits an equivalent

Algorithm 1 ASFP

Input: Training set, batch size B, number of source domains N, number of class C, number of perturbed classifiers M, maximum iteration T, the temperature τ, and the perturbation magnitude γ

Parameters: An encoder h_ϕ, and the classifier $g_\omega^{(0)}$

Output: Encoder h_ϕ, and the classifiers $\{g_\omega^{(m)}\}_{m=0}^{M}$

Randomly generate the normalized perturbed coefficients $\{\boldsymbol{\alpha}_c^{(m)}\}_{m=1}^{M}$
for $t \leftarrow 1$ to T **do**
  Randomly sample a batch $S = \{(\boldsymbol{x}_b, y_b)\}_{b=1}^{B}$
  # Extract the latent representation
  $\boldsymbol{z} \leftarrow h(\boldsymbol{x})$
  # Reconstruct the perturbed classifiers
  $\mathcal{G}_c \leftarrow$ Eq. (1) with $g_\omega^{(0)}$
  $\Delta\boldsymbol{\omega}_c^{(m)} \leftarrow$ Eq. (3) with $\mathcal{G}_c$ and $\boldsymbol{\alpha}_c^{(m)}$
  # Achieve m-th predictive score
  $\boldsymbol{s}^{(m)} \leftarrow$ Eq. (4) with $\boldsymbol{z}$, $g_\omega^{(0)}$ and $\Delta\boldsymbol{W}^{(m)}$
  # Overall Objective Function
  $\mathcal{L} \leftarrow$ Eq. (6) with $\{\boldsymbol{s}^{(m)}\}_{m=0}^{M}$
  Update network parameters
end for

variational interpretation as an importance-weighted loss regularized by a KL divergence from the uniform prior, eliminating the need for additional hand-crafted head-wise weights.

It is worth noting that, during training, all heads reuse the same backbone representations, while their perturbations are computed algebraically from the current classification weights. As a result, no additional gradient-ascent search is required for perturbation construction, resulting in low training overhead. Detailed algorithm is provided in Algorithm 1.

2.3 Theoretical Interpretation

We next reveal the conditions under which the perturbed function family can facilitate generalization to unseen domains. In particular, this benefit depends on whether the induced logit perturbations sufficiently cover the predictive variations arising from domain shift.

For analysis, consider a source example $(\boldsymbol{x}_i, y_i)$ and its label-preserving counterpart $T_\mathcal{T}(\boldsymbol{x}_i)$ under an unseen target domain. For brevity, let $\boldsymbol{s} = f_\theta(\boldsymbol{x}_i)$ and $y = y_i$. The corresponding target-induced predictive deviation can be then expressed as

$$\boldsymbol{e}_\mathcal{T} = f_\theta(T_\mathcal{T}(\boldsymbol{x}_i)) - f_\theta(\boldsymbol{x}_i). \tag{8}$$

Using the explored perturbations $\{\boldsymbol{e}^{(m)}\}_{m=0}^{M}$, with $\boldsymbol{e}^{(0)} = \mathbf{0}$, we can decompose this deviation as

$$\boldsymbol{e}_\mathcal{T} = \sum_{m=0}^{M} q_m \boldsymbol{e}^{(m)} + \boldsymbol{v}, \quad q_m \geq 0, \quad \sum_{m=0}^{M} q_m = 1. \tag{9}$$

Here, q_m characterizes the contribution of each explored variation, while $\boldsymbol{v}$ represents the residual component not captured by their convex combination. Accordingly, a smaller $\|\boldsymbol{v}\|_2$ indicates better coverage of the target-induced deviation by the explored functional variations. Let $\mathcal{Q}_M$ denote the probability simplex over the $M+1$ heads and let $\pi_m = 1/(M+1)$ denote the uniform head weights.

Proposition 1 (Loss Control under Incomplete Coverage)**.** *For the cross-entropy loss and the decomposition in Eq. (9), the following holds for any $\boldsymbol{q} \in \mathcal{Q}_M$ and $\tau > 0$:*

$$\ell(\boldsymbol{s} + \boldsymbol{e}_\mathcal{T}, y) \leq \mathcal{L}_i + \underbrace{\tau\,\mathrm{KL}(\boldsymbol{q}\|\boldsymbol{\pi})}_{\text{weighting discrepancy}} + \underbrace{\sqrt{2}\|\boldsymbol{v}\|_2}_{\text{uncovered deviation}}. \tag{10}$$

With the convex hull $\mathcal{C}_i = \left\{\sum_{m=0}^{M} q_m \boldsymbol{e}^{(m)} : \boldsymbol{q} \in \mathcal{Q}_M\right\}$, minimizing the residual over $\boldsymbol{q}$ then yields

$$\ell(\boldsymbol{s} + \boldsymbol{e}_\mathcal{T}, y) \leq \mathcal{L}_i + \tau \log(M+1) + \sqrt{2}\,\mathrm{dist}(\boldsymbol{e}_\mathcal{T}, \mathcal{C}_i). \tag{11}$$

The looser bound in Eq. (11) demonstrates that, for fixed M and τ, the target loss is bounded by the training objective, a constant term, and the distance from the target-induced deviation to the convex hull of the explored variations. This distance vanishes when the target deviation lies within that hull. Therefore, the SmoothMax objective in Eq. (6), together with the constructed perturbations $\Delta \boldsymbol{W}^{(m)}$, provides explicit loss control for domain changes whose predictive effects are well covered by the explored functional variations.

Proposition 2 (Local Implicit Regularization)**.** *Fix $\boldsymbol{s}$ and y, and suppose the loss has a locally Lipschitz Hessian in a neighborhood of $\boldsymbol{s}$. Let $\boldsymbol{e}^{(0)} = \mathbf{0}$ and $\max_m \|\boldsymbol{e}^{(m)}\|_2 \leq \varepsilon$. All expectations below are uniform over $m \in \{0, \ldots, M\}$. We define*

$$\boldsymbol{\mu}_e = \mathbb{E}_m[\boldsymbol{e}^{(m)}], \quad \boldsymbol{Q}_e = \mathbb{E}_m[\boldsymbol{e}^{(m)}\boldsymbol{e}^{(m)\top}], \quad \boldsymbol{\Sigma}_e = \boldsymbol{Q}_e - \boldsymbol{\mu}_e\boldsymbol{\mu}_e^\top.$$

Let $\ell_0 = \ell(\boldsymbol{s}, y)$, $\boldsymbol{g} = \nabla_{\boldsymbol{s}}\ell(\boldsymbol{s}, y)$, and $\boldsymbol{H} = \nabla_{\boldsymbol{s}}^2\ell(\boldsymbol{s}, y)$. For fixed $\tau > 0$, as $\varepsilon \to 0$,

$$\mathcal{L}_i = \ell_0 + \boldsymbol{g}^\top\boldsymbol{\mu}_e + \frac{1}{2}\operatorname{Tr}(\boldsymbol{H}\boldsymbol{Q}_e) + \frac{1}{2\tau}\boldsymbol{g}^\top\boldsymbol{\Sigma}_e\boldsymbol{g} + \mathcal{O}(\varepsilon^3).$$

If the constructed family is centered, i.e., $\boldsymbol{\mu}_e = \mathbf{0}$, this becomes

$$\mathcal{L}_i = \ell_0 + \underbrace{\frac{1}{2}\operatorname{Tr}(\boldsymbol{H}\boldsymbol{\Sigma}_e)}_{\text{curvature term}} + \underbrace{\frac{1}{2\tau}\boldsymbol{g}^\top\boldsymbol{\Sigma}_e\boldsymbol{g}}_{\text{sensitivity term}} + \mathcal{O}(\varepsilon^3).$$

This proposition reveals that, locally, our objective can be interpreted as the standard loss augmented with curvature-and sensitivity-dependent regularization along the perturbation geometry characterized by $\boldsymbol{\Sigma}_e$. Minimizing these additional terms suppresses both the local curvature and the loss sensitivity in the explored functional directions, thereby encouraging a flatter loss landscape around the reference predictor. As characterized in Lemma 1, these directions follow the classifier-induced geometry associated with pairwise discrimination.

Their corresponding proofs are provided in Appendix A.3 and A.4, respectively.

3 Experiments

In this section, extensive experiments have been constructed to comprehensively evaluate the effectiveness of ASFP on both toy data and five standard Benchmarks.

3.1 Results on Toy Datasets

To provide an intuitive examination of the perturbation geometry induced by ASFP, we construct a controlled toy experiment with progressively rotated domains. Specifically, we generate a three-class classification problem and obtain different domains by rotating the input distribution with different angles. The domains at -15°, 0°, and 15° are utilized as source domains for training, while those at 5°, 30°, and 45° serve as unseen target domains with different degrees of deviation from the source domains. A more detailed description of the toy dataset is provided in Appendix C. This controlled setting allows us to explicitly compare the target-induced predictive variation with the perturbation geometry constructed from the source-trained classifier, and to examine whether the generated perturbations provide meaningful coverage of unseen target variations.

As illustrated in Fig. 3, the three target domains exhibit clearly different levels of distribution shift. In SubFig. (A), the 5° target domain closely follows the source domains throughout training, whereas increasingly larger gaps are observed for the 30° and 45° target domains. A similar ordering is reflected in SubFig. (E) and (F), where the magnitude of the target-induced variation increases substantially with the rotation angle. Since the 5° domain lies within the angular range spanned by the source domains, it behaves as an interpolative target and is well covered by the source-domain variations, resulting in nearly vanishing geometric discrepancies. In contrast, the increasingly distant 30° and 45° domains induce progressively stronger functional deviations.

Interestingly, the curvature and sensitivity terms in SubFig. (B) and (C) exhibit a characteristic rise-and-fall pattern. At the early training stage, the classifier geometry and the induced perturbation

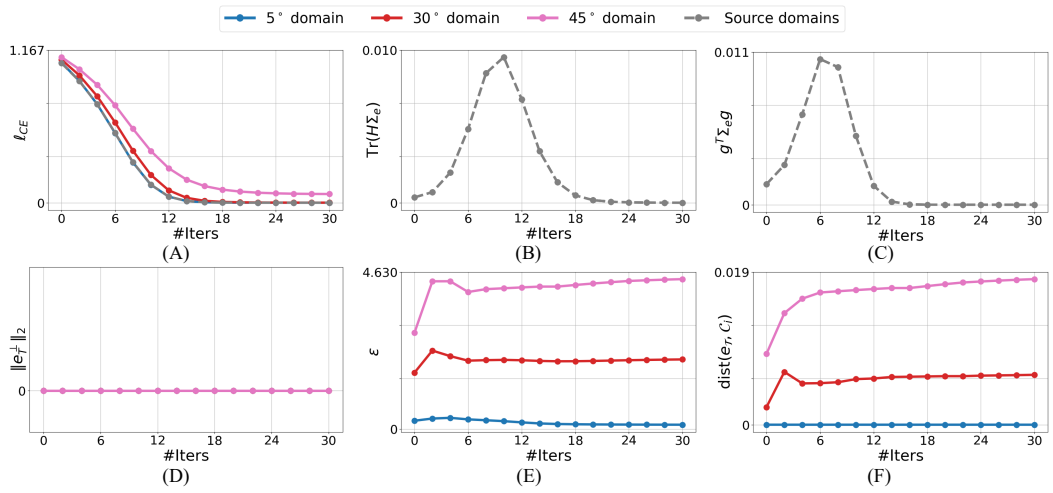


Figure 3: Toy illustration of the functional perturbation geometry under rotated domain shifts, which consists of (A) the classification loss, (B) the curvature regularization term, (C) the sensitivity regularization term, (D) the norm of target orthogonal residual, (E) the magnitude of target variation, and (F) the distance to the finite perturbation hull throughout training.

structure are not yet well established, leading to relatively weak directional regularization. As training proceeds, the classifier becomes increasingly discriminative while remaining sensitive to the explored perturbations, causing both terms to increase. They subsequently decrease as optimization progresses and approach zero near convergence, indicating that the loss becomes progressively less curved and less sensitive along the explored functional directions. This behavior is consistent with Prop. 2, where the two terms act as local curvature-and sensitivity-dependent regularizers and jointly encourage functional flatness along the induced perturbation geometry.

SubFig. (D) further demonstrates that the orthogonal component $\|e_{\mathcal{T}}^{\perp}\|_2$ remains nearly zero across the target domains. This indicates that the target-induced variations contain little component outside the explored perturbation subspace, supporting the classifier-induced geometry characterized in Lemma 1. Finally, SubFig. (E) and (F) jointly characterize the magnitude and coverage of unseen target variations, which can gradually stabilize during training. Hence, increasingly severe target shifts lead to larger functional deviations and somewhat larger approximation errors, but the dominant target variations remain well captured by the perturbations from the source-trained classifier.

3.2 Results on Five Benchmarks

For the architecture, our ASFP aligns with the settings outlined in DomainBed Gulrajani & Lopez-Paz (2021), wherein ResNet-50 has been selected as the backbone and the training-domain validation method has been employed for model selection. Extensive experiments are constructed on five Benchmarks, i.e., VLCS, PACS, OfficeHome, TerraIncognita, and DomainNet. The average results on these dataset are reported in Tab. 1.

Nineteen recent strong comparison methods and two other representative methods are deployed to compare with our ASFP. These methods can be divided into seven categories: 1) *distribution robust method* (GroupDRO Sagawa et al. (2020)), 2) *causal methods learning invariance* (IRM Arjovsky et al. (2019), VREx Krueger et al. (2021), EQRM Eastwood et al. (2022), TCRI Salaudeen & Koyejo (2024)), 3) *gradient matching* (Fish Shi et al. (2022), Fishr Rame et al. (2022), GGA Ballas & Diou (2025)), 4) *representation distribution matching* (MMD Li et al. (2018b), CORAL Sun & Saenko (2016), DANNGanin et al. (2016), RDM Nguyen et al. (2024), I^{3}C Zhou et al. (2025), 5) *meta-learning* (MLDG Li et al. (2018a)) 6) *sharpness-aware methods* (SAGM Wang et al. (2023a), CRSAM Wu et al. (2024), FSAM Li et al. (2024), DA-SAM He et al. (2026)) 7) and *other variants* (Mixup Yan et al. (2020), Mixstyle Zhou et al. (2024)). For more details on datasets, comparison methods, and experimental settings, please refer to our Appendix D.

Table 1: Overall accuracy results on five Benchmarks. The **bold** and underline items are the best and the second-best results, respectively. † denotes results reproduced by us, while ‡ denotes results reported in He et al. (2026).

Methods	VLCS	PACS	OfficeHome	TerraIncognita	DomainNet	Avg
ERM (1998)	77.5 ± 0.4	85.5 ± 0.2	66.5 ± 0.3	46.1 ± 1.8	40.9 ± 0.3	63.3
Mixup (2020)	77.4 ± 0.6	84.6 ± 0.6	68.1 ± 0.3	47.9 ± 0.8	39.2 ± 0.4	63.4
DANN (2016)	78.6 ± 0.7	83.6 ± 1.1	65.9 ± 0.6	46.7 ± 1.6	38.3 ± 0.3	62.6
MLDG (2018a)	77.2 ± 0.4	84.9 ± 1.0	66.8 ± 0.6	47.7 ± 0.9	41.2 ± 0.3	63.6
GroupDRO (2020)	76.7 ± 0.6	84.4 ± 0.8	66.0 ± 0.7	43.2 ± 1.1	33.3 ± 0.2	60.7
IRM (2019)	78.5 ± 0.5	83.5 ± 0.8	64.3 ± 2.2	47.6 ± 0.8	33.9 ± 1.9	61.6
VREx (2021)	78.3 ± 0.2	84.9 ± 0.6	66.4 ± 0.6	46.4 ± 0.6	33.6 ± 3.0	61.9
EQRM (2022)	77.8 ± 0.6	86.5 ± 0.2	67.5 ± 0.3	47.8 ± 0.6	41.0 ± 0.6	64.1
Fish (2022)	77.8 ± 0.3	85.5 ± 0.3	68.6 ± 0.4	45.1 ± 1.3	42.7 ± 0.2	63.9
Fishr (2022)	77.8 ± 0.1	85.5 ± 0.4	67.8 ± 0.1	47.4 ± 1.6	41.7 ± 0.3	64.0
CORAL (2016)	78.8 ± 0.6	86.2 ± 0.3	68.7 ± 0.3	47.6 ± 1.0	41.5 ± 0.2	64.6
MMD (2018b)	77.5 ± 0.9	84.6 ± 0.5	66.3 ± 0.1	42.2 ± 1.6	23.4 ± 9.5	58.8
Mixstyle (2024)	77.6 ± 0.9	85.2 ± 1.1	60.4 ± 0.5	44.0 ± 1.3	34.0 ± 0.3	60.2
RDM (2024)	78.4 ± 0.4	87.2 ± 0.7	67.3 ± 0.4	47.5 ± 1.0	43.4 ± 0.4	64.8
TCRI† (2024)	$\underline{79.9 \pm 0.6}$	87.0 ± 0.5	67.9 ± 0.4	$\underline{51.2 \pm 0.8}$	42.2 ± 0.1	65.6
I^3C (2025)	79.6 ± 0.9	87.1 ± 0.9	$\mathbf{70.2 \pm 0.5}$	49.6 ± 1.9	42.6 ± 0.6	65.8
GGA (2025)	$\underline{79.9 \pm 0.4}$	87.3 ± 0.4	68.5 ± 0.2	50.6 ± 0.1	$\mathbf{45.2 \pm 0.2}$	$\underline{66.3}$
SAGM‡ (2023a)	79.1 ± 1.0	86.9 ± 0.3	69.4 ± 0.1	48.6 ± 1.5	44.7 ± 0.2	65.7
CRSAM‡ (2024)	79.1 ± 0.6	85.4 ± 0.7	68.9 ± 1.0	45.3 ± 0.4	44.3 ± 0.1	64.6
FSAM‡ (2024)	79.4 ± 0.1	86.5 ± 0.3	$\mathbf{70.2 \pm 0.1}$	46.1 ± 0.6	$\underline{44.9 \pm 0.1}$	65.4
DA-SAM (2026)	79.6 ± 0.8	$\underline{87.5 \pm 0.4}$	68.4 ± 0.3	50.0 ± 0.2	44.5 ± 0.2	66.0
ASFP (*ours*)	$\mathbf{80.4 \pm 0.0}$	$\mathbf{88.3 \pm 0.4}$	$\underline{69.8 \pm 0.1}$	$\mathbf{53.6 \pm 0.7}$	44.3 ± 0.1	**67.3**

Table 2: Ablation study on five Benchmarks. **Ave. Loss** replaces SmoothMax with the average loss over all heads during training, while **Only** $g_\omega^{(0)}$ and **Top-**K denote inference using only the reference head and the K most confident heads, respectively.

	VLCS	PACS	OfficeHome	TerraIncognita	DomainNet
baseline	77.5 ± 0.4	85.5 ± 0.2	66.5 ± 0.3	46.1 ± 1.8	40.9 ± 0.3
Ave. Loss	78.0 ± 0.5	86.3 ± 0.5	66.5 ± 0.2	47.3 ± 2.1	41.1 ± 0.5
Only $g_\omega^{(0)}$	80.2 ± 0.2	88.3 ± 0.4	69.7 ± 0.2	53.7 ± 0.6	44.3 ± 0.3
Top-K	80.5 ± 0.9	88.6 ± 0.8	70.2 ± 0.7	53.6 ± 1.2	44.5 ± 0.6
ours	80.4 ± 0.0	88.3 ± 0.4	69.8 ± 0.1	53.6 ± 0.7	44.3 ± 0.1

From Tab. 1, we can observe the following findings: *1)* Overall, our ASFP achieves the best average performance across the five benchmarks. In particular, it outperforms the second-best method by 1.0%, which is substantially larger than the 0.3% gap between the second- and third-best methods. The improvement is especially pronounced on TerraIncognita, outperforming the strongest competitor by 2.4%. On PACS, ASFP also improves over the previous best result by 0.8%. Although the gains on OfficeHome and DomainNet are less pronounced, our ASFP remains competitive. These results demonstrate that the overall improvement is not driven by a single benchmark, but is consistently maintained across diverse domain shifts. *2)* Compared with existing sharpness-aware methods, our ASFP consistently achieves stronger overall performance. Notably, compared with the strongest baseline DA-SAM, ASFP obtains clear gains on almost benchmarks. These results suggest that exploring local variations within the classifier-induced functional geometry provides an effective alternative to directly seeking sharpness-aware solutions through parameter-space perturbations. *3)* In addition, our ASFP exhibits consistently small standard deviations across all five benchmarks, demonstrating stable performance over repeated runs.

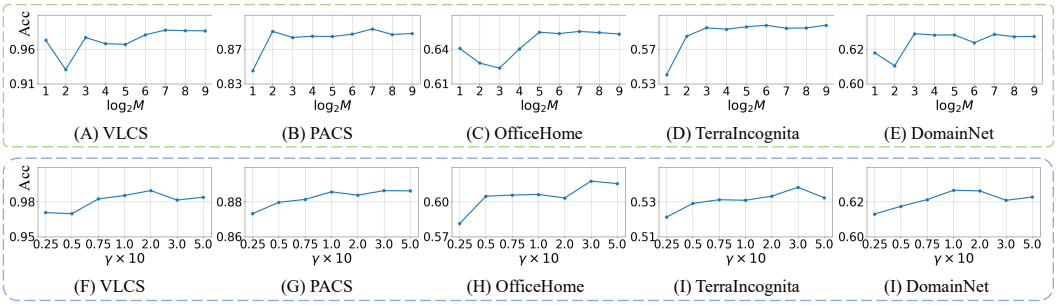


Figure 4: Hyperparameter sensitivity analysis of ASFP. The top panel evaluates the number of perturbed classifiers M, while the bottom panel examines the perturbation magnitude γ.

Table 3: The time cost(s) per batch compared to recent studies on representative Benchmarks.

	SAGM	TCRI	I^{3}C	GGA	DA-SAM	*Ours*
PACS	0.486	0.432	<u>0.217</u>	0.230	0.543	**0.175**
OfficeHome	0.500	0.445	<u>0.283</u>	0.296	0.554	**0.185**
DomainNet	0.714	0.623	<u>0.406</u>	0.461	0.766	**0.293**

From Tab. 2, we can observe the following findings: *1)* Replacing SmoothMax with the average loss over all heads leads to a clear performance drop, confirming the benefit of emphasizing high-loss functional variations during training. *2)* At inference, using only $g_\omega^{(0)}$ achieves nearly identical accuracy to ASFP, suggesting that the perturbed heads mainly benefit the model through training-time functional regularization. Top-K selection provides slightly higher accuracy on some benchmarks but with noticeably larger variance, while uniform averaging offers a more stable inference strategy.

From Fig. 4, ASFP exhibits generally stable performance over a broad range of M and γ. Increasing the number of perturbed classifiers M generally improves performance, while the gains gradually saturate as more functional directions are explored. This suggests that a moderate number of perturbed classifiers is sufficient to characterize the local perturbation geometry. Regarding the perturbation magnitude γ, overly small perturbations tend to provide insufficient functional exploration, whereas moderate values consistently achieve stronger performance across benchmarks. Further increasing γ brings limited gains and occasionally causes slight degradation, which is consistent with the local nature of our perturbation-based regularization.

From Tab. 3, ASFP achieves the lowest per-batch training cost across all three benchmarks, demonstrating the highest overall efficiency among the compared methods. Compared with sharpness-aware methods, e.g., SAGM and DA-SAM, ASFP reduces the computational cost substantially, benefiting from avoiding repeated gradient-based perturbation search. Moreover, even against recent strong competitors, e.g., I^{3}C and GGA, ASFP remains markedly more efficient, indicating a favorable balance between generalization performance and training cost.

4 Conclusion

We develop a lightweight framework, namely Anticipating Shifts via Functional Probing (ASFP), to promote local flatness along diverse explored functional directions. ASFP constructs perturbations guided by the classifier geometry and jointly optimizes them through a SmoothMax objective. Theoretically, our analysis characterizes the induced perturbation structure and its implicit local regularization toward flatness. Empirically, extensive experiments on five DG benchmarks demonstrate that ASFP consistently improves generalization while maintaining high training efficiency.

A current limitation is that our theoretical analysis mainly focuses on local functional variations induced by the classifier geometry. More complex or distant shifts may involve variations beyond this locally explored neighborhood, which remain to be further investigated in future work.

AI USE STATEMENT

In this work, we used generative AI tools for assisting with the presentation and writing of this paper, including refining the paper structure, drafting and editing text for clarity and concision, and improving the writing of the proofs. We have not used generative AI tools for generating experimental data, implement the evaluated methods, or execute experiment. We have reviewed all AI-assisted work. We manually ensure the citations are correct. LLM-generated code was verified and tested for correctness by the authors. The core research idea and scientific contributions were developed by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Our code is available in the supplementary materials.

REFERENCES

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Aristotelis Ballas and Christos Diou. Gradient-guided annealing for domain generalization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 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 (ECCV)*, pp. 456–473, 2018.

Meng Cao and Songcan Chen. A simple yet lightweight module for enhancing domain generalization through relative representation. *Pattern Recognition*, pp. 112423, 2025.

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021.

Ziyang Chen, Yiwen Ye, Feilong Tang, Yongsheng Pan, and Yong Xia. Meta curvature-aware minimization for domain generalization. *arXiv preprint arXiv:2412.11542*, 2024.

Cian Eastwood, Alexander Robey, Shashank Singh, Julius Von Kügelgen, Hamed Hassani, George J Pappas, and Bernhard Schölkopf. Probable domain generalization via quantile risk minimization. *Advances in Neural Information Processing Systems*, 35:17340–17358, 2022.

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*, pp. 1657–1664, 2013.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Antoine Gonon, Nicolas Brisebarre, Elisa Riccietti, and Rémi Gribonval. A path-norm toolkit for modern networks: consequences, promises and challenges. In *International Conference on Learning Representations*, volume 2024, pp. 41108–41152, 2024.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, pp. 11856–11884, 2021.

Junwen He, Yang He, Lebing Zheng, Zirui Yin, Hong-Yu Zhang, and Yulong Wang. Navigating the flatlands: Dual adaptive sharpness-aware minimization for domain generalization. In *Forty-third International Conference on Machine Learning*, 2026.

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*, pp. 770–778, 2016.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pp. 5815–5826. PMLR, 2021.

Aodi Li, Liansheng Zhuang, Xiao Long, Minghong Yao, and Shafei Wang. Seeking consistent flat minima for better domain generalization via refining loss landscapes. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15349–15359. IEEE, 2025.

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*, pp. 5542–5550, 2017.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, pp. 3490–3497, 2018a.

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.

Tao Li, Pan Zhou, Zhengbao He, Xinwen Cheng, and Xiaolin Huang. Friendly sharpness-aware minimization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5631–5640. IEEE, 2024.

Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. On tilted losses in machine learning: Theory and applications. *Journal of Machine Learning Research*, 24(142):1–79, 2023.

Toan Nguyen, Kien Do, Bao Duong, and Thin Nguyen. Domain generalisation via risk distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2790–2799, 2024.

Vardan Papyan, Xiao Y Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019.

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

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, pp. 2661–2679, 2020.

Olawale Salaudeen and Sanmi Koyejo. Causally inspired regularization enables domain general representations. In *International Conference on Artificial Intelligence and Statistics*, pp. 3124–3132. PMLR, 2024.

Yuge Shi, Jeffrey Seely, Philip Torr, Siddharth N, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. In *International Conference on Learning Representations*, pp. 15803–15830, 2022.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pp. 443–450. Springer, 2016.

Vladimir Naumovich Vapnik, Vlamimir Vapnik, et al. Statistical learning theory. 1998.

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*, pp. 5018–5027, 2017.

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

Zhe Wang, Jake Grigsby, and Yanjun Qi. PGrad: Learning principal gradients for domain generalization. In *The Eleventh International Conference on Learning Representations*, pp. 17807–17827, 2023b.

Tao Wu, Tie Luo, and Donald C Wunsch II. Cr-sam: Curvature regularized sharpness-aware minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 6144–6152, 2024.

Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve unsupervised domain adaptation with mixup training. *arXiv preprint arXiv:2001.00677*, 2020.

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

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.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Mixstyle neural networks for domain generalization and adaptation. *International Journal of Computer Vision*, 132(3):822–836, 2024.

A Theoretically Analysis

A.1 Proof of Lemma 1

Proof. For each class c and head m, define $\boldsymbol{t}_c^{(m)} \in \mathbb{R}^C$ by

$$t_{c,j}^{(m)} = \begin{cases} \alpha_{c,j}^{(m)}, & j \neq c, \\ -\sum_{k \neq c} \alpha_{c,k}^{(m)}, & j = c. \end{cases}$$

Then, we can have

$$\begin{cases} \mathbf{1}^\top \boldsymbol{t}_c^{(m)} & = 0, \\ \boldsymbol{W}^\top \boldsymbol{t}_c^{(m)} & = \sum_{j \neq c} \alpha_{c,j}^{(m)} (\boldsymbol{w}_j - \boldsymbol{w}_c) = \mathcal{G}_c \boldsymbol{\alpha}_c^{(m)} = \boldsymbol{v}_c^{(m)}. \end{cases}$$

Set $\kappa_c^{(m)} = \gamma \|\boldsymbol{w}_c\|_2 / \|\boldsymbol{v}_c^{(m)}\|_2$ when $\boldsymbol{v}_c^{(m)} \neq \mathbf{0}$, and set $\kappa_c^{(m)} = 0$ otherwise. Define row c of $\boldsymbol{A}^{(m)}$ as $\kappa_c^{(m)} \boldsymbol{t}_c^{(m)\top}$. It follows that $\boldsymbol{A}^{(m)} \mathbf{1} = \mathbf{0}$ and $\Delta \boldsymbol{W}^{(m)} = \boldsymbol{A}^{(m)} \boldsymbol{W}$. Multiplying by $\boldsymbol{z}$ gives

$$\boldsymbol{e}^{(m)}(\boldsymbol{z}) = \boldsymbol{A}^{(m)} \boldsymbol{W} \boldsymbol{z} = \boldsymbol{A}^{(m)} (\boldsymbol{s}^{(0)} - \boldsymbol{b}).$$

where the c-th component can be denoted as

$$e_c^{(m)}(\boldsymbol{z}) = \kappa_c^{(m)} \sum_{j \neq c} \alpha_{c,j}^{(m)} \big[(s_j^{(0)} - s_c^{(0)}) - (b_j - b_c) \big].$$

Thus, each logit perturbation is expressed as a linear combination of the feature-dependent components of pairwise class-score differences. Moreover, since $\boldsymbol{A}^{(m)} \mathbf{1} = \mathbf{0}$, the transformation captures class-relative contrasts rather than defining a probability transition matrix.

We next show that the constructed perturbations are confined to the class-contrast subspace and have bounded magnitude. By definition, $\boldsymbol{v}_c^{(m)}$ is a linear combination of pairwise class-weight differences and therefore belongs to $\mathcal{S}(\boldsymbol{W})$ defined in Eq. (5). Since $\Delta \boldsymbol{w}_c^{(m)}$ is obtained by scaling $\boldsymbol{v}_c^{(m)}$, we have $\Delta \boldsymbol{w}_c^{(m)} \in \mathcal{S}(\boldsymbol{W})$. As $\boldsymbol{\Pi}_{\mathcal{S}}$ is the orthogonal projector onto $\mathcal{S}(\boldsymbol{W})$, it follows that

$$\Delta \boldsymbol{W}^{(m)} = \Delta \boldsymbol{W}^{(m)} \boldsymbol{\Pi}_{\mathcal{S}}.$$

Consequently,

$$\boldsymbol{e}^{(m)}(\boldsymbol{z}) = \Delta \boldsymbol{W}^{(m)} \boldsymbol{\Pi}_{\mathcal{S}} \boldsymbol{z} = \boldsymbol{e}^{(m)}(\boldsymbol{\Pi}_{\mathcal{S}} \boldsymbol{z}),$$

which proves the first identity in Eq. (1).

To characterize feature variations orthogonal to this subspace, let $\delta \boldsymbol{z} \in \mathcal{S}(\boldsymbol{W})^\perp$. By the definition of $\mathcal{S}(\boldsymbol{W})$, we can have

$$(\boldsymbol{w}_i - \boldsymbol{w}_j)^\top \delta \boldsymbol{z} = 0, \quad \forall\, i, j.$$

Hence, $\boldsymbol{w}_i^\top \delta \boldsymbol{z}$ is identical across all classes, so there exists a scalar a such that $\boldsymbol{W} \delta \boldsymbol{z} = a \mathbf{1}$. Moreover, the projection identity above implies

$$\Delta \boldsymbol{W}^{(m)} \delta \boldsymbol{z} = \Delta \boldsymbol{W}^{(m)} \boldsymbol{\Pi}_{\mathcal{S}} \delta \boldsymbol{z} = \mathbf{0}.$$

Consequently,

$$\begin{aligned} \boldsymbol{s}^{(0)}(\boldsymbol{z} + \delta \boldsymbol{z}) &= \boldsymbol{s}^{(0)}(\boldsymbol{z}) + a \mathbf{1}, \\ \boldsymbol{e}^{(m)}(\boldsymbol{z} + \delta \boldsymbol{z}) &= \boldsymbol{e}^{(m)}(\boldsymbol{z}). \end{aligned}$$

Since softmax is invariant to a common additive shift,

$$\operatorname{softmax}\Big(\boldsymbol{s}^{(0)}(\boldsymbol{z} + \delta \boldsymbol{z}) \Big) = \operatorname{softmax}\Big(\boldsymbol{s}^{(0)}(\boldsymbol{z}) \Big).$$

Thus, feature variations in $\mathcal{S}(\boldsymbol{W})^\perp$ preserve the relative logits and softmax predictions of the reference classifier, while leaving the induced logit perturbations unchanged.

Furthermore, the perturbation construction ensures that $\|\Delta \boldsymbol{w}_c^{(m)}\|_2 \leq \gamma \|\boldsymbol{w}_c\|_2$ for every c, implying $\|\Delta \boldsymbol{W}^{(m)}\|_F \leq \gamma \|\boldsymbol{W}\|_F$. Combining this, we can yield

$$\begin{aligned} \|\boldsymbol{e}^{(m)}(\boldsymbol{z})\|_2 &\leq \|\Delta \boldsymbol{W}^{(m)}\|_F \|\boldsymbol{\Pi}_{\mathcal{S}} \boldsymbol{z}\|_2 \\ &\leq \gamma \|\boldsymbol{W}\|_F \|\boldsymbol{\Pi}_{\mathcal{S}} \boldsymbol{z}\|_2, \end{aligned}$$

□

Despite being constructed from distinct reference classes, all $\mathcal{G}_c$ span a shared class-contrast subspace $\mathcal{S}(\boldsymbol{W})$ of dimension at most $\min(r, C-1)$. Consequently, ASFP confines the generated class-specific perturbation directions to this common discriminative subspace. Since pairwise class-weight differences govern how relative class scores change with the features, this restriction grounds functional probing in the classifier's learned discriminative geometry. This geometric constraint inherently serves as a discriminative inductive bias for functional probing, enabling the model to anticipate diverse predictive variations in unseen domains.

A.2 A Reweighting Interpretation of SmoothMax Objective

In the main manuscript, we adopt the per-sample SmoothMax objective in Eq. (6), which averages the exponentiated losses across the perturbed predictive functions before applying a logarithmic transformation. Its derivatives with respect to the individual head losses show that higher-loss heads receive greater emphasis for each example. Here, we clarify this implicit weighting through an equivalent formulation as loss reweighting with a KL penalty.

Let $l_m = \ell(\boldsymbol{s} + \boldsymbol{e}^{(m)}, y)$, $Z = \sum_{m=0}^{M} \pi_m \exp(l_m/\tau)$, and $p_m = \frac{\pi_m \exp(l_m/\tau)}{Z}$, where $\pi_m = \frac{1}{M+1}$. For any $\boldsymbol{q} \in \mathcal{Q}_M$, where $\mathcal{Q}_M$ denotes the probability simplex over all heads, we have

$$\begin{aligned}
\tau D_{\mathrm{KL}}(\boldsymbol{q}\|\boldsymbol{p}) &= \tau \sum_{m=0}^{M} q_m \log \frac{q_m}{p_m} \\
&= \tau \sum_{m=0}^{M} q_m \left(\log \frac{q_m}{\pi_m} - \frac{l_m}{\tau} + \log Z \right) \\
&= \tau D_{\mathrm{KL}}(\boldsymbol{q}\|\boldsymbol{\pi}) - \sum_{m=0}^{M} q_m l_m + \mathcal{L}_i,
\end{aligned}$$

Since the KL divergence $D_{\mathrm{KL}}(\boldsymbol{q}\|\boldsymbol{p})$ is non-negative, we can yield

$$\sum_{m} q_m l_m \le \mathcal{L}_i + \tau D_{\mathrm{KL}}(\boldsymbol{q}\|\boldsymbol{\pi}). \tag{12}$$

Here, equality holds when $\boldsymbol{q} = \boldsymbol{p}$, giving the exact identity

$$\mathcal{L}_i = \max_{\boldsymbol{q} \in \mathcal{Q}_M} \left\{ \sum_{m=0}^{M} q_m l_m - \tau D_{\mathrm{KL}}(\boldsymbol{q}\|\boldsymbol{\pi}) \right\}.$$

Therefore, the SmoothMax objective can be interpreted as maximizing a weighted average of the head losses while penalizing divergences from uniform weights. The weighted-loss term favors placing more weight on high-loss heads, whereas the KL penalty encourages the weights to remain close to $\boldsymbol{\pi}$. Moreover, the maximizing weights satisfy

$$\frac{\partial \mathcal{L}_i}{\partial l_m} = \frac{\tau}{Z} \frac{\partial Z}{\partial l_m} = p_m,$$

so they coincide exactly with the weights identified by the gradient analysis in the main manuscript. Since the maximizing weights admit a closed-form expression, this variational representation requires no additional inner optimization loop.

Finally, Eq. (12) holds for any $\boldsymbol{q} \in \mathcal{Q}_M$ and can therefore be instantiated with the coefficients used to approximate target-domain predictive variations, which need not coincide with the maximizing weights $\boldsymbol{p}$. This establishes the link between the training objective and the weighted combination of head losses employed in the proof of Prop. 1, as detailed in the following subsection.

A.3 Proof of Proposition 1

Proof. For any $\boldsymbol{q} \in \mathcal{Q}_M$, we can define

$$\boldsymbol{a}_{\boldsymbol{q}} = \sum_{m=0}^{M} q_m \boldsymbol{e}^{(m)}, \quad \boldsymbol{v}_{\boldsymbol{q}} = \boldsymbol{e}_{\mathcal{T}} - \boldsymbol{a}_{\boldsymbol{q}}.$$

where $\boldsymbol{a_q}$ approximates the target deviation using the explored perturbations, while $\boldsymbol{v_q}$ is the residual. Since $\sum_m q_m = 1$, we can have

$$\boldsymbol{s} + \boldsymbol{a_q} = \sum_{m=0}^{M} q_m \left(\boldsymbol{s} + \boldsymbol{e}^{(m)} \right).$$

Meanwhile, the cross-entropy objective is convex and globally $\sqrt{2}$-Lipschitz in logits under the Euclidean norm Gonon et al. (2024). Therefore, we obtain the following inequality

$$\ell(\boldsymbol{s} + \boldsymbol{a_q}, y) \leq \sum_{m=0}^{M} q_m \ell(\boldsymbol{s} + \boldsymbol{e}^{(m)}, y).$$

Then, combining Eq. (12) yields

$$\begin{aligned}
\ell(\boldsymbol{s} + \boldsymbol{e}_{\mathcal{T}}, y) &= \ell(\boldsymbol{s} + \boldsymbol{a_q} + \boldsymbol{v_q}, y) \\
&\leq \ell(\boldsymbol{s} + \boldsymbol{a_q}, y) + \sqrt{2}\|\boldsymbol{v_q}\|_2 && \triangleleft\ \sqrt{2}\text{-Lipschitz continuity} \\
&\leq \sum_{m=0}^{M} q_m \ell(\boldsymbol{s} + \boldsymbol{e}^{(m)}, y) + \sqrt{2}\|\boldsymbol{v_q}\|_2 && \triangleleft\ \text{Convexity} \\
&\leq \mathcal{L}_i + \tau D_{\mathrm{KL}}(\boldsymbol{q}\|\boldsymbol{\pi}) + \sqrt{2}\|\boldsymbol{v_q}\|_2. && \triangleleft\ \text{Eq. (12)}
\end{aligned}$$

□

To further demonstrate its geometric intuition, we can define the convex hull as

$$\mathcal{C}_i = \mathrm{conv}\{\boldsymbol{e}^{(0)}, \ldots, \boldsymbol{e}^{(M)}\}.$$

Hence, a closest point $\boldsymbol{e}^{\dagger}$ to $\boldsymbol{e}_{\mathcal{T}}$ exists and can be represented by some $\boldsymbol{q}^{\dagger} \in \mathcal{Q}_M$, satisfying

$$\left\|\boldsymbol{e}_{\mathcal{T}} - \boldsymbol{e}^{\dagger}\right\|_2 = \left\|\boldsymbol{e}_{\mathcal{T}} - \sum_{m=0}^{M} q_m^{\dagger} \boldsymbol{e}^{(m)}\right\|_2 = \mathrm{dist}(\boldsymbol{e}_{\mathcal{T}}, \mathcal{C}_i).$$

For the uniform reference weights $\pi_m = 1/(M+1)$,

$$\begin{aligned}
D_{\mathrm{KL}}(\boldsymbol{q}^{\dagger}\|\boldsymbol{\pi}) &= \sum_{m=0}^{M} q_m^{\dagger} \log q_m^{\dagger} + \log(M+1) \\
&\leq \log(M+1),
\end{aligned}$$

where the inequality follows from $q_m^{\dagger} \in [0, 1]$. Combining with Eq. (10) yields

$$\ell(\boldsymbol{s} + \boldsymbol{e}_{\mathcal{T}}, y) \leq \mathcal{L}_i + \tau \log(M+1) + \sqrt{2}\,\mathrm{dist}(\boldsymbol{e}_{\mathcal{T}}, \mathcal{C}_i),$$

In particular, the distance term vanishes whenever $\boldsymbol{e}_{\mathcal{T}} \in \mathcal{C}_i$. And, the result holds globally in logit space for every $\tau > 0$.

A.4 Proof of Proposition 2

Proof. Let $a_m = \boldsymbol{g}^{\top}\boldsymbol{e}^{(m)}$ and $b_m = \frac{1}{2}\boldsymbol{e}^{(m)\top}\boldsymbol{H}\boldsymbol{e}^{(m)}$. By the local Lipschitz continuity of the Hessian and $\|\boldsymbol{e}^{(m)}\|_2 \leq \varepsilon$, a second-order Taylor expansion gives

$$\ell(\boldsymbol{s} + \boldsymbol{e}^{(m)}, y) - \ell_0 = a_m + b_m + \mathcal{O}(\varepsilon^3)$$

uniformly over the finite perturbation family. Then, define $d_m = \ell(\boldsymbol{s} + \boldsymbol{e}^{(m)}, y) - \ell_0$. For fixed $\tau > 0$, expanding the exponential and subsequently the logarithm around $d_m = 0$ yields

$$\begin{aligned}
\tau \log \mathbb{E}_m[\exp(d_m/\tau)] &= \mathbb{E}_m[d_m] + \frac{1}{2\tau}\left(\mathbb{E}_m[d_m^2] - (\mathbb{E}_m[d_m])^2\right) + \mathcal{O}(\varepsilon^3) \\
&= \mathbb{E}_m[a_m] + \mathbb{E}_m[b_m] + \frac{1}{2\tau}\,\mathrm{Var}_m(a_m) + \mathcal{O}(\varepsilon^3).
\end{aligned}$$

where the second equality follows from $a_m = \mathcal{O}(\varepsilon)$ and $b_m = \mathcal{O}(\varepsilon^2)$.

Finally, using $\mathbb{E}_m[a_m] = \boldsymbol{g}^{\top}\boldsymbol{\mu}_e$, $\mathbb{E}_m[b_m] = \frac{1}{2}\,\mathrm{Tr}(\boldsymbol{H}\boldsymbol{Q}_e)$, and $\mathrm{Var}_m(a_m) = \boldsymbol{g}^{\top}\boldsymbol{\Sigma}_e\boldsymbol{g}$, we obtain the stated expansion, which completes the proof. □

This expansion separates the effect of the mean perturbation from the regularization induced by variations across the perturbed functions. A simple average of the perturbed losses already introduces the curvature term $\frac{1}{2}\operatorname{Tr}(\boldsymbol{H}\boldsymbol{Q}_e)$, whereas SmoothMax further contributes $\frac{1}{2\tau}\boldsymbol{g}^\top\boldsymbol{\Sigma}_e\boldsymbol{g}$, which penalizes first-order loss sensitivity across the explored functions. For cross-entropy, both terms are nonnegative, while the mean-shift term $\boldsymbol{g}^\top\boldsymbol{\mu}_e$ may have either sign and vanishes for centered perturbations. Therefore, when the perturbations are centered, jointly optimizing the perturbed functions with SmoothMax suppresses both local curvature and first-order sensitivity along the explored perturbation geometry, thereby promoting local functional flatness in these directions.

To further examine whether this local regularization can bound the loss increase under unseen perturbations, we first define the following term arising from the local expansion in Prop. 2:

$$\Omega_{\mathrm{NF}} = \frac{1}{2}\operatorname{Tr}(\boldsymbol{H}\boldsymbol{\Sigma}_e) + \frac{1}{2\tau}\boldsymbol{g}^\top\boldsymbol{\Sigma}_e\boldsymbol{g}.$$

where $\boldsymbol{\Sigma}_e$ characterizes the local geometry explored by the perturbed predictive functions.

Furthermore, with a slight abuse of notation, we revisit the decomposition of $\boldsymbol{e}$ introduced in A.3 as

$$\boldsymbol{e} = \boldsymbol{a} + \boldsymbol{v}, \quad \boldsymbol{a} = \boldsymbol{\Sigma}_e^{1/2}\boldsymbol{u}, \quad \|\boldsymbol{u}\|_2 \le \epsilon,$$

where $\boldsymbol{a}$ denotes the component represented by the explored perturbation geometry, while $\boldsymbol{v}$ is the residual component not captured by it. Equivalently, $\boldsymbol{a}$ lies in the covariance-induced ellipsoid

$$\mathcal{E}_\epsilon(\boldsymbol{\Sigma}_e) = \left\{\boldsymbol{\Sigma}_e^{1/2}\boldsymbol{u} : \|\boldsymbol{u}\|_2 \le \epsilon\right\}.$$

With the L_H-Lipschitz assumption, a second-order Taylor expansion gives

$$\begin{aligned}
\ell(\boldsymbol{s}+\boldsymbol{e}, y) - \ell(\boldsymbol{s}, y) &\le \boldsymbol{g}^\top\boldsymbol{a} + \frac{1}{2}\boldsymbol{a}^\top\boldsymbol{H}\boldsymbol{a} + \frac{L_H}{6}\|\boldsymbol{a}\|_2^3 + \sqrt{2}\|\boldsymbol{v}\|_2 \\
&\le \epsilon\sqrt{\boldsymbol{g}^\top\boldsymbol{\Sigma}_e\boldsymbol{g}} + \frac{\epsilon^2}{2}\operatorname{Tr}(\boldsymbol{H}\boldsymbol{\Sigma}_e) + \frac{L_H}{6}\|\boldsymbol{a}\|_2^3 + \sqrt{2}\|\boldsymbol{v}\|_2 \\
&\le \epsilon\sqrt{2\tau\Omega_{\mathrm{NF}}} + \epsilon^2\Omega_{\mathrm{NF}} + \frac{L_H}{6}\|\boldsymbol{a}\|_2^3 + \sqrt{2}\|\boldsymbol{v}\|_2. \qquad (13)
\end{aligned}$$

The second inequality follows from Cauchy-Schwarz and

$$\lambda_{\max}\left(\boldsymbol{\Sigma}_e^{1/2}\boldsymbol{H}\boldsymbol{\Sigma}_e^{1/2}\right) \le \operatorname{Tr}(\boldsymbol{H}\boldsymbol{\Sigma}_e),$$

where $\boldsymbol{\Sigma}_e^{1/2}\boldsymbol{H}\boldsymbol{\Sigma}_e^{1/2} \succeq 0$. The last inequality follows directly from the nonnegativity of the two components of Ω_{NF}.

Consistent with the discussion in A.3, Eq. (13) explicitly connects Ω_{NF} to robustness within the explored perturbation geometry, while deviations outside this geometry are captured by the residual term $\boldsymbol{v}$.

Importantly, this result does not assume that $\boldsymbol{\Sigma}_e$ spans the entire logit space. When $\boldsymbol{\Sigma}_e$ is singular, the ellipsoid is restricted to $\operatorname{Range}(\boldsymbol{\Sigma}_e)$, and deviations outside this range cannot be characterized solely by the Mahalanobis radius $\boldsymbol{e}^\top\boldsymbol{\Sigma}_e^\dagger\boldsymbol{e}$. Such components are precisely represented by the residual $\boldsymbol{v}$. For example, a single antithetic perturbation pair produces a covariance matrix of rank at most one, illustrating that the resulting local robustness is inherently directional rather than isotropic.

Finally, Ω_{NF} should not be interpreted in isolation as a robustness measure, since its magnitude depends on the scale of the perturbation covariance. Indeed, rescaling $\boldsymbol{\Sigma}_e$ by c^2 scales Ω_{NF} by c^2, while the ellipsoidal radius required to represent the same covered deviation scales from ρ to ρ/c. Hence, the two leading terms in Eq. (13) remain unchanged. This shows that robustness depends jointly on the regularization strength and the perturbation geometry it characterizes.

B Related Works

Existing domain generalization (DG) methods improve generalization to unseen domains from several perspectives. 1) Distribution-matching methods seek domain-invariant representations by reducing distribution discrepancies among domains. MMD Li et al. (2018b) aligns distributions

through kernel mean embeddings, while CORAL Sun & Saenko (2016) matches second-order representation statistics. DANN Ganin et al. (2016) introduced an additional domain classifier to interfere with the representations, so that they do not contain domain information. I^3C Zhou et al. (2025) incorporates class-aware invariance into cross-domain representation learning. Meanwhile, multiple researches improve the invariance at the risk level. For example, RDM Nguyen et al. (2024) matches domain-wise risk distributions. GroupDRO Sagawa et al. (2020) optimizes worst-group performance, while VREx Krueger et al. (2021) reduces variation in risks across environments and EQRM Eastwood et al. (2022) minimizes a high quantile of the domain-risk distribution. Another invariance-based methods, such as IRM Arjovsky et al. (2019) and TCRI Salaudeen & Koyejo (2024), exploit invariant predictors or causal regularization to suppress environment-dependent correlations. 2) Optimization-based methods instead regularize the learning dynamics across domains via gradient information. Fish Shi et al. (2022) encourages agreement between domain-wise gradients, while Fishr Rame et al. (2022) aligns their gradient variances. PGrad Wang et al. (2023b) extracts robust principal directions from optimization trajectories, and GGA Ballas & Diou (2025) searches for parameter regions with improved cross-domain gradient alignment. 3) Meta-learning-based methods, such as MLDG Li et al. (2018a), simulate domain shifts to improve generalizability. 4) Augmentation-based methods enlarge source-domain diversity through sample or feature interpolation to reduce domain gap, including Mixup Yan et al. (2020) and MixStyle Zhou et al. (2024). In summary, these methods typically improve generalization through data-level manipulation or distribution-level regularization.

Another line of work, which is most closely related to our study, explicitly seeks flat solutions whose predictions are less sensitive to local parameter perturbations. SWAD Cha et al. (2021) promotes flat minima through dense and overfit-aware weight averaging, establishing a direct connection between flatness and domain generalization. Building on sharpness-aware optimization, SAGM Wang et al. (2023a) combines neighborhood flatness with gradient matching to improve out-of-domain robustness. MeCAM Chen et al. (2024) further introduces a curvature-aware objective to encourage convergence toward flatter minima, while SFT Li et al. (2025) seeks flat minima that are more consistent across domain-specific loss landscapes. More recently, DA-SAM He et al. (2026) accounts for domain heterogeneity by adaptively adjusting the directions and magnitudes of sharpness-aware perturbations. Despite their effectiveness, many perturbation-based flatness-aware methods characterize local neighborhoods using generic parameter-space metrics and rely on additional gradient-based search to identify informative perturbations. Such neighborhoods do not explicitly exploit the classifier's learned inter-class geometry when exploring predictive variations. In contrast, our method constructs multiple classifier-conditioned perturbation directions directly from pairwise weight differences, inducing diverse functional probes without gradient-based perturbation search.

C The Details for Toy Data

As illustrated in Fig. 5 (A), we construct a three-class two-dimensional toy dataset with class centers

$$\boldsymbol{\mu}_1 = (2, 0), \quad \boldsymbol{\mu}_2 = (-1, \sqrt{3}), \quad \boldsymbol{\mu}_3 = (-1, -\sqrt{3}),$$

which are evenly distributed on a circle and form an equilateral triangular configuration.

For each class c, latent samples are independently drawn from an isotropic Gaussian distribution,

$$\boldsymbol{u} = \boldsymbol{\mu}_c + 0.25\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}).$$

Different domains are generated by rotating the latent samples while preserving their class labels. Specifically, as illustrated in Fig. 5 (B), a domain with rotation angle θ is obtained by

$$\boldsymbol{x}_\theta = \boldsymbol{R}_\theta \boldsymbol{u}, \quad \boldsymbol{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

We construct three source domains with rotation angles -15°, 0°, and 15°. Each source domain is independently sampled and contains 500 examples per class, resulting in $1,500$ training samples per domain. For evaluation, we independently sample a shared set of 300 examples per class and rotate the same latent samples to 0°, 5°, 30°, and 45°. The 0° domain only serves as the reference domain, while the remaining three are treated as unseen target domains. Using the same latent evaluation samples across different rotation angles ensures that their differences arise solely from the controlled rotational shift.

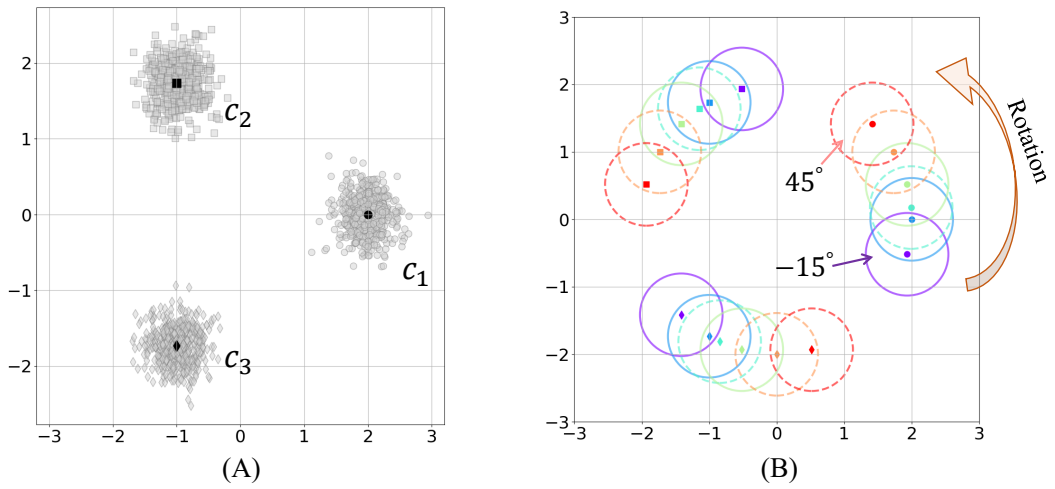


Figure 5: Illustration of rotated toy data. (A) represents samples randomly generated around three predefined class centers. (B) represents the rotated class distributions, where the dashed circle denotes the target domain and the others denote the source domains.

D Detailed Description of Experiments on Five Benchmarks

Datasets Five commonly used DG benchmarks have been included in our main manuscript. VLCS Fang et al. (2013) consists of images from 4 different domains: Voc2007, LabelMe, Caltech101, Sun09. It contains around 10,729 images from 5 different classes. PACS Li et al. (2017) consists of images from 4 different domains: Art, Cartoon, Photo, Sketch. It contains around 9,991 images from 7 different classes. Office-Home (Venkateswara et al., 2017) consists of images from 4 different domains: Art, Clipart, Product and Real-World. It contains around 15,500 images from 65 different categories. TerraIncognita (Beery et al., 2018) consists of images from 4 different domains: L100, L38, L43, L46. It contains around 24,778 images from 10 different classes. DomainNet (Peng et al., 2019), which is the largest and most complex dataset currently available, consists of images from 6 domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. It contains over 600,000 images from 345 different classes.

Settings Following the settings on DomainBed Gulrajani & Lopez-Paz (2021), we adopt ResNet-50 architecture as the backbone, and each input image is resized and cropped to 224×224. Adam is utilized as the optimizer. The training data are randomly split into two parts: 80% for training and 20% for validation. The best model on the validation split is selected to evaluate the target domain. The training batch size is set to 32 per source domain for all benchmarks except OfficeHome and DomainNet, in which it is set to 64. The test batch size is set to 64 for all benchmarks. Maximum iteration T is set to 5,000. The default learning rate is set to $5e$-5. Three independent trials are conducted, each with 20 randomly sampled hyperparameter configurations. In ASFP, the number of perturbed classifiers M is sampled from $[2^5, 2^7]$, with a default value of 64. The perturbation magnitude γ is set to 0.2 for all benchmarks except OfficeHome, in which it is set to 0.35. Meanwhile, the temperature τ is set to 0.1 for all benchmarks. The codes are run on Python 3.10 and PyTorch 2.7 on Arch Linux with NVIDIA GeForce RTX 4090 and RTX 5090 GPUs.

E Full Experimental Results

In this section, we provide the complete results on all benchmarks in Tab. 4-8. ASFP achieves the best performance in most cases, further demonstrating its effectiveness across diverse domain shifts.

Algorithm	C	L	S	V	Avg
ERM	97.7 ± 0.4	64.3 ± 0.9	73.4 ± 0.5	74.6 ± 1.3	77.5
Mixup	98.3 ± 0.6	64.8 ± 1.0	72.1 ± 0.5	74.3 ± 0.8	77.4
DANN	99.0 ± 0.3	65.1 ± 1.4	73.1 ± 0.3	77.2 ± 0.6	78.6
MLDG	97.4 ± 0.2	65.2 ± 0.7	71.0 ± 1.4	75.3 ± 1.0	77.2
GroupDRO	97.3 ± 0.3	63.4 ± 0.9	69.5 ± 0.8	76.7 ± 0.7	76.7
IRM	98.6 ± 0.1	64.9 ± 0.9	73.4 ± 0.6	77.3 ± 0.9	78.6
VREx	98.4 ± 0.3	64.4 ± 1.4	74.1 ± 0.4	76.2 ± 1.3	78.3
EQRM	98.3 ± 0.0	63.7 ± 0.8	72.6 ± 1.0	76.7 ± 1.1	77.8
Fish	-	-	-	-	77.8
Fishr	98.9 ± 0.3	64.0 ± 0.5	71.5 ± 0.2	76.8 ± 0.7	77.8
CORAL	98.3 ± 0.1	66.1 ± 1.2	73.4 ± 0.3	77.5 ± 1.2	78.8
MMD	97.7 ± 0.1	64.0 ± 1.1	72.8 ± 0.2	75.3 ± 3.3	77.4
Mixstyle	98.6 ± 0.3	64.5 ± 1.1	72.6 ± 0.5	75.7 ± 1.7	77.8
RDM	98.1 ± 0.2	64.9 ± 0.7	72.6 ± 0.5	77.9 ± 1.2	78.4
TCRI	**99.1 ± 0.4**	67.2 ± 0.6	74.4 ± 0.6	79.0 ± 0.6	79.9
I^3C	98.8 ± 0.2	67.5 ± 0.6	72.1 ± 2.0	**80.2 ± 0.8**	79.6
GGA	**99.1 ± 0.2**	67.5 ± 0.6	75.1 ± 0.3	78.0 ± 0.1	79.9
SAGM	98.4 ± 0.6	65.1 ± 1.1	74.1 ± 0.5	78.6 ± 2.4	79.0
CRSAM	99.0 ± 0.4	63.7 ± 0.8	73.8 ± 0.5	79.8 ± 1.4	79.1
FSAM	98.9 ± 0.2	64.5 ± 0.4	74.1 ± 1.0	80.0 ± 1.0	79.4
DA-SAM	98.7 ± 0.2	65.9 ± 1.3	75.2 ± 0.8	78.8 ± 1.3	79.6
ASFP	99.0 ± 0.0	**67.6 ± 0.7**	**75.8 ± 0.7**	79.1 ± 0.5	**80.4**

Table 4: Overall results on VLCS Benchmark. The bold and underline items are the best and the second-best results, respectively.

Algorithm	A	C	P	S	Avg
ERM	84.7 ± 0.4	80.8 ± 0.6	97.2 ± 0.3	79.3 ± 1.0	85.5
Mixup	86.1 ± 0.5	78.9 ± 0.8	97.6 ± 0.1	75.8 ± 1.8	84.6
DANN	86.4 ± 0.8	77.4 ± 0.8	97.3 ± 0.4	73.5 ± 2.3	83.6
MLDG	85.5 ± 1.4	80.1 ± 1.7	97.4 ± 0.3	76.6 ± 1.1	84.9
GroupDRO	83.5 ± 0.9	79.1 ± 0.6	96.7 ± 0.3	78.3 ± 2.0	84.4
IRM	84.8 ± 1.3	76.4 ± 1.1	96.7 ± 0.6	76.1 ± 1.0	83.5
VREx	86.0 ± 1.6	79.1 ± 0.6	96.9 ± 0.5	77.7 ± 1.7	84.9
EQRM	86.5 ± 0.4	82.1 ± 0.7	96.6 ± 0.2	80.8 ± 0.2	86.5
Fish	-	-	-	-	85.5
Fishr	88.4 ± 0.2	78.7 ± 0.7	97.0 ± 0.1	77.8 ± 2.0	85.5
CORAL	88.3 ± 0.2	80.0 ± 0.5	97.5 ± 0.3	78.8 ± 1.3	86.2
MMD	86.1 ± 1.4	79.4 ± 0.9	96.6 ± 0.2	76.5 ± 0.5	84.6
Mixstyle	86.8 ± 0.5	79.0 ± 1.4	96.9 ± 0.1	78.5 ± 2.3	85.3
RDM	88.4 ± 0.2	81.3 ± 1.6	97.1 ± 0.1	81.8 ± 1.1	87.1
TCRI	87.8 ± 0.7	81.8 ± 0.6	**98.1 ± 0.4**	80.3 ± 0.4	87.0
I^3C	89.2 ± 1.1	80.6 ± 1.3	97.7 ± 0.6	81.4 ± 0.6	87.2
GGA	88.8 ± 0.2	80.1 ± 0.3	97.3 ± 0.2	81.2 ± 0.5	86.8
SAGM	87.6 ± 0.4	81.8 ± 0.5	97.2 ± 0.1	81.3 ± 2.1	87.0
CRSAM	86.7 ± 0.5	80.5 ± 1.6	96.4 ± 0.5	77.8 ± 2.4	85.4
FSAM	87.6 ± 1.1	80.2 ± 0.7	95.4 ± 0.4	**82.8 ± 0.7**	86.5
DA-SAM	88.6 ± 0.8	82.9 ± 0.4	96.7 ± 0.3	81.9 ± 1.2	87.5
ASFP	**89.4 ± 0.4**	**84.0 ± 0.5**	98.0 ± 0.3	81.9 ± 0.8	**88.3**

Table 5: Overall results on PACS Benchmark.

Algorithm	A	C	P	R	Avg
ERM	61.3 ± 0.7	52.4 ± 0.3	75.8 ± 0.1	76.6 ± 0.3	66.5
Mixup	62.4 ± 0.8	54.8 ± 0.6	76.9 ± 0.3	78.3 ± 0.2	68.1
DANN	59.9 ± 1.3	53.0 ± 0.3	73.6 ± 0.7	76.9 ± 0.5	65.8
MLDG	61.5 ± 0.9	53.2 ± 0.6	75.0 ± 1.2	77.5 ± 0.4	66.8
GroupDRO	60.4 ± 0.7	52.7 ± 1.0	75.0 ± 0.7	76.0 ± 0.7	66.0
IRM	58.9 ± 2.3	52.2 ± 1.6	72.1 ± 2.9	74.0 ± 2.5	64.3
VREx	60.7 ± 0.9	53.0 ± 0.9	75.3 ± 0.1	76.6 ± 0.5	66.4
EQRM	60.5 ± 0.1	56.0 ± 0.2	76.1 ± 0.4	77.4 ± 0.3	67.5
Fish	-	-	-	-	68.6
Fishr	62.4 ± 0.5	54.4 ± 0.4	76.2 ± 0.5	78.3 ± 0.1	67.8
CORAL	65.3 ± 0.4	54.4 ± 0.5	76.5 ± 0.1	78.4 ± 0.5	68.6
MMD	60.4 ± 0.2	53.3 ± 0.3	74.3 ± 0.1	77.4 ± 0.6	66.4
Mixstyle	51.1 ± 0.3	53.2 ± 0.4	68.2 ± 0.7	69.2 ± 0.6	60.4
RDM	61.1 ± 0.4	55.1 ± 0.3	75.7 ± 0.5	77.3 ± 0.3	67.3
TCRI	62.9 ± 0.6	54.0 ± 0.0	76.5 ± 0.5	78.0 ± 0.4	67.8
I^3C	**67.5 ± 0.8**	56.1 ± 0.4	77.5 ± 0.4	79.8 ± 0.4	**70.2**
GGA	64.3 ± 0.1	54.4 ± 0.2	76.5 ± 0.3	78.9 ± 0.2	68.5
SAGM	64.2 ± 0.6	56.2 ± 0.4	78.2 ± 0.2	79.2 ± 0.1	69.4
CRSAM	60.8 ± 3.7	56.9 ± 0.4	78.3 ± 0.5	79.5 ± 0.7	68.9
FSAM	64.7 ± 0.3	**57.3 ± 0.8**	**78.8 ± 0.1**	**79.9 ± 0.2**	**70.2**
DA-SAM	64.2 ± 0.7	53.8 ± 1.0	77.4 ± 0.2	78.1 ± 0.7	68.4
ASFP	65.7 ± 0.2	56.2 ± 0.2	77.9 ± 0.1	79.4 ± 0.1	69.8

Table 6: Overall results on OfficeHome Benchmark.

Algorithm	L100	L38	L43	L46	Avg
ERM	49.8 ± 4.4	42.1 ± 1.4	56.9 ± 1.8	35.7 ± 3.9	46.1
Mixup	**59.6 ± 2.0**	42.2 ± 1.4	55.9 ± 0.8	33.9 ± 1.4	47.9
DANN	51.1 ± 3.5	40.6 ± 0.6	57.4 ± 0.5	37.7 ± 1.8	46.7
MLDG	54.2 ± 3.0	44.3 ± 1.1	55.6 ± 0.3	36.9 ± 2.2	47.8
GroupDRO	41.2 ± 0.7	38.6 ± 2.1	56.7 ± 0.9	36.4 ± 2.1	43.2
IRM	54.6 ± 1.3	39.8 ± 1.9	56.2 ± 1.8	39.6 ± 0.8	47.6
VREx	48.2 ± 4.3	41.7 ± 1.3	56.8 ± 0.8	38.7 ± 3.1	46.3
EQRM	47.9 ± 1.9	45.2 ± 0.3	59.1 ± 0.3	38.8 ± 0.6	47.8
Fish	-	-	-	-	45.1
Fishr	50.2 ± 3.9	43.9 ± 0.8	55.7 ± 2.2	39.8 ± 1.0	47.4
CORAL	51.6 ± 2.4	42.2 ± 1.0	57.0 ± 1.0	39.8 ± 2.9	47.7
MMD	41.9 ± 3.0	34.8 ± 1.0	57.0 ± 1.9	35.2 ± 1.8	42.2
Mixstyle	54.3 ± 1.1	34.1 ± 1.1	55.9 ± 1.1	31.7 ± 2.1	44.0
RDM	52.9 ± 1.2	43.1 ± 1.0	58.1 ± 1.3	36.1 ± 2.9	47.6
TCRI	58.9 ± 1.6	45.0 ± 1.0	59.3 ± 0.2	41.5 ± 0.5	51.2
I^3C	57.0 ± 3.5	43.2 ± 2.4	56.4 ± 1.7	41.9 ± 1.3	49.6
GGA	55.9 ± 0.1	45.5 ± 0.1	59.7 ± 0.1	41.5 ± 0.1	50.7
SAGM	52.2 ± 3.9	42.3 ± 0.2	59.9 ± 0.3	39.9 ± 2.2	48.6
CRSAM	50.8 ± 0.5	35.3 ± 1.1	56.8 ± 0.1	38.4 ± 1.4	45.3
FSAM	47.7 ± 3.1	39.3 ± 3.6	57.8 ± 1.7	39.5 ± 2.1	46.1
DA-SAM	59.0 ± 0.6	39.2 ± 1.1	58.5 ± 1.9	43.3 ± 1.5	50.0
ASFP	58.5 ± 2.0	**50.2 ± 0.5**	**61.1 ± 0.7**	**44.5 ± 0.0**	**53.6**

Table 7: Overall results on TerraIncognita Benchmark.

Algorithm	C	I	P	Q	R	S	Avg
ERM	58.1 ± 0.3	18.8 ± 0.3	46.7 ± 0.3	12.2 ± 0.4	59.6 ± 0.1	49.8 ± 0.4	40.9
Mixup	55.7 ± 0.3	18.5 ± 0.5	44.3 ± 0.5	12.5 ± 0.4	55.8 ± 0.3	48.2 ± 0.5	39.2
DANN	53.1 ± 0.2	18.3 ± 0.1	44.2 ± 0.7	11.8 ± 0.1	55.5 ± 0.4	46.8 ± 0.6	38.3
MLDG	59.1 ± 0.2	19.1 ± 0.3	45.8 ± 0.7	13.4 ± 0.3	59.6 ± 0.1	50.2 ± 0.4	41.2
GroupDRO	47.2 ± 0.5	17.5 ± 0.4	33.8 ± 0.5	9.3 ± 0.3	51.6 ± 0.4	40.1 ± 0.6	33.2
IRM	48.5 ± 2.8	15.0 ± 1.5	38.3 ± 4.3	10.9 ± 0.5	48.2 ± 5.2	42.3 ± 3.1	33.9
VREx	47.3 ± 3.5	16.0 ± 1.5	35.8 ± 4.6	10.9 ± 0.3	49.6 ± 4.9	42.0 ± 3.0	33.6
EQRM	56.1 ± 1.3	19.6 ± 0.1	46.3 ± 1.5	12.9 ± 0.3	61.1 ± 0.0	50.3 ± 0.1	41.1
Fish	-	-	-	-	-	-	42.7
Fishr	58.2 ± 0.5	20.2 ± 0.2	47.7 ± 0.3	12.7 ± 0.2	60.3 ± 0.2	50.8 ± 0.1	41.7
CORAL	59.2 ± 0.1	19.7 ± 0.2	46.6 ± 0.3	13.4 ± 0.4	59.8 ± 0.2	50.1 ± 0.6	41.5
MMD	32.1 ± 13.3	11.0 ± 4.6	26.8 ± 11.3	8.7 ± 2.1	32.7 ± 13.8	28.9 ± 11.9	23.4
Mixstyle	51.9 ± 0.4	13.3 ± 0.2	37.0 ± 0.5	12.3 ± 0.1	46.1 ± 0.3	43.4 ± 0.4	34.0
RDM	62.1 ± 0.2	20.7 ± 0.1	49.2 ± 0.4	14.1 ± 0.2	63.0 ± 1.3	51.4 ± 0.1	43.4
TCRI	58.6 ± 0.3	18.6 ± 0.1	49.4 ± 0.2	12.9 ± 0.1	61.5 ± 0.2	50.8 ± 0.1	42.0
I^3C	59.4 ± 0.7	20.8 ± 0.4	49.1 ± 1.1	13.2 ± 0.4	61.8 ± 0.4	51.1 ± 0.8	42.6
GGA	**64.0 ± 0.2**	**22.2 ± 0.3**	**51.7 ± 0.1**	14.3 ± 0.2	64.1 ± 0.4	**54.3 ± 0.3**	**45.1**
SAGM	63.6 ± 0.5	21.6 ± 0.2	51.4 ± 0.4	14.2 ± 0.3	64.2 ± 0.2	53.1 ± 0.2	44.7
CRSAM	62.9 ± 0.3	20.7 ± 0.2	50.8 ± 0.2	15.2 ± 0.4	62.5 ± 0.4	53.5 ± 0.2	44.3
FSAM	63.3 ± 0.5	21.5 ± 0.1	51.6 ± 0.3	**15.4 ± 0.5**	63.5 ± 0.2	**54.3 ± 0.2**	44.9
DA-SAM	63.4 ± 0.2	21.9 ± 0.2	50.6 ± 0.6	14.2 ± 0.6	**64.4 ± 0.3**	52.7 ± 0.7	44.5
ASFP	62.5 ± 0.5	21.6 ± 0.1	50.9 ± 0.3	15.3 ± 0.1	63.2 ± 0.3	52.5 ± 0.1	44.3

Table 8: Overall results on DomainNet Benchmark.