LATENT-GEOMETRY-AWARE TRAINING FOR DOMAIN GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning-based decision-making models tend to underutilize information from less frequent scenarios, often resulting in poor domain generalization. Most existing domain generalization methods focus on learning domain-invariant representations, typically assuming access to domain information during training. In this work, we develop a two-phase training strategy for generic classifiers that enhances domain generalization *without* requiring explicit domain information. The key idea is to first (*i*) jointly learn a feature extractor and a low-dimensional affine subspace that minimizes within-class latent dispersion while maximizing between-class latent dispersion, and then (*ii*) further fine-tune the model by minimizing within-class logit dispersion while maximizing classification accuracy. Experiments on the ColoredMNIST, Waterbirds, and UrbanCars benchmarks show that the proposed training approach, which requires no domain information during training, achieves domain-generalization performance comparable to state-of-the-art methods that leverage domain information during training.

1 INTRODUCTION

Domain generalization (DG) aims to learn models that generalize to unseen domains, where the distribution of domain-dependent input variations may differ from those observed during training (Wang et al., 2022; Gulrajani & Lopez-Paz, 2021). Under such distribution shifts, a learned model may exhibit substantially degraded performance on certain domains or groups. Hence, it is important to improve worst-group accuracy without sacrificing overall predictive accuracy (Sagawa et al., 2020; Li et al., 2023). Standard empirical risk minimization (ERM) minimizes the average empirical loss over the training samples (Vapnik, 1991), but does not explicitly optimize accuracy on less frequent groups. Many existing domain-generalization methods address this limitation using environment-, domain-, or group-dependent objectives, for example by enforcing invariant predictors across environments or penalizing variation in environment-wise risks (Arjovsky et al., 2019; Krueger et al., 2021). Such approaches hence generally require the training data to be organized according to explicit environment, domain, or group assignments. In practice, however, these assignments may be unavailable, costly to obtain, or difficult to define when the training data contain a mixture of latent domains (Matsuura & Harada, 2020; Zhang et al., 2021; Wang et al., 2024). Moreover, when multiple spurious factors coexist, a single domain partition may not directly identify the particular input variation that should be made irrelevant to the prediction (Li et al., 2023). This motivates us to pursue the *same* domain-generalization objective *without* relying on explicit domain or group labels.

Recent work has explored DG without explicit domain labels through different mechanisms, including discovering latent domains, aligning class-conditioned predictions, or encouraging prediction stability across subsets of the pooled training data (Matsuura & Harada, 2020; Zhang et al., 2021; Wang et al., 2024). In this work, we take a different approach by assuming access to label-preserving matched sets constructed from the available training data instead of inferring or using domain assignments. Each matched set contains samples that share the same class label while differing in a variation that should not affect the model decision. These matched sets therefore specify the variation of interest directly without assigning every training sample to a domain or group.

The remaining question is how this specified variation should be organized in the learned representation. Pair- or correspondence-based objectives can reduce the sensitivity of the model to variations

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.

that should not affect the prediction or encourage corresponding samples to receive similar representations (Mahajan et al., 2021; Surner et al., 2026). However, consistency between matched samples does not by itself require the corresponding variation to occupy an identifiable and removable subspace in the latent representation. This distinction is important because, in an ERM representation, variation within matched sets can overlap with variation that separates different matched-set centers. Removing the former directly can therefore also remove latent structure used for classification. Representation-erasure methods such as INLP and LEACE explicitly remove specified information from an already learned representation, but operate on the geometry produced by the preceding training procedure (Ravfogel et al., 2020; Belrose et al., 2023). That is, if the specified variation is distributed across multiple directions or overlaps with task-relevant structure, direct information removal based on the fixed representation can either leave residual dependence or remove useful information.

We address this problem by *jointly* training a feature extractor, together with a representation eraser. Rather than directly forcing matched representations to coincide or removing specified information from a fixed representation, we first reorganize the latent space so that the within-class dispersion and between-class dispersion can be appropriately separated. Once this structure is obtained, the specified variation can be removed by an explicit reference-centered affine projection. A separate post-intervention stage then adapts the classifier while constraining the quantities that preserve the intended invariance and class separation.

We implement this idea using a two-phase training pipeline. In Phase 1, we first train the feature extractor and classifier head using standard ERM. Then, the feature extractor is fine-tuned jointly with the orthogonal low-dimension projector using a new fine-tuning loss designed based on latent-level dispersions. Then, in Phase 2, we fine-tune the remaining classification head using another new fine-tuning loss designed based on logit-level dispersion as well as classification accuracy.

Although individual components of the proposed framework bear strong resemblance to prior work – for example, orthogonal projection for eliminating harmful dispersion has been extensively studied in Phase 1 (Belrose et al., 2023), while Wang et al. (2024) theoretically showed that reducing predictive-level dispersion can improve domain generalization, which is also the key idea underlying Phase 2 – their non-trivial integration and modifications enable the proposed framework to achieve performance comparable to, and in some cases surpassing, that of existing DG methods that leverage domain information, despite requiring no domain data.

We evaluate the proposed training pipeline on ColoredMNIST, Waterbirds, and UrbanCars (Arjovsky et al., 2019; Sagawa et al., 2020; Li et al., 2023). Across the three benchmarks, the method improves worst-group accuracy while maintaining competitive overall predictive accuracy. On UrbanCars, where background and co-occurring objects vary independently, a single joint specification reduces prediction sensitivity to each factor and to their simultaneous variation. The matched-data study further shows that the method reaches a predefined worst-group accuracy threshold with fewer matched sets than ERM. The ablation studies isolate the role of each stage: applying the affine projection directly to the ERM representation does not reproduce the Phase 1 endpoint, removing the Phase 1 preservation mechanisms produces distinct failures, and removing the Phase 2 constraints allows the decision-space quantities to exceed their prescribed tolerances.

Our contributions are:

- We formulate DG without explicit domain or group labels as a two-phase training pipeline that improves worst-group accuracy while preserving overall predictive accuracy using label-preserving matched sets instead of domain assignments.
- We introduce a Phase 1 objective that reduces within-class latent dispersion outside a removable subspace while increasing between-class latent dispersion inside the remaining subspace.
- We introduce a Phase 2 objective that reduces within-class logit dispersion while maximizing classification accuracy.
- We evaluate the resulting training pipeline on ColoredMNIST, Waterbirds, and UrbanCars, including multiple simultaneously varying specified factors, matched-data efficiency, and ablations of the Phase 1 and Phase 2 mechanisms.

2 METHOD

2.1 OVERALL GOAL

Consider a generic classifier that consists of *feature extractor* $f_\theta : \mathcal{X} \to \mathbb{R}^d$ parametrized by $\theta \in \Theta$ and *classification head* $h_\phi : \mathbb{R}^d \to \mathbb{R}^C$ parametrized by $\phi \in \Phi$ where C is the number of candidate classes. Given an input $X \in \mathcal{X}$, the feature extractor yields a d-dimensional latent vector $Z := f_\theta(X)$. Then, using this latent vector, the classifier head yields a C-dimensional logit vector $\ell := h_\phi(Z)$, which then passes through a softmax operator to form a likelihood vector $g_{\theta,\phi}(X) := \text{softmax}(h_\phi(f_\theta(X)))$. The final hard estimate is then obtained by the maximum-likelihood, i.e., $\hat{Y}_{\theta,\phi}(X) := \arg\max_{c\in\{1,\dots,C\}} [g_{\theta,\phi}(X)]_c$ where $[v]_c$ is the c-th element of an arbitrary C-dimensional vector v.

Standard goal of learning is to find a pair (θ, ϕ) that maximizes the following *average accuracy*, i.e.,

$$\theta_0, \phi_0 \in \arg\max_{\theta,\phi} \mathbb{E}[\mathbb{1}\{Y = \hat{Y}_{\theta,\phi}(X)\}], \tag{1}$$

where $Y \sim P_{Y|X}$ is the ground-truth label (class) of the input $X \sim P_X$. However, high average accuracy does not guarantee high accuracy on every feasible subset of the data, as aggregate performance can hide poor performance on specific subsets. To see this more clearly, let us consider the case where the input distribution P_X admits the form of a mixture of domain-specific distribution $P_{X|g}$, i.e.,

$$P_X = \sum_{g\in\mathcal{G}} \pi_g \cdot P_{X|G=g}, \tag{2}$$

where $\pi_g := P_G(G = g)$ is the mixture weight of domain $g \in \mathcal{G}$ under the domain distribution P_G. And it is obvious that, unless all the domains are equally likely, i.e., $\pi_g = 1/|\mathcal{G}|$, the optimal model parameters (θ_0, ϕ_0) need not guarantee the optimal *worst-group accuracy*, $\min_{g\in\mathcal{G}} \mathbb{E}[\mathbb{1}\{Y = \hat{Y}_{\theta,\phi}(X)\}|G = g]$. To address the limitation of the original problem (1), DG literature generally aims at achieving the following modified goal

$$\begin{aligned} &\max_{\theta,\phi} \min_{g\in\mathcal{G}} \mathbb{E}\big[\mathbb{1}\{Y = \hat{Y}_{\theta,\phi}(X)\}|G = g\big], \\ &\qquad \text{s.t. } \mathbb{E}[\mathbb{1}\{Y = \hat{Y}_{\theta,\phi}(X)\}] \geq \mathbb{E}[\mathbb{1}\{Y = \hat{Y}_{\theta_0,\phi_0}(X)\}] - \epsilon, \end{aligned} \tag{3}$$

i.e., maximize the worst-group accuracy while maintaining the average accuracy similar with the optimal one (1) up to some tolerance level $\epsilon \geq 0$.

In this work, we aim at solving (3) with n number of i.i.d. training examples $(X_i, Y_i) \overset{\text{i.i.d.}}{\sim} P_{XY}$ for $i = 1, ..., n$ *that does not contain any domain information*. In the sequel, let us consider a random variable $(X', Y') \sim \hat{P}_{XY}$ where $\hat{P}_{XY} := (1/n)\sum_{i=1}^n \delta_{(X_i,Y_i)}$ is the empirical distribution and δ_a denotes a point mass at a. This will allow us to simplify the description of proposed fine-tuning strategy.

2.2 PHASE 1: FINE-TUNE THE FEATURE EXTRACTOR TO CONSTRUCT A REMOVABLE LATENT SUBSPACE

We first obtain the initial parameters (θ_0, ϕ_0) of the feature extractor f_θ and the classification head h_ϕ, respectively, by standard empirical risk minimization over $\hat{P}_{XY}$, i.e.,

$$\theta_0, \phi_0 \in \arg\max_{\theta,\phi} \mathbb{E}\left[\log[g_{\theta,\phi}(X')]_{Y'}\right]. \tag{4}$$

As alluded in the previous sections, our goal in Phase 1 is to reduce the within-class latent dispersion while preserving, or increasing, the between-class latent dispersion. To elaborate, let us first define the considered latent-level dispersion precisely as follows

$$\underbrace{\Delta^{\text{f, within}}_{y|\theta} := f_\theta(X'_y) - f_\theta(X''_y)}_{\text{within-class latent dispersion}}, \quad \underbrace{\Delta^{\text{f, between}}_{y|\theta} := \mathbb{E}\big[f_\theta(X'_y)\big] - \mathbb{E}[f_\theta(X')]}_{\text{between-class latent dispersion}}, \quad X'_y, X''_y \overset{\text{i.i.d.}}{\sim} \hat{P}_{X|Y=y}. \tag{5}$$

Note that $\Delta^{\text{f, within}}_{y|\theta}$ captures the pairwise difference of the latent vectors coming from the same-labeled input data, while $\Delta^{\text{f, between}}_{y|\theta}$ captures the difference between the class centroid of label y from the global centroid.

2.2.1 Dimension and Subspace Identification

Ideally, we want to minimize $\Delta^{\text{f,within}}_{y|\theta}$ while maximizing $\Delta^{\text{f,between}}_{y|\theta}$. However, the entire latent space is often too large hence directly fine-tuning f_θ may easily suffer from overfitting to the empirical distribution. To address this issue, we propose to fine-tune f_θ *under the dimension-reduced restricted subspace*. Our key idea stems from PCA (Jolliffe, 2002) in particular supervised PCA (Barshan et al., 2011; Papazoglou & Yin, 2025) to find a few number of principle component directions that best describe the empirical data. Unlike existing works, we find principle components so as to minimize $\Delta^{\text{f,within}}_{y|\theta}$ while maximizing $\Delta^{\text{f,between}}_{y|\theta}$. Note however that, the former part resembles the focus of (Belrose et al., 2023).

Specifically, we aim at reducing the dimension of the latent space by eliminating the subspace associated with span(B) in which only the (harmful) within-class latent dispersion lies. In other words, we obtain the reduced-dimension latent space via an orthogonal projection onto span$(B)^\perp$ during which we would like to preserve (or increase) between-class latent dispersion while alleviating within-class latent dispersion. Finding such B, along with joint fine-tuning of f_θ can be summarized into the following objective:

$$\max_{r\in\{1,\dots,d\}} \max_{\substack{\theta, B\in\mathbb{R}^{d\times r}\\ B^\top B = I_r}} \mathbb{E}\Big[\big\|\Pi_B^\perp \Delta^{\text{f, between}}_{Y'|\theta}\big\|_2^2\Big] \quad \text{s.t. } \mathbb{E}\Big[\big\|\Pi_B \Delta^{\text{f, within}}_{Y'|\theta}\big\|_2^2\Big] = 0, \tag{6}$$

where we have denoted as $\Pi_B := BB^\top$ the orthogonal projection onto span(B), and as $\Pi_B^\perp := I_d - BB^\top$ the orthogonal projection onto span$(B)^\perp$.

2.2.2 Constrained joint optimization

In practice, solving (6) is a challenging task due to (*i*) the coupling between $\theta^\star$ and $B^\star$ and (*ii*) the presence of the constraints that forces within-class latent dispersion to be zero. We address the challenges using the standard (*i*) alternative optimization and (*ii*) Lagrangian dual approaches.

Specifically, we propose the following update rule:

- for $i = 1, 2, ...,$
  - ▷ let $C(\theta_{i-1}) := \text{Cov}(\Delta^{\text{f, between}}_{Y'|\theta_{i-1}}) + \beta_{i-1} \cdot \mathbb{E}\big[\Delta^{\text{f, within}}_{Y'|\theta_{i-1}} (\Delta^{\text{f, within}}_{Y'|\theta_{i-1}})^\top\big]$
  - ▷ update $B_i = [u_1, ..., u_{r_i}]$, $U\Lambda U^\top = C(\theta_{i-1})$, $r_i = \max\Big\{k \le \bar{r} : \lambda_k \ge \dfrac{\lambda_1}{C}\Big\}$
  - ▷ update $\theta_i = \theta_{i-1} - \eta \nabla_{\theta_{i-1}} \text{tr}\big(B_i^\top C(\theta_{i-1}) B_i\big)$, (7)

where $U = [u_1, ..., u_d]$ is an orthogonal matrix and $\Lambda = \text{diag}(\lambda_1, ..., \lambda_d)$ with $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_d$, given some $\beta_i \ge 0$ for $i = 1, 2, \ldots$, and some hyperparameters $C \ge 1$ and $\bar{r} \le d$.

Note that $C(\theta)$ can also be more explicitly written using the empirical distribution $\hat{P}_{XY} = (1/n)\sum_{i=1}^n \delta_{(X_i,Y_i)}$. Specifically, denoting as $\bar{f}_{c|\theta} := \frac{1}{n_c}\sum_{i=1}^n f_\theta(X_i)\mathbb{1}\{Y_i = c\}$ the class centroid of label c whenever $n_c = \sum_{i=1}^n \mathbb{1}\{Y_i = c\} > 0$ and $\bar{f}_\theta := \frac{1}{n}\sum_{i=1}^n f_\theta(X_i)$ the global centroid, we can rewrite the covariance matrix $\text{Cov}(\Delta^{\text{f, between}}_{Y'|\theta})$ under the empirical distribution $\hat{P}_{XY}$ as

$$\begin{aligned}\text{Cov}(\Delta^{\text{f, between}}_{Y'|\theta}) &= \mathbb{E}\Big[\big(\mathbb{E}[f_\theta(X'_{Y'})] - \mathbb{E}[f_\theta(X')]\big)\big(\mathbb{E}[f_\theta(X'_{Y'})] - \mathbb{E}[f_\theta(X')]\big)^\top\Big] \\ &= \sum_{c=1}^C \frac{n_c}{n}\big(\bar{f}_{c|\theta} - \bar{f}_\theta\big)\big(\bar{f}_{c|\theta} - \bar{f}_\theta\big)^\top. \end{aligned} \tag{8}$$

Similarly, the term $\mathbb{E}[\Delta^{\text{f, within}}_{Y'|\theta}(\Delta^{\text{f, within}}_{Y'|\theta})^\top]$ can be explicitly written as

$$\mathbb{E}\big[\Delta^{\text{f, within}}_{Y'|\theta}(\Delta^{\text{f, within}}_{Y'|\theta})^\top\big] = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\big(f_\theta(X_i) - f_\theta(X_j)\big)\big(f_\theta(X_i) - f_\theta(X_j)\big)^\top \mathbb{1}\{Y_i = Y_j = c\}. \tag{9}$$

We refer to Appendix C for further details.

2.2.3 Projection onto the updated affine low-dimensional subspace

Let us now denote as $\theta^\star = \theta_I$ and $B^\star = B_I$ the final updated parameters after sufficient number of iterations I following (7). Then the overall mapping $f_{\theta^\star, B^\star}$ from any arbitrary raw input $x \in \mathcal{X}$ to the input of the classification head $h_\phi(\cdot)$ can now be explicitly expressed as follows

$$f^\star(x) \equiv f_{\theta^\star, B^\star}(x) := z - B^\star(B^\star)^\top\Big(z - \frac{1}{n}\sum_{i=1}^{n} f_{\theta^\star}(X_i)\Big), \quad z = f_{\theta^\star}(x). \tag{10}$$

2.3 Phase 2: Fine-tune the classifier head based on logit-level dispersion

During Phase 2, we will fix the mapping $f^\star$ in (10) and focus on fine-tuning the classification head h_ϕ. As mentioned earlier, the focus here is to minimize the within-class logit dispersion while maximizing classification accuracy. In particular, we focus on *logit margin* defined as follows

$$[\ell_\phi(X)]^{\infty}_{Y} := [\ell_\phi(X)]_Y - \max_{c \neq Y}[\ell_\phi(X)]_c, \quad \ell_\phi(\cdot) := h_\phi(f^\star(\cdot)), \tag{11}$$

where we have denoted as $[v]_i$ the i-th element of an arbitrary vector v. In general, it is desirable to have sufficiently large logit margin. Note also that negative logit margin indicates a wrong classification.

Equipped with the logit margin definition in (11), we can define the considered logit-level dispersion precisely as follows

$$\underbrace{\Delta^{\text{c, within}}_{y|\phi} := \big[\ell_\phi(X'_y)\big]^{\infty}_{y} - \big[\ell_\phi(X''_y)\big]^{\infty}_{y}}_{\text{within-class logit dispersion}}, \quad \underbrace{\Delta^{\text{c, worst}}_{y|\phi} := \min_{x'_y : \hat{P}_{X|Y=y}(x'_y) > 0} \big[\ell_\phi(x'_y)\big]^{\infty}_{y}}_{\text{within-class worst logit}}. \tag{12}$$

Basically the first term quantifies how logit margin differs for different inputs within the same class, which we aim at reducing to achieve better DG. This shares the similar spirit with (Wang et al., 2024) where the authors showed that decreasing dispersion in the predictive likelihood space can help enhancing DG. The second term captures the worst classification ability for each class that can also help enhancing DG via fairness, as similarly considered in (Krueger et al., 2021).

Mathematically, the objective can be summarized as follows

$$\begin{aligned} \phi^\star \in \arg\max_{\phi}\ & \mathbb{E}\big[\log[\text{softmax}(\ell_\phi(X'))]_{Y'}\big] \\ \text{s.t. }\ & \mathbb{P}\big\{\Delta^{\text{c, within}}_{y|\phi} \leq \Delta^{\text{c, within}}_{y|\phi_0} + \epsilon^{\text{within}}_{y}\big\} = 1 \\ & \Delta^{\text{c, worst}}_{y|\phi} \geq \Delta^{\text{c, worst}}_{y|\phi_0} - \epsilon^{\text{worst}}_{y}, \ \ \forall y \in \{1, ..., C\}, \end{aligned} \tag{13}$$

given some tolerance hyperparameters $\epsilon^{\text{within}}_y, \epsilon^{\text{worst}}_y \geq 0$ for $y \in \{1, ..., C\}$. Note that, we impose constraint under the comparison with the pre-trained model in a manner somewhat similar to the original objective (3). In a manner similar to Sec. 2.2.2, solving (13) is generally hard due to the constraints though it does not involve any joint optimization. We use gradient-based Lagrangian approach to appropriately solve (13). We refer to Appendix E for further details.

3 Related Work

DG methods are commonly organized by algorithmic mechanism, such as data manipulation, representation learning, or optimization strategy (Wang et al., 2022). For the setting considered in this work, however, it is useful to distinguish methods by the supervision required to characterize distributional variation during training.

3.1 Domain- and Group-Supervised Generalization

A large class of DG methods explicitly conditions their objectives on training environments or groups. Invariant Risk Minimization (IRM) (Arjovsky et al., 2019) seeks a representation for which the same classifier is optimal across training environments, while Risk Extrapolation (REx) (Krueger et al., 2021) reduces variation in environment-wise risks. Group Distributionally Robust Optimization (GroupDRO) (Sagawa et al., 2020) instead minimizes the worst training loss over predefined groups. These objectives differ technically, but all require the training data to be partitioned into environments or groups so that environment-specific risks or group losses can be computed.

Domain identities are also used by methods that impose invariance through optimization dynamics, feature statistics, or cross-domain relationships. Fish and Fishr align domain-wise gradient information (Shi et al., 2021; Rame et al., 2022), while CORAL align feature statistics across domains (Sun & Saenko, 2016; Gulrajani & Lopez-Paz, 2021). Other approaches exploit domain-conditioned interpolation, matching, or parameter selection (Yao et al., 2022; Mahajan et al., 2021; Pan et al., 2026). Despite these different mechanisms, their training signals are defined relative to explicit domain or group partitions.

Such supervision can be restrictive when domain labels are costly, or only coarsely describe the underlying sources of variation. Prior work has therefore motivated domain-agnostic learning from the burden of domain annotation, the mismatch between dataset identities and latent domains, and the difficulty of specifying an appropriate partition (Matsuura & Harada, 2020; Zhang et al., 2021; Wang et al., 2024). Moreover, multiple shortcuts may coexist within the same nominal domain, so a single partition need not capture all variations associated with poor generalization (Li et al., 2023).

3.2 Generalization without Explicit Domain Labels

Domain-agnostic objectives and latent-domain discovery. Several DG methods avoid using explicit domain labels. ERM is domain-agnostic by construction and remains a strong baseline when carefully tuned (Vapnik, 1991; Gulrajani & Lopez-Paz, 2021), while SWAD improves generalization through flat-minimum seeking (Cha et al., 2021). Other methods remove explicit domain supervision through class-level alignment, latent-domain discovery, or subset-level prediction regularization (Zhang et al., 2021; Matsuura & Harada, 2020; Wang et al., 2024). Although these approaches differ substantially in mechanism, they aim to obtain robust representations without specifying which particular input variation should be removed.

Matched-pair supervision. A complementary line of work replaces domain assignments with relational information between observations. Weakly supervised representation learning has shown that paired observations can provide useful supervision without explicit group or factor annotations (Locatello et al., 2020). More closely related, Invariance Pair Guidance (IPG) (Surner et al., 2026) uses sparse pairs that preserve semantic content while changing a specified spurious attribute, and enforces invariance through pair-based corrective updates. Our method uses similar label-preserving matched evidence for a different purpose: the matched variation is used to reorganize the latent representation so that it becomes concentrated in a compact removable subspace, which is then eliminated by an affine projection.

Representation intervention without domain labels. Representation-editing methods directly remove specified information from learned features without requiring domain partitions. INLP (Ravfogel et al., 2020) iteratively projects representations onto the null space of predictors for a target attribute, while LEACE (Belrose et al., 2023) derives a closed-form affine transformation for linear concept erasure. Related intervention mechanisms have also been explored for language models, diffusion models, and vision–language–action policies (Gilligan-Lee et al., 2026; Saha et al., 2026; Khan et al., 2025). Unlike these post-hoc editing approaches, our method first reorganizes the upstream representation so that the specified variation becomes removable while preserving task-relevant structure, and then applies an explicit affine projection to improve worst-group generalization without domain or group labels.

Method	ColoredMNIST		Waterbirds		UrbanCars	
	Avg. ↑	WG ↑	Avg. ↑	WG ↑	Adj. ↑	WG_8 ↑
IRM	$\underline{85.0 \pm 0.1}$	0.3 ± 0.5	87.7 ± 1.4	64.7 ± 3.0	83.6 ± 1.4	73.3 ± 5.3
LISA	72.5 ± 1.4	$\mathbf{70.2 \pm 0.3}$	90.8 ± 0.3	$\mathbf{87.2 \pm 0.8}$	81.1 ± 1.6 / 74.8 ± 1.0	55.2 ± 2.1 / 43.2 ± 2.1
MatchDG	$\mathbf{85.1 \pm 0.1}$	0.0 ± 0.0	89.4 ± 0.4	60.4 ± 6.9	78.8 ± 0.6	28.3 ± 7.8
LEACE	58.3	54.8	92.8	76.4	76.0	42.9
ERM	84.4 ± 0.5	1.4 ± 1.5	89.0 ± 0.4	60.2 ± 6.4	80.6 ± 1.6	41.1 ± 15.1
Fish	84.4 ± 0.5	2.0 ± 1.6	$\underline{94.1 \pm 0.3}$	80.2 ± 3.8	87.9 ± 2.9	77.1 ± 9.1
CORAL	84.5 ± 0.7	2.6 ± 3.8	93.6 ± 0.8	80.7 ± 0.4	$\underline{88.3 \pm 0.9}$	$\underline{79.7 \pm 4.4}$
VREx	84.3 ± 0.4	2.9 ± 0.8	89.6 ± 1.0	61.7 ± 3.9	81.3 ± 0.1	45.9 ± 10.1
Fishr	84.1 ± 0.6	2.7 ± 3.4	89.1 ± 0.4	61.2 ± 5.5	80.6 ± 1.0	45.6 ± 8.9
Ours	72.6 ± 0.1	$\underline{61.7 \pm 3.7}$	$\mathbf{94.8 \pm 0.4}$	$\underline{83.2 \pm 0.4}$	$\mathbf{92.3 \pm 1.0}$	$\mathbf{86.9 \pm 3.9}$

Table 1: Main benchmark comparison on the official validation sets. Values are percentages; mean ± sample standard deviation over three seeds where available. Best and second-best results in each column are shown in bold and underlined, respectively. For UrbanCars, the two LISA values correspond to the BG and CoObj configurations, respectively.

4 Experiments

4.1 Experimental Setup

We compare our method with representative domain-generalization, correspondence-based, and representation-intervention methods, including IRM (Arjovsky et al., 2019), LISA (Yao et al., 2022), MatchDG (Mahajan et al., 2021), LEACE (Belrose et al., 2023), ERM (Gulrajani & Lopez-Paz, 2021), Fish (Shi et al., 2021), CORAL (Sun & Saenko, 2016), VREx (Krueger et al., 2021), and Fishr (Rame et al., 2022), with the complete set of baselines reported in Table 1. Each baseline is evaluated using its original training objective and supervision setting, without modifying it to use the matched correspondences provided to our method.

4.2 Overall Robustness

We first ask whether constructing an dimension-reduced subspace and explicitly removing the specified variation along it yields robust predictive performance across different spurious-correlation settings. Table 1 reports the main benchmark comparison on ColoredMNIST (Arjovsky et al., 2019), Waterbirds (Sagawa et al., 2020), and UrbanCars (Li et al., 2023). We report both average predictive performance and worst-group performance, using the latter as the primary measure of robustness.

On ColoredMNIST, our method achieves a worst-group accuracy of 0.697, compared with 0.702 for LISA, while substantially improving over ERM and MatchDG. The corresponding average accuracies are also nearly identical (0.716 versus 0.718). On Waterbirds, our method obtains the highest average accuracy (0.917) and a worst-group accuracy of 0.850, remaining competitive with LISA (0.872). These results indicate that explicitly constructing and removing the specified variation does not require sacrificing overall predictive utility to obtain strong group robustness.

The largest difference appears on UrbanCars, where background and co-occurring-object correlations coexist. Our method achieves an adjusted accuracy of 0.905 ± 0.007 and an eight-cell worst-group accuracy of 0.827 ± 0.023. The stronger of the two factor-specific LISA variants reaches 0.811 adjusted accuracy and 0.552 WG_8, while ERM and MatchDG achieve substantially lower worst-group performance. Thus, the proposed intervention remains competitive in the single-factor settings while showing its largest advantage when robustness must extend across multiple nuisance factors.

The aggregate metrics in Table 1, however, do not reveal whether the UrbanCars improvement is distributed across both specified factors or is driven by only one conflict condition. We examine this question directly in Section 4.3.

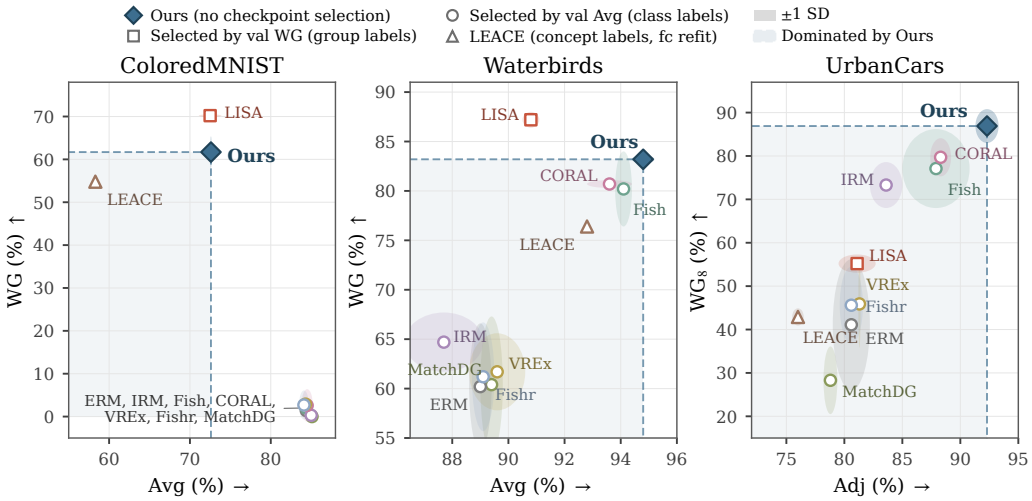


Figure 1: **Average Accuracy–robustness trade-off across benchmarks.** Each point corresponds to the same model checkpoint reported in Table 1. The horizontal axis reports overall predictive accuracy (Avg. for ColoredMNIST and Waterbirds; Adj. for UrbanCars), and the vertical axis reports worst-group accuracy. Higher is better on both axes.

4.3 Robustness under Multiple Specified Variations

UrbanCars allows background (BG) and co-occurring object (CoObj) to vary independently, providing a setting in which robustness must extend across more than one specified variation. We use this structure to ask whether a single intervention constructed from a joint specification can remain robust to each factor individually as well as to their simultaneous variation.

We evaluate four strata of the balanced UrbanCars validation set: *aligned*, where both specified factors follow the training correlation; *BG-conflict* and *CoObj-conflict*, where only one factor conflicts with the target; and *double-conflict*, where both factors conflict simultaneously. In addition to stratum accuracy, we measure the prediction flip rate under label-preserving counterfactual interventions,

$$\mathrm{Flip}_j = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[\hat{y}(x_i) \neq \hat{y}(x_i^{(j)})\right], \tag{14}$$

where $j \in \{\mathrm{BG}, \mathrm{CoObj}, \mathrm{BC}\}$ denotes background, co-occurring-object, or joint intervention. Lower flip rate indicates lower prediction sensitivity to the corresponding specified variation. For compact visualization, we also summarize single-factor sensitivity by $\mathrm{Flip}^{\mathrm{worst}}_{\mathrm{single}} = \max(\mathrm{Flip}_{\mathrm{BG}}, \mathrm{Flip}_{\mathrm{CoObj}})$.

Figure 2(a) shows that the factor-specific LISA configurations exhibit complementary behavior under individual conflicts: LISA-BG favors robustness to background conflict, whereas LISA-CoObj favors robustness to co-occurring-object conflict. In contrast, our joint specification remains strong on both factors, placing it near the upper-right of the single-factor robustness plane. This indicates that the learned intervention does not merely shift robustness from one specified factor to the other, but supports robustness to both.

This behavior persists when the two factors vary simultaneously. Figure 2(b) shows that our method achieves a double-conflict accuracy of 0.861, compared with 0.560 for the strongest baseline. Thus, the gain is not limited to a single conflict condition, but extends to the more challenging setting in which both specified factors oppose the target at the same time.

The same pattern appears at the prediction level. Figure 2(c) shows that our method reduces the worst single-factor flip rate and joint flip rate to 0.033 and 0.034, respectively. In contrast, the competing methods exhibit substantially higher joint than single-factor sensitivity, indicating additional prediction instability under simultaneous variation. Our method remains close to the origin and near

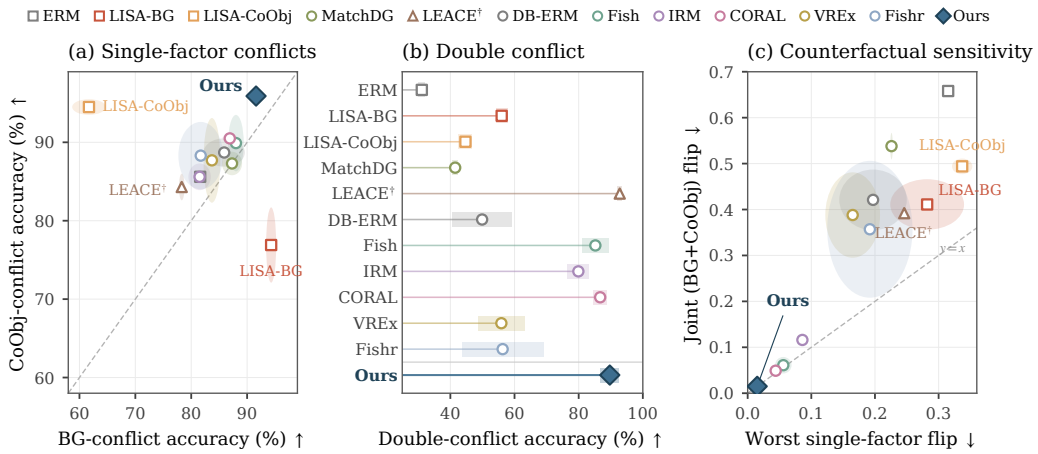


Figure 2: **Robustness under multiple specified variations on UrbanCars. (a)** Accuracy under individual background (BG) and co-occurring-object (CoObj) conflicts. The dashed diagonal indicates equal robustness to the two specified factors. Factor-specific LISA variants favor different factors, whereas our joint specification remains strong under both. **(b)** Accuracy when BG and CoObj conflict simultaneously. Our method retains substantially higher robustness under the double-conflict condition. **(c)** Counterfactual prediction sensitivity, measured by the worst single-factor flip rate and the joint BG–CoObj flip rate. The dashed diagonal indicates equal joint and worst single-factor sensitivity; points above it exhibit additional sensitivity under simultaneous variation. Lower-left is better.

the diagonal, showing both low single-factor sensitivity and little additional amplification under joint intervention.

These gains are not obtained uniformly across all strata: our method attains an aligned accuracy of 0.941, compared with 0.971–0.991 for the strongest competing configurations. The joint intervention therefore trades some performance in the dominant aligned regime for substantially reduced dependence on the specified nuisance factors. Taken together with the aggregate UrbanCars result in Section 4.2, these results show that the robustness gain extends across individually and jointly varying specified factors rather than arising from a single favorable conflict case.

Additional ablation studies supporting the necessity of each component of the proposed training pipeline are provided in Appendix A.

5 Conclusion

In this work, we have presented a new two-phase training pipeline for domain generalization that requires no domain information during training. The key idea is to assign distinct optimization objectives to the feature extractor and classifier head by fully exploiting the available label information. The proposed two-phase training pipeline can be summarized as follows: (*i*) After training a baseline model with the standard average-accuracy objective, Phase 1 fine-tunes the feature extractor within a dimension-reduced subspace to reduce within-class dispersion while increasing between-class dispersion in the latent space; (*ii*) Phase 2 subsequently fine-tunes the classifier head to reduce within-class logit-level dispersion while improving the worst-case logit margin and average classification performance. Across ColoredMNIST, Waterbirds, and UrbanCars, the proposed training algorithm improves worst-group generalization while maintaining competitive overall performance. Ablation studies further confirm the importance of each step in the proposed training pipeline. Future work may consider extending the proposed framework to large-scale pretrained models (Ramé et al., 2022) and/or developing a statistical analysis (Hellström et al., 2025) to better understand the empirically observed domain-generalization gains.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology and experimental design during the validation of research ideas developed by the authors, and to assist with parts of the implementation of the proposed training algorithm. Generative AI tools were not used to originate the central research idea or hypothesis. All AI-assisted methodological feedback and implementations were independently reviewed, tested, and validated by the authors. The authors take full responsibility for the content, claims, and experimental results of this work.

REFERENCES

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

Elnaz Barshan, Ali Ghodsi, Zohreh Azimifar, and Mansoor Zolghadri Jahromi. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition*, 44(7):1357–1371, 2011.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: Perfect linear concept erasure in closed form. In *Advances in Neural Information Processing Systems*, volume 36, pp. 66044–66063, 2023.

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.

Ciarán M Gilligan-Lee, Joseph Egan, Yuchen Zhu, and Michael O'Riordan. Unsupervised identification and removal of spurious correlations during fine-tuning. *arXiv preprint arXiv:2605.27676*, 2026.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=lQdXeXDoWtI.

Fredrik Hellström, Giuseppe Durisi, Benjamin Guedj, and Maxim Raginsky. Generalization bounds: Perspectives from information theory and pac-bayes. *Foundations and Trends in Machine Learning*, 18(1):1–223, 2025.

Ian T Jolliffe. *Principal component analysis*. Springer, 2002.

Momin Ahmad Khan, Novak Boskov, Fatima M Anwar, and Manzoor A Khan. Controlling vision-language-action policies through sparse latent directions. In *Mechanistic Interpretability Workshop at NeurIPS 2025*, 2025.

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 *Proceedings of the 38th International Conference on Machine Learning*, pp. 5815–5826. PMLR, 2021.

Zhiheng Li, Ivan Evtimov, Albert Gordo, Caner Hazirbas, Tal Hassner, Cristian Canton Ferrer, Chenliang Xu, and Mark Ibrahim. A whac-a-mole dilemma: Shortcuts come in multiples where mitigating one amplifies others. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20071–20082, 2023.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 6348–6359. PMLR, 2020.

Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 7313–7324. PMLR, 2021.

Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 11749–11756, 2020.

Bin Pan, Shiyu Shen, Zongbin Wang, Zhenwei Shi, and Xia Xu. Preserving domain generalization in fine-tuning via joint parameter selection. *IEEE Transactions on Multimedia*, 2026.

Theodosios Papazoglou and Guosheng Yin. Covariance supervised principal component analysis. *arXiv preprint arXiv:2506.19247*, 2025.

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

Alexandre Ramé, Jianyu Zhang, Léon Bottou, and David Lopez-Paz. Pre-train, fine-tune, interpolate: a three-stage strategy for domain generalization. In *First Workshop on Interpolation Regularizers and Beyond at NeurIPS 2022*, 2022.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7237–7256, 2020.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020.

Shaswati Saha, Rajasekhar Anguluri, and Manas Gaur. To erase, or not to erase: Robust training-free concept erasure with preservation aware adaptive ranked subspace expansion. In *European Conference on Computer Vision*, pp. 605–623. Springer, 2026.

Yuge Shi, Jeffrey Seely, Philip H. S. Torr, N. Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.

Baochen Sun and Kate Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pp. 443–450. Springer, 2016.

Martin Surner, Abdelmajid Khelil, and Ludwig Bothmann. Invariance pair guidance: Robustness to spurious correlations via corrective gradients. *Machine Learning*, 115(8):188, 2026.

Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in Neural Information Processing Systems*, volume 4, pp. 831–838, 1991.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8052–8072, 2022.

Zongbin Wang, Bin Pan, and Zhenwei Shi. Domain agnostic conditional invariant predictions for domain generalization. *arXiv preprint arXiv:2406.05616*, 2024.

Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 25407–25437. PMLR, 2022.

Wenyu Zhang, Mohamed Ragab, and Ramon Sagarna. Robust domain-free domain generalization with class-aware alignment. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2870–2874. IEEE, 2021.

A Ablation Studies

Our method consists of a sequence of components with distinct roles: reorganization constructs an intervention-compatible geometry, a reference-centered affine projection removes sample-dependent specified variation along the committed subspace, and subsequent constrained adaptation recovers task utility while preserving the effect of the projection. We therefore ablate these components to determine whether the final gains can be reproduced by simpler alternatives and to identify the failure mode addressed at each stage.

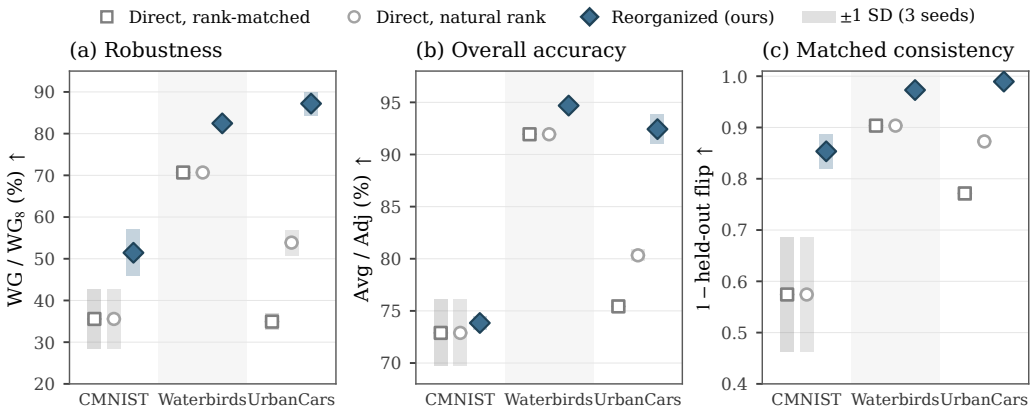


Figure 3: **Post-projection endpoints with and without representation reorganization.** Direct, rank-matched uses the committed rank of the reorganized model, whereas Direct, natural rank selects its rank directly from the frozen ERM geometry.

A.1 Is Reorganization Necessary Before Intervention?

A natural alternative to our pipeline is to skip reorganization and estimate the projection subspace directly from the ERM representation. We compare against two such controls. **Direct, rank-matched** uses the same committed rank as our method, while **Direct, natural rank** selects its rank directly from the ERM geometry. All variants use the same reference-centered affine projection and exclude post-projection adaptation, isolating the role of representation reorganization.

As shown in Fig. 3, direct projection provides partial gains but remains below the endpoint obtained after reorganization. The gap is largest on UrbanCars, where WG_8 reaches 0.349 and 0.539 for the rank-matched and natural-rank controls, respectively, compared with 0.872 after reorganization. This gap cannot be explained solely by projection rank, indicating that reorganization is necessary to construct a latent geometry in which the specified variation can be removed while preserving predictive accuracy.

A.2 What Makes Reorganization Intervention-Safe?

We ablate two constraints used during reorganization: between-class preservation and the logit-response constraint. All variants are evaluated on UrbanCars at the committed state T_0 and immediately after the affine projection at T_1, before Phase 2 adaptation.

The two constraints address distinct failure modes. Without between-class preservation, two of three runs diverge and the remaining run collapses to near-chance accuracy, yielding no valid projection geometry across the three seeds. Thus, concentrating the specified variation without preserving task-relevant structure is insufficient to construct a safely removable subspace.

Removing the logit-response constraint leaves reorganization numerically stable, but increases the T_0 logit response to approximately $2.1\times$ that of the full method. After projection, the held-out matched-pair flip rate increases from 0.011 to 0.021, while WG_8 decreases from 87.2 to 82.9. The comparable change in adjusted accuracy ($+12.4$ vs. $+12.0$ points) indicates that this constraint primarily limits sensitivity to the specified variation rather than utility recovery.

A.3 Is Constraint-Preserving Adaptation Necessary?

We next isolate the role of the Phase 2 constraints. Starting from the same T_1 state, we compare the full constrained adaptation with an unconstrained variant using the same objective, optimizer, learning rate, data, and optimization budget. The two variants therefore differ only in whether the Phase 2 constraints are enforced.

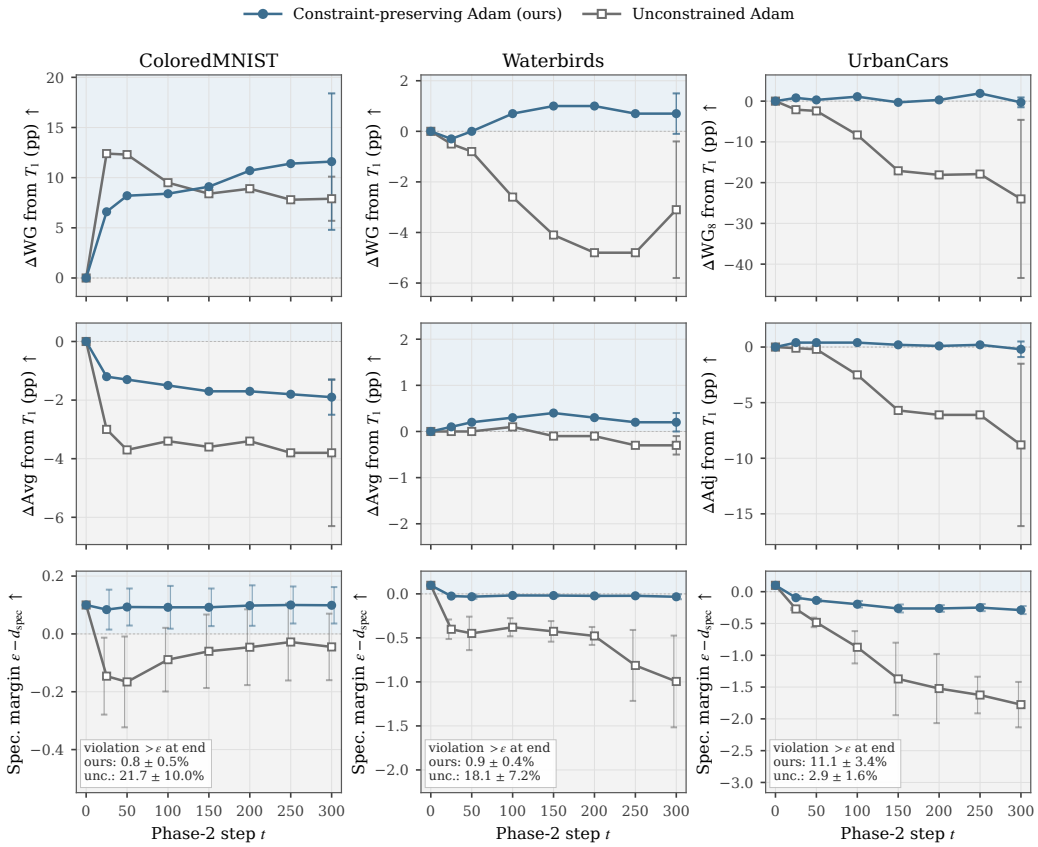


Figure 4: **Effect of the Phase 2 constraints across the three benchmarks.** Both variants start from the same T_1 state and use the same optimization objective and budget. The trajectories report worst-group accuracy and the normalized increase in within-class logit differences relative to T_1. The dashed line denotes the Phase 2 tolerance $\epsilon = 0.10$. Curves show mean values over three seeds. For unconstrained adaptation on UrbanCars, the trajectory is shown only through the last step shared by all three runs.

As shown in Fig. 4, removing the constraints allows substantially larger changes in within-class logit differences on all three benchmarks. On ColoredMNIST, unconstrained adaptation initially increases worst-group accuracy but subsequently loses part of this gain, whereas constrained adaptation reaches a higher final worst-group accuracy. On Waterbirds, unconstrained adaptation decreases worst-group accuracy by 3.0 points, while constrained adaptation increases it by 0.8 points.

The difference is largest on UrbanCars. Without the constraints, two of the three runs terminate after a substantial decrease in predictive accuracy; at the last common step, worst-group accuracy has decreased by 17.1 points. The constrained variant completes all runs while maintaining worst-group accuracy close to its T_1 value. These results show that the Phase 2 constraints limit changes in the quantities fixed at T_1 and prevent subsequent adaptation from substantially degrading worst-group accuracy.

B Multi-Factor Counterfactual Contrasts

Section 2.2.1 defines the specified response using a zero-sum contrast over label-preserving counterfactual views. Here we give the explicit construction used when multiple binary specified factors vary within the same counterfactual orbit.

Consider two specified factors B and C, each taking levels in $\{-1, +1\}$. For instance i, let

$$\mathcal{O}_i = \{x_i^{--}, x_i^{+-}, x_i^{-+}, x_i^{++}\} \tag{15}$$

denote the four-view factorial orbit, where the first sign indexes the level of B and the second sign indexes the level of C. All four views preserve the task label and underlying instance identity while varying only the specified factors. Let

$$h_i^{bc} = f_\theta(x_i^{bc}), \qquad b, c \in \{-1, +1\}.$$

The orbit admits the orthogonal Hadamard decomposition

$$\mu_i = \frac{1}{4}\left(h_i^{--} + h_i^{+-} + h_i^{-+} + h_i^{++}\right), \tag{16}$$

$$a_{i,B} = \frac{1}{4}\left(-h_i^{--} + h_i^{+-} - h_i^{-+} + h_i^{++}\right), \tag{17}$$

$$a_{i,C} = \frac{1}{4}\left(-h_i^{--} - h_i^{+-} + h_i^{-+} + h_i^{++}\right), \tag{18}$$

$$a_{i,BC} = \frac{1}{4}\left(h_i^{--} - h_i^{+-} - h_i^{-+} + h_i^{++}\right). \tag{19}$$

The first term μ_i represents structure shared across the entire orbit. The contrasts $a_{i,B}$ and $a_{i,C}$ isolate the main effects associated with the two specified factors, while $a_{i,BC}$ captures their interaction. Each contrast has zero-sum coefficients and is therefore orthogonal to the shared orbit mean in the factorial coding.

More generally, for K binary specified factors, let $s^{(v)} = (s_1^{(v)}, \ldots, s_K^{(v)}) \in \{-1, +1\}^K$ denote the factor coding of view v. For any nonempty subset of factors $S \subseteq \{1, \ldots, K\}$, the corresponding factorial contrast is

$$a_{i,S} = \frac{1}{2^K} \sum_{v=1}^{2^K} \left(\prod_{j \in S} s_j^{(v)} \right) h_i^{(v)}. \tag{20}$$

Singleton sets $S = \{j\}$ give factor-specific main effects, whereas $|S| > 1$ yields interaction contrasts.

When multiple specified effects are jointly targeted, we aggregate their response moments rather than collapsing the orbit into a single arbitrary pairwise difference. For a selected set of effects $\mathcal{S}$, we use

$$C_{\text{spec}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{S \in \mathcal{S}} a_{i,S} a_{i,S}^\top. \tag{21}$$

The retained statistic is computed from the orbit centers,

$$C_{\text{ret}} = \frac{1}{N-1} \sum_{i=1}^{N} (\mu_i - \bar{\mu})(\mu_i - \bar{\mu})^\top. \tag{22}$$

For UrbanCars, B corresponds to background and C to the co-occurring object. The four-view orbit therefore provides separate latent response contrasts for background, co-occurring object, and their interaction. These contrast statistics are subsequently passed to the subspace construction in Section 2.2.1; the factorial coding itself does not assume that any of these effects already occupies a distinct linear subspace.

C Detailed Reorganization Procedure

This section provides the complete optimization and commit procedure underlying Section 2.2.1. The main text describes the geometric principle of the method, whereas here we specify the running statistics, adaptive-rank rule, exact safety constraints, and removability gate used in the implementation.

For compactness, throughout this appendix we write

$$C_R \equiv C_{\text{spec}}, \qquad C_T \equiv C_{\text{ret}},$$

for the specified-response and retained-structure statistics, respectively.

C.1 Live Geometry Statistics

Because the encoder changes during reorganization, the counterfactual geometry is re-estimated throughout training. Given mini-batch estimates C_R^{batch} and C_T^{batch}, we maintain exponential moving averages

$$C_R^{(t)} = (1-\eta)C_R^{(t-1)} + \eta C_R^{\text{batch}}, \tag{23}$$

$$C_T^{(t)} = (1-\eta)C_T^{(t-1)} + \eta C_T^{\text{batch}}, \tag{24}$$

with $\eta = 0.02$.

At each basis-refresh step, we form the normalized contrast operator

$$M = \frac{C_R}{\operatorname{tr}(C_R)} - \beta\frac{C_T}{\operatorname{tr}(C_T)}, \qquad M \leftarrow \frac{1}{2}(M + M^\top), \tag{25}$$

where $\beta = 1$. The explicit symmetrization removes numerical asymmetry before the eigendecomposition.

To avoid estimating a subspace after near-complete response collapse, a basis refresh is suspended whenever

$$\operatorname{tr}(C_R) < 0.01\ \operatorname{tr}(C_R^{(0)}). \tag{26}$$

We additionally monitor $\operatorname{tr}(C_R) > 1.5\ \operatorname{tr}(C_R^{(0)})$ as an energy-growth diagnostic, but this monitor does not itself trigger a commit.

C.2 Adaptive Rank and Shadow-Basis Estimation

Let

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$$

denote the eigenvalues of M with corresponding eigenvectors $v_1, \ldots, v_d$. For rank r, the candidate intervention basis is

$$B_r = [v_1, \ldots, v_r] = \arg\max_{B^\top B = I_r} \operatorname{tr}(B^\top M B). \tag{27}$$

The rank proposal is determined from the relative spectral support:

$$r_{\text{prop}} = \max(r_{\min}, \min[r_{\max}, |\{k \leq r_{\max} : \lambda_k \geq \tau\lambda_1\}|]), \tag{28}$$

with

$$\tau = 0.05, \qquad r_{\min} = 1, \qquad r_{\max} = 32.$$

If $\lambda_1 \leq 0$, we instead propose $r_{\min}$. To reduce rank oscillation, a proposed change is accepted only after the same rank proposal is observed at two consecutive basis refreshes.

During reorganization, the resulting basis is maintained as a shadow estimate B_{sh} and does not yet modify the model forward path. Two useful geometric diagnostics are

$$\text{cap}(B) = \frac{\operatorname{tr}(B^\top C_R B)}{\operatorname{tr}(C_R)}, \tag{29}$$

$$\text{ovl}(B) = \frac{\operatorname{tr}(B^\top C_T B)}{\operatorname{tr}(C_T)}. \tag{30}$$

The former measures how much specified-response energy is captured by the candidate span, while the latter measures retained structure contained within that span.

C.3 Exact Reorganization Objective and Safety Constraints

The geometric objective in Section 2.2.1 is implemented by separating its two components. With $\text{sg}(\cdot)$ denoting stop-gradient, the residual specified response outside the candidate span is

$$\mathcal{L}_{\text{leak}} = \frac{\operatorname{tr}(C_R) - \operatorname{tr}\left[\text{sg}(B)^\top C_R\, \text{sg}(B)\right]}{\operatorname{tr}(C_R^{\text{ref}}) + \epsilon}. \tag{31}$$

The retained-structure overlap penalty is

$$\mathcal{L}_{\text{ovl}} = \frac{\operatorname{tr}\left[\operatorname{sg}(B)^\top C_T \operatorname{sg}(B)\right]}{\operatorname{tr}(C_T) + \epsilon}. \tag{32}$$

Stopping gradients through B ensures that these losses reshape the representation rather than optimize the objective by rotating the basis alone.

For paired counterfactual observations, prediction-level consistency is encouraged using

$$\mathcal{L}_{\text{M1}} = \mathbb{E}_i \left\| \operatorname{softmax}(z_i^A) - \operatorname{softmax}(z_i^B) \right\|_2^2. \tag{33}$$

Concentration by itself admits degenerate solutions. We therefore impose three base-anchored safety constraints.

Retained-structure exclusion. We measure retained energy inside the candidate intervention span by

$$E_T(\theta) = \frac{\operatorname{tr}\left[\operatorname{sg}(B)^\top C_T(\theta) \operatorname{sg}(B)\right]}{\operatorname{tr}(C_T^{\text{ref}}) + \epsilon}, \tag{34}$$

and impose

$$c_T(\theta) = E_T(\theta) - E_T^{(0)} \leq 0. \tag{35}$$

This prevents retained structure from being moved into the subspace that will later be removed.

Specified-response energy floor. To prevent the method from satisfying the concentration objective by globally suppressing the specified response, we impose

$$c_E(\theta) = 1 - \frac{\operatorname{tr}(C_R(\theta))}{\operatorname{tr}(C_R^{\text{ref}})} \leq 0. \tag{36}$$

Hence, reduction of response outside B must occur through redistribution rather than collapse of the total specified-response energy.

Prediction-level response constraint. For a counterfactual pair, define

$$R(\theta) = \sqrt{\mathbb{E}_i \left\| \frac{z_i^A - z_i^B}{2} \right\|_2^2}. \tag{37}$$

We constrain this response relative to its initialization,

$$c_R(\theta) = \frac{R(\theta)}{R^{\text{init}}} - 1 \leq 0, \tag{38}$$

so that reorganization cannot increase prediction-level counterfactual sensitivity.

The complete Stage-R_1 objective is

$$\begin{aligned} \mathcal{L}_{R_1} = \mathcal{L}_{\text{CE}} + \lambda_{\text{pair}} \mathcal{L}_{\text{M1}} + \lambda_{\text{conc}} \left(\mathcal{L}_{\text{leak}} + \beta_{\text{ovl}} \mathcal{L}_{\text{ovl}} \right) \\ + \sum_{c \in \{c_T, c_E, c_R\}} \left(\eta_c c + \frac{\rho}{2} [c]_+^2 \right), \end{aligned} \tag{39}$$

with

$$\lambda_{\text{pair}} = 0.2, \qquad \lambda_{\text{conc}} = 0.16, \qquad \beta_{\text{ovl}} = 1, \qquad \rho = 1.$$

The corresponding dual variables are updated at basis refreshes according to

$$\eta_c \leftarrow [\eta_c + \rho c]_+, \qquad c \in \{c_T, c_E, c_R\}. \tag{40}$$

C.4 Removability and Commit Criteria

The shadow basis is committed only when the current representation provides evidence that the identified span is both geometrically stable and safe to remove. Let m index basis refreshes. We require the following five conditions.

C1: rank stability. The selected rank must remain unchanged for K_c consecutive refreshes:

$$r^{(m)} = r^{(m-1)} = \cdots = r^{(m-K_c+1)}, \qquad K_c = 3. \tag{41}$$

C2: spectral separation. At the selected rank, the relative eigengap must satisfy

$$\frac{\lambda_r - \lambda_{r+1}}{|\lambda_r| + \epsilon} \geq 0.25 \tag{42}$$

for the most recent K_c refreshes.

C3: subspace stability. Let B_{prev} denote the previous shadow basis with the same rank. We measure projector drift by

$$d_S = \frac{\left\| B_{\mathrm{sh}} B_{\mathrm{sh}}^\top - B_{\mathrm{prev}} B_{\mathrm{prev}}^\top \right\|_F}{\sqrt{2r}}, \tag{43}$$

and require

$$d_S \leq 0.10. \tag{44}$$

C4: specified-response capture. The candidate span must explain at least 95% of the current specified response:

$$\mathrm{cap}(B_{\mathrm{sh}}) \geq 0.95. \tag{45}$$

C5: safety feasibility. All three reorganization constraints must be feasible:

$$c_T \leq 0, \qquad c_E \leq 0, \qquad c_R \leq 0. \tag{46}$$

A hard commit is performed only when

$$\mathrm{C1} \wedge \mathrm{C2} \wedge \mathrm{C3} \wedge \mathrm{C4} \wedge \mathrm{C5} \tag{47}$$

is satisfied within the allowed commit window. At that point,

$$B^\star = B_{\mathrm{sh}}. \tag{48}$$

The committed basis is subsequently frozen and converted into the explicit projection described in Section 2.2.3. If the gate is not satisfied before the commit deadline, no intervention basis is deployed.

C.5 Full Reorganization Procedure

Algorithm 1 gives the complete execution order used during Stage R_1.

D Reference-Centered Affine Intervention

This section provides the geometric properties of the affine intervention introduced in Section 2.2.3. The committed basis $B^\star$ identifies the directions along which the specified variation is removed, while the counterfactual centers define the reference coordinate that is retained along those directions.

D.1 Reference Coordinate and Affine Target Set

Let

$$B^\star \in \mathbb{R}^{d \times r}, \qquad B^{\star\top} B^\star = I_r,$$

denote the committed intervention basis. From the counterfactual centers $\{\mu_i\}_{i=1}^N$ defined in Section 2.2.1, we compute

$$\bar{\mu} = \frac{1}{N} \sum_{i=1}^{N} \mu_i, \qquad a^\star = B^{\star\top} \bar{\mu}. \tag{49}$$

The reference coordinate $a^\star$ is computed once at commit from the training counterfactual evidence and is subsequently frozen together with $B^\star$.

Algorithm 1 Full Reorganization and Commit Procedure

Require: Encoder f_θ, counterfactual sets $\{\mathcal{V}_i\}$, reorganization start t_0, commit deadline t_{latest}
Ensure: Committed basis $B^\star$, or NoCommit

```
Initialize C_R, C_T, shadow basis B_sh, rank state, and dual variables η_T, η_E, η_R
Keep the deployed forward path unchanged
for training step t do
    Compute counterfactual representations, contrasts, and centers
    Form C_R^batch and C_T^batch
    Update C_R, C_T using Eqs. equation 23–equation 24
    if t ≥ t_0 and t is a basis-refresh step then
        if tr(C_R) ≥ 0.01 tr(C_R^(0)) then
            Construct M using Eq. equation 25
            Compute its eigendecomposition
            Propose rank using Eq. equation 28
            Apply rank hysteresis
            Update shadow basis B_sh
        end if
    end if
    Compute L_M1, L_leak, and L_ovl
    Evaluate c_T, c_E, c_R
    Compute L_R1 using Eq. equation 39
    Update encoder parameters θ
    if t is a basis-refresh step then
        Update dual variables using Eq. equation 40
        Evaluate C1–C5
        if C1 ∧ C2 ∧ C3 ∧ C4 ∧ C5 then
            B* ← B_sh
            return B*
        end if
    end if
    if t > t_latest then
        return NoCommit
    end if
end for
```

The orthogonal-complement projector associated with $B^\star$ is

$$P^\star = I - B^\star B^{\star\top}. \tag{50}$$

The intervention maps a representation h to

$$\Pi_{\text{aff}}(h) = h - B^\star \left(B^{\star\top} h - a^\star\right), \tag{51}$$

or equivalently,

$$\Pi_{\text{aff}}(h) = P^\star h + B^\star a^\star = \bar{\mu} + P^\star(h - \bar{\mu}). \tag{52}$$

Thus, the target of the intervention is the affine set

$$\mathcal{S}_{\text{aff}} = \left\{z \in \mathbb{R}^d : B^{\star\top} z = a^\star\right\}. \tag{53}$$

Every intervened representation therefore satisfies

$$B^{\star\top} \Pi_{\text{aff}}(h) = a^\star. \tag{54}$$

D.2 Removal of Variation Along the Committed Subspace

Although the intervention preserves a nonzero reference coordinate, variation along the committed subspace is removed exactly. For any two representations h_i and h_j,

$$\Pi_{\text{aff}}(h_i) - \Pi_{\text{aff}}(h_j) = P^\star h_i + B^\star a^\star - P^\star h_j - B^\star a^\star \tag{55}$$

$$= P^\star(h_i - h_j). \tag{56}$$

Since

$$B^{\star\top} P^\star = 0,$$

it follows that

$$B^{\star\top} \left[\Pi_{\text{aff}}(h_i) - \Pi_{\text{aff}}(h_j)\right] = 0. \tag{57}$$

In particular, for any label-preserving counterfactual pair (h_i, h_i^{cf}),

$$B^{\star\top} \left[\Pi_{\text{aff}}(h_i^{\text{cf}}) - \Pi_{\text{aff}}(h_i)\right] = 0. \tag{58}$$

Hence the preserved reference offset does not reintroduce sample-dependent variation along $\text{span}(B^\star)$.

D.3 Orthogonal Minimum-Distance Interpretation

The affine intervention is the Euclidean projection of h onto $\mathcal{S}_{\text{aff}}$. Specifically,

$$\Pi_{\text{aff}}(h) = \arg\min_z \frac{1}{2}\|z - h\|_2^2 \quad \text{s.t.} \quad B^{\star\top} z = a^\star. \tag{59}$$

To see this, consider the Lagrangian

$$\mathcal{L}(z, \lambda) = \frac{1}{2}\|z - h\|_2^2 + \lambda^\top \left(B^{\star\top} z - a^\star\right). \tag{60}$$

Stationarity with respect to z gives

$$z = h - B^\star \lambda.$$

Substituting this expression into the affine constraint and using $B^{\star\top} B^\star = I_r$ yields

$$\lambda = B^{\star\top} h - a^\star.$$

Therefore,

$$z = h - B^\star \left(B^{\star\top} h - a^\star\right) = \Pi_{\text{aff}}(h). \tag{61}$$

The correction

$$h - \Pi_{\text{aff}}(h) = B^\star \left(B^{\star\top} h - a^\star\right) \tag{62}$$

lies entirely in $\text{span}(B^\star)$, which is orthogonal to the tangent space of $\mathcal{S}_{\text{aff}}$. The intervention therefore satisfies the specified reference-coordinate constraint with minimum Euclidean distortion.

D.4 Idempotence and Relation to Linear Projection

Because the first application of the intervention already places the representation in $\mathcal{S}_{\text{aff}}$,

$$\Pi_{\text{aff}}\left(\Pi_{\text{aff}}(h)\right) = \Pi_{\text{aff}}(h). \tag{63}$$

Thus, Π_{aff} is idempotent, as expected for an orthogonal projection onto an affine set.

The standard zero-centered orthogonal projection is recovered as the special case $a^\star = 0$. In that case,

$$\Pi_{\text{aff}}(h) = h - B^\star B^{\star\top} h = P^\star h. \tag{64}$$

The affine formulation therefore generalizes the usual linear projection by separating the directions of removed variation, represented by $B^\star$, from the reference coordinate preserved along those directions, represented by $a^\star$.

E Detailed Post-Intervention Optimization

This section gives the complete implementation of the post-intervention adaptation stage described in Section 2.3. Once the intervention basis has been committed, both $B^\star$ and

$$P^\star = I - B^\star B^{\star\top}$$

remain fixed. No additional intervention basis or projector is introduced during this stage. The purpose of the optimizer is instead to recover task utility from the projected representation while preventing uncontrolled re-entry of the specified variation.

E.1 Post-Intervention Re-Anchoring and Task Objective

Let T_1 denote the model state immediately after inserting the committed projector $P^\star$. All post-intervention reference quantities are reset at this state. In particular, for a counterfactual representation difference δ and shared center μ, we use

$$\delta_\perp = P^\star \delta, \qquad \mu_\perp = P^\star \mu. \tag{65}$$

The residual geometric and safety terms from the reorganization stage are therefore re-evaluated on $(\delta_\perp, \mu_\perp)$ using reference values anchored at T_1. We denote the resulting residual objective by $\mathcal{L}_{R+A}$.

To encourage recovery through information shared across counterfactual views, let $h_i^{\mathrm{pen},(v)}$ denote the penultimate representation of view $v \in \mathcal{V}_i$, and define

$$\bar{h}_i^{\mathrm{pen}} = \frac{1}{|\mathcal{V}_i|} \sum_{v \in \mathcal{V}_i} h_i^{\mathrm{pen},(v)}. \tag{66}$$

The task-common objective is

$$\mathcal{L}_{\mathrm{TC}} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{CE}\left(\mathrm{fc}\left(\bar{h}_i^{\mathrm{pen}}\right), y_i\right), \tag{67}$$

and the complete post-intervention objective is

$$\mathcal{L}_{\mathrm{post}} = \mathcal{L}_{R+A} + \lambda_{\mathrm{TC}} \mathcal{L}_{\mathrm{TC}}. \tag{68}$$

The task-common coefficient is calibrated once at T_1 from gradient magnitudes rather than tuned independently:

$$\lambda_{\mathrm{TC}} = \rho_{\mathrm{TC}} \frac{\|g_R\|_2}{\|g_T\|_2 + \epsilon}, \qquad \rho_{\mathrm{TC}} = 0.25, \tag{69}$$

where

$$g_R = \nabla_\theta \mathcal{L}_{R+A}, \qquad g_T = \nabla_\theta \mathcal{L}_{\mathrm{TC}}.$$

E.2 Adam Task-Descent Proposal

At each post-intervention step, the optimizer first constructs an ordinary Adam proposal from

$$g_t = \nabla_\theta \mathcal{L}_{\mathrm{post}}(\theta_t).$$

The Adam moments are

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \tag{70}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^{\odot 2}, \tag{71}$$

with bias-corrected moments

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t}, \qquad \widehat{v}_t = \frac{v_t}{1 - \beta_2^t}.$$

The raw task-descent displacement is

$$d_A = -\eta_{\mathrm{post}} \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon_A}, \tag{72}$$

where

$$\eta_{\mathrm{post}} = 3 \times 10^{-6}, \qquad (\beta_1, \beta_2) = (0.9, 0.999), \qquad \epsilon_A = 10^{-8}.$$

The constrained solver acts on the displacement d_A rather than replacing the task objective itself.

E.3 Decision-Space Constraint Construction

The implementation defines post-intervention preservation constraints directly in decision space. For the binary classifier, let

$$m_i^{(v)}(\theta) = z_1(x_i^{(v)};\theta) - z_0(x_i^{(v)};\theta) \tag{73}$$

denote the unsigned binary decision margin for view v.

For a reference view $x_i^{(0)}$ and specified counterfactual view $x_i^{(j)}$, define the magnitude of the corresponding decision response as

$$\delta m_{ij}(\theta) = \left| m_i^{(j)}(\theta) - m_i^{(0)}(\theta) \right|. \tag{74}$$

The worst correct-class margin over all views in the counterfactual orbit is

$$m_i^{\min}(\theta) = \min_v (2y_i - 1) m_i^{(v)}(\theta). \tag{75}$$

Robust normalization scales are computed once from the full anchor pool at T_1:

$$s_{N,j} = \operatorname{med}_i \delta m_{ij}(T_1) + \operatorname{MAD}_i \delta m_{ij}(T_1), \tag{76}$$

$$s_T = \operatorname{med}_i \left| m_i^{\min}(T_1) \right|. \tag{77}$$

Using a common normalized slack

$$\varepsilon = 0.10,$$

we define the nuisance-drift constraints

$$\widetilde{c}_{ij}^{N}(\theta) = \frac{\delta m_{ij}(\theta) - \delta m_{ij}(T_1)}{s_{N,j}} - \varepsilon, \tag{78}$$

and the task-margin constraints

$$\widetilde{c}_i^{T}(\theta) = \frac{m_i^{\min}(T_1) - m_i^{\min}(\theta)}{s_T} - \varepsilon. \tag{79}$$

A constraint is feasible when its normalized residual is non-positive.

The first family prevents the post-intervention decision response to the specified variation from drifting substantially above its committed value. The second prevents excessive deterioration of the worst-view task margin.

All constraint-evaluation forward passes used for steering are performed in the fixed post-intervention reference frame. In particular, BatchNorm running statistics are frozen at T_1:

$$\mu_{\mathrm{BN},t} = \mu_{\mathrm{BN},T_1}, \qquad \sigma^2_{\mathrm{BN},t} = \sigma^2_{\mathrm{BN},T_1}. \tag{80}$$

E.4 Constraint States and Active Set

At each step, constraints are evaluated on the current anchor window. A violation threshold

$$\delta_V = \frac{\varepsilon}{2} = 0.05 \tag{81}$$

separates ordinary task-descent operation from explicit restoration.

If

$$\max\left\{ \max_{i,j} \widetilde{c}_{ij}^{N}, \max_i \widetilde{c}_i^{T} \right\} > \delta_V, \tag{82}$$

the solver enters the restoration state described below. Otherwise it performs task descent using either an unconstrained or boundary-corrected step.

For task-descent states, constraints sufficiently close to their boundaries are included proactively:

$$\mathcal{A}_t = \{k : \widetilde{c}_k(\theta_t) \geq -\delta_{\mathrm{act}}\}, \qquad \delta_{\mathrm{act}} = 1.5\varepsilon. \tag{83}$$

The nuisance and task rows are capped at

$$K_N = 48, \qquad K_T = 16,$$

respectively, retaining the largest normalized residuals when the cap is exceeded.

For each active constraint k, define

$$q_k = \nabla_\theta \widetilde{c}_k(\theta_t), \tag{84}$$

and stack the row gradients into

$$G_t = \begin{bmatrix} q_{k_1}^\top \\ \vdots \\ q_{k_r}^\top \end{bmatrix}. \tag{85}$$

For nuisance constraints, differentiation through the absolute decision response includes the corresponding sign of the decision contrast. For the worst-view task constraint, the gradient is taken through the currently active minimum-margin view; this is the subgradient implied by the minimum operation.

E.5 Feasible-Direction Correction

When $\mathcal{A}_t = \varnothing$, the optimizer is in the free state and uses the Adam proposal directly:

$$d^\star = d_A. \tag{86}$$

Otherwise, the solver enters the boundary state and computes the displacement closest to the Adam proposal that satisfies the first-order constraint model:

$$d^\star = \arg\min_d \frac{1}{2}\|d - d_A\|_2^2 \quad \text{s.t.} \quad \widetilde{c} + G_t d \le 0. \tag{87}$$

The final implementation uses the Euclidean metric $M = I$.

The associated dual problem is

$$\lambda^\star = \arg\min_{\lambda \ge 0} \left[\frac{1}{2}\lambda^\top (G_t G_t^\top)\lambda - \lambda^\top (G_t d_A + \widetilde{c}) \right], \tag{88}$$

and the corrected displacement is

$$d^\star = d_A - G_t^\top \lambda^\star. \tag{89}$$

The non-negative QP in Eq. equation 88 is solved using Hildreth coordinate descent with at most 300 sweeps. For numerical conditioning, the constraint Gram matrix

$$K_t = G_t G_t^\top$$

uses a ridge of

$$\gamma_{\text{ridge}} = 10^{-12}\,\text{tr}(K_t). \tag{90}$$

E.6 Actual-Value Guard and Backtracking

The QP in Eq. equation 87 relies on the local approximation

$$\widetilde{c}(\theta_t + d) \approx \widetilde{c}(\theta_t) + G_t d.$$

Because this approximation need not remain accurate for a finite neural-network update, every proposed task step is checked using an actual forward pass.

For each constraint family $F \in \{N, T\}$, a candidate displacement $\alpha d^\star$ is accepted if either

$$\max \widetilde{c}^F(\theta_t + \alpha d^\star) \le \epsilon_{\text{num}}, \tag{91}$$

or

$$\max \widetilde{c}^F(\theta_t + \alpha d^\star) < \max \widetilde{c}^F(\theta_t) - 10^{-4}, \tag{92}$$

where

$$\epsilon_{\text{num}} = 3 \times 10^{-3}.$$

We choose the largest accepted step multiplier from

$$\alpha \in \left\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\right\}. \tag{93}$$

The feasible-or-improving rule in Eqs. equation 91–equation 92 is applied separately to the nuisance and task constraint families.

If no multiplier is accepted, the task step is skipped. Both the parameters and the provisional Adam state are then rolled back.

E.7 Violation Restoration

If Eq. equation 82 holds, ordinary task descent is temporarily suspended. Let $\widetilde{c}_{\text{viol}}$ contain the selected violated constraints and let G_{viol} contain their gradients. The restoration step is chosen to reduce the current violations in the local model while minimizing parameter displacement.

Its dual form is

$$\lambda_R = \arg\min_{\lambda \geq 0} \left[\frac{1}{2}\lambda^\top \left(G_{\text{viol}} G_{\text{viol}}^\top\right)\lambda - \lambda^\top \left(\kappa \widetilde{c}_{\text{viol}}\right)\right], \tag{94}$$

with

$$\kappa = 0.5,$$

and the resulting restoration direction is

$$d_R = -G_{\text{viol}}^\top \lambda_R. \tag{95}$$

The restoration direction is checked by actual forward evaluation with

$$\alpha_R \in \left\{1, \frac{1}{2}, \frac{1}{4}\right\}. \tag{96}$$

A restoration step is accepted when it reduces the maximum normalized violation by at least 10^{-4}.

Restoration is deliberately separated from Adam momentum. If a restoration step is accepted, the parameter update is retained but the provisional Adam moments are rolled back. Thus repair-only motion does not accumulate as task-descent momentum. For an accepted task step, both the parameter update and provisional Adam moments are committed; for a rejected task step, both are rolled back.

E.8 Violation Memory and Anchor Management

Evaluating every counterfactual anchor at every optimization step is unnecessary and expensive, but using only a rotating local window allows a previously violated anchor to disappear from the constraint set. We therefore combine window rotation with violation memory.

At each step, the base guard window contains

$$N_G = 48$$

rotating counterfactual anchors. Any anchor–constraint pair with positive normalized residual is inserted into a violation-memory buffer. The buffer is capped at 64 entries; the 24 highest-priority remembered quartets are merged with the current rotating window.

Entries are refreshed when revisited and removed after returning to feasibility. In addition, a full anchor-pool audit is performed every

$$K_{\text{audit}} = 10$$

optimization steps. During the audit, previously unseen global violations satisfying

$$\widetilde{c}_k > 2\epsilon_{\text{num}} \tag{97}$$

are inserted into violation memory.

This mechanism keeps locally problematic anchors available to the solver even when they leave the rotating window.

Algorithm 2 Full Constraint-Preserving Post-Intervention Adaptation

Require: Committed parameters θ_{T_1}, fixed projector $P^\star$, counterfactual anchor pool $\mathcal{G}$
Ensure: Adapted parameters θ

Re-anchor residual geometry and all behavioral references at T_1
Freeze $P^\star$ and BatchNorm running statistics at their T_1 values
Compute $s_{N,j}$, s_T, and λ_{TC}
Initialize violation memory $\mathcal{M} \leftarrow \varnothing$
for $t = 1, \ldots, 300$ **do**
    Compute $g_t = \nabla_\theta \mathcal{L}_{\mathrm{post}}(\theta_t)$
    Form provisional Adam moments and proposal d_A
    Select $N_G = 48$ rotating anchors
    Merge the highest-priority remembered anchors from $\mathcal{M}$
    Evaluate $\widetilde{c}^N$ and $\widetilde{c}^T$
    if $\max\{\max \widetilde{c}^N, \max \widetilde{c}^T\} > \delta_V$ **then** ▷ Violation state
        Select highest-priority violated nuisance/task rows subject to caps $K_N = 48$, $K_T = 16$
        Construct G_{viol}
        Compute d_R using Eqs. equation 94 –equation 95
        Search $\alpha_R \in \{1, \frac{1}{2}, \frac{1}{4}\}$
        if maximum actual violation decreases by at least 10^{-4} **then**
            $\theta \leftarrow \theta + \alpha_R d_R$
            Roll back provisional Adam moments
        else
            Restore θ and roll back provisional Adam moments
        end if
    else
        Construct active set $\mathcal{A}_t = \{k : \widetilde{c}_k \geq -\delta_{\mathrm{act}}\}$
        if $\mathcal{A}_t = \varnothing$ **then** ▷ Free state
            $d^\star \leftarrow d_A$
        else ▷ Boundary-correction state
            Construct active constraint Jacobian G_t
            Solve Eq. equation 88 using Hildreth NNQP
            $d^\star \leftarrow d_A - G_t^\top \lambda^\star$
        end if
        Search $\alpha \in \{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$
        if both constraint families satisfy the feasible-or-improving guard **then**
            $\theta \leftarrow \theta + \alpha d^\star$
            Commit provisional Adam moments
        else
            Restore θ
            Roll back provisional Adam moments
        end if
    end if
    Update violation memory using current positive residuals
    if $t \bmod K_{\mathrm{audit}} = 0$ **then**
        Audit the full anchor pool
        Add violations with $\widetilde{c}_k > 2\epsilon_{\mathrm{num}}$ to $\mathcal{M}$
    end if
end for
return θ

E.9 Full Post-Intervention Solver

Algorithm 2 summarizes the complete post-intervention optimizer. The post stage is run for 300 optimization steps while $P^\star$ remains fixed.

E.10 Frozen Post-Intervention Hyperparameters

The post-intervention solver uses the following frozen values:

Quantity	Meaning	Value
η_{post}	Adam learning rate	3×10^{-6}
ρ_{TC}	task-common gradient ratio	0.25
ε	normalized constraint slack	0.10
δ_{act}	active-set width	1.5ε
δ_V	restoration threshold	$\varepsilon/2 = 0.05$
κ	restoration contraction factor	0.5
K_N	nuisance-row cap	48
K_T	task-row cap	16
ϵ_{num}	actual-value tolerance	3×10^{-3}
N_G	rotating guard window	48
$\lvert\mathcal{M}\rvert_{\max}$	violation-memory capacity	64
N_{vmem}	remembered quartets merged per step	24
K_{audit}	full-pool audit period	10 steps
NNQP sweeps	Hildreth solver limit	300
QP ridge	numerical regularization	$10^{-12}\,\text{tr}(G_t G_t^\top)$
post steps	adaptation horizon	300

F Experimental Details

This section provides the benchmark construction, counterfactual evidence, baseline configurations, training settings, and evaluation protocol used throughout our experiments. Method-specific optimization details are described separately in Appendices C and E.

F.1 Benchmark Construction and Data Splits

ColoredMNIST. We use the IRM-style ColoredMNIST construction. Digits are converted into a binary classification task according to $Y = \mathbb{1}[\text{digit} \geq 5]$, followed by 25% label noise. A binary color attribute is then correlated with the noisy label, with environment-specific color-flip probabilities of 0.2 and 0.1 for the two training environments. Each training environment contains $20{,}000$ examples. Validation environments follow the same correlation settings, while the held-out OOD environment uses a color-flip probability of 0.9.

Evaluation groups are defined by the Cartesian product of the binary target and color attribute, yielding four groups. Images are scaled to $[0, 1]$ and resized to 224×224 to support the common ResNet-50 architecture.

Waterbirds. We use the standard `waterbird_complete95_forest2water2` benchmark, where bird class is spuriously correlated with background type in the training distribution. The released train, validation, and test split assignments are retained. Groups are defined by bird class $\{\text{waterbird}, \text{landbird}\}$ and background $\{\text{water}, \text{land}\}$.

For constructing the counterfactual training assets used by our method, the minority-group training examples are replaced by background-swapped composites while preserving the original group counts. This keeps the group imbalance and spurious-correlation structure of the original training distribution unchanged.

Images are resized to 256 pixels along the shorter side, center-cropped to 224×224, and normalized using ImageNet statistics.

UrbanCars. UrbanCars contains two independently varying nuisance factors: background (BG) and co-occurring object (CoObj). Both are correlated with the target car class at probability 0.95 in the training distribution. We follow the official Whac-A-Mole data construction and use the provided validation and test protocols.

The validation set contains eight balanced cells defined by

$$(Y, B, C) \in \{0, 1\}^3,$$

where Y denotes the target class, B the background factor, and C the co-occurring-object factor. Each validation cell contains 125 examples. The official deterministic ResNet-50 preprocessing is used for both training and evaluation.

Across all benchmarks, development decisions are made using training and validation data only; the final test splits are kept separate from method development.

F.2 Counterfactual Evidence Construction

Our method receives label-preserving counterfactual evidence constructed from training assets. We maintain separate registries for counterfactual evidence used during optimization and for held-out counterfactual examples used only for diagnostic evaluation. Views within the same counterfactual set are processed with deterministic transformations to avoid introducing unrelated augmentation noise into the measured representation differences.

ColoredMNIST. For each selected digit, the counterfactual view is obtained by swapping the red and green color channels while leaving the underlying digit unchanged. This produces an exact label-preserving intervention on the specified color factor. The training registry contains $2{,}400$ counterfactual pairs, with an additional $1{,}000$ pairs reserved for held-out diagnostics.

Waterbirds. Counterfactual pairs preserve the foreground bird and replace only the background using the available segmentation masks. Foreground identities are disjoint between the training and held-out counterfactual registries so that held-out evaluation measures transfer of background invariance to unseen foreground instances. We use $2{,}400$ training pairs and $1{,}915$ held-out pairs.

UrbanCars. For each selected car instance, we construct a complete 2×2 factorial orbit over background and co-occurring-object factors,

$$\mathcal{O}_i = \left\{ x_i^{00}, x_i^{10}, x_i^{01}, x_i^{11} \right\}.$$

The foreground car is identical across all four views, while only BG and CoObj are changed. This construction provides separate counterfactual variations for background, co-occurring object, and their joint intervention. The corresponding factorial representation contrasts are defined in Appendix B.

The training registry contains $6{,}000$ quartets and is used by the post-intervention training and constraint-evaluation streams. A separate set of $2{,}000$ quartets is reserved for factor-wise diagnostic evaluation.

Explicit counterfactual correspondences constructed for our method are not provided to ERM, LISA, or MatchDG; each baseline retains its native supervision interface.

F.3 Baseline Implementations

We compare against ERM, LISA, and MatchDG. Each method is trained using its native objective and supervision structure rather than being reformulated to use the explicit counterfactual correspondences available to our method. Within each benchmark, all final checkpoints are evaluated using the same data split, preprocessing, and metric implementation.

ERM. ERM minimizes the standard cross-entropy objective and serves both as a baseline and as the initialization for our intervention pipeline. We retain the benchmark-specific model-selection protocol used in the corresponding training configuration.

LISA. LISA is trained using its native augmentation and attribute-supervision mechanism. For UrbanCars, which contains two nuisance factors, we evaluate two factor-specific configurations: LISA-BG and LISA-CoObj. These variants target background and co-occurring-object correlations, respectively.

MatchDG. MatchDG is trained using its native correspondence-learning formulation. We use the benchmark-appropriate domain construction without supplying the explicit counterfactual pairs or quartets used by our method.

F.4 Training and Hyperparameter Configuration

All primary experiments use a ResNet-50 backbone with a two-class classification head. Our method begins from a converged ERM model and applies the sequence

$$\text{ERM} \to \text{Reorganization} \to \text{Projection} \to \text{Post-intervention Adaptation}.$$

The intervention is applied at the output of ResNet-50 `layer3`, whose channel dimension is 1024. Once the intervention basis $B^\star$ is committed, the corresponding projector

$$P^\star = I - B^\star B^{\star\top}$$

is fixed for all subsequent optimization and inference.

During the reorganization stage, we use SGD with learning rate 10^{-3}, momentum 0.9, weight decay 10^{-4}, and a cosine learning-rate schedule. The classification and counterfactual batch sizes are 128 and 48, respectively. The main geometry coefficients are

$$\lambda_{\text{pair}} = 0.2, \qquad \lambda_{\text{conc}} = 0.16.$$

The detailed reorganization objective, adaptive-rank procedure, safety constraints, and commit criteria are given in Appendix C.

Following the hard intervention, the model is adapted for a fixed 300 optimization steps using Adam with learning rate 3×10^{-6} and $(\beta_1, \beta_2) = (0.9, 0.999)$. The post-intervention task-common coefficient is calibrated once at the intervention state from the relative gradient magnitudes rather than selected through an additional hyperparameter sweep. The complete constrained adaptation procedure is described in Appendix E.

For the proposed method, the reported checkpoint is the final state after the fixed post-intervention adaptation horizon; intermediate checkpoints are not selected based on validation robustness.

F.5 Evaluation Metrics and Reporting Protocol

Predictive performance. For evaluation groups $\mathcal{G}$, worst-group accuracy is

$$A_{\text{WG}} = \min_{g \in \mathcal{G}} A_g.$$

For multi-seed experiments, worst-group accuracy is first computed separately for each seed and then aggregated across seeds.

For ColoredMNIST, average accuracy is the unweighted mean over the four target–color groups. For Waterbirds, it is the unweighted mean over the four bird–background groups. For UrbanCars, we report

$$A^8_{\text{WG}} = \min_{y,b,c} A_{ybc}$$

and adjusted accuracy

$$A_{\text{Adj}} = \frac{1}{8} \sum_{y,b,c} A_{ybc}.$$

For factor-level UrbanCars evaluation, the eight cells are additionally grouped into aligned, BG-conflict, CoObj-conflict, and double-conflict strata.

Counterfactual diagnostics. Counterfactual sensitivity is measured only on held-out counterfactual registries. We report prediction flip rate,

$$\text{Flip} = \mathbb{P}\left[\hat{y}(x) \neq \hat{y}(x^{\text{cf}})\right],$$

and, where applicable, the mean absolute change in decision margin,

$$D_{\text{held}} = \mathbb{E}\left|m(x^{\text{cf}}) - m(x)\right|.$$

For UrbanCars, these diagnostics are computed separately for BG, CoObj, and joint interventions.

Aggregation and reporting. Our ColoredMNIST results are aggregated over six committed runs (three random seeds with two independent repetitions), while Waterbirds and UrbanCars results are aggregated over three seeds. Unless otherwise stated, we report mean and standard deviation across these runs.

All final models are evaluated through a common benchmark-specific evaluator. Training counterfactual registries may be used by the proposed optimization procedure, whereas held-out counterfactual registries and final test data are excluded from training, hyperparameter selection, threshold selection, and early stopping.