DOMAIN GENERALIZATION GUIDED BY CLASS-WISE DOMAIN-INVARIANT SEMANTIC SUBSPACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain generalization aims to learn a model from multiple source domains that generalizes to unseen target domains. Existing domain generalization methods typically model each class as a single prototype vector, that is, a point in Euclidean space, making it difficult to explicitly characterize intra-class variation under domain shift at the class level. To this end, we model each class as a semantic subspace. Specifically, we use multiple class-related prompts to estimate the class semantic center and its principal variation directions, extending the point representation to a low-dimensional affine subspace. The model combines prototype similarity with subspace projection similarity, thereby providing semantic discriminative evidence beyond a single-point prototype for samples of the same class across different domains. To preserve the semantic structure of the subspace and improve prediction stability against variations not represented by it, we combine semantic geometry constraints with residual-space robustness learning. Theoretical analysis characterizes the representational properties of the subspace and prediction stability under bounded residual perturbations. Experiments on multiple domain generalization benchmarks show that our method achieves stable and competitive performance.

1 INTRODUCTION

Traditional machine learning methods have achieved strong performance on tasks(Redmon et al., 2016; Zou et al., 2023; Zhang et al., 2024a) such as image classification(He et al., 2016), but they typically rely on the assumption that training and test data are independently drawn from the same distribution(Cha et al., 2022). In real-world scenarios, however, changes in imaging conditions, background environments, and visual styles may induce substantial distribution shifts, making the discriminative patterns learned from source domains difficult to generalize to unseen data. This motivates the problem of domain generalization(Zhou et al., 2023). Domain generalization aims to train a model using only one or multiple source domains such that it generalizes well to target domains that are inaccessible during training. Therefore, learning representations that remain semantically consistent and robust to domain shifts continues to be a central challenge in domain generalization.

Vision-language models such as CLIP(Radford et al., 2021) provide a new paradigm for domain generalization through their strong cross-domain semantic representations. Existing CLIP-based methods mainly exploit this knowledge through prompt engineering or prompt tuning(Zhou et al., 2022b;a). However, they often represent each class using a single text prototype or compress multiple prompts into one representation. Such point-wise representations may discard the intra-class semantic variations conveyed by diverse descriptions, attributes, and contexts.

We argue that representing each class by a single text prototype(Khattak et al., 2023) may be restrictive under domain shift, since an unseen-domain image need not remain close to one fixed textual direction. Multiple prompts describe complementary class characteristics, and their principal variation directions provide a low-rank extension of the class center. We model each class as an affine textual subspace, allowing an image to receive class evidence through its compatibility with the text-derived variation directions rather than its proximity to a single prototype. Based on this observation, we represent each class as an affine textual semantic subspace, as illustrated in Figure 1.

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.

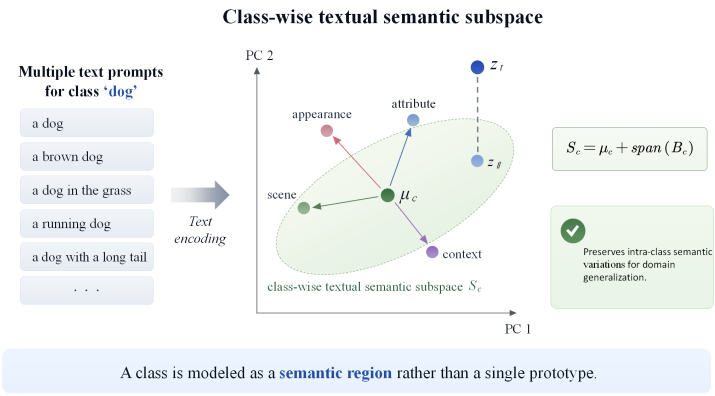

Figure 1: Motivation for class-wise textual semantic subspaces. Multiple prompts define a class center and principal variation directions, extending class representation from a single prototype to an affine subspace.

Specifically, we encode multiple class-related prompts using the frozen CLIP(Radford et al., 2021) text encoder and construct a class-wise low-rank textual subspace from their semantic center and principal variation directions. During training, we learn adaptable textual contexts and impose textual knowledge and subspace geometry preservation constraints, allowing adaptation to source domains while retaining CLIP's pre-trained semantic knowledge.

For classification, we do not use the Euclidean distance from an image feature to a textual subspace. Instead, we combine the prototype similarity between the image feature and the class center with the similarity between the image feature and its class-specific subspace projection, thereby exploiting both central class semantics and intra-class semantic variations. Moreover, we decompose each image feature into its projection onto the ground-truth class textual subspace and the corresponding out-of-subspace residual. We use a weak subspace-proximity regularizer and enforce prediction consistency under bounded adversarial perturbations in the orthogonal complement, improving robustness to variations not represented by the current textual subspace.

The main contributions of this work are summarized as follows. **(1)** We propose a class-wise affine textual subspace representation that extends each class center along the principal variation directions extracted from multiple prompts, retaining dominant intra-class textual variations. **(2)** We introduce a subspace-projection-similarity-based CLIP classifier that combines prototype similarity with the angular compatibility between an image feature and its class-wise subspace projection. **(3)** We introduce textual knowledge preservation, subspace geometry preservation, out-of-subspace residual regularization, and adversarial residual-space prediction invariance to improve robustness to bounded variations not directly represented by the textual subspaces.

2 Related Work

2.1 Domain Generalization

Domain generalization (DG) aims to learn a model from one or multiple source domains that can generalize to previously unseen target domains. Existing DG methods can generally be categorized into three groups: data manipulation(Yue et al., 2019; Huang et al., 2021), representation learning(Gan et al., 2016; Ghifary et al., 2017), and learning strategies(Cha et al., 2021; Zhang et al., 2024b). Data manipulation methods improve model robustness by applying data augmentation(Wang et al., 2022; 2024) or synthesizing domain(Li et al., 2026; 2021) variations. Representation learning methods(Ganin & Lempitsky, 2015) seek to learn domain-invariant(Yao et al., 2022) representations through distribution alignment(Li et al., 2018), disentanglement of semantic(Peng et al., 2019b) and domain-specific factors, or invariant risk minimization(Mitrovic et al., 2021). Learning-strategy-based methods improve model generalization by designing more robust training and optimization schemes, such as meta-learning (Qiao & Peng, 2021) and gradient-aware

optimization that alleviates gradient conflicts across source domains (Xu et al., 2025). Although these unimodal approaches have achieved substantial progress, they typically rely solely on visual supervision from the source domains. Consequently, models may still capture source-specific visual correlations, such as textures, backgrounds, or styles, limiting robustness to unseen target domains.

2.2 Text-Based Domain Generalization

In recent years, vision-language models such as CLIP have introduced a new paradigm for domain generalization. Through large-scale image-text contrastive pre-training, CLIP learns a shared visual-textual embedding space and exhibits strong zero-shot transferability across diverse visual domains, supporting both domain generalization(Lew et al., 2023) and domain adaptation(Chen et al., 2023). Existing CLIP-based DG methods primarily focus on prompt learning or generation. Soft-prompt methods replace manually designed prompts with learnable prompt tokens and adapt them using source-domain data. For example, Soft Prompt Generation(Bai et al., 2024) reformulates prompt learning from a generative perspective and generates diverse instance-specific soft prompts for unseen target domains. Multiple Local Prompts Distillation(Lyu et al., 2025) learns multiple local prompts from different subsets of the training data and distills their knowledge into a global prompt to improve generalization. These methods demonstrate that textual prompts can activate the semantic knowledge encoded in CLIP and alleviate performance degradation under distribution shifts.

Despite their promising performance, existing text-enhanced domain generalization methods typically represent each class using a single text prototype or an aggregated prompt representation, potentially discarding the intra-class semantic variations conveyed by multiple textual descriptions. In contrast, we construct a class-wise textual semantic subspace from multiple class-related prompts and combine prototype similarity with subspace projection similarity for classification, thereby retaining both central class semantics and text-derived variation directions. We introduce a weak subspace-proximity constraint and adversarial prediction consistency in out-of-subspace directions, complementing the subspace-based classifier with local robustness regularization.

3 Method

3.1 Problem Setup and Method Overview

Let $\mathcal{D}_S = \{\mathcal{D}_S^m\}_{m=1}^{N_s}$ and $\mathcal{D}_T = \{\mathcal{D}_T^m\}_{m=1}^{N_t}$ denote the collections of source and target domains, respectively, where N_s and N_t are the corresponding numbers of domains. Each source domain is represented as $\mathcal{D}_S^m = \{(x_i^m, y_i^m)\}_{i=1}^{n_m}$, where $x_i^m \in \mathcal{X}$ is an input image and $y_i^m \in \mathcal{Y} = \{1, \ldots, C\}$ is its class label, with C denoting the number of classes. The target domains are inaccessible during training, and N_t is typically set to one under the leave-one-domain-out evaluation protocol. The goal of domain generalization is to learn a parameterized classifier $h_\theta : \mathcal{X} \to \mathcal{Y}$ using only $\mathcal{D}_S$, such that it generalizes well to the unseen target domain $\mathcal{D}_T$.

We study CLIP-based domain generalization for image classification. As illustrated in Figure 2, our method represents each class as a textual semantic subspace derived from the principal variation directions of multiple textual prompts, rather than a single text prototype. It then combines prototype similarity with subspace projection similarity for classification and treats bounded out-of-subspace directions as a local uncertainty set for prediction-invariance learning. This formulation preserves the pre-trained textual structure of CLIP while improving prediction robustness to variations not directly represented by the textual subspaces.

3.2 Class-wise Textual Semantic Subspace

CLIP-based classifiers typically represent each class with a single text prototype from manual or learnable prompts. However, a single prototype captures only the central semantic direction of a class. In domain generalization, images from unseen domains may exhibit different attributes, appearances, and contexts for the same class. To preserve such intra-class semantic variations, we represent each class with a low-rank textual semantic subspace instead of a single text feature.

For each class $c \in \mathcal{Y}$, we construct a set of K class-related textual prompts, denoted by $\mathcal{P}_c = \{p_{c,k}\}_{k=1}^{K}$. These prompts describe the same class from different semantic perspectives, such as

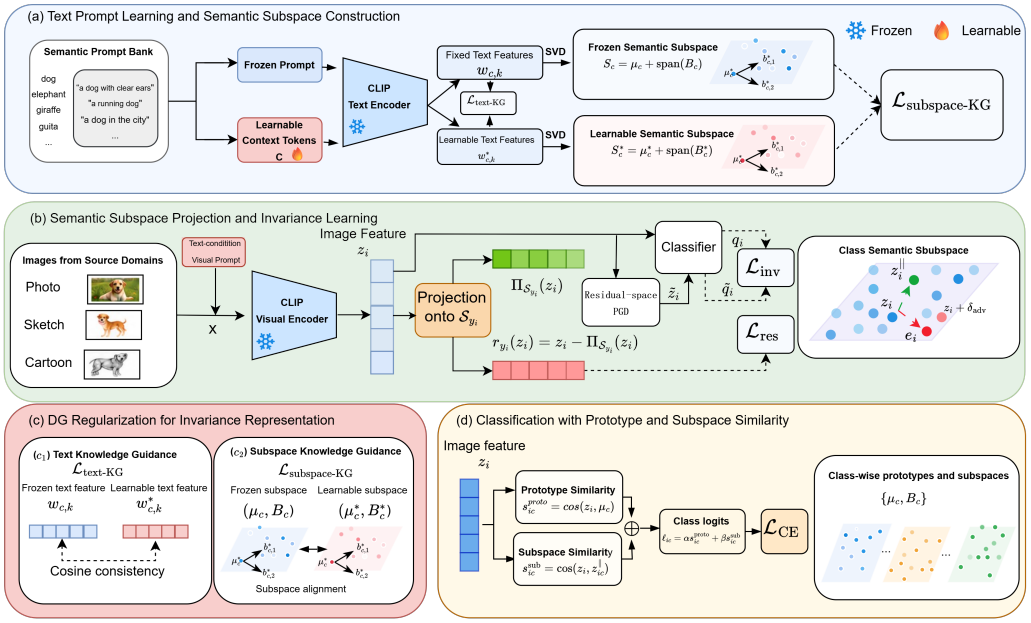

Figure 2: Overview of the proposed framework. (a) Multiple prompts construct frozen and learnable class-wise textual semantic subspaces. (b) Visual features are decomposed into textual-subspace projections and out-of-subspace residuals, and bounded residual-space perturbations are used to encourage prediction invariance. (c) Textual knowledge and subspace geometry regularize prompt learning. (d) Prototype similarity and subspace projection similarity are combined for classification.

object appearance, visual attributes, scene context, and common domain variations. Given the frozen CLIP text encoder $f_t(\cdot)$, each prompt is encoded as $t^{\mathrm{fr}}_{c,k} = f_t(p_{c,k})/\|f_t(p_{c,k})\|_2$, where $k = 1, \ldots, K$. The frozen textual center of class c is computed as $\mu^{\mathrm{fr}}_c = K^{-1}\sum_{k=1}^{K} t^{\mathrm{fr}}_{c,k}$ and is further normalized when used as a class prototype.

Instead of compressing all prompt features into the single textual center μ^{fr}_c, we preserve their principal semantic variation directions. We first construct the centered text-feature matrix

$$A^{\mathrm{fr}}_c = \left[t^{\mathrm{fr}}_{c,1} - \mu^{\mathrm{fr}}_c,\ t^{\mathrm{fr}}_{c,2} - \mu^{\mathrm{fr}}_c,\ \ldots,\ t^{\mathrm{fr}}_{c,K} - \mu^{\mathrm{fr}}_c\right] \in \mathbb{R}^{d\times K}. \tag{1}$$

where d denotes the dimensionality of the CLIP embedding space. We perform singular value decomposition as $A^{\mathrm{fr}}_c = U^{\mathrm{fr}}_c \Sigma^{\mathrm{fr}}_c (V^{\mathrm{fr}}_c)^\top$ and retain the first r left singular vectors as the class-specific subspace basis $B^{\mathrm{fr}}_c = U^{\mathrm{fr}}_c[:, 1:r] \in \mathbb{R}^{d\times r}$. Consequently, class c is represented by the affine textual semantic subspace

$$\mathcal{S}^{\mathrm{fr}}_c = \left\{\mu^{\mathrm{fr}}_c + B^{\mathrm{fr}}_c \mathbf{a} \mid \mathbf{a} \in \mathbb{R}^r\right\}. \tag{2}$$

The center μ^{fr}_c captures central class semantics, while B^{fr}_c retains the principal variation directions among the prompt features. Singular values determine which directions are retained, whereas the subsequent orthogonal projection treats the selected directions uniformly. Our formulation models text-derived variation directions rather than a bounded prompt-density region, since the finite prompt bank is not used to estimate a calibrated distribution of unseen-domain variations.

3.3 Subspace Projection Similarity Classifier

Given an input image x, the frozen CLIP image encoder $f_v(\cdot)$ produces the normalized visual feature $z = f_v(x)/\|f_v(x)\|_2 \in \mathbb{R}^d$. We project z onto the textual semantic subspace of class c as

$$\pi_c(z)\Pi_{\mathcal{S}_c}(z) = \mu_c + B_c B_c^\top (z - \mu_c). \tag{3}$$

where $\mathcal{S}_c = \{\mu_c + B_c\mathbf{a} \mid \mathbf{a} \in \mathbb{R}^r\}$ is the learnable textual subspace used by the classifier.

Although the Euclidean projection error characterizes how well a textual subspace approximates a visual feature, CLIP performs image-text matching primarily through angular similarity between normalized features. We therefore do not directly use the Euclidean distance as a classification logit. Instead, we compute the cosine similarity between the image feature and its projection:

$$s_c^{\mathrm{sub}}(z) = \cos\left(z, \pi_c(z)\right) = \frac{z^\top \pi_c(z)}{\|z\|_2 \|\pi_c(z)\|_2}. \tag{4}$$

This projection similarity retains the geometric interpretation of the subspace representation while remaining consistent with the angular matching principle of CLIP.

The textual center μ_c retains the dominant class semantics in the CLIP text space. We therefore also compute the prototype similarity as $s_c^{\mathrm{proto}}(z) = z^\top \mu_c / (\|z\|_2 \|\mu_c\|_2)$. The classification logit for class c is defined as

$$\ell_c(z) = \tau \left(\alpha s_c^{\mathrm{proto}}(z) + \beta s_c^{\mathrm{sub}}(z) \right). \tag{5}$$

where τ is the logit scale, and α and β control the contributions of prototype similarity and subspace projection similarity, respectively.

The predictive probability is $p_\theta(y = c \mid x) = \exp(\ell_c(z)) / \sum_{j=1}^{C} \exp(\ell_j(z))$. Accordingly, the classification loss is $\mathcal{L}_{\mathrm{ce}} = -B^{-1} \sum_{i=1}^{B} \log p_\theta(y_i \mid x_i)$, where B denotes the mini-batch size.

Prototype similarity preserves the discriminative semantics of the class center, whereas subspace projection similarity measures the compatibility between a visual feature and the semantic variation directions of each class. The proposed classifier therefore makes more comprehensive use of the class-level semantic structure provided by multiple textual prompts.

3.4 Subspace Proximity and Adversarial Residual Invariance

For a visual feature z_i with label y_i, we decompose the feature relative to its ground-truth class subspace:

$$z_i = \pi_{y_i}(z_i) + r_{y_i}(z_i), \qquad r_{y_i}(z_i) = z_i - \pi_{y_i}(z_i). \tag{6}$$

Since the subspace basis is orthonormal, the residual is orthogonal to its learned variation directions. It measures what the current prompt-derived subspace does not represent, without requiring that this information be exclusively domain-specific or class-irrelevant. We encourage proximity to the ground-truth subspace through

$$\mathcal{L}_{\mathrm{res}} = \frac{1}{B} \sum_{i=1}^{B} \|z_i - \pi_{y_i}(z_i)\|_2^2 . \tag{7}$$

This weak constraint acts on the adapted visual features and the learnable textual subspaces, while the pretrained CLIP encoder parameters remain frozen. The ground-truth label selects the subspace for this training loss; classification still compares each feature against all classes.

Subspace proximity alone does not control how predictions change under nearby variations outside the learned directions. We therefore consider bounded perturbations orthogonal to the ground-truth subspace basis:

$$\Delta_{y_i}(\epsilon) = \left\{ \delta \in \mathbb{R}^d : B_{y_i}^\top \delta = 0,\ \|\delta\|_2 \le \epsilon \right\}, \qquad T_\delta(z_i) = \frac{z_i + \delta}{\|z_i + \delta\|_2}. \tag{8}$$

The perturbation is added to the full feature before normalization, after which the fused classifier recomputes logits for every class. Let $q_i = \mathrm{softmax}(\ell(z_i))$ and $\widetilde{q}_i(\delta) = \mathrm{softmax}(\ell(T_\delta(z_i)))$. We seek the admissible perturbation that changes the complete prediction distribution most:

$$\delta_i^\star \in \arg\max_{\delta \in \Delta_{y_i}(\epsilon)} D_{\mathrm{KL}} \left(\mathrm{sg}(q_i) \,\|\, \widetilde{q}_i(\delta) \right), \tag{9}$$

where sg stops gradients through the clean distribution. We approximate the maximization by projected gradient ascent from a random nonzero initialization, keeping the perturbation within $\Delta_{y_i}(\epsilon)$. Given the resulting $\widehat{\delta}_i$, we minimize

$$\mathcal{L}_{\mathrm{inv}} = \frac{1}{B} \sum_{i=1}^{B} D_{\mathrm{KL}} \left(\mathrm{sg}(q_i) \,\middle\|\, \widetilde{q}_i(\widehat{\delta}_i) \right). \tag{10}$$

Cross-entropy supplies label supervision, while this objective reduces the prediction change found by the residual-space attack. Thus, $\mathcal{L}_{\text{res}}$ encourages subspace proximity, and $\mathcal{L}_{\text{inv}}$ promotes local prediction stability beyond the represented directions.

3.5 Training Objective

During training, we freeze all pretrained CLIP image and text encoder parameters, while optimizing the learnable textual contexts, the text-conditioned deep visual prompts, and their cross-modal projection module. Consequently, the visual representation z_i remains adaptive through the visual prompting pathway. In addition to $\mathcal{L}_{\text{ce}}$, we introduce textual knowledge preservation and subspace geometry preservation constraints to prevent the learnable textual representations from drifting excessively from the original CLIP textual structure.

Let $t^{\text{fr}}_{c,k}$ and $t_{c,k}$ denote the text features obtained from the fixed and learnable prompts, respectively. The prompt-level textual knowledge preservation loss is

$$\mathcal{L}_{\text{text}} = \frac{1}{CK}\sum_{c=1}^{C}\sum_{k=1}^{K}\left[1-\cos\left(t_{c,k}, t^{\text{fr}}_{c,k}\right)\right]. \tag{11}$$

This loss preserves the correspondence between each learnable prompt and its original CLIP semantics. Let $(\mu^{\text{fr}}_c, B^{\text{fr}}_c)$ and (μ_c, B_c) denote the textual centers and subspace bases constructed from the frozen and learnable text features, respectively. Their projection matrices are $\mathbf{P}^{\text{fr}}_c = B^{\text{fr}}_c (B^{\text{fr}}_c)^\top$ and $\mathbf{P}_c = B_c B_c^\top$. The class-level subspace geometry preservation loss is

$$\mathcal{L}_{\text{sub}} = \frac{1}{C}\sum_{c=1}^{C}\left[\left\|\mathbf{P}_c - \mathbf{P}^{\text{fr}}_c\right\|_F^2 + 1 - \cos\left(\mu_c, \mu^{\text{fr}}_c\right)\right]. \tag{12}$$

Unlike $\mathcal{L}_{\text{text}}$, which preserves individual prompt features, $\mathcal{L}_{\text{sub}}$ preserves the class-level geometric structure jointly induced by multiple prompts.

The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \lambda_{\text{text}}\mathcal{L}_{\text{text}} + \lambda_{\text{sub}}\mathcal{L}_{\text{sub}} + \lambda_{\text{res}}\mathcal{L}_{\text{res}} + \lambda_{\text{inv}}\mathcal{L}_{\text{inv}},$$

where λ_{text}, λ_{sub}, λ_{res}, and λ_{inv} are the corresponding weighting coefficients.

3.6 METHOD-ALIGNED GEOMETRIC AND ROBUSTNESS ANALYSIS

We provide a method-aligned analysis to clarify the geometric and robustness implications of the proposed design. Specifically, we characterize (i) the effect of extending a class prototype to an affine textual subspace, (ii) the relation between projection similarity and CLIP's angular matching geometry, and (iii) the prediction stability induced by residual-space adversarial consistency and its connection to OOD risk. Formal statements and proofs are provided in Appendix B.

Geometric characterization. Because the class center belongs to the affine textual subspace, orthogonal projection onto the subspace cannot increase the Euclidean approximation error relative to the center alone. Moreover, projection similarity admits a direct angular interpretation consistent with CLIP's cosine-based matching. Together, these properties characterize the geometry of the proposed representation and scoring rule.

Prediction stability under residual perturbations. Define

$$\kappa_h(z, y; \epsilon) = \sup_{\delta \in \Delta_y(\epsilon)} D_{\text{KL}}\left(q_h(\cdot \mid z) \| q_h(\cdot \mid T_\delta(z))\right)$$

and

$$m_h(z, y) = q_h(y \mid z) - \max_{c \neq y} q_h(c \mid z).$$

Theorem B.3 shows that if

$$m_h(z, y) > \sqrt{2\kappa_h(z, y; \epsilon)},$$

then every perturbation in $\Delta_y(\epsilon)$ preserves the predicted class. Corollary B.4 further lifts this sample-level condition to a distribution-level residual-robust error decomposition.

OOD risk decomposition. Under the assumptions specified in Appendix B,

$$R_Q(h) \le \widehat{R}_P(h) + 2\widehat{\mathfrak{R}}_n(\Phi) + 3\sqrt{\frac{\log(2/\delta)}{2n}} + L_q\sqrt{2K_{P,\epsilon}(h)} + \frac{2L_z}{1-\epsilon}D_{S,\epsilon}(P,Q;h).$$

Here, $K_{P,\epsilon}(h)$ captures residual-space prediction sensitivity, while $D_{S,\epsilon}(P,Q;h)$ captures projection shift and residual shift not covered by the perturbation radius. This result provides an explanatory decomposition of target risk rather than a directly computable training objective.

4 Experiments

4.1 Datasets and protocol

Following the DomainBed experimental setting, we evaluate our method on five widely used domain generalization benchmarks. PACS(Li et al., 2017) contains 9,991 images from seven classes across four visual domains: Art Painting, Cartoon, Photo, and Sketch. VLCS(Torralba & Efros, 2011) consists of 10,729 images from five classes across four domains: VOC2007, LabelMe, Caltech101, and SUN09. OfficeHome(Venkateswara et al., 2017) contains 15,588 images from 65 classes across four domains: Art, Clipart, Product, and Real World. TerraIncognita(Beery et al., 2018) contains 24,788 images from 10 classes captured at four camera locations and exhibits substantial variations in natural environments. DomainNet(Peng et al., 2019a) contains 586,575 images from 345 classes across six visually diverse domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch.

We adopt the leave-one-domain-out evaluation protocol. For each dataset, one domain is treated as the unseen target domain, while the remaining domains are used as source domains for training. No target-domain data are accessible during training. We repeat each experiment using three random seeds and report the mean classification accuracy averaged over all target domains.

4.2 Implementation Details

We adopt CLIP ViT-B/16 as the backbone and freeze its pretrained image and text encoder parameters. Following MaPLe (Khattak et al., 2023), the learnable textual context is mapped to layer-specific deep visual prompts, inserted into each visual Transformer layer. Thus, the visual backbone remains frozen while its output features are adapted through trainable prompts. Both the textual context length and the number of visual prompt tokens are set to 16. For each class, 30 textual descriptions are used to construct a textual semantic subspace with a default rank of $r = 8$. The training objective includes cross-entropy, textual knowledge preservation, subspace geometry preservation, out-of-subspace residual regularization, and adversarial residual-space prediction invariance. We set $\lambda_{\text{text}} = \lambda_{\text{sub}} = 1.0$, $\lambda_{\text{res}} = 0.01$, and $\lambda_{\text{inv}} = 0.1$ for all datasets. The residual-space attack uses random initialization, three projected gradient-ascent steps with step size $\eta = 0.05$, and radius $\epsilon = 0.05$. We use Adam with a learning rate of 5×10^{-5}, zero weight decay, and training/test batch sizes of 32/128. The fusion coefficients are fixed to $\alpha = \beta = 1$, and all models are trained for $4{,}800$ optimization steps. All experiments are repeated over three random seeds, and we report the mean accuracy and standard deviation.

4.3 Comparison with State-of-the-Art Methods

We compare our method with representative CLIP-based approaches in Table 1, including CLIP(Radford et al., 2021), CoOp(Zhou et al., 2022b), CoCoOp(Zhou et al., 2022a), VPT(Jia et al., 2022), KgCoOp(Yao et al., 2023), MaPLe(Khattak et al., 2023), VL2V-SD(Addepalli et al., 2024), DPL(Zhang et al., 2023), GESTUR(Lew et al., 2023)and MLPD(Lyu et al., 2025). All methods use ViT-B/16 as the backbone and follow the DomainBed leave-one-domain-out protocol. Methods marked with * are reproduced under the same experimental protocol, while the remaining results are taken from their original publications. Average accuracy is computed over all five benchmarks.

Under the unified reproduction protocol, our method consistently outperforms all reproduced baselines across the five benchmarks. In particular, compared with MaPLe, our method improves the average accuracy from 75.3% to 77.9%.

For reference, our 77.9% average accuracy is numerically higher than the published 77.1% result of MLPD. Since MLPD is not reproduced in our implementation, this comparison serves as a protocol-

Table 1: Average classification accuracy (%) on five domain generalization benchmarks. All methods use ViT-B/16. Methods marked with * are reproduced under our protocol; the others are published results reported as protocol-level references.

Method	PACS	VLCS	OfficeHome	TerraInc.	DomainNet	Avg.
Reproduced under the same protocol						
CLIP*	96.1 ± 0.1	82.8 ± 0.1	82.2 ± 0.1	32.0 ± 0.1	56.9 ± 0.1	70.0
CoOp*	96.5 ± 0.1	81.4 ± 0.1	84.5 ± 0.2	45.5 ± 0.8	59.8 ± 0.1	73.5
VPT*	96.8 ± 0.2	83.1 ± 0.3	84.6 ± 0.2	55.0 ± 1.1	59.9 ± 0.1	75.9
KgCoOp*	97.0 ± 0.1	81.9 ± 0.3	85.0 ± 0.2	42.1 ± 0.3	60.1 ± 0.1	73.2
MaPLe*	96.7 ± 0.2	82.7 ± 0.3	83.4 ± 0.3	53.6 ± 0.9	60.1 ± 0.1	75.3
VL2V-SD*	95.6 ± 0.2	82.5 ± 0.2	84.8 ± 0.2	41.3 ± 0.1	57.5 ± 0.2	72.3
Reported in the original publications						
CoCoOp	96.0 ± 0.4	82.4 ± 0.2	83.1 ± 0.5	49.1 ± 1.6	59.7 ± 0.4	74.1
GESTUR	96.0 ± 0.0	82.8 ± 0.1	84.2 ± 0.1	55.7 ± 0.2	58.9 ± 0.1	75.5
DPL	97.3 ± 0.2	84.3 ± 0.4	84.2 ± 0.2	52.6 ± 0.6	56.7 ± 0.4	75.0
MLPD	97.3 ± 0.1	84.8 ± 0.3	85.2 ± 0.1	57.3 ± 0.3	60.7 ± 0.8	77.1
Ours	97.6 ± 0.1	85.9 ± 0.2	86.2 ± 0.1	58.7 ± 0.3	61.3 ± 0.4	77.9

level reference rather than an implementation-controlled comparison. Since our method leverages multiple class descriptions, we control for textual information in Table 3, where all multi-description variants use the same 30 descriptions per class and training setting.

4.4 Ablation Study

Contribution of auxiliary objectives. Table 2 evaluates the auxiliary objectives by removing them one at a time from the full model while retaining $\mathcal{L}_{\text{CE}}$ throughout. Removing any auxiliary objective reduces accuracy on both benchmarks. Excluding subspace geometry preservation causes the largest drop on PACS, by 1.1 percentage points, whereas excluding textual knowledge preservation has the greatest effect on OfficeHome, reducing accuracy by 1.4 points. The other removals lower accuracy by 0.6–0.9 points. The full model achieves 97.6% on PACS and 86.2% on OfficeHome, exceeding the classification-only baseline by 1.4 and 2.1 points, respectively. Thus, the final improvement is not attributable to a single auxiliary objective.

Table 2: Auxiliary-objective ablation on PACS and OfficeHome. Accuracy in percent, averaged over three runs.

Training objective	PACS	OH
$\mathcal{L}_{\text{CE}}$ only	96.2 ± 0.2	84.1 ± 0.1
Full $-\mathcal{L}_{\text{text}}$	96.8 ± 0.2	84.8 ± 0.2
Full $-\mathcal{L}_{\text{sub}}$	96.5 ± 0.1	85.5 ± 0.4
Full $-\mathcal{L}_{\text{res}}$	97.0 ± 0.3	85.3 ± 0.2
Full $-\mathcal{L}_{\text{inv}}$	96.7 ± 0.1	85.6 ± 0.1
Full model	97.6 ± 0.1	86.2 ± 0.1

Effectiveness of the textual semantic subspace. To disentangle the benefit of the proposed subspace representation from that of simply using multiple textual descriptions, we compare four class representations under a controlled setting. Prompt averaging, maximum similarity, and our textual semantic subspace use exactly the same 30 descriptions per class and the same MaPLe-based prompt adaptation. Because L_{sub}, L_{res}, and L_{inv} are specific to the subspace formulation, the three multi-description representations are compared under their shared training objective, $L_{\text{CE}} + \lambda_{\text{text}} L_{\text{text}}$, while the contributions of the subspace-specific objectives are evaluated separately in Table 2. The single-prototype variant serves as a one-prompt reference. Hence, Table 3 reports controlled rather than full-model results. As shown in Table 3, our subspace achieves 55.8%, 84.8%, and 85.2% on TerraIncognita, VLCS, and OfficeHome, respectively. Compared with maximum prompt similar-

Table 3: Controlled comparison of class representations on TerraIncognita, VLCS, and OfficeHome. All multi-description variants use the same 30 descriptions per class.

Method	TerraInc.	VLCS	OfficeHome
MaPLe with a single text prototype	53.9 ± 0.2	83.1 ± 0.1	83.6 ± 0.4
MaPLe with prompt averaging	54.2 ± 0.1	83.4 ± 0.3	84.0 ± 0.1
MaPLe with prompt maximum	54.7 ± 0.5	83.9 ± 0.2	84.3 ± 0.1
Ours with textual semantic subspace	55.8 ± 0.2	84.8 ± 0.1	85.2 ± 0.1

ity, it improves accuracy by 1.1, 0.9, and 0.9 percentage points. These results support the benefit of modeling variation directions among multiple descriptions rather than matching the descriptions independently.

Effect of class-specific textual geometry. We examine whether performance depends on the correspondence between classes and textual variation directions, rather than merely on the presence of low-rank matching directions. For the class-shuffled control, we retain each class center μ_c but replace its basis B_c with $B_{\pi(c)}$ from another class, where $\pi(c) \neq c$. The permutation is used during training and inference, with other settings unchanged. As shown in Table 4, shuffling the bases reduces accuracy by 3.39 and 2.45 percentage points on VLCS and OfficeHome, respectively. Since the rank, class centers, prompt bank, and training protocol remain fixed, this indicates that matching each class to its own textual variation directions matters beyond introducing low-rank directions alone.

Table 4: Effect of class-specific textual geometry on VLCS and OfficeHome. Mean accuracy and standard deviation (%) over three runs.

Subspace basis	VLCS	OfficeHome
Class-shuffled	82.51 ± 0.40	83.75 ± 0.20
Class-specific	$\mathbf{85.90 \pm 0.20}$	$\mathbf{86.20 \pm 0.10}$

Sensitivity to the subspace rank. We evaluate $r \in \{1, 2, 4, 8, 16\}$ on VLCS and OfficeHome while keeping the prompt bank and other training settings fixed. As shown in Figure 3, accuracy increases from 83.6% to 85.9% on VLCS and from 84.7% to 86.2% on OfficeHome as r grows from 1 to 8. Both datasets then show a 0.5-point decrease at $r = 16$. This shared trend is consistent with a moderate rank capturing useful prompt variation, whereas additional lower-variance directions may offer less discriminative information. We use $r = 8$ by default in the remaining experiments.

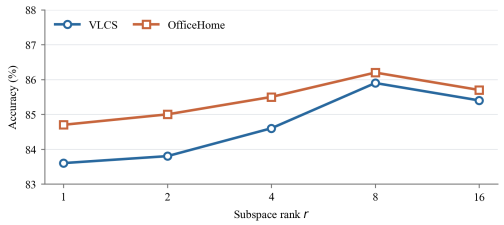

Figure 3: Sensitivity to the textual semantic subspace rank r on VLCS and OfficeHome.

5 Conclusion

We presented a textual semantic subspace learning approach for domain generalization. Our method represents each class as a low-rank subspace derived from the principal variation directions of multiple textual prompts and combines prototype similarity with subspace projection similarity for classification, thereby preserving intra-class semantic variations. We decompose visual features into textual-subspace projections and out-of-subspace residuals, and introduce textual knowledge preservation, subspace geometry preservation, residual regularization, and adversarial residual-space prediction invariance to improve robustness against bounded variations not directly represented by the textual subspaces. Our method-aligned theoretical analysis characterizes the geometry of the textual-subspace representation and projection-based scoring rule, and connects residual-space adversarial consistency to prediction stability and a projection–residual decomposition of OOD risk. Experiments and ablation studies across multiple domain generalization benchmarks demonstrate consistent improvements over CLIP-based prompt-learning methods, supporting textual semantic subspaces as an effective class-level representation for domain generalization.

REFERENCES

Sravanti Addepalli, Ashish Ramayee Asokan, Lakshay Sharma, and R. Venkatesh Babu. Leveraging vision-language models for improving domain generalization in image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23922–23932, 2024.

Shuanghao Bai, Yuedi Zhang, Wanqi Zhou, Zhirong Luan, and Badong Chen. Soft prompt generation for domain generalization. In *Computer Vision – ECCV 2024*, volume 15061, pp. 434–450, 2024.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Computer Vision – ECCV 2018*, volume 11220, pp. 472–489, 2018.

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

Junbum Cha, Kyungjae Lee, Sungrae Park, and Sanghyuk Chun. Domain generalization by mutual-information regularization with pre-trained models. In *Computer Vision – ECCV 2022*, volume 13683, pp. 440–457, 2022.

Haoran Chen, Xintong Han, Zuxuan Wu, and Yu-Gang Jiang. Multi-prompt alignment for multi-source unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Chuang Gan, Tianbao Yang, and Boqing Gong. Learning attributes equals multi-source domain generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 87–97, 2016.

Yaroslav Ganin and Victor S. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pp. 1180–1189, 2015.

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

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.

Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Fsdr: Frequency space domain randomization for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6891–6902, 2021.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge J. Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Computer Vision – ECCV 2022*, volume 13693, pp. 709–727, 2022.

Muhammad Uzair Khattak, Hanoona Abdul Rasheed, Muhammad Maaz, Salman H. Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19113–19122, 2023.

Byounggyu Lew, Donghyun Son, and Buru Chang. Gradient estimation for unseen domain risk minimization with pre-trained models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 4438–4448, 2023.

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. 5543–5551, 2017.

Hao Li, Yubin Xiao, Ke Liang, Mengzhu Wang, Long Lan, Kenli Li, and Xinwang Liu. Let synthetic data shine: Domain reassembly and soft-fusion for single domain generalization. *International Journal of Computer Vision*, 134(5):231, 2026.

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, 2018.

Lei Li, Ke Gao, Juan Cao, Ziyao Huang, Yepeng Weng, Xiaoyue Mi, Zhengze Yu, Xiaoya Li, and Boyang Xia. Progressive domain expansion network for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 224–233, 2021.

Huaihai Lyu, Hantao Yao, and Changsheng Xu. Multiple local prompts distillation for domain generalization. *IEEE Transactions on Multimedia*, 27:8555–8564, 2025.

Jovana Mitrovic, Brian McWilliams, Jacob C. Walker, Lars Holger Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. In *International Conference on Learning Representations*, 2021.

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, 2019a.

Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 5102–5112, 2019b.

Fengchun Qiao and Xi Peng. Uncertainty-guided model generalization to unseen domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6790–6800, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8748–8763, 2021.

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

Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1521–1528, 2011.

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. 5385–5394, 2017.

Mengzhu Wang, Jianlong Yuan, Qi Qian, Zhibin Wang, and Hao Li. Semantic data augmentation based distance metric learning for domain generalization. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 3214–3223, 2022.

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

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

Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6757–6767, 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.

Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto L. Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2100–2110, 2019.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 56(3):64:1–64:37, 2024a.

Jian Zhang, Lei Qi, Yinghuan Shi, and Yang Gao. Exploring flat minima for domain generalization with large learning rates. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6145–6158, 2024b.

Xin Zhang, Shixiang Shane Gu, Yutaka Matsuo, and Yusuke Iwasawa. Domain prompt learning for efficiently adapting clip to unseen domains. *Transactions of the Japanese Society for Artificial Intelligence*, 38(6):B–MC2_1, 2023.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16795–16804, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022b.

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.

Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jie Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3):257–276, 2023.

A Appendix

A.1 Construction of the Semantic Prompt Bank

Two-stage offline generation. We construct the semantic prompt pool using GPT-5.6 in two separate offline stages. In the first stage, GPT-5.6 is used to prepare the class-name set corresponding to the predefined label space of each benchmark. This stage produces class names only and does not generate textual descriptions.

In the second stage, the resulting class-name set is provided to GPT-5.6 in a separate generation request. The model receives only the class names and is not provided with the benchmark name, domain names, domain composition, held-out domain, or any other benchmark-specific metadata. For each class, GPT-5.6 generates a fixed pool of 40 descriptions covering visually recognizable characteristics, including overall appearance, shape, constituent parts, attributes, and common visual context. The main experiments use a fixed subset of $K = 30$ descriptions from this pool.

Effectiveness of subspace projection similarity. To evaluate the contribution of subspace projection similarity, we retrain three variants using prototype similarity, projection similarity, and their combination, respectively, while keeping the textual subspaces, training objectives, and other settings unchanged. As shown in Table 5, prototype similarity alone achieves 57.2% and 85.0% on TerraIncognita and OfficeHome, while projection similarity alone obtains 57.7% and 85.3%. Combining the two achieves 58.7% and 86.2%, improving upon prototype similarity alone by 1.5 and 1.2 percentage points, respectively. These results indicate that subspace projection similarity provides a complementary classification signal to class-center similarity.

Description-generation instruction. The description-generation request follows the same instruction for all datasets and classes:

Table 5: Ablation of similarity terms for classification. Results are average accuracies (%) over three runs.

Logit formulation	TerraIncognita	OfficeHome
Prototype similarity only	57.2 ± 0.3	85.0 ± 0.1
Projection similarity only	57.7 ± 0.2	85.3 ± 0.1
Prototype + projection similarity	58.7 ± 0.3	86.2 ± 0.1

> Given the following class names, generate exactly 40 distinct textual descriptions for each class. The descriptions should characterize visually recognizable properties, including overall appearance, shape, constituent parts, attributes, and common visual context. Do not mention any dataset, benchmark, domain, domain label, image style, or target environment. Return only the generated descriptions grouped by class name.

No dataset-specific generation rule or manually designed description template is used.

Fixed prompt bank and information boundaries. The 40-description pool is generated once and retained without manual selection, rewriting, ranking, or performance-based filtering. Its ordering is fixed before model training and remains unchanged across all leave-one-domain-out splits, random seeds, and model variants. The default model uses the first $K = 30$ descriptions from this fixed ordering. For the prompt-number sensitivity analysis, we use nested subsets from the same pool, such that $P_5 \subset P_{10} \subset P_{20} \subset P_{30} \subset P_{40}$.

Neither generation stage uses source- or target-domain images, sample-level annotations, target-domain identities, target-domain statistics, validation performance, or test results. In particular, the description-generation stage has access only to the class-name set and does not receive the benchmark name or domain information. The complete generation instruction and prompt banks are included in the supplementary material to support reproducibility.

Construction of class-wise textual subspaces. Let $\mathcal{P}_c = \{p_{c,k}\}_{k=1}^{K}$ denote the fixed prompt bank for class c, where $K = 30$. Each description is independently encoded by the frozen CLIP text encoder:

$$t_{c,k}^{\mathrm{fr}} = \frac{f_t(p_{c,k})}{\|f_t(p_{c,k})\|_2}. \tag{13}$$

The class center is computed as $\mu_c^{\mathrm{fr}} = K^{-1} \sum_{k=1}^{K} t_{c,k}^{\mathrm{fr}}$. We center the textual features and apply singular value decomposition to extract the leading r variation directions, yielding the class-wise affine textual semantic subspace $\mathcal{S}_c^{\mathrm{fr}} = \mu_c^{\mathrm{fr}} + \mathrm{span}(B_c^{\mathrm{fr}})$.

A.2 Hyperparameter and Model Selection Protocol

All hyperparameter tuning and model selection are performed exclusively using the source-domain data. No target-domain samples, labels, or performance statistics are used during training, hyperparameter selection, or checkpoint selection.

Specifically, the number of descriptions K, the subspace rank r, and loss weights λ_{text}, λ_{sub}, λ_{res}, and λ_{inv} are selected based on source-domain validation performance. Once determined, the same configuration is kept fixed across all benchmarks without dataset-specific or target-specific tuning. Model checkpoints are likewise selected according to source-domain validation accuracy.

The held-out target domain is used only for final evaluation. The sensitivity analyses reported in the paper, such as the study of the subspace rank r, are conducted post hoc to analyze the robustness of the method and are not used for hyperparameter or model selection.

A.3 SENSITIVITY TO THE NUMBER OF PROMPTS.

We study the sensitivity to the number of prompts per class by varying $K \in \{5, 10, 20, 30, 40\}$ on VLCS and OfficeHome. All variants use nested subsets of the same fixed 40-description pool described in Appendix A.1, following its predetermined ordering, i.e., $P_5 \subset P_{10} \subset P_{20} \subset P_{30} \subset P_{40}$. The default value $K = 30$ is selected exclusively using source-domain validation and is fixed before evaluating any unseen target domain. Following our implementation, the effective subspace rank is $r_K = \min(r, K - 1, D)$.

As shown in Figure 4, performance consistently improves as K increases from 5 to 30 on both datasets, suggesting that additional descriptions provide more comprehensive class-level semantic variation. Increasing K further to 40 leads to a slight decrease, suggesting diminishing returns from additional prompts. Importantly, these target-domain results are reported only as a post-hoc sensitivity analysis and are not used for hyperparameter or model selection.

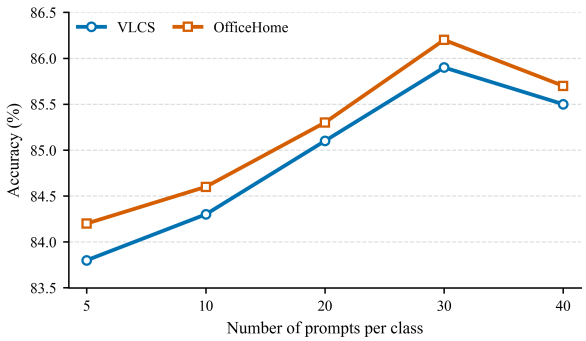

Figure 4: Sensitivity to the number of prompts per class on VLCS and OfficeHome. Results are mean accuracies over three runs.

A.4 BASELINE IMPLEMENTATION AND COMPARISON PROTOCOL.

Methods marked with * are reproduced using the same DomainBed-based codebase and evaluation pipeline as our method, with only method-specific components replaced. Results of methods not included in our unified reproduction, such as DPL and MLPD, are taken from their original publications. For MLPD, no official implementation was available at the time of our experiments. Its reported setup closely matches ours in the DomainBed framework, ViT-B/16 backbone, leave-one-domain-out protocol, source-domain model selection, and three-run evaluation. We therefore report its published results as protocol-level references without claiming implementation-level identity. The five-dataset averages are recomputed from the reported per-dataset accuracies.

A.5 SENSITIVITY TO THE NUMBER OF INNER OPTIMIZATION STEPS.

We study the sensitivity to the number of projected gradient-ascent steps used in the residual-space attack by varying $T \in \{1, 3, 5, 10\}$ on OfficeHome. The default setting uses $T = 3$ steps, while all other training and attack configurations are kept unchanged.

As shown in Figure 5, performance remains stable across the tested range of inner optimization steps, with only minor variations as the number of steps increases. The default three-step configuration achieves the highest accuracy, while further increasing the number of inner steps yields no further improvement in this sensitivity analysis. Overall, these results indicate that the proposed training objective is relatively insensitive to the exact number of inner optimization steps.

B SUPPORTING GEOMETRIC AND ROBUSTNESS ANALYSIS

This appendix provides method-aligned theoretical characterizations of three central components of our approach: class-wise textual semantic subspaces, projection-similarity classification, and adversarial prediction consistency under out-of-subspace perturbations. The analysis follows a progressive structure. Propositions B.1–B.2 characterize the geometry of the textual-subspace representation and its scoring rule. Theorem B.3 and Corollary B.4 connect the residual-space KL objective to sample-level prediction stability and distribution-level residual robustness. Finally, Theorem B.6 places this residual sensitivity within a projection–residual decomposition of OOD risk.

Throughout the analysis, h denotes the trained classifier, and the learned centers and bases $\{(\mu_c, B_c)\}_{c=1}^{C}$ are treated as fixed. The analysis is aligned with the implemented classifier and

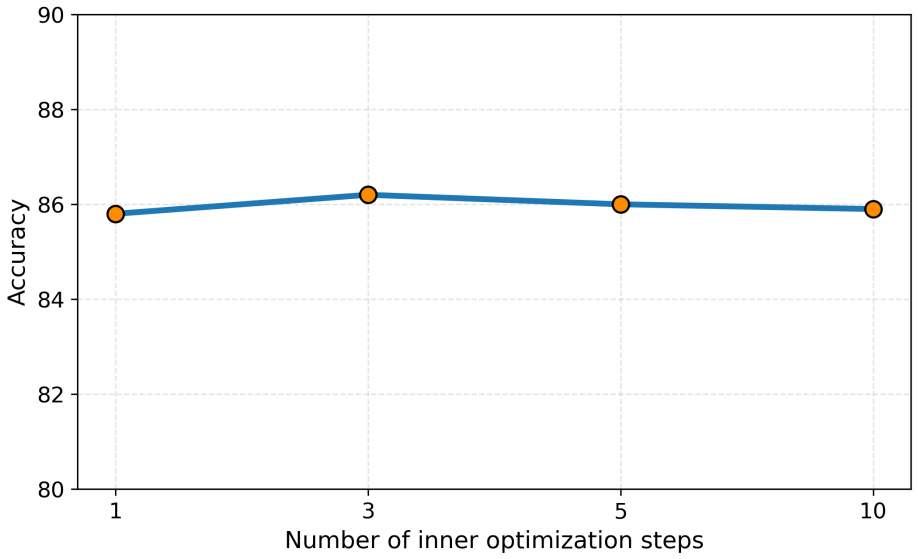

Figure 5: Sensitivity to the number of inner optimization steps on OfficeHome.

KL-based residual perturbation objective, independently of the numerical procedure used to obtain these parameters.

B.1 Notation and Implemented Objective

Let $z \in \mathbb{R}^d$ denote a normalized CLIP image feature. For class c, let $\{t_{c,k}\}_{k=1}^K$ denote the learned textual prompt features used by the final classifier. The class center and the corresponding rank-r affine textual semantic subspace are defined as

$$\mu_c = \frac{1}{K}\sum_{k=1}^{K} t_{c,k}, \tag{14}$$

$$\mathcal{S}_c = \{\mu_c + B_c\mathbf{a} : \mathbf{a} \in \mathbb{R}^r\}, \tag{15}$$

where $B_c \in \mathbb{R}^{d\times r}$ contains the leading left singular vectors of the centered prompt-feature matrix $A_c = [t_{c,1} - \mu_c, \ldots, t_{c,K} - \mu_c] \in \mathbb{R}^{d\times K}$, following the convention in Section 3.2. Since the implementation applies QR orthogonalization after singular value decomposition, $B_c^\top B_c = I_r$.

The projection of z onto $\mathcal{S}_c$ and the corresponding residual are

$$\pi_c(z)\Pi_{\mathcal{S}_c}(z) = \mu_c + B_c B_c^\top (z - \mu_c), \tag{16}$$

$$r_c(z) = z - \pi_c(z). \tag{17}$$

By construction, $B_c^\top r_c(z) = 0$.

Let $\mathcal{N}(v) = v/\|v\|_2$ for $v \neq 0$. The implemented classifier combines prototype similarity and subspace projection similarity:

$$\ell_c(z) = \tau\left[\alpha\left\langle\mathcal{N}(z), \mathcal{N}(\mu_c)\right\rangle + \beta\left\langle\mathcal{N}(z), \mathcal{N}(\pi_c(z))\right\rangle\right], \tag{18}$$

where $\tau > 0$ is the logit scale and $\alpha, \beta \geq 0$ are the fusion weights. The resulting predictive distribution is

$$q_h(\cdot \mid z) = \mathrm{softmax}(\ell(z)), \tag{19}$$

where h denotes the complete textual-subspace classifier.

For a training sample with label y, define the orthogonal-complement projector

$$\mathbf{P}_y^\perp = I - B_y B_y^\top \tag{20}$$

and the admissible residual perturbation set

$$\Delta_y(\epsilon) = \left\{\delta \in \mathbb{R}^d : B_y^\top \delta = 0,\ \|\delta\|_2 \le \epsilon\right\}, \qquad 0 < \epsilon < 1. \tag{21}$$

The perturbed feature evaluated by the classifier is

$$T_\delta(z) = \mathcal{N}(z + \delta). \tag{22}$$

Therefore, the perturbation is constrained to the residual space before feature renormalization, consistently with the implementation.

The worst-case predictive discrepancy is defined as

$$\kappa_h(z, y; \epsilon) = \sup_{\delta \in \Delta_y(\epsilon)} D_{\mathrm{KL}}\left(q_h(\cdot \mid z) \,\|\, q_h(\cdot \mid T_\delta(z))\right). \tag{23}$$

The implementation approximates this supremum using projected gradient ascent with a random nonzero initialization. Specifically,

$$g^{(t)} = \mathbf{P}_y^\perp \nabla_\delta D_{\mathrm{KL}}\left(q_h(\cdot \mid z) \,\|\, q_h(\cdot \mid T_{\delta^{(t)}}(z))\right), \tag{24}$$

$$\delta^{(t+1)} = \Pi_{\Delta_y(\epsilon)}\left(\delta^{(t)} + \eta \frac{g^{(t)}}{\|g^{(t)}\|_2 + \nu}\right), \tag{25}$$

where η is the ascent step size and $\nu > 0$ is a numerical constant.

Let $\widehat{\delta}_i$ denote the perturbation obtained by this inner optimization. The implemented invariance objective is

$$\mathcal{L}_{\mathrm{inv}} = \frac{1}{n}\sum_{i=1}^{n} D_{\mathrm{KL}}\left(\mathrm{sg}\left[q_h(\cdot \mid z_i)\right] \,\middle\|\, q_h(\cdot \mid T_{\widehat{\delta}_i}(z_i))\right), \tag{26}$$

where sg denotes stop-gradient. Both distributions are computed using the complete fused classifier in Equation equation 18.

B.2 GEOMETRIC CHARACTERIZATION OF TEXTUAL SUBSPACES

Proposition B.1 (Projection-error decomposition)**.** *For any feature z and class c,*

$$\|z - \mu_c\|_2^2 = \|z - \pi_c(z)\|_2^2 + \|B_c^\top (z - \mu_c)\|_2^2.$$

Consequently,

$$\|z - \pi_c(z)\|_2^2 \le \|z - \mu_c\|_2^2,$$

with strict inequality whenever $B_c^\top (z - \mu_c) \neq 0$.

Proof. Since $B_c^\top B_c = I_r$, $B_c B_c^\top$ and $I - B_c B_c^\top$ are mutually orthogonal projectors. We have

$$z - \mu_c = B_c B_c^\top (z - \mu_c) + (I - B_c B_c^\top)(z - \mu_c).$$

Moreover,

$$\pi_c(z) - \mu_c = B_c B_c^\top (z - \mu_c),$$
$$z - \pi_c(z) = (I - B_c B_c^\top)(z - \mu_c).$$

The two components are orthogonal. Therefore, the Pythagorean theorem gives

$$\|z - \mu_c\|_2^2 = \|B_c B_c^\top (z - \mu_c)\|_2^2 + \|z - \pi_c(z)\|_2^2$$
$$= \|B_c^\top (z - \mu_c)\|_2^2 + \|z - \pi_c(z)\|_2^2.$$

Dropping the first nonnegative term proves the inequality. The inequality is strict whenever this term is nonzero. □

Proposition B.1 makes explicit the geometric effect of extending a single class center to an affine textual subspace containing that center. The center-based error decomposes into the residual projection error and the component explained by the learned textual variation directions. Thus, the subspace preserves the class center while explicitly representing additional directions of variation.

B.3 ANGULAR INTERPRETATION OF PROJECTION SIMILARITY

Proposition B.2 (Projection similarity equals angular agreement)**.** *Suppose that $z \neq 0$ and $\pi_c(z) \neq 0$, and define*

$$\rho_c(z) = \|N(z) - N(\pi_c(z))\|_2^2.$$

Then

$$\langle N(z), N(\pi_c(z))\rangle = 1 - \frac{1}{2}\rho_c(z).$$

Proof. Since both normalized vectors have unit norm,

$$\begin{aligned}\rho_c(z) &= \|N(z)\|_2^2 + \|N(\pi_c(z))\|_2^2 - 2\langle N(z), N(\pi_c(z))\rangle \\ &= 2 - 2\langle N(z), N(\pi_c(z))\rangle.\end{aligned}$$

Rearranging proves the result. □

This result shows that maximizing projection similarity is equivalent to minimizing the angular discrepancy between an image feature and its class-specific textual-subspace projection. It therefore retains the angular matching principle used by CLIP without treating Euclidean projection error as the classification score. Together, Propositions B.1 and B.2 characterize the geometry of the proposed representation and scoring rule: the affine subspace represents text-derived variation around the class center, while projection similarity evaluates compatibility with this structure in CLIP's normalized angular geometry.

B.4 PREDICTION STABILITY UNDER RESIDUAL PERTURBATIONS

For a labeled sample (z, y), define the clean probability margin as

$$m_h(z, y) = q_h(y \mid z) - \max_{c \neq y} q_h(c \mid z). \tag{27}$$

Theorem B.3 (Adversarial KL consistency controls prediction changes)**.** *For any (z, y) satisfying $m_h(z, y) > 0$, if*

$$m_h(z, y) > \sqrt{2\kappa_h(z, y; \epsilon)}, \tag{28}$$

then every $\delta \in \Delta_y(\epsilon)$ preserves the prediction:

$$\arg\max_c q_h(c \mid T_\delta(z)) = y. \tag{29}$$

Proof. Fix $\delta \in \Delta_y(\epsilon)$ and let $q = q_h(\cdot \mid z)$ and $\widetilde{q} = q_h(\cdot \mid T_\delta(z))$. Pinsker's inequality gives

$$\mathrm{TV}(q, \widetilde{q}) \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}(q \| \widetilde{q})} \leq \sqrt{\frac{1}{2}\kappa_h(z, y; \epsilon)}. \tag{30}$$

For every class c, $|q(c) - \widetilde{q}(c)| \leq \mathrm{TV}(q, \widetilde{q})$. Thus, for every $c \neq y$,

$$\begin{aligned}\widetilde{q}(y) - \widetilde{q}(c) &\geq q(y) - q(c) - |\widetilde{q}(y) - q(y)| - |\widetilde{q}(c) - q(c)| && (31)\\ &\geq m_h(z, y) - 2\,\mathrm{TV}(q, \widetilde{q}) && (32)\\ &\geq m_h(z, y) - \sqrt{2\kappa_h(z, y; \epsilon)}. && (33)\end{aligned}$$

Under Equation equation 28, this quantity is positive for every competing class. Hence the perturbed prediction remains y. □

This theorem corresponds directly to the latest implementation: cross-entropy encourages a positive clean margin, while $\mathcal{L}_{\mathrm{inv}}$ reduces the KL change produced by the residual attack.

Corollary B.4 (Residual-robust error decomposition)**.** *Define*

$$R_P^{01}(h) = \Pr_P\left[\arg\max_c q_h(c \mid z) \neq y\right], \tag{34}$$

$$R_{P,\epsilon}^{\mathrm{rob}}(h) = \Pr_P\left[\exists \delta \in \Delta_y(\epsilon) : \arg\max_c q_h(c \mid T_\delta(z)) \neq y\right]. \tag{35}$$

For every $\gamma > 0$,

$$R_{P,\epsilon}^{\mathrm{rob}}(h) \leq R_P^{01}(h) + \Pr_P\left(0 \leq m_h(z, y) \leq \gamma\right) + \frac{2}{\gamma^2}\mathbb{E}_P\left[\kappa_h(z, y; \epsilon)\right]. \tag{36}$$

Proof. A residual-robust error occurs if the clean prediction is wrong, if it is correct but has margin at most γ, or if an admissible perturbation changes a correct prediction whose margin exceeds γ. By Theorem B.3, the last event requires $\kappa_h(z, y; \epsilon) \geq \gamma^2/2$. Markov's inequality gives

$$\Pr_P\left[\kappa_h(z, y; \epsilon) \geq \frac{\gamma^2}{2}\right] \leq \frac{2}{\gamma^2}\mathbb{E}_P\left[\kappa_h(z, y; \epsilon)\right]. \tag{37}$$

Combining the three events proves the result. □

B.5 PROJECTION–RESIDUAL OOD RISK DECOMPOSITION

Let P and Q denote source- and target-domain distributions over normalized features and labels. For clarity, assume that P and Q share the same class prior. If their class priors differ, an additional label-shift term $\mathrm{TV}(P_Y, Q_Y)$ is required.

For class y, let $\Gamma(P_y, Q_y)$ be the set of couplings between $P(z \mid y)$ and $Q(z \mid y)$. Define

$$\mathcal{D}_{\mathcal{S},\epsilon}(P, Q; h) = \sum_{y=1}^{C} P(y) \inf_{\xi_y \in \Gamma(P_y, Q_y)} \mathbb{E}_{(z_P, z_Q)\sim\xi_y}\Big[\|\pi_y(z_Q) - \pi_y(z_P)\|_2 + \big(\|r_y(z_Q) - r_y(z_P)\|_2 - \epsilon\big)_+\Big], \tag{38}$$

where $(a)_+ = \max(a, 0)$. The first term measures class-conditional shift between textual-subspace projections, whereas the second measures the out-of-subspace shift not covered by the perturbation radius.

Let $\mathcal{J}(q, y) \in [0, 1]$ be a bounded classification loss and define $\phi_h(z, y) = \mathcal{J}(q_h(\cdot \mid z), y)$. Assume that

$$|\mathcal{J}(q, y) - \mathcal{J}(q', y)| \leq L_q\|q - q'\|_1, \tag{39}$$

$$|\phi_h(z, y) - \phi_h(z', y)| \leq L_z\|z - z'\|_2 \tag{40}$$

for normalized z and z'. Define

$$R_D(h) = \mathbb{E}_{(z,y)\sim D}[\phi_h(z, y)], \tag{41}$$

$$\mathcal{K}_{P,\epsilon}(h) = \mathbb{E}_{(z,y)\sim P}[\kappa_h(z, y; \epsilon)]. \tag{42}$$

Let $\Phi = \{(z, y) \mapsto \phi_h(z, y) : h \in \mathcal{H}\}$ be the loss class induced by the complete classifier.

Lemma B.5 (Lipschitz continuity of normalization)**.** *For any nonzero vectors a and b,*

$$\|\mathcal{N}(a) - \mathcal{N}(b)\|_2 \leq \frac{2\|a - b\|_2}{\min(\|a\|_2, \|b\|_2)}. \tag{43}$$

Proof. Using the triangle inequality,

$$\left\|\frac{a}{\|a\|_2} - \frac{b}{\|b\|_2}\right\|_2 \leq \frac{\|a - b\|_2}{\|a\|_2} + \|b\|_2\left|\frac{1}{\|a\|_2} - \frac{1}{\|b\|_2}\right| \tag{44}$$

$$\leq \frac{2\|a - b\|_2}{\|a\|_2}, \tag{45}$$

where the reverse triangle inequality is used in the second step. Exchanging a and b and taking the tighter bound proves Equation equation 43. □

Theorem B.6 (OOD risk bound under projection–residual shift)**.** *Let $0 < \epsilon < 1$ and suppose the preceding assumptions hold. For an i.i.d. source sample of size n, with probability at least $1 - \delta$, every $h \in \mathcal{H}$ satisfies*

$$R_Q(h) \leq \widehat{R}_P(h) + 2\widehat{\mathfrak{R}}_n(\Phi) + 3\sqrt{\frac{\log(2/\delta)}{2n}} + L_q\sqrt{2\mathcal{K}_{P,\epsilon}(h)} + \frac{2L_z}{1 - \epsilon}\mathcal{D}_{\mathcal{S},\epsilon}(P, Q; h). \tag{46}$$

Proof. Fix a class y and a coupling $\xi_y \in \Gamma(P_y, Q_y)$. For a coupled pair (z_P, z_Q), write

$$z_P = \pi_y(z_P) + r_y(z_P), \tag{47}$$

$$z_Q = \pi_y(z_Q) + r_y(z_Q). \tag{48}$$

Let $\Delta r = r_y(z_Q) - r_y(z_P)$ and define

$$\delta_{P,Q} = \begin{cases} \min\left(1, \dfrac{\epsilon}{\|\Delta r\|_2}\right) \Delta r, & \Delta r \neq 0, \\ 0, & \Delta r = 0. \end{cases} \tag{49}$$

Since both residuals are orthogonal to B_y, $B_y^\top \Delta r = 0$. Therefore, $\delta_{P,Q} \in \Delta_y(\epsilon)$. Furthermore,

$$\|z_Q - (z_P + \delta_{P,Q})\|_2 \leq \|\pi_y(z_Q) - \pi_y(z_P)\|_2 \tag{50}$$

$$+ \big(\|r_y(z_Q) - r_y(z_P)\|_2 - \epsilon\big)_+. \tag{51}$$

Because $\|z_P\|_2 = \|z_Q\|_2 = 1$ and $\|\delta_{P,Q}\|_2 \leq \epsilon$, we have $\|z_P + \delta_{P,Q}\|_2 \geq 1 - \epsilon$. Applying Lemma B.5 gives

$$\|z_Q - T_{\delta_{P,Q}}(z_P)\|_2 \leq \frac{2}{1-\epsilon}\Big[\|\pi_y(z_Q) - \pi_y(z_P)\|_2 \tag{52}$$

$$+ \big(\|r_y(z_Q) - r_y(z_P)\|_2 - \epsilon\big)_+\Big]. \tag{53}$$

Using the Lipschitz assumptions and Pinsker's inequality,

$$\phi_h(z_Q, y) \leq \phi_h(T_{\delta_{P,Q}}(z_P), y) + L_z\|z_Q - T_{\delta_{P,Q}}(z_P)\|_2 \tag{54}$$

$$\leq \phi_h(z_P, y) + L_q \left\|q_h(\cdot \mid T_{\delta_{P,Q}}(z_P)) - q_h(\cdot \mid z_P)\right\|_1 \tag{55}$$

$$+ L_z\|z_Q - T_{\delta_{P,Q}}(z_P)\|_2 \tag{56}$$

$$\leq \phi_h(z_P, y) + L_q\sqrt{2\kappa_h(z_P, y; \epsilon)} \tag{57}$$

$$+ \frac{2L_z}{1-\epsilon}\Big[\|\pi_y(z_Q) - \pi_y(z_P)\|_2 \tag{58}$$

$$+ \big(\|r_y(z_Q) - r_y(z_P)\|_2 - \epsilon\big)_+\Big]. \tag{59}$$

Taking expectations, applying Jensen's inequality, and taking the infimum over the class-conditional couplings yields

$$R_Q(h) \leq R_P(h) + L_q\sqrt{2\mathcal{K}_{P,\epsilon}(h)} + \frac{2L_z}{1-\epsilon}\mathcal{D}_{\mathcal{S},\epsilon}(P, Q; h). \tag{60}$$

Since every function in Φ takes values in $[0, 1]$, the standard uniform Rademacher bound gives

$$R_P(h) \leq \widehat{R}_P(h) + 2\widehat{\mathfrak{R}}_n(\Phi) + 3\sqrt{\frac{\log(2/\delta)}{2n}} \tag{61}$$

with probability at least $1 - \delta$. Substitution into Equation equation 60 completes the proof. □

Theorem B.6 provides an explanatory decomposition of target risk into empirical source risk, statistical complexity, worst-case prediction variation within the residual perturbation radius, and projection plus uncovered residual shift between domains. Here, $K_{P,\epsilon}(h)$ directly captures the residual-space predictive sensitivity controlled by the KL invariance objective, whereas $D_{S,\epsilon}(P, Q; h)$ accounts for source–target variation not absorbed by this local robustness region. The result therefore connects the preceding stability analysis to domain generalization through a projection–residual view of distribution shift.

Scope of the analysis. The geometric results characterize the proposed representation and scoring rule, while the stability and OOD results characterize the prediction behavior of the resulting classifier. These results are not intended as a direct guarantee of lower classification risk. In particular, the OOD decomposition conditions on the trained classifier and contains the target-dependent discrepancy $D_{S,\epsilon}(P, Q; h)$; it therefore serves as an explanatory decomposition rather than a quantity directly computable during domain-generalization training. Moreover, Equation (23) uses an exact supremum, whereas the implemented objective employs a finite-step projected attack. Accordingly, the stability statements should not be interpreted as certified robustness guarantees for the finite-step optimizer.

AI USE STATEMENT

Large language models were used to support non-technical aspects of manuscript preparation, including translation, linguistic refinement, structural organization, and LaTeX formatting, as well as the offline generation of textual descriptions used in the experiments. They were not used to develop or implement the proposed method, generate or preprocess datasets, perform model training or result analysis, or perform mathematical derivations and proofs. All AI-assisted content was carefully reviewed and revised by the authors. The authors take full responsibility for the correctness, scientific validity, and integrity of the final manuscript, including all claims, analyses, and experimental results.