FLATNESS AND GRADIENT ALIGNMENT ARE BOTH NECESSARY: SPECTRAL-AWARE GRADIENT-ALIGNED EXPLORATION FOR MULTI-DISTRIBUTION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Sharpness-aware and gradient-alignment methods have been shown to improve generalization, however each family of methods targets a single geometric property of the loss landscape, while ignoring the other. In this paper, we show that this omission is structurally unavoidable and that both *flatness* and *gradient alignment* should be considered in multi-distribution learning settings. Specifically, we derive an excess-risk decomposition that yields two additive leading-order terms: (i) an alignment term, controlled by the trace of $\bar{H}^{-1}\Sigma_g$ and (ii) a curvature term, controlled by $\bar{H}$, where $\bar{H}$ is the average Hessian and Σ_g is the covariance of the gradient across distributions. Notably, $\bar{H}$ appears inverted in one and non-inverted in the other. We further show, via a counterexample, that neither quantity bounds the other in general, so no algorithm targeting only one term can guarantee low excess risk. Motivated by this decomposition, we propose SAGE (Spectral-Aware Gradient-Aligned Exploration) that targets both terms. The curvature component replaces SAM's gradient-scaled perturbation with the polar factor of each layer's gradient matrix, computed via Newton–Schulz iteration, so that the ascent step probes all directions with similar magnitude. On the other hand, the alignment component injects isotropic noise at the descent step, the magnitude of which scales with cross-distribution gradient disagreement. Experiments on five domain-generalization and two multi-task learning benchmarks show that the proposed method establishes a new state-of-the-art on DomainBed and acts as a general-purpose improvement to base MTL solvers, remaining competitive with, or even surpassing, state-of-the-art methods[1].

1 INTRODUCTION

Understanding the geometry of loss landscapes Li et al. (2018a); Fort & Jastrzebski (2019) in overparameterized neural networks has become a central lens for interpreting optimization behavior and why certain solutions generalize better than others Wu et al. (2017); Li et al. (2018a); Fort & Jastrzebski (2019); Rangamani et al. (2020); Keskar et al. (2016); Jiang et al. (2019). Specifically, two geometric properties have emerged as particularly influential: the *flatness* Foret et al. (2021); Kwon et al. (2021); Kim et al. (2022); Zhang et al. (2024); Ban et al. (2025) of the loss surface around a minimum, and the *alignment* Mansilla et al. (2021); Shi et al. (2022); Le & Woo (2024); Ballas & Diou (2025); Yu et al. (2020); Liu et al. (2021a); Senushkin et al. (2023) of gradients computed across different training distributions, e.g., domains in the case of Domain Generalization (DG) Wang et al. (2022); Zhou et al. (2023), or tasks, in Multi-Task Learning (MTL) Zhang & Yang (2022).

Even though both flatness and gradient alignment have been studied extensively in isolation, existing methods typically pursue each property as a standalone objective Foret et al. (2021); Yu et al. (2020); Mansilla et al. (2021); Shi et al. (2022); Li & Giannakis (2023); Zhang et al. (2025); Ballas & Diou (2025), while their interplay has received limited formal treatment Wang et al. (2023); Kim et al. (2022). Intuitively, a flat minimum may sit in a region where distribution-specific gradients conflict, while an aligned minimum may lie in a sharp basin where small perturbations inflate the loss. We argue that neither property alone is sufficient and that robust generalization benefits from minima that

[1]Code provided as supplemental material and will be published upon acceptance.

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.

are simultaneously flat across directions and exhibit improved gradient agreement across training sub-objectives.

To justify this claim, we derive an excess-risk decomposition (Theorem 1) that yields two additive leading-order terms: (i) an alignment term scaling as $\mathrm{tr}(\bar{H}^{-1}\Sigma_g)$, where $\bar{H}$ is the average curvature and Σ_g is the gradient covariance matrix, across distributions and (ii) a curvature term scaling as $\mathrm{tr}(\bar{H})$. $\bar{H}$ appears *inverted* in the first term and *non-inverted* in the other and, as a result, targeting only one of the two during optimization can leave the other unconstrained. We formalize this independence in Counterexample 1, which shows that neither quantity bounds the other, even in the simple quadratic family.

Motivated by this decomposition, and based on the fact that first-order optimization methods (including first-order flatness-aware methods) ignore curvature, we propose **SAGE** (Spectral-Aware Gradient-Aligned Exploration) a method that jointly targets both curvature and gradient agreement via principled heuristics. For the curvature term ($\mathrm{tr}(\bar{H})$), SAGE replaces SAM's Foret et al. (2021) gradient-scaled ascent perturbation with the polar factor UV^T of each layer's gradient matrix, computed efficiently via Newton–Schulz iteration Bernstein & Newhouse (2024), scaled by the layer Frobenius norm so that all spectral directions are probed with comparable magnitude [2] (details in Section 3). For the alignment term, SAGE injects isotropic Gaussian noise at the descent step with magnitude scaling with cross-distribution gradient disagreement, biasing the optimizer away from high-conflict regions. Empirical evaluation on DG Wang et al. (2022); Zhou et al. (2023) and MTL Zhang & Yang (2022) benchmarks indicates that SAGE compares favorably to both sharpness-aware and gradient-alignment baselines across seven datasets.

Our key contributions can be summarized as follows:

- We derive an excess-risk decomposition for multi-distribution learning (Theorem 1) that splits the leading-order risk into two additive terms, an alignment term scaling as $\mathrm{tr}(\bar{H}^{-1}\Sigma_g)$ and a curvature term scaling as $\mathrm{tr}(\bar{H})$, with $\bar{H}$ appearing inverted in one and non-inverted in the other.
- We provide a separation result (Counterexample 1) showing that the two terms are independently controllable for quadratic losses, showing that even in a simple scenario, targeting each property in isolation may prove insufficient.
- Based on the above, we propose SAGE, a method targeting both terms of the decomposition; a spectral perturbation (curvature) and a gradient alignment-driven noise injection (alignment).
- We evaluate our proposed method on five DG and two MTL benchmarks, demonstrating competitive results against both sharpness-aware and gradient-alignment baselines.

2 Theoretical Analysis: Flatness and Gradient Alignment in Multi-Distribution Learning

2.1 Setup and Notation

Let $\mathcal{X}$ be the input space and $\mathcal{Y}$ the label space. Let e be a joint distribution P^e_{XY} over $\mathcal{X} \times \mathcal{Y}$ (denoted as *environment* in Arjovsky et al. (2019)). For example, different distributions/environments may correspond to different *domains* in the domain generalization literature or *tasks* in the multi-task literature. Let $\mathcal{E}$ be a set of distributions and $\mathcal{P}$ a meta-distribution over $\mathcal{E}$. For each $e \in \mathcal{E}$, let $\mathcal{L}_e : \Theta \to \mathbb{R}$ denote a per-distribution loss, where $\Theta \subseteq \mathbb{R}^d$ is an open parameter space. The *population risk* is:

$$R(\theta) := \mathbb{E}_{e\sim\mathcal{P}}[\mathcal{L}_e(\theta)], \tag{1}$$

and the *empirical multi-distribution risk* over K iid training distributions $e_1, \ldots, e_K \sim \mathcal{P}$ is $\hat{R}(\theta) := \frac{1}{K}\sum_{k=1}^{K} \mathcal{L}_{e_k}(\theta)$. This is the standard objective minimized by ERM. At any base point $\theta_0 \in \Theta$ we

[2] We do not explicitly address the decomposition via second-order approximations Kim et al. (2022); Zhang et al. (2023); Shin et al. (2025) to avoid additional computational burden. In addition, the same orthogonalization in the gradient matrix singular basis has been used in recent optimizers Jordan et al. (2024); Gupta et al. (2018) and corresponds to the steepest-descent step under the spectral norm Bernstein & Newhouse (2024).

introduce the per-distribution gradient and Hessian,

$$g_e := \nabla_\theta \mathcal{L}_e(\theta_0), \qquad H_e := \nabla^2_\theta \mathcal{L}_e(\theta_0), \tag{2}$$

and the following cross-distribution statistics,

$$\bar{g} := \mathbb{E}_e[g_e], \qquad \bar{H} := \mathbb{E}_e[H_e], \qquad \Sigma_g := \mathrm{Cov}_e(g_e) = \mathbb{E}_e\left[(g_e - \bar{g})(g_e - \bar{g})^T\right]. \tag{3}$$

$\bar{H}$ is the average curvature targeted by sharpness-aware methods, Σ_g is the cross-distribution gradient covariance targeted (implicitly or explicitly) by gradient-alignment methods, and $\bar{g}$ is the expected gradient direction, vanishing ($\bar{g} = 0$) at stationary points of R.

Assumption 1 (Regularity)**.** Each $\mathcal{L}_e$ is three times continuously differentiable (i.e., C^3) in θ, with third derivatives uniformly bounded in a neighborhood of $\theta^\star := \arg\min_\theta R(\theta)$, and $\bar{H}(\theta^\star) \succ 0$ (Positive-Definite). The gradients $g_e(\theta^\star)$ have finite second moments under $\mathcal{P}$.

2.2 Excess-Risk Decomposition

The main result of our analysis is a decomposition of the expected excess risk of the first-order empirical minimizer into two additive terms, one of which depends on Σ_g through $\bar{H}^{-1}$ and the other of which depends on $\bar{H}$ directly.

Theorem 1 (Multi-distribution excess-risk decomposition)**.** *Under Assumption 1, let $\hat{\theta}$ denote the minimizer of $\hat{R}$ over K iid training distributions, and let $u \sim \mathcal{N}(0, \sigma^2 I)$ denote isotropic Gaussian noise of scale $\sigma > 0$. Then, for a parameter $\theta = \hat{\theta} + u$:*

$$\mathbb{E}[\mathbb{E}_u[R(\theta)]] - R(\theta^\star) \;=\; \underbrace{\frac{1}{2K}\,\mathrm{tr}\big(\bar{H}^{-1}\Sigma_g\big)}_{\text{alignment term}} \;+\; \underbrace{\frac{\sigma^2}{2}\,\mathrm{tr}(\bar{H})}_{\text{curvature term}} \;+ O(K^{-3/2}) \;+\; O(\sigma^3), \tag{4}$$

where the outer expectation is over the iid sampling of training distributions, $\bar{H} := \nabla^2 R(\theta^\star)$, and $\Sigma_g := \mathrm{Cov}_e(\nabla \mathcal{L}_e(\theta^\star))$.

Proof sketch. To prove Theorem 1, we expand R quadratically around $\theta^\star$, where the linear term vanishes since $\bar{g}(\theta^\star) = 0$. We then express $\hat{\theta} - \theta^\star = -\bar{H}^{-1}\hat{g} + O_P(K^{-1})$ via the implicit-function argument standard in M-estimator asymptotics Van der Vaart (2000). Substituting into the quadratic and taking expectation over the iid distribution sampling, with $\mathrm{Cov}(\hat{g}) = \frac{1}{K}\Sigma_g$, yields the alignment term. Separately, expanding R around $\hat{\theta}$ under random perturbations $u \sim \mathcal{N}(0, \sigma^2 I)$, e.g. due to optimization noise, produces the curvature term. Full proof in Appendix B.2.

Remark 1. The two terms exhibit a structural trade-off, i.e., $\bar{H}$ appears inverted in the alignment term and non-inverted in the curvature term. As a result, minimizing one term by reshaping the Hessian strictly inflates the other. Considered in isolation, the alignment term implies that excess risk is minimized by maximizing $\bar{H}$. The curvature term mitigates this by explicitly penalizing excessive sharpness, ensuring a balanced optimization objective.

2.3 Decoupling of Flatness and Alignment

Theorem 1 gives an additive decomposition into a flatness-controlled term and an alignment-controlled term. The natural next question is whether the two terms are linked, i.e. an inequality of the form "small $\mathrm{tr}(\bar{H})$ implies small $\mathrm{tr}(\bar{H}^{-1}\Sigma_g)$" that would allow a single criterion to bound both. The following counterexample shows that such link may not exist, even in simple settings.

Counterexample 1 (Decoupling of flatness and gradient alignment)**.** Consider the family of multi-distribution learning problems with per-distribution losses $\mathcal{L}_e(\theta) = \frac{1}{2}\theta^T A \theta + b_e^T \theta, \qquad \mathbb{E}_e[b_e] = 0,$ parameterized by a symmetric PD matrix $A \in \mathbb{R}^{d\times d}$ and a collection of linear coefficients $\{b_e\}$ satisfying $\mathbb{E}_e[b_e] = 0$. For every $M > 0$, there exist instances such that:

1. **Flat but misaligned:** $\mathrm{tr}(\bar{H}) \le M^{-1}$ and $\mathrm{tr}(\bar{H}^{-1}\Sigma_g) \ge M$.
2. **Aligned but sharp:** $\mathrm{tr}(\bar{H}^{-1}\Sigma_g) \le M^{-1}$ and $\mathrm{tr}(\bar{H}) \ge M$.

In particular, neither $\mathrm{tr}(\bar{H})$ nor $\mathrm{tr}(\bar{H}^{-1}\Sigma_g)$ can be bounded above by any function of the other alone.

Proof sketch. For the quadratic family, $\bar{H} = A$ depends only on A and $\Sigma_g = \mathbb{E}_e[b_e b_e^T]$ depends only on $\{b_e\}$, so the two quantities can be specified independently. Setting $A = \lambda I$ and choosing $\{b_e\}$ supported on a single direction with prescribed variance yields both constructions, by an appropriate choice of λ. Full proof details in Appendix B.3.

Remark 2. This decoupling result in the quadratic setting indicates that an algorithm targeting only one of the two quantities may increase the generalization error resulting from the other. Thus, both flatness and alignment terms need to be considered during optimization.

To make Theorem 1 and Counterexample 1 concrete, we construct in Appendix C a two-dimensional learning problem in which the support of Σ_g coincides with the flat eigendirection of $\bar{H}$. The example realizes and exhibits two distinct failure modes along orthogonal eigendirections.

3 Spectral-Aware Gradient-Aligned Exploration

In this section, we describe SAGE, a single algorithm that targets the excess-risk decomposition terms of Theorem 1 via two separate principled heuristic mechanisms. The two components introduced below are tractable surrogates designed to probe each term and are introduced to avoid 2nd-order computational overhead.

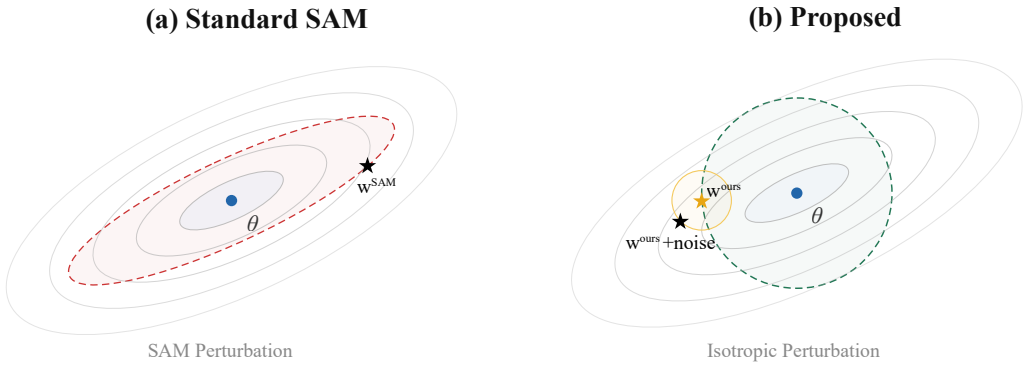


Figure 1: **Comparison of standard SAM and our proposed method.** (a) Standard SAM computes the perturbation as $\epsilon_{\text{SAM}} = \rho\,\nabla_\theta \mathcal{L}/\|\nabla_\theta \mathcal{L}\|_2$, determined solely by the gradient direction (dashed ellipse), and the descent step lands at w^{SAM} (★). (b) We replace the gradient with its orthogonal polar factor ($G \to UV^T$), setting all singular values to one and scaling by the layer norm, yielding a perturbation $\epsilon' = \rho\|W\|_F \cdot UV^T$ that probes all directions with equal magnitude (dashed circle). After the descent step lands at w^{ours} (★), isotropic noise scaled by cross-distribution gradient disagreement perturbs the update (orange circle), discouraging convergence to regions of gradient conflict.

3.1 Targeting the Curvature Term

Motivated by the curvature term in Theorem 1 we follow the logic of SAM Foret et al. (2021) and SAM-like Zhuang et al. (2022); Wang et al. (2023); Kwon et al. (2021); Li & Giannakis (2023); Zhang et al. (2024); Ban et al. (2025) methods, which have empirically proven to be effective in multi-distribution settings, but propose an alternative for the ascent perturbation step. Specifically, SAM-like methods derive the perturbation by approximating the inner maximization $\max_{\|\epsilon\| \le \rho} \mathcal{L}(\theta + \epsilon)$ with a first-order Taylor expansion, $\mathcal{L}(\theta + \epsilon) \approx \mathcal{L}(\theta) + \epsilon^T \nabla \mathcal{L}(\theta)$, whose maximizer over the L_2 ball admits the closed form $\epsilon_{\text{SAM}} = \rho\,\nabla \mathcal{L}/\|\nabla \mathcal{L}\|_2$. As apparent, the above perturbation direction ignores curvature information, is determined by the gradient vector alone, and is therefore probed by dominating gradients during training[3]. To address the above limitation, we propose replacing SAM's perturbation with an isotropic perturbation during the ascent step, inspired by preconditioning

[3]Even though iterated SAM dynamics have been shown to implicitly minimize $\lambda_{\max}(\bar{H})$ Wen et al. (2023), it is a different functional of the Hessian from the trace $\text{tr}(\bar{H})$ identified by Theorem 1 as the relevant target.

methods such as Shampoo Gupta et al. (2018) and Muon Jordan et al. (2024). These methods have empirically shown to achieve effective preconditioning by orthogonalizing each $2D$ weight (i.e., parameter) matrix (e.g., layer) of a network independently. We follow the same heuristic to expand the exploration of additional perturbation directions of SAM.

Spectral perturbation via orthogonalized gradients. For each weight matrix W with gradient $G = \nabla_W \mathcal{L}$, we compute the orthogonal polar factor $Q = UV^T$, where $G = U\Sigma V^T$ is the SVD of G. This amounts to setting every singular value of the gradient to one, so that $Q = \sum_{i=1}^{r} u_i v_i^T$ retains the directional structure of G, i.e. the left and right singular vectors, while equalizing the magnitude across all singular directions. A perturbation along Q therefore probes the loss surface with equal magnitude in every spectral direction of the gradient. Since computing the exact polar factor via SVD adds a computational burden of $\mathcal{O}(\min\{m, n\} \cdot mn)$ per weight matrix, we instead approximate Q with the Newton–Schulz iteration Bernstein & Newhouse (2024); Guo & Higham (2006); Björck & Bowie (1971); Kovarik (1970), which converges cubically to the orthogonal factor using only matrix multiplications:

$$X_0 = \frac{G}{\|G\|_F}, \qquad X_{k+1} = \tfrac{1}{2}\, X_k\big(3I - X_k^T X_k\big). \tag{5}$$

After T iterations (T=5 suffices in practice), $X_T \approx UV^T$. Because each step involves only two matrix products of the same shape as G, the overhead on modern GPU hardware is negligible. For bias vectors and other one-dimensional parameter groups, where matrix polar decomposition is not applicable, we fall back to the standard L_2-normalized perturbation $\epsilon = \rho\, g/\|g\|_2$, consistent with vanilla SAM.

Scale-adaptive radius. Furthermore, to address the limitation from the well-documented interaction between SAM's fixed perturbation radius and the scale invariance of modern architectures Dinh et al. (2017), we scale the orthogonalized radius by the Frobenius norm of each weight matrix $\|W\|_F$, following the intuition of adaptive sharpness Kwon et al. (2021). The final resulting spectral perturbation for weight matrix W is therefore $\epsilon_W^{\text{spec}} = \rho\, \|W\|_F\, UV^T$, where U, V are from the SVD of $G = \nabla_W \mathcal{L}$. Under this formulation, every singular direction receives a perturbation of identical magnitude $\rho\|W\|_F$, and the perturbation rescales with the current parameter norm, ensuring that the sharpness measure remains invariant to reparameterizations of the form $W \mapsto \alpha W$. More details in Appendix D.

3.2 Targeting the Alignment Term

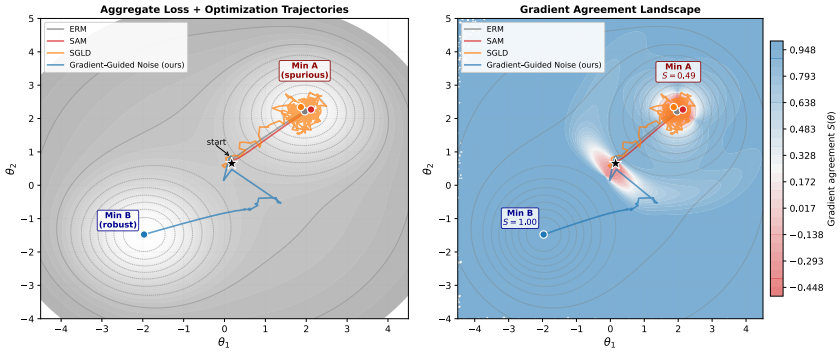


Figure 2: *Left:* Aggregate loss contours with optimization trajectories from a shared starting point ($\star$). ERM (grey), SAM (red), and SGLD (orange) converge to Minimum A, whereas only gradient-aligned noise (blue, ours) escapes to Minimum B. *Right:* The same trajectories overlaid on the gradient agreement landscape $S(\theta)$, where blue and red indicate high and low cross-domain gradient agreement, respectively.

Motivated by the alignment term in Theorem 1, we introduce a gradient-agreement-based regularizer at the descent step. Since the cross-distribution covariance Σ_g is intractable to estimate and differentiate through at the scale of modern networks, we replace it with a tractable scalar proxy of cross-distribution gradient *disagreement* and use it to modulate noise that biases the optimizer away from high-conflict regions, rather than minimizing $\mathrm{tr}(\bar{H}^{-1}\Sigma_g)$ directly.

In several settings, such as domain generalization Wang et al. (2022); Zhou et al. (2023); Ballas & Diou (2024), multi-task learning Zhang & Yang (2022), the training signal decomposes into K distinct distributions, each corresponding to a different domain, data source, or task. The overarching goal is to learn a model that performs well on all sub-tasks and unseen OOD data. Intuitively, *directions of parameter update that are beneficial across all distributions are more likely to generalize to new ones*. This intuition has been formalized under gradient alignment Yu et al. (2020); Mansilla et al. (2021); Shi et al. (2022); Le & Woo (2024), where it has been shown that under certain assumptions Ballas & Diou (2025) it is a necessary condition for domain-invariance. Specifically, when gradients of different tasks or domains point in similar directions, the shared gradient direction captures features that are invariant across distributions. On the contrary, when they conflict the model is being pulled toward domain- or task-specific shortcuts.

Formally, as defined in Section 2.1, given distribution loss functions $\mathcal{L}_{e_k}(\theta)$ for K *source* distributions $e_1, \ldots, e_K \sim \mathcal{P}$, where in our case a distribution or environment e_k may represent a domain, a task, or any other meaningful partition of the training data, we measure the agreement among gradients $g_{e_k} = \nabla_\theta \mathcal{L}_{e_k}(\theta)$ using the average pairwise cosine similarity, $S(\theta) = \frac{2}{K(K-1)} \sum_{i<j} \frac{\langle g_{e_i}, g_{e_j} \rangle}{\|g_{e_i}\| \, \|g_{e_j}\|}$.

When $S(\theta) \approx 1$, all distributions agree on the update direction, whereas when $S(\theta) \leq 0$, gradients conflict, indicating that the optimizer is navigating a region where domain- or objective-specific features dominate. Rather than modifying the gradient direction directly (as in PCGrad Yu et al. (2020)) and to avoid further computational burden via second-order approximation techniques, we adopt a noise-injection strategy inspired by the recently proposed Gradient-Guided Annealing (GGA) framework Ballas & Diou (2025). We inject isotropic Gaussian noise into the gradient at the descent step, with magnitude inversely proportional to the gradient agreement $\beta = \gamma\,(1 - S(\theta))$, where $\gamma > 0$ is a hyperparameter controlling the maximum noise scale. The updated gradient then becomes $g' = \nabla_\theta \mathcal{L}(\theta + \epsilon) + \beta\,\xi, \quad \xi \sim \mathcal{N}(0, I)$.

The intuition behind this design is twofold. When gradients are well-aligned ($S \approx 1$) the additive noise vanishes, whereas when gradients conflict ($S \ll 1$) the noise discourages update directions that benefit some distributions at the expense of others. We illustrate the behavior between ERM, SAM Foret et al. (2021), SGLD[4] Welling & Teh (2011); Teh et al. (2016), and the proposed noise mechanism in Fig. 2.

3.3 Spectral-Aware Gradient-Aligned Exploration

We now describe the complete SAGE algorithm, which unifies the spectral perturbation and gradient-alignment into a single training procedure. At each iteration, SAGE: (i) computes per-distribution gradients g_{e_k} and aggregate gradient from a single forward pass, (ii) evaluates the gradient agreement $S(\theta)$ and sets the noise scale $\beta = \gamma(1 - S(\theta))$, (iii) constructs the spectral perturbation by orthogonalizing each weight matrix's gradient via the Newton-Schulz iteration and scaling by $\rho\|W\|_F$, (iv) performs a forward-backward pass at the perturbed point $\theta + \epsilon$, and (v) restores the clean weights, adds isotropic gradient alignment-driven Gaussian noise $\beta\,\mathcal{N}(0, I)$ to the adversarial gradient, and finally applies the base optimizer update step. A detailed summary in pseudocode of the SAGE algorithm is provided in Appendix H, whereas a conceptual illustration is provided in Fig. 1.

4 Related Work

In this section we situate SAGE within the literature of DG and MTL, and provide an overview of the most related works. An extended discussion regarding the landscape of the two fields is provided in Appendix A.

Sharpness-aware minimization. The observation that flat minima generalize better than sharp ones Keskar et al. (2016) motivated Sharpness-Aware Minimization (SAM) Foret et al. (2021), which performs a gradient ascent step followed by a descent step so that the optimizer preferentially converges to flat regions of the loss landscape. Several refinements have since been proposed, such as GSAM Zhuang et al. (2022) which introduces a surrogate gap objective, while ASAM Kwon et al. (2021) adapts perturbation magnitudes to the parameter scale, and SAGM Wang et al. (2023)

[4]SGLD or Stochastic Gradient Langevin Dynamics adds constant isotropic noise during optimization.

additionally encourages the clean-loss and perturbed-loss gradients to remain aligned, jointly minimizing sharpness and the surrogate gap. In the DG setting, DISAM Zhang et al. (2024) showed that SAM's perturbation is biased toward the domain with the largest gradient and proposed a domain-loss variance constraint to counteract this effect, while SWAD Cha et al. (2021) achieves flatness passively by averaging model weights along the training trajectory. In MTL, SAMO Ban et al. (2025) combines global and per-task perturbation information through a zeroth-order estimator. An important note is the majority of all sharpness-aware algorithms incorporate the perturbation method of the initially proposed SAM Foret et al. (2021).

Gradient-alignment methods. A complementary line of work uses agreement among per-environment or per-task gradients as an indicator of domain invariance. In DG, Fish Shi et al. (2022) approximately maximizes the inner product of per-domain gradients, while Mansilla et al. Mansilla et al. (2021) adapt gradient surgery Yu et al. (2020) to mute gradient dimensions where domains disagree in sign. More recently, GGA Ballas & Diou (2025) searches for parameter configurations with well-aligned domain gradients via a simulated-annealing-inspired procedure. In MTL, PCGrad Yu et al. (2020) projects conflicting task gradients, CAGrad Liu et al. (2021a) maximizes worst-case per-task improvement, and Nash-MTL Navon et al. (2022) frames the problem as a bargaining game. FAMO Liu et al. (2023) circumvents the $\mathcal{O}(K)$ per-iteration cost of gradient manipulation by working in log-loss space with amortized weight updates. However, these methods enforce gradient alignment without considering the curvature of the resulting solution.

Positioning of SAGE. The most closely related prior work is SAGM Wang et al. (2023), which also augments SAM with a gradient-matching term. However, SAGM's matching aligns the clean-loss and perturbed-loss gradients, both computed on the *aggregate* data, and does not involve per-environment gradient information. SAGE differs in two respects: (i) its spectral perturbation replaces SAM's gradient-scaled ascent with the polar factor of each layer's gradient, computed via Newton–Schulz iteration and scaled by the layer's Frobenius norm, yielding a reparameterization-invariant sharpness probe; and (ii) its noise injection at the descent step is scaled by the degree of cross-environment gradient conflict, targeting the alignment term $\mathrm{tr}(\bar{H}^{-1}\Sigma_g)$ identified in Theorem 1. Unlike methods that modify the gradient direction (PCGrad, Fish) or solve auxiliary optimization problems (Nash-MTL, FAMO), SAGE's noise injection is lightweight and requires no per-environment gradient storage beyond what is needed for computing pairwise cosine similarities. SAGE can also operate as a drop-in replacement for the optimizer's ascent and descent steps, making it compatible with existing gradient manipulation methods in MTL (as demonstrated in Table 2) and with the standard DomainBed protocol in DG (Table 1).

5 Experiments

5.1 Experimental setting

Datasets and Protocol. Our method is evaluated in two different experimental settings, namely Domain Generalization (DG) Zhou et al. (2023) and Multi-Task learning Zhang & Yang (2022). For DG, we follow the protocol of the widely adopted and challenging DomainBed Gulrajani & Lopez-Paz (2021) benchmark, and conduct experiments on five image classification datasets. Specifically, we follow the leave-one-domain-out evaluation protocol for PACS Li et al. (2017), VLCS Fang et al. (2013), OfficeHome Venkateswara et al. (2017), TerraIncognita Beery et al. (2018), and DomainNet Peng et al. (2019), and report the average top-1 accuracy over 3 runs based on training-domain split validation. For the MTL setting, we follow the standard protocol in recent literature Ban et al. (2025); Liu et al. (2023); Navon et al. (2022) and evaluate SAGE on Cityscapes Cordts et al. (2016) (semantic segmentation and depth estimation), and NYU-v2 Silberman et al. (2012) (semantic segmentation, depth estimation, and surface normal prediction).

Implementation Details. For the DG experiments, we follow recent literature Li et al. (2026); Zhang et al. (2024) and finetune a CLIP Radford et al. (2021) pretrained ViT-B/16 Dosovitskiy et al. (2021) model. For the MTL setting, we follow Ban & Ji (2024); Liu et al. (2021a; 2023); Navon et al. (2022) and employ MTAN Liu et al. (2019) as the shared backbone, with task-specific attention modules built on top of SegNet Badrinarayanan et al. (2017). Dataset and training hyperparameter details are provided in Appendix E.

Table 1: **Comparison of SAGE on DomainBed**. The top out-of-domain accuracies on five domain generalization benchmarks averaged over three trials, are presented. Results of previously proposed methods are from Li et al. (2026).

Algorithm	PACS	VLCS	OfficeHome	TerraInc	DomainNet	Avg.
ERM Vapnik (1998)	95.9	81.9	84.1	56.1	59.5	75.5
IRM Arjovsky et al. (2019)	96.1	82.9	83.2	56.7	59.1	75.6
DANN Ganin et al. (2016)	96.3	81.8	83.0	56.0	58.4	75.1
CDANN Li et al. (2018c)	96.5	82.4	82.9	55.6	58.4	75.2
MMD Li et al. (2018b)	95.8	82.3	83.6	57.4	59.9	75.8
RSC Huang et al. (2020)	96.5	82.2	83.2	58.2	59.0	75.8
CORAL Sun & Saenko (2016)	96.4	82.6	83.8	57.5	59.8	76.0
IIB Li et al. (2022)	96.5	82.3	84.2	58.2	58.6	76.0
FISH Shi et al. (2022)	96.9	82.7	85.0	58.0	59.1	75.0
SAM Foret et al. (2021)	96.6	82.9	85.4	56.2	59.8	76.2
GSAM Zhuang et al. (2022)	96.6	82.9	85.6	55.4	59.8	76.1
GAM Zhang et al. (2023)	96.4	83.6	85.5	55.3	59.5	76.1
SAGM Wang et al. (2023)	96.8	82.8	85.2	58.0	59.1	76.4
GGABallas & Diou (2025)	97.2	84.1	85.7	58.8	59.8	77.1
DISAM Zhang et al. (2024)	97.1	82.7	85.4	57.3	59.8	76.5
BOA (SAM pretrained) Li et al. (2026)	**97.4**	**86.5**	86.0	60.3	60.2	78.1
SAGE (Ours)	**97.4**	85.5	**87.1**	**61.1**	**63.4**	**78.9**

Table 2: Results on NYU-v2 (3-task) and Cityscapes (2-task) datasets. The best results are highlighted in **bold** and second-best are underlined. Green plus signs (+) and Red minuses (−) indicate whether the addition of SAGE on top of base solvers improves or hinders training, respectively.

Method	NYU-v2										Cityscapes				
	Segmentation		Depth		Surface Normal					$\Delta m\% \downarrow$	Segmentation		Depth		$\Delta m\% \downarrow$
	mIoU ↑	Pix Acc ↑	Abs Err ↓	Rel Err ↓	Angle Distance ↓		Within t° ↑				mIoU ↑	Pix Acc ↑	Abs Err ↓	Rel Err ↓	
					Mean	Median	<11.25	<22.5	<30						
STL	38.30	63.76	0.6754	0.2780	25.01	19.21	30.14	57.20	69.15		74.01	93.16	0.0125	27.77	
LS	39.29	65.33	0.5493	0.2263	28.15	23.96	22.09	47.50	61.08	5.59	75.18	93.49	0.0155	46.77	22.60
SI	38.45	64.27	0.5354	0.2201	27.60	23.37	22.53	48.57	62.32	4.39	70.95	91.73	0.0161	33.83	14.11
RLW Lin et al. (2021)	37.17	63.77	0.5759	0.2410	28.27	24.18	22.26	47.05	60.62	7.78	74.57	93.41	0.0158	47.79	24.38
DWA Liu et al. (2019)	39.11	65.31	0.5510	0.2285	27.61	23.18	24.17	50.18	62.39	3.57	75.24	93.52	0.0160	44.37	21.45
UW Kendall et al. (2018)	36.87	63.17	0.5446	0.2260	27.04	22.61	23.54	49.05	63.65	4.05	72.02	92.85	0.0140	30.13	5.89
MGDA Sener & Koltun (2018)	30.47	59.90	0.6070	0.2555	24.88	19.45	29.18	56.88	69.36	1.38	68.84	91.54	0.0309	33.50	44.14
PCGrad Yu et al. (2020)	38.06	64.64	0.5550	0.2325	27.41	22.80	23.86	49.83	63.14	3.97	75.13	93.48	0.0154	42.07	18.29
GradDrop Chen et al. (2020)	39.39	65.12	0.5455	0.2279	27.48	22.96	23.38	49.44	62.87	3.58	75.27	93.53	0.0157	47.54	23.73
IMTL-G Liu et al. (2021b)	39.35	65.60	0.5426	0.2256	26.02	21.19	26.20	53.13	66.24	-0.76	75.33	93.49	0.0135	38.41	11.10
CAGrad Liu et al. (2021a)	39.79	65.49	0.5486	0.2250	26.31	21.58	25.61	52.36	65.58	0.20	75.16	93.48	0.0141	37.60	11.64
MoCo Fernando et al. (2023)	40.30	**66.07**	0.5575	0.2135	26.67	21.83	25.61	51.78	64.85	0.16	75.42	93.55	0.0149	34.19	9.90
Nash-MTL Navon et al. (2022)	40.13	65.93	**0.5261**	0.2171	25.26	20.08	28.40	55.47	68.15	-4.04	75.41	93.66	0.0129	35.02	6.82
FAMO Liu et al. (2023)	38.88	64.90	0.5474	0.2194	25.06	19.57	29.21	56.61	68.98	-4.10	74.54	93.29	0.0145	32.59	8.13
FairGrad Ban & Ji (2024)	38.80	65.29	0.5572	0.2322	24.55	18.97	30.50	57.94	70.14	-4.96	74.10	93.03	0.0135	29.92	3.90
F-MTL Phan et al. (2025)	**40.42**	65.61	0.5389	0.2121	25.03	19.75	28.90	56.19	68.72	-4.77	76.63	93.76	0.0124	31.17	1.87
SAMO-LS Ban et al. (2025)	39.59	65.72	0.5514	0.2246	27.38	22.78	24.09	49.82	63.01	2.88	**76.46**	**93.76**	0.0147	39.85	14.30
SAMO-MGDA Ban et al. (2025)	29.85	60.83	0.6111	0.2388	**24.11**	18.18	32.16	59.59	71.15	-2.19	73.28	93.26	0.0133	30.57	4.30
SAMO-FairGrad Ban et al. (2025)	39.05	65.06	0.5359	0.2137	24.43	18.79	30.98	58.35	70.42	**-6.55**	74.37	93.14	0.0129	**26.30**	**-0.62**
SAGE-LS	39.67 +	65.03 −	0.5331 +	0.2180 +	27.40 +	23.06 +	23.62 +	49.28 +	62.87 +	3.81 +	75.69 +	93.57 +	0.0146 +	42.84 +	19.92 +
SAGE-MGDA	33.24 +	61.88 +	0.5707 +	0.2266 +	24.55 +	18.99 +	30.15 +	57.91 +	70.07 +	-1.77 +	73.46 +	93.28 +	**0.0123** +	29.59 +	3.82 +
SAGE-FairGrad	40.32 +	65.34 +	0.5301 +	**0.2082** +	24.53 +	18.97 ±	30.28 −	57.44 −	69.66 −	-5.63 +	75.40 +	93.63 +	0.0129 +	29.33 +	2.63 +
SAGE-SAMO-LS	40.15 +	65.59 −	0.5508 +	0.2251 −	27.40 −	22.76 +	23.34 −	49.86 +	63.56 +	3.82 −	76.07 −	93.73 −	0.0149 −	42.15 −	18.71 −
SAGE-SAMO-MGDA	32.32 +	62.78 +	0.5651 +	0.2201 +	24.23 −	**18.10** +	**32.41** +	**59.68** +	**71.16** +	-3.66 +	73.92 +	93.20 −	0.0133 ±	32.46 −	9.85 −
SAGE-SAMO-FairGrad	39.76 +	65.80 +	0.5413 −	0.2164 −	24.34 +	18.69 +	31.01 +	58.38 +	70.46 +	-6.19 −	75.08 +	93.52 +	0.0131 −	32.02 −	4.88 −

5.2 Main Results

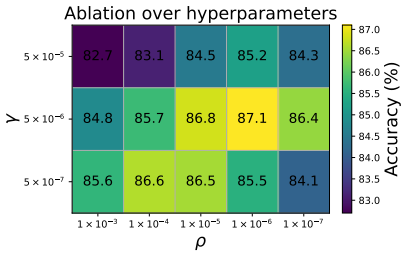
Table 1 presents the results on the DomainBed benchmark. SAGE achieves an average accuracy of 78.9%, establishing a new state-of-the-art in this particular setting. Notably, SAGE even surpasses recent sharpness-aware adaptations tailored for DG, including SAGM (76.4%), DISAM (76.5%), and the BOA regularizer (78.1%), which is applied to SAM-pretrained backbones. Extending beyond domain shift, we demonstrate SAGE's versatility on the NYU-v2 and Cityscapes MTL benchmarks (Table 2). The results illustrate that integrating SAGE improves the performance of underlying base MTL solvers. For instance, augmenting FairGrad Ban & Ji (2024) with SAGE (SAGE-FairGrad) yields improvements, notably reducing the per-task performance drop against the single-task (STL) ($\Delta m\% \downarrow$) from -4.96 to -5.63 on NYU-v2, and from 3.90 to 2.63 on Cityscapes. Similar relative gains are observed when SAGE is applied to Linear Scalarization (LS) and MGDA Sener & Koltun (2018).

Summary. Across the two benchmark settings, SAGE establishes a new state-of-the-art on the DomainBed DG benchmark (78.9% average) and acts as a general-purpose improvement for base MTL solvers. On MTL specifically, specialized methods such as SAMO-FairGrad remain the strongest on absolute $\Delta m\%$, while SAGE provides consistent gains over the underlying solvers (LS, MGDA, FairGrad) without requiring task-routing or solver-specific design. In combination, the DG experiments demonstrate that targeting both curvature and gradient alignment can yield

state-of-the-art results, while the MTL results demonstrate that the same mechanisms can be added to existing gradient-balancing methods and yield improved results.

5.3 Ablations

Figure 3 (left) reports OfficeHome accuracy over selections of the perturbation radius ρ and the noise scale γ. Performance is stable across selection, with the exception of the selection of either relatively low or high values. In the same, figure on the right we compare the performance of SAGE when removing each of its components and against the Muon Jordan et al. (2024) optimizer. As apparent from the results, each component of SAGE, i.e. orthogonalization via Newton-Schulz, weight scaling in each layer and the additive noise, contributes positively during training. Further plots indicating training dynamics, i.e. validation accuracy, gradient agreement and approximate Hessian trace curves, are provided in Appendix F.



Method	OfficeHome	TerraIncognita
No Newton-Schulz	86.3	60.1
No Scaling	86.6	60.1
No Noise	86.8	61.0
Muon Jordan et al. (2024)	85.5	54.2
SAGE	**87.1**	**61.1**

Figure 3: *Left*: Sensitivity analysis for the ρ and γ hyperparameters on OfficeHome and *right*: Ablations regarding each component of SAGE, on the OfficeHome and TerraIncognita datasets.

6 Conclusions, Limitations & Future Work

In this paper, we presented an excess-risk decomposition for multi-distribution learning that exposes a structural trade-off between the curvature of the loss surface and the alignment of per-distribution gradients, along with a separation result showing that neither quantity bounds the other. Motivated by this decomposition, we proposed SAGE, whose two components target the two terms via: a spectral perturbation that probes all singular directions of each layer's gradient with comparable magnitude, and a gradient-disagreement-scaled noise injection that biases the optimizer away from high-conflict regions. Across five DG and two MTL benchmarks, SAGE compares favorably with both sharpness-aware and gradient-alignment baselines.

One limitation of SAGE is that it incurs additional per-step cost relative to vanilla SAM, as Newton–Schulz orthogonalization adds matrix multiplications on every weight matrix, and the noise mechanism requires per-distribution gradient computation (Appendix G). Additionally, the empirical gains are not uniformly state-of-the-art across all benchmarks. In the MTL setting in particular, when paired with base solvers SAGE improves performance in almost all cases, but does not dominate every metric compared to methods that have been specifically developed for MTL. More fundamentally, the connection between SAGE's mechanisms and the terms of Theorem 1 is motivational rather than formal: we have not shown that either component provably reduces a tractable surrogate of its target term, and Counterexample 1 is an argument in the quadratic family rather than a guarantee for the non-convex deep-learning regime. These limitations point to natural directions for future work. For example, methods that directly estimate or bound the two terms of Theorem 1 could replace the spectral perturbation with a target that more closely matches the curvature term, while estimators of Σ_g could be coupled with regularizers that penalize the alignment term directly. Finally, whether the curvature–alignment trade-off identified also applies to other settings such as continual learning or federated optimization, where multi-distribution structure arises from different generative processes than DG and MTL, is an interesting open question which we aim to research in future work.

AI USE STATEMENT

In this work, we used generative AI tools for editing (e.g., grammar, spelling, word choice). We have not used generative AI tools for scientific writing, reference generation or literature review. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Our experiments follow standard, well-documented benchmarks: DomainBed Gulrajani & Lopez-Paz (2021) for domain generalization and the established MTL protocols of Ban & Ji (2024); Navon et al. (2022). The full training hyperparameters, dataset splits, and evaluation protocol are given in Appendix E. The theoretical claims in the main text are supported by complete proofs in Appendix B. We will upload an anonymous code release as supplementary material, including all training and evaluation scripts, to facilitate reproduction of the reported results.

REFERENCES

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

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

Aristotelis Ballas and Christos Diou. Towards domain generalization for ecg and eeg classification: Algorithms and benchmarks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8(1):44–54, 2024. doi: 10.1109/TETCI.2023.3306253.

Aristotelis Ballas and Christos Diou. Gradient-guided annealing for domain generalization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 20558–20568, 2025.

Hao Ban and Kaiyi Ji. Fair resource allocation in multi-task learning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Hao Ban, Gokul Ram Subramani, and Kaiyi Ji. Samo: A lightweight sharpness-aware approach for multi-task optimization with joint global-local perturbation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 785–795, 2025.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 456–473, 2018.

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. *arXiv preprint arXiv:2409.20325*, 2024.

Åke Björck and Clazett Bowie. An iterative algorithm for computing the best estimate of an orthogonal matrix. *SIAM Journal on Numerical Analysis*, 8(2):358–364, 1971.

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

Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *Advances in Neural Information Processing Systems*, 33:2039–2050, 2020.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus. Mathématique*, 350(5-6):313–318, 2012.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pp. 1019–1028. PMLR, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

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

Heshan Devaka Fernando, Han Shen, Miao Liu, Subhajit Chaudhury, Keerthiram Murugesan, and Tianyi Chen. Mitigating gradient bias in multi-objective learning: A provably convergent approach. In *The eleventh international conference on learning representations*, 2023.

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

Stanislav Fort and Stanislaw Jastrzebski. Large scale structure of neural network loss landscapes. *Advances in Neural Information Processing Systems*, 32, 2019.

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

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

Chun-Hua Guo and Nicholas J Higham. A schur–newton method for the matrix\boldmath p th root and its inverse. *SIAM Journal on Matrix Analysis and Applications*, 28(3):788–804, 2006.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pp. 1842–1850. PMLR, 2018.

Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. *European Conference on Computer Vision*, 2020.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Minyoung Kim, Da Li, Shell X Hu, and Timothy Hospedales. Fisher sam: Information geometry and sharpness aware minimisation. In *International Conference on Machine Learning*, pp. 11148–11161. PMLR, 2022.

Zdislav Kovarik. Some iterative methods for improving orthonormality. *SIAM Journal on Numerical Analysis*, 7(3):386–389, 1970.

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

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International conference on machine learning*, pp. 5905–5914. PMLR, 2021.

Binh M Le and Simon S Woo. Gradient alignment for cross-domain face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 188–199, 2024.

Aodi Li, Liansheng Zhuang, Xiao Long, Houqiang Li, and Shafei Wang. Exploring mode connectivity in krylov subspace for domain generalization. In *The Fourteenth International Conference on Learning Representations*, 2026.

Bingcong Li and Georgios Giannakis. Enhancing sharpness-aware optimization through variance suppression. *Advances in Neural Information Processing Systems*, 36:70861–70879, 2023.

Bo Li, Yifei Shen, Yezhen Wang, Wenzhen Zhu, Dongsheng Li, Kurt Keutzer, and Han Zhao. Invariant information bottleneck for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7399–7407, 2022.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018a.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Computer Vision and Pattern Recognition*, 2018b.

Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Domain generalization via conditional invariant representations. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018c.

Baijiong Lin, Feiyang Ye, Yu Zhang, and Ivor W Tsang. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *arXiv preprint arXiv:2111.10603*, 2021.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in neural information processing systems*, 34:18878–18890, 2021a.

Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. Famo: Fast adaptive multitask optimization. *Advances in Neural Information Processing Systems*, 36:57226–57243, 2023.

Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=IMPnRXEWpvr.

Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1871–1880, 2019.

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

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In *International Conference on Machine Learning*, pp. 16428—-16446. PMLR, 2022.

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

Hoang Phan, Lam Tran, Quyen Tran, Ngoc Tran, Tuan Truong, Qi Lei, Nhat Ho, Dinh Phung, and Trung Le. Beyond losses reweighting: Empowering multi-task learning via the generalization perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2440–2450, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

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

Akshay Rangamani et al. *Loss landscapes and generalization in neural networks: Theory and applications*. PhD thesis, Johns Hopkins University, 2020.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

Dmitry Senushkin, Nikolay Patakin, Arseny Kuznetsov, and Anton Konushin. Independent component alignment for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20083–20093, June 2023.

Yuge Shi, Jeffrey Seely, Philip Torr, Siddharth N, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vDwBW49HmO.

Dahun Shin, Dongyeop Lee, Jinseok Chung, and Namhoon Lee. Sassha: Sharpness-aware adaptive second-order optimization with stable hessian approximation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=7bgqx5OoVe.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pp. 746–760. Springer, 2012.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, 2016.

Yee Whye Teh, Alexandre Thiéry, and Sebastian J Vollmer. Consistency and fluctuations for stochastic gradient langevin dynamics. *Journal of Machine Learning Research*, 17(7), 2016.

Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

V Vapnik. Statistical learning theory. *NY: Wiley*, 1998.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5018–5027, 2017.

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

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

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. How does sharpness-aware minimization minimize sharpness?, 2023. URL https://arxiv.org/abs/2211.05729.

Lei Wu, Zhanxing Zhu, et al. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33: 5824–5836, 2020.

Ruipeng Zhang, Ziqing Fan, Jiangchao Yao, Ya Zhang, and Yanfeng Wang. Domain-inspired sharpness aware minimization under domain shifts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=I4wB3HA3dJ.

Xingxuan Zhang, Renzhe Xu, Han Yu, Hao Zou, and Peng Cui. Gradient norm aware minimization seeks first-order flatness and improves generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20247–20257, 2023.

Yilang Zhang, Bingcong Li, and Georgios B Giannakis. Preconditioned sharpness-aware minimization: Unifying analysis and a novel learning algorithm. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2022. doi: 10.1109/TKDE.2021.3070203.

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, April 2023. ISSN 1939-3539. doi: 10.1109/TPAMI.2022.3195549. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha C Dvornek, sekhar tatikonda, James s Duncan, and Ting Liu. Surrogate gap minimization improves sharpness-aware training. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=edONMAnhLu-.

Appendix

Table of Contents

A Extended Related Work **16**
- A.1 Domain Generalization 16
- A.2 Multi-Task Learning 17
- A.3 Positioning of SAGE 18

B Proofs **18**
- B.1 Setup Reminder: 18
- B.2 Proof of Theorem 1: Multi-distribution excess-risk decomposition 19
- B.3 Proof of Counterexample 1: Decoupling of flatness and gradient alignment . . . 22
- B.4 Limitations 23

C Motivating Example: Why Both Properties Are Necessary **23**

D Scale Invariance and Scale-Dependent Flatness in SAM **25**
- D.1 Illustrative Example 25

E Experimental details & Hyperparameters **26**
- E.1 Details & Hyperparameters 26
- E.2 Infrastructure 26

F Additional Ablations **26**

G Computational Analysis **26**

H Pseudo code of the proposed SAGE algorithm **27**

A Extended Related Work

A.1 Domain Generalization

Domain generalization (DG) aims to train a model on data from multiple source domains such that it generalizes to previously unseen target domains whose distributions may differ substantially from those encountered during training Wang et al. (2022); Zhou et al. (2023). The core challenge is that Empirical Risk Minimization (ERM), which minimizes the average loss over all training data, may converge to solutions that exploit domain-specific shortcuts rather than domain-invariant features, resulting in poor out-of-distribution performance. In this section, we will focus on the methodological groups that are closest in nature with our proposed method, namely domain-invariant representation learning, sharpness-aware optimization, gradient-based alignment, and loss-landscape exploration.

Domain-invariant and robust optimization methods. A central line of work found in the literature seeks representations that are invariant across training domains. In one of the most fundamental works, Invariant Risk Minimization (IRM) Arjovsky et al. (2019) constrains the optimal classifier to be identical across all data domains, while V-REx Krueger et al. (2021) penalizes the variance of per-domain risks. Fishr Rame et al. (2022) on the other hand enforces consistency of per-domain gradient variances, and CORAL Sun & Saenko (2016) aligns second-order feature statistics across domains. Although these methods directly modify the training objective to encourage domain-invariant solutions, they do not directly consider the geometry of the loss landscape.

Sharpness-aware minimization for DG. As already mentioned in the main text, the observation that flat minima tend to generalize better than sharp ones Foret et al. (2021); Keskar et al. (2016) has motivated the application of SAM and its variants to domain generalization. Specifically, GSAM Zhuang et al. (2022) refines the sharpness estimate via a surrogate gap objective, ASAM Kwon et al. (2021) introduces adaptive, scale-aware perturbation magnitudes, GAM Zhang et al. (2023) regularizes the gradient norm to seek first-order flatness, and VaSSO Li & Giannakis (2023) addresses the noise introduced by mini-batch sampling in the perturbation direction. Furthermore, SAGM Wang et al. (2023) simultaneously minimizes the empirical loss, the perturbed loss, and the surrogate gap between them, encouraging the clean-loss gradient $\nabla\mathcal{L}(\theta)$ and the perturbed-loss gradient $\nabla\mathcal{L}_p(\theta)$ to remain aligned so that the optimizer converges to regions that are both flat and low-loss. Importantly, the gradient matching is between the clean and perturbed objectives at the *aggregate* level and does not involve per-domain gradient information. Finally, Fisher SAM Kim et al. (2022) re-parameterizes the perturbation using the Fisher information metric to better capture the intrinsic geometry of the model.

A key limitation of vanilla SAM under domain shifts was identified by Zhang et al. Zhang et al. (2024), who observed that when domains differ in difficulty or data quantity, SAM's gradient-based perturbation is biased toward whichever domain has the largest gradient, disrupting training for the others. To address this, they proposed DISAM (Domain-Inspired SAM), which incorporates a domain-loss variance minimization constraint into the perturbation generation step. The resulting perturbation adaptively up-weights well-converged domains and down-weights under-converged ones, yielding faster overall convergence and improved generalization. DISAM is compatible with other SAM variants (GSAM, SAGM) and achieves consistent improvements on the DomainBed benchmark, particularly on datasets with large domain gaps such as TerraIncognita. Finally, complementary ensemble-based approaches such as SWAD Cha et al. (2021) achieve flatness passively by averaging model weights densely along the training trajectory, rather than explicitly optimizing a sharpness objective.

Gradient-alignment methods for DG. A parallel family of approaches leverages the agreement of per-domain gradients as a signal for domain invariance. The intuition is straightforward: if a model's internal representation does not depend on the domain, then the expected gradients of the loss with respect to different domains should point in similar directions. Shi et al. Shi et al. (2022) proposed Fish, which approximately maximizes the inner product of per-domain gradients. Mansilla et al. Mansilla et al. (2021) adapted gradient surgery from multi-task learning Yu et al. (2020) to DG by muting or randomizing gradient dimensions where domains disagree in sign. Le and Woo Le & Woo (2024) applied gradient alignment to cross-domain face anti-spoofing, demonstrating the generality of the approach beyond standard image classification benchmarks. More recently,

Gradient-Guided Annealing (GGA) Ballas & Diou (2025) was proposed, for searching parameter configurations via a Simulated Annealing-inspired process, where domain gradients are well-aligned before continuing standard optimization.

Beyond local flatness: mode connectivity. While the methods above focus on properties local to a single minimum (flatness, gradient alignment), Li et al. Li et al. (2026) recently shifted attention to a global property of the loss landscape: *mode connectivity*, the phenomenon whereby distinct local minima are connected by continuous low-loss pathways. They demonstrated empirically that models with poor and strong out-of-domain generalization can be connected via such pathways and proposed the Billiard Optimization Algorithm (BOA), which navigates these pathways by alternating line search (to locate loss contour boundaries) and reflection (to redirect the trajectory along the contour). To overcome the curse of dimensionality in high-dimensional parameter spaces, BOA operates within a low-dimensional Krylov subspace constructed from training gradients and Hessian–vector products, which they showed to be well-aligned with test gradients across diverse datasets and architectures. BOA achieves state-of-the-art results on DomainBed with ViT backbones, outperforming all sharpness-aware baselines, and represents a qualitatively different approach to loss-landscape exploitation for DG.

A.2 Multi-Task Learning

Multi-task learning (MTL) trains a single model to perform multiple tasks simultaneously, with the goal of exploiting shared structure across tasks to improve data efficiency and generalization Zhang & Yang (2022). The standard approach minimizes the average loss across all tasks, however, this simple objective often leads to suboptimal solutions because the gradients of different tasks may differ substantially in magnitude or direction, a phenomenon commonly referred to as *task conflict* or *conflicting gradients* Yu et al. (2020); Liu et al. (2021a). When one task's gradient dominates the shared update, other tasks may stagnate or even regress, resulting in performance that falls short of what independent single-task models can achieve. Addressing this fundamental optimization challenge has motivated a rich body of work, which can be broadly organized into loss-weighting methods, gradient manipulation methods, and more recent approaches that integrate sharpness-aware minimization into the MTL framework.

Loss weighting methods. A natural strategy for balancing tasks is to assign scalar weights to each task's loss and adapt those weights during training. Kendall et al. Kendall et al. (2018) proposed Uncertainty Weighting (UW), which uses learned task-dependent homoscedastic uncertainty as a proxy for task difficulty. Liu et al. Liu et al. (2019) introduced Dynamic Weight Average (DWA), a heuristic that adjusts task weights based on the rate of change of each task's loss over recent iterations. Lin et al. Lin et al. (2021) showed that randomly sampling task weights at each step, i.e., Random Loss Weighting (RLW), can be surprisingly effective. A scale-invariant (SI) baseline that minimizes the sum of log-losses has also been considered, as it is invariant to multiplicative rescaling of individual task losses Liu et al. (2023). While computationally lightweight, these methods adjust only the loss aggregation and do not directly resolve geometric conflicts in the gradient space.

Gradient manipulation methods. A more principled family of approaches directly modifies the update direction to ensure balanced progress across tasks. Désidéri Désidéri (2012) introduced the Multiple Gradient Descent Algorithm (MGDA), which finds a convex combination of task gradients lying in the common descent cone, and Sener and Koltun Sener & Koltun (2018) adapted MGDA for deep networks. PCGrad Yu et al. (2020) projects each task's gradient onto the normal plane of conflicting gradients from other tasks. GradDrop Chen et al. (2020) randomly zeroes gradient dimensions where tasks disagree in sign. CAGrad Liu et al. (2021a) constrains the update within a ball around the average gradient while maximizing worst-case per-task improvement. IMTL-G Liu et al. (2021b) finds an update direction with equal cosine similarity to all task gradients, while ICA Senushkin et al. (2023) decomposes task gradients into independent components. Nash-MTL Navon et al. (2022) formulates the problem as a bargaining game, and FairGrad Ban & Ji (2024) extends the bargaining framework with a fairness-aware conic combination. MoCo Fernando et al. (2023) corrects gradient bias through a momentum-based mechanism. A common limitation of all these methods is their $\mathcal{O}(K)$ time and space overhead per iteration, which becomes prohibitive when both the number of tasks and the model size are large.

Efficient multi-task optimization. To address this scalability bottleneck, Liu et al. Liu et al. (2023) proposed FAMO, which decreases all task losses at approximately equal *rates* by working in log-loss space and amortizing the weight update using only the change in task losses between consecutive iterations. This achieves $\mathcal{O}(1)$ space and time overhead per iteration, matching the complexity of standard average-loss training while remaining competitive with gradient manipulation methods on standard benchmarks.

Sharpness-aware minimization in MTL. Ban et al. Ban et al. (2025) provided empirical evidence that SAM Foret et al. (2021) mitigates task conflicts by guiding optimization toward flatter loss regions where changes in one task's loss do not substantially affect others. Their analysis revealed that applying SAM to the average loss (global SAM) and to each task individually (local SAM) both contribute to conflict reduction, but neither is consistently superior. Phan et al. Phan et al. (2025) proposed F-MTL, which applies SAM independently to each task but incurs K additional backpropagations. Ban et al. Ban et al. (2025) introduced SAMO, which combines global and local perturbation information via a weighted average and approximates expensive local gradients using a zeroth-order SPSA estimator requiring only forward passes. A layerwise normalization scheme stabilizes the estimates, achieving the benefits of joint global–local perturbations at a cost comparable to global-only SAM.

A.3 Positioning of SAGE

The methods reviewed above target individual geometric properties of the loss landscape, i.e., flatness, gradient alignment, mode connectivity, or task balancing, but limited methods address more than one simultaneously. Sharpness-aware methods (SAM, GSAM, SAGM, DISAM) seek flat minima without explicitly considering whether per-distribution gradients agree at those minima. Gradient-alignment methods (Fish, GGA, PCGrad in DG; PCGrad, CAGrad, Nash-MTL in MTL) enforce gradient agreement without considering the curvature of the resulting solution. The closest prior work is that of SAGM Wang et al. (2023), which also augments SAM with a gradient-matching term. However, the two methods differ in three respects. First, SAGM's gradient matching aligns the clean-loss and perturbed-loss gradients, which are both computed on the *aggregate* data, to ensure joint descent toward flat, low-loss regions and does not compute per-distribution gradients.

SAGE addresses the curvature through a spectral perturbation that replaces SAM's gradient-scaled ascent with the polar factor of each layer's gradient matrix, computed via Newton–Schulz iteration and scaled by the layer's Frobenius norm. Regarding gradient alignment term, SAGE injects isotropic Gaussian noise at the descent step whose magnitude scales with the degree of cross-environment gradient conflict, $\beta = \gamma(1 - S(\theta))$, discouraging convergence to regions of high gradient disagreement. Unlike methods that directly modify the gradient direction (PCGrad, Fish) or solve auxiliary optimization problems (Nash-MTL, FAMO), SAGE's noise injection is lightweight and requires no per-environment gradient storage beyond what is needed for computing the pairwise cosine similarity. This noise injection was inspired from the recent GGA Ballas & Diou (2025) work.

Importantly, SAGE operates as a drop-in replacement for the optimizer's ascent and descent steps, making it compatible with existing gradient manipulation methods in MTL (as demonstrated by the SAGE-MGDA, SAGE-FairGrad, and SAGE-SAMO combinations in Table 2) and with the standard DomainBed protocol in DG (Table 1).

B Proofs

B.1 Setup Reminder:

For each $e \in \mathcal{E}$, let $\mathcal{L}_e : \Theta \to \mathbb{R}$ denote a per-distribution loss, where $\Theta \subseteq \mathbb{R}^d$ is an open parameter space. The *population risk* is:

$$R(\theta) := \mathbb{E}_{e\sim\mathcal{P}}[\mathcal{L}_e(\theta)], \tag{6}$$

and the *empirical multi-distribution risk* over K iid training distributions $e_1, \ldots, e_K \sim \mathcal{P}$ is

$$\hat{R}(\theta) := \frac{1}{K}\sum_{k=1}^{K} \mathcal{L}_{e_k}(\theta). \tag{7}$$

At any base point $\theta_0 \in \Theta$ we introduce the per-distribution gradient and Hessian,

$$g_e := \nabla_\theta \mathcal{L}_e(\theta_0), \qquad H_e := \nabla^2_\theta \mathcal{L}_e(\theta_0), \tag{8}$$

and the following cross-distribution statistics,

$$\bar{g} := \mathbb{E}_e[g_e], \qquad \bar{H} := \mathbb{E}_e[H_e], \qquad \Sigma_g := \mathrm{Cov}_e(g_e) = \mathbb{E}_e\left[(g_e - \bar{g})(g_e - \bar{g})^T\right]. \tag{9}$$

B.2 Proof of Theorem 1: Multi-distribution excess-risk decomposition

Proof of Theorem 1. We restate the Theorem and Assumption included in the main text for completeness.

Theorem 1 (Multi-distribution excess-risk decomposition)**.** *Under Assumption 1, let $\hat{\theta}$ denote the minimizer of $\hat{R}$ over K iid training distributions, and let $\xi \sim \mathcal{N}(0, \sigma^2 I)$ denote isotropic Gaussian noise of scale $\sigma > 0$. Then, for a parameter $\theta = \hat{\theta} + \xi$:*

$$\mathbb{E}[\mathbb{E}_\xi[R(\theta)]] - R(\theta^\star) \;=\; \underbrace{\frac{1}{2K}\,\mathrm{tr}\big(\bar{H}^{-1}\Sigma_g\big)}_{\text{alignment term}} \;+\; \underbrace{\frac{\sigma^2}{2}\,\mathrm{tr}(\bar{H})}_{\text{curvature term}} \;+\; O(K^{-3/2}) \;+\; O(\sigma^3), \tag{10}$$

where the outer expectation is over the iid sampling of training distributions, $\bar{H} := \nabla^2 R(\theta^\star)$, and $\Sigma_g := \mathrm{Cov}_e(\nabla \mathcal{L}_e(\theta^\star))$.

Assumption 1 (Regularity)**.** Each $\mathcal{L}_e$ is three times continuously differentiable in θ, with third derivatives uniformly bounded in a neighborhood of $\theta^\star := \arg\min_\theta R(\theta)$, and $\bar{H}(\theta^\star) \succ 0$ (Positive-Definite). The gradients $g_e(\theta^\star)$ have finite second moments under $\mathcal{P}$.

Assumption 1 is the standard regularity condition under which quadratic excess-risk expansions are valid, which is however strictly weaker than the convexity assumptions used in classical M-estimation theory Van der Vaart (2000). The C^3 requirement is used only to calculate and control the remainder in a second-order Taylor expansion, while the PD-Hessian requirement ensures the quadratic form $\bar{H}^{-1}$ is well defined.

The proof proceeds in three steps: (i) expand R around $\theta^\star$ to obtain the excess risk as a quadratic form in $\hat{\theta} - \theta^\star$, (ii) characterize $\hat{\theta} - \theta^\star$ as a sample-mean quantity and take expectations to produce the alignment term, (iii) expand R around $\hat{\theta}$ under random perturbations $\xi \sim \mathcal{N}(0, \sigma^2 I)$ to produce the curvature term.

Step 1: Quadratic expansion of R at $\theta^\star$. Because $\theta^\star$ minimizes the population risk R, we have $\nabla R(\theta^\star) = \bar{g} = 0$. Under Assumption 1, Taylor's theorem gives, for any $\delta \in \mathbb{R}^d$ sufficiently small,

$$R(\theta^\star + \delta) = R(\theta^\star) + \nabla R(\theta^\star)\delta + \tfrac{1}{2}\delta^T \nabla^2 R(\theta^\star)\delta + r_3(\delta), \tag{11}$$

where $|r_3(\delta)| \le C\|\delta\|^3$ for some constant C depending on the uniform bound on third derivatives, by definition $\nabla R(\theta^\star) = \bar{g} = 0$, and $\nabla^2 R(\theta^\star) = \bar{H}$. Therefore, subtracting $R(\theta^\star)$ yields:

$$R(\theta^\star + \delta) - R(\theta^\star) = \tfrac{1}{2}\delta^T \bar{H}\delta + O(\|\delta\|^3). \tag{12}$$

Equation equation 12 is the standard quadratic excess-risk formula, which holds for any δ. We now look at $\delta = \hat{\theta} - \theta^\star$ and analyze the distance between the empirical minimizer ($\hat{\theta}$) and population minimizer ($\theta^\star$).

Step 2: Characterization of $\hat{\theta} - \theta^\star$. Note that the empirical minimizer satisfies $\nabla \hat{R}(\hat{\theta}) = 0$ across K distributions. Expanding $\nabla \hat{R}$ around $\theta^\star$ using Taylor's theorem yields,

$$\nabla \hat{R}(\hat{\theta}) = \nabla \hat{R}(\theta^\star) + \nabla^2 \hat{R}(\theta^\star)(\hat{\theta} - \theta^\star) + O(\|\hat{\theta} - \theta^\star\|^2), \tag{13}$$

or,

$$0 = \nabla \hat{R}(\hat{\theta}) = \hat{g} + \hat{H}(\hat{\theta} - \theta^\star) + O(\|\hat{\theta} - \theta^\star\|^2), \tag{14}$$

where $\hat{g} := \frac{1}{K}\sum_{k=1}^{K} g_{e_k}(\theta^\star)$ and $\hat{H} := \frac{1}{K}\sum_{k=1}^{K} H_{e_k}(\theta^\star)$.

Solving for $\hat{\theta} - \theta^\star$:

$$\begin{aligned} \hat{H}(\hat{\theta} - \theta^\star) &= -\hat{g} - O(\|\hat{\theta} - \theta^\star\|^2) \\ \hat{\theta} - \theta^\star &= -\hat{H}^{-1}\hat{g} - \hat{H}^{-1}O(\|\hat{\theta} - \theta^\star\|^2) \\ \hat{\theta} - \theta^\star &= -\hat{H}^{-1}\hat{g} + O(\|\hat{\theta} - \theta^\star\|^2) \end{aligned} \tag{15}$$

Where from Assumption 1, $\bar{H}$ is PD, so assuming enough data so will $\hat{H}$. To find the "size" of the $(\hat{\theta} - \theta^\star) \approx -\hat{H}^{-1}\hat{g}$, we take the vector norm (magnitude) of both sides. Using the standard property of matrix norms and the triangle inequality, we can separate the matrix from the vector to create an upper bound:

$$||\hat{\theta} - \theta^\star|| \approx || - \hat{H}^{-1}\hat{g}|| \leq ||\hat{H}^{-1}|| \cdot ||\hat{g}||. \tag{16}$$

The eigenvalues of an inverse matrix are simply $1/\lambda$. Because the original eigenvalues (λ) of $\hat{H}$ are bounded away from zero, the eigenvalues of the inverse matrix cannot blow up to infinity. As the those eigenvalues are capped, $||\hat{H}^{-1}||$ is just some finite constant number c (based on the largest eigenvalue). Therefore:

$$\begin{aligned} ||\hat{\theta} - \theta^\star|| &\leq c||\hat{g}|| \\ ||\hat{\theta} - \theta^\star|| &\leq O(||\hat{g}||) \\ ||\hat{\theta} - \theta^\star||^2 &\leq O(||\hat{g}||^2) \end{aligned} \tag{17}$$

And substituting 17 to 15, yields:

$$\hat{\theta} - \theta^\star = -\hat{H}^{-1}\hat{g} + O(||\hat{g}||^2) \tag{18}$$

This is the standard implicit-function argument used in M-estimator asymptotics Van der Vaart (2000). We now continue to replace $\hat{H}$ with $\bar{H}$, which concentrates to $||\hat{H} - \bar{H}||$, as $K \to \infty$. By the law of large numbers, as K increases, the empirical average converges to the true expectation: $\hat{H} \to \bar{H}$. Recall that $\hat{H}$ is the empirical Hessian, calculated by averaging the actual Hessians observed across K randomly sampled training distributions: $\hat{H} = \frac{1}{K}\sum_{k=1}^{K} H_{e_k}$. Because the training distributions $e_1, \ldots, e_K$ are sampled independently and identically (iid) from the meta-distribution $\mathcal{P}$, each observed Hessian H_{e_k} is an independent random matrix. This means $\hat{H}$ is simply a sample mean, and we want to know how far this sample mean deviates from the true mean $\bar{H}$. Let $\hat{H}_{ij}$ denote the scalar entry in the i-th row and j-th column of the empirical Hessian matrix, and $\bar{H}_{ij}$ denote the corresponding entry in the population Hessian. By definition, the empirical entry is a simple sample mean of the individual distribution Hessians:

$$\hat{H}_{ij} = \frac{1}{K}\sum_{k=1}^{K}(H_{e_k})_{ij}.$$

Since the third derivatives of the loss are uniformly bounded in a neighborhood of θ^*, as per 1, each entry $(H_{e_k})_{ij}$ is itself bounded, and there exists some finite variance $\sigma_{ij}^2 < \infty$ such that:

$$Var((H_{e_k})_{ij}) = \sigma_{ij}^2.$$

Since each distribution is sampled i.i.d, from the Central Limit Theorem (or Chebyshev's Inequality) for each element; as $K \to \infty$:

$$\sqrt{K}(\hat{H}_{ij} - \bar{H}_{ij}) \to \mathcal{N}(0, \sigma_{ij}^2).$$

Where by calculating the Frobenius norm of the difference, it can be easily shown that $||\hat{H} - \bar{H}||_F = O_P(K^{-1/2})$ and we can formally bound the random error matrix:

$$||\hat{H} - \bar{H}|| = O_P(K^{-1/2}) \tag{19}$$

Therefore the distance between the empirical sample $\hat{H}$ and the true population $\overline{H}$ shrinks at a rate proportional to $1/\sqrt{K}$.

Again, following the same logic, by the Central Limit Theorem $\sqrt{K}\hat{g} \to \mathcal{N}(0, \Sigma_g)$, and the magnitude of the average gradient $||\hat{g}||$ shrinks at a rate of $O_P(K^{-1/2})$ and in turn the squared gradient term becomes $O(||\hat{g}||^2) = O_P(K^{-1})$.

Thus, Equation 18 becomes:

$$\hat{\theta} - \theta^\star = -\bar{H}^{-1}\hat{g} + O_P(K^{-1}). \tag{20}$$

Taking expectations. Substitute Eq. equation 20 into Eq. equation 12:

$$\begin{aligned} R(\hat{\theta}) - R(\theta^\star) &= \tfrac{1}{2}(\bar{H}^{-1}\hat{g})^T \bar{H}(\bar{H}^{-1}\hat{g}) + O_P(K^{-3/2}) \\ &= \tfrac{1}{2}\hat{g}^T \bar{H}^{-1}\hat{g} + O_P(K^{-3/2}), \end{aligned} \tag{21}$$

where we used $\bar{H}^{-1}\bar{H}\bar{H}^{-1} = \bar{H}^{-1}$ and the symmetry $(\bar{H}^{-1})^T = \bar{H}^{-1}$. The $\|\delta\|^3$ term in Eq. equation 12 becomes $O_P(K^{-3/2})$ because $\|\hat{g}\| = O_P(K^{-1/2})$ by the central limit theorem. Now take expectation over the iid sampling of the K training distributions. Using the matrix identity

$$\mathbb{E}[x^T A x] = \operatorname{tr}(A \operatorname{Cov}(x)) + \mathbb{E}[x]^T A \, \mathbb{E}[x] \tag{22}$$

(valid for any random vector x with finite second moments and any symmetric A) with $x = \hat{g}$ and $A = \bar{H}^{-1}$, we compute:

- $\mathbb{E}[\hat{g}] = \mathbb{E}\big[\frac{1}{K}\sum_k g_{e_k}\big] = \bar{g} = 0$ (since $\theta^\star$ is stationary for R), so the mean term vanishes.
- $\operatorname{Cov}(\hat{g}) = \frac{1}{K^2}\sum_k \operatorname{Cov}(g_{e_k}) = \frac{1}{K}\Sigma_g$ by independence of the distributions and due to the fact that each distribution is drawn from $\mathcal{P}$.

Substituting:

$$\mathbb{E}[\hat{g}^T \bar{H}^{-1}\hat{g}] = \operatorname{tr}\big(\bar{H}^{-1} \cdot \tfrac{1}{K}\Sigma_g\big) = \tfrac{1}{K}\operatorname{tr}(\bar{H}^{-1}\Sigma_g). \tag{23}$$

Combining with Eq. equation 21:

$$\mathbb{E}[R(\hat{\theta}) - R(\theta^\star)] = \tfrac{1}{2K}\operatorname{tr}(\bar{H}^{-1}\Sigma_g) + O(K^{-3/2}). \tag{24}$$

This establishes the alignment term.

Step 3: Expansion of R at $\hat{\theta}$ under random perturpations. We want to account for the fact that the learner does not exactly realize $\hat{\theta}$ and to capture the sharpness of the local landscape, we evaluate the expected risk of the empirical minimizer under random perturbations, e.g. due to optimization noise or finite-precision arithmetic. Let the deployed parameter be $\theta = \hat{\theta} + u$, where $u \sim \mathcal{N}(0, \sigma^2 I)$ is an isotropic Gaussian perturbation of scale $\sigma > 0$.

By expanding R at $\hat{\theta}$, from Taylor's theorem:

$$R(\hat{\theta} + u) = R(\hat{\theta}) + \nabla R(\hat{\theta})\xi + \frac{1}{2}\xi^T \nabla^2 R(\hat{\theta}) u + O(||\xi||^3), \tag{25}$$

and taking expectation with respect to the noise u,

$$\mathbb{E}_u[R(\hat{\theta}+\xi)] = R(\hat{\theta}) + \nabla R(\hat{\theta})\mathbb{E}[u] + \frac{1}{2}\mathbb{E}[u^T\nabla^2 R(\hat{\theta})u] + O(\sigma^3) \tag{26}$$

The linear term vanishes because the perturbation is zero-mean ($\mathbb{E}[u] = 0$). For the quadratic term, apply the identity equation 22 with $x = u$ (zero mean) and $A = \nabla^2 R(\hat{\theta})$, using $\mathrm{Cov}(u) = \sigma^2 I$:

$$\frac{1}{2}\mathbb{E}[u^T\nabla^2 R(\hat{\theta})u] = \frac{\sigma^2}{2}tr(\nabla^2 R(\hat{\theta})) \tag{27}$$

Since $\nabla^2 R(\hat{\theta}) = \bar{H} + O_P(K^{-1/2})$ by continuity of the Hessian and the fact that $\hat{\theta} \to \theta^\star$, we can replace $\nabla^2 R(\hat{\theta})$ with the population Hessian $\bar{H}$ up to a correction that multiplies the existing $\sigma^2/2$ factor and contributes at order $O(\sigma^2 K^{-1/2})$, which is absorbed into the stated error. Thus, the expected risk under perturbation u is:

$$\mathbb{E}_u[R(\hat{\theta})] = R(\hat{\theta}) + \frac{\sigma^2}{2}tr(\bar{H}) + O(\sigma^3) + O(\sigma^2 K^{-1/2}) \tag{28}$$

Combining. Taking expectation of Eq. equation 28 over the distribution sampling and adding Eq. equation 24:

$$\mathbb{E}[\mathbb{E}_u[R(\hat{\theta})]] - R(\theta^*) = \frac{1}{2K}tr(\bar{H}^{-1}\Sigma_g) + \frac{\sigma^2}{2}tr(\bar{H}) + O(K^{-3/2}) + O(\sigma^3), \tag{29}$$

which is Eq. equation 10. □ □

B.3 Proof of Counterexample 1: Decoupling of flatness and gradient alignment

Proof of Counterexample 1. We restate the counterexample included in the main text for completeness.

Counterexample 1 (Decoupling of flatness and gradient alignment)**.** Consider the family of multi-distribution learning problems with per-distribution losses

$$\mathcal{L}_e(\theta) = \tfrac{1}{2}\theta^T A\theta + b_e^T\theta, \qquad \mathbb{E}_e[b_e] = 0, \tag{30}$$

parameterized by a symmetric PD matrix $A \in \mathbb{R}^{d\times d}$ and a collection of linear coefficients $\{b_e\}$ satisfying $\mathbb{E}_e[b_e] = 0$. For every $M > 0$, there exist instances of Eq. equation 30 such that:

1. **Flat but misaligned:** $\mathrm{tr}(\bar{H}) \le M^{-1}$ and $\mathrm{tr}(\bar{H}^{-1}\Sigma_g) \ge M$.
2. **Aligned but sharp:** $\mathrm{tr}(\bar{H}^{-1}\Sigma_g) \le M^{-1}$ and $\mathrm{tr}(\bar{H}) \ge M$.

In particular, neither $\mathrm{tr}(\bar{H})$ nor $\mathrm{tr}(\bar{H}^{-1}\Sigma_g)$ can be bounded above by any function of the other alone.

For the family in the counterexample we have $\nabla\mathcal{L}_e(\theta) = A\theta + b_e$ and $\nabla^2\mathcal{L}_e(\theta) = A$. Therefore $\bar{H} = A$ is fixed by the choice of A alone, and the population minimizer is $\theta^\star = -A^{-1}\mathbb{E}_e[b_e] = 0$. At $\theta^\star$, the per-environment gradients are $g_e = b_e$, so

$$\Sigma_g = \mathbb{E}_e[b_e b_e^T], \tag{31}$$

which is fixed by the choice of $\{b_e\}$ alone. The two quantities are thus independently controllable: A determines $\bar{H}$ without affecting Σ_g, and $\{b_e\}$ determines Σ_g without affecting $\bar{H}$.

Construction of (i): flat but misaligned. Take $d = 2$, $A = \lambda I$ with $\lambda = (2M)^{-1}$, and let b_e be supported on e_2 with variance $v = 1$. Then $\bar{H} = \lambda I$, so

$$\mathrm{tr}(\bar{H}) = 2\lambda = M^{-1}, \tag{32}$$

satisfying the first inequality. Also $\Sigma_g = \mathrm{diag}(0, v)$, so

$$\mathrm{tr}(\bar{H}^{-1}\Sigma_g) = \mathrm{tr}\big(\lambda^{-1}I \cdot \mathrm{diag}(0, v)\big) = v/\lambda = 2Mv. \tag{33}$$

Choosing $v = 1$ gives $\mathrm{tr}(\bar{H}^{-1}\Sigma_g) = 2M \ge M$. Concretely, two environments with $b_1 = (0, +1)^T$ and $b_2 = (0, -1)^T$ realize this Σ_g.

Construction of (ii): aligned but sharp. Take $d = 2$, $A = \lambda I$ with $\lambda = M/2$, and b_e supported on e_1 with variance $v = 1/2$. Then

$$\mathrm{tr}(\bar{H}) = 2\lambda = M, \qquad \mathrm{tr}(\bar{H}^{-1}\Sigma_g) = v/\lambda = M^{-1}. \tag{34}$$

Two environments with $b_1 = (+1/\sqrt{2}, 0)^T$ and $b_2 = (-1/\sqrt{2}, 0)^T$ realize the required Σ_g.

Conclusion. In both constructions, $\bar{H}$ and Σ_g are chosen independently to drive one quantity arbitrarily small and the other arbitrarily large. If there existed a function ϕ such that $\mathrm{tr}(\bar{H}^{-1}\Sigma_g) \leq \phi(\mathrm{tr}(\bar{H}))$ for all instances in the family, then construction (i) with $M \to \infty$ would contradict $\phi(M^{-1}) < \infty$, symmetrically for the reverse direction. □ □

B.4 Limitations

Local, not global. Theorem 1 is an asymptotic expansion valid in a neighborhood of $\theta^\star$ where the quadratic approximation holds. In non-convex deep learning landscapes, there may be many $\bar{g} = 0$ points, and the theorem does not address which one the optimizer reaches. This is the "hypothesis selection" gap: among the manifold of stationary points, flatness and alignment act as refinement criteria, but the decomposition alone does not prove that optimization converges to the jointly-optimal refinement.

Not a PAC-Bayes bound. Although Step 3 of the proof of Theorem 1 echoes the structure of Gaussian PAC-Bayes analyses the result is an *expectation-form* excess-risk decomposition, not a high-probability uniform bound. A full PAC-Bayes statement would add a KL-divergence term, bound the failure probability explicitly, and require a different argument for the alignment term. The result stated here is sufficient for its purpose, i.e, identifying which quantities appear in the excess risk and in what functional form, but should not be read as a finite-sample generalization guarantee.

Quadratic family is toy. The decoupling construction in Counterexample 1 uses a purely quadratic family, in which $\bar{H}$ is globally constant and independent of $\{b_e\}$. A skeptical reader might object that real deep networks are highly non-convex and that the independence of $\bar{H}$ and Σ_g observed in the quadratic family may not hold globally. The correct response is that Counterexample 1 provides a strong indication, but no guarantees. If even in a simple setting (exact quadratic, infinite data, closed-form) neither criterion alone suffices, then a fortiori neither is expected to suffice in the harder, non-convex setting.

C Motivating Example: Why Both Properties Are Necessary

Theorem 1 decomposes the excess risk into an alignment term and a curvature term, while Theorem 1 shows that neither term can be bounded as a function of the other. The example below illustrates how this is reflected in a concrete setting where each failure mode appears along an orthogonal eigendirection of $\bar{H}$. The example uses a linear model on a Gaussian data-generating process with two domains, chosen so that the loss landscape admits closed-form analysis.

Data-generating process. Consider a binary classification task with label $y \in \{-1, +1\}$ drawn uniformly, and input $x = (x_{\text{inv}}, x_{\text{spur}}) \in \mathbb{R}^2$ consisting of an invariant feature and a spurious feature. For domain $k \in \{1, 2\}$, the features are generated conditionally on y as:

$$x_{\text{inv}} \mid y \sim \mathcal{N}(y \cdot \mu_{\text{inv}},\ \sigma^2_{\text{inv}}), \tag{35}$$

$$x^{(k)}_{\text{spur}} \mid y \sim \mathcal{N}(y \cdot c_k,\ \sigma^2_{\text{spur}}), \tag{36}$$

where $c_1 = +\mu_{\text{spur}}$ and $c_2 = -\mu_{\text{spur}}$. The invariant feature is causally linked to y identically in both domains, while the spurious feature's correlation with y reverses across domains. Crucially, we set $\sigma^2_{\text{inv}} \gg \sigma^2_{\text{spur}}$: the invariant feature is noisy (high variance) while the spurious feature is clean (low variance). We adopt the concrete values $\mu_{\text{inv}} = 1$, $\sigma^2_{\text{inv}} = 9$, $\mu_{\text{spur}} = 2$, $\sigma^2_{\text{spur}} = 0.01$ throughout the example.

Loss landscape. We train a linear model $f(x;\theta) = \theta_{\text{inv}}\, x_{\text{inv}} + \theta_{\text{spur}}\, x_{\text{spur}}$ with mean squared error. The expected loss for domain k expands to:

$$\mathcal{L}_k(\theta) = \frac{1}{2} - \theta_{\text{inv}}\mu_{\text{inv}} - \theta_{\text{spur}}c_k + \frac{1}{2}\theta_{\text{inv}}^2(\sigma_{\text{inv}}^2 + \mu_{\text{inv}}^2) + \frac{1}{2}\theta_{\text{spur}}^2(\sigma_{\text{spur}}^2 + \mu_{\text{spur}}^2) + \theta_{\text{inv}}\theta_{\text{spur}}\,\mu_{\text{inv}}c_k, \quad (37)$$

using the moments $\mathbb{E}[y^2] = 1$, $\mathbb{E}[y\, x_{\text{inv}}] = \mu_{\text{inv}}$, $\mathbb{E}[x_{\text{inv}}^2] = \sigma_{\text{inv}}^2 + \mu_{\text{inv}}^2$, and so on. Because each sub-objective loss is quadratic, the Hessian of $\mathcal{L}_k$ is constant:

$$H_k = \begin{pmatrix} \sigma_{\text{inv}}^2 + \mu_{\text{inv}}^2 & \mu_{\text{inv}}\, c_k \\ \mu_{\text{inv}}\, c_k & \sigma_{\text{spur}}^2 + \mu_{\text{spur}}^2 \end{pmatrix}. \quad (38)$$

Substituting our numerical values gives $H_{11} = 10$, $H_{22} = 4.01$, and $H_{12}^{(1)} = +2$, $H_{12}^{(2)} = -2$. Two features of this landscape are critical. First, the curvature along θ_{inv} is $2.5\times$ larger than along θ_{spur}, meaning the invariant direction forms a narrow valley while the spurious direction is comparatively flat. Second, the off-diagonal term $\mu_{\text{inv}}c_k$ flips sign across domains, coupling the two parameters differently in each domain's loss surface.

Aggregate loss and its minima. The aggregate (ERM) loss $\bar{\mathcal{L}} = \frac{1}{2}(\mathcal{L}_1 + \mathcal{L}_2)$ has the Hessian:

$$\bar{H} = \frac{1}{2}(H_1 + H_2) = \begin{pmatrix} 10 & 0 \\ 0 & 4.01 \end{pmatrix}, \quad (39)$$

where the cross-domain averaging cancels the off-diagonal terms. The aggregate minimum is $\theta^* = \bar{H}^{-1}\nabla_\theta\bar{\mathcal{L}}(0) = (0.1, 0)^T$, confirming that the ERM optimum correctly assigns zero weight to the spurious feature.

However, the individual domain minima tell a different story. Setting $\nabla_\theta\mathcal{L}_k = 0$ and solving yields domain-specific optima that place substantial weight on θ_{spur} with opposite signs across domains. In each domain individually, the spurious feature is a low-noise, high-signal predictor and can lead to training models that will not generalize.

Failure mode 1: Flatness without alignment. A method that searches for low-curvature solutions within a single domain finds the per-domain optima $\theta_k^* = H_k^{-1}b_k \approx (0.0003, \pm 0.499)^T$, which lie almost entirely along the spurious axis with reversed signs across domains. This is a structural consequence of the within-domain feature statistics, as the spurious feature has signal-to-noise ratio $\mu_{\text{spur}}^2/\sigma_{\text{spur}}^2 = 400$, far exceeding the invariant feature's $\mu_{\text{inv}}^2/\sigma_{\text{inv}}^2 \approx 0.11$. Within either domain individually, the spurious feature is the better predictor. A flatness-only criterion cannot distinguish a flat minimum that uses spurious features from one that uses invariant ones, and the alignment term of Theorem 1 is what penalizes the difference.

Failure mode 2: Alignment without flatness. A method that enforces gradient agreement across domains correctly identifies the shared direction. The per-domain gradients at the origin are $\nabla\mathcal{L}_1(\mathbf{0}) = (-1, -2)^T$ and $\nabla\mathcal{L}_2(\mathbf{0}) = (-1, +2)^T$, which agree on the invariant component and conflict on the spurious one. The optimizer converges to $\theta^* = (0.1, 0)^T$, i.e. the aggregate minimum, which uses exclusively the invariant feature. However, θ^* sits on the *sharp* eigendirection of $\bar{H}$ (eigenvalue 10), so a parameter displacement δ along θ_{inv} leads to a loss increase of $\frac{1}{2}\cdot 10\cdot\delta^2 = 5\delta^2$. Therefore, under distribution shift the optimal coefficient moves and the steep curvature amplifies the resulting error. The curvature term of Theorem 1 is what penalizes this.

Connection to Counterexample 1. The two failure modes occur along orthogonal eigendirections of $\bar{H}$, which is what makes this two-dimensional construction minimal. The cross-domain gradient covariance at the aggregate minimum is $\Sigma_g = \text{diag}(0, 3.24)$, supported entirely on the flat eigendirection of $\bar{H}$. The alignment term evaluates to $\text{tr}(\bar{H}^{-1}\Sigma_g) = 3.24/4.01 \approx 0.81$, which is dominated by the ratio of gradient variance to the small eigenvalue. This is, structurally, an instance of construction (i) of Counterexample 1, as the support of Σ_g coincides with the flat eigendirection of $\bar{H}$, so the alignment term is amplified by the inverse curvature in that direction. The two failure modes the example exhibits are not just numerical coincidences but the structural examples of the connection between Σ_g and $\bar{H}$.

D SCALE INVARIANCE AND SCALE-DEPENDENT FLATNESS IN SAM

A well-documented limitation that SAGE addresses is with regards to the interaction between SAM's fixed perturbation radius and the scale invariance of modern architectures Dinh et al. (2017). Neural networks employing ReLU activations and normalization layers exhibit positive scale invariance, meaning that scaling a weight matrix W by $\alpha > 0$ to obtain $W' = \alpha W$ (with the subsequent layer scaled by $1/\alpha$) leaves the network's output unchanged. The loss landscape is therefore functionally identical, yet the gradient scales inversely:

$$\nabla_{W'}\mathcal{L} = \frac{1}{\alpha}\nabla_W\mathcal{L}. \tag{40}$$

Because SAM uses a fixed L_2-radius ρ, the perturbation's size relative to the parameter norm changes drastically with α. Shrinking the weights ($\alpha < 1$) makes the fixed-size perturbation push parameters proportionally further from their current values, inflating the apparent sharpness; conversely, scaling weights up ($\alpha \gg 1$) makes the same perturbation negligibly small relative to the parameter norm, producing artificially flat sharpness estimates. This phenomenon, termed "scale-dependent flatness" Kwon et al. (2021), means that SAM's sharpness measure conflates the true geometry of the loss surface with the arbitrary scale of the parameterization.

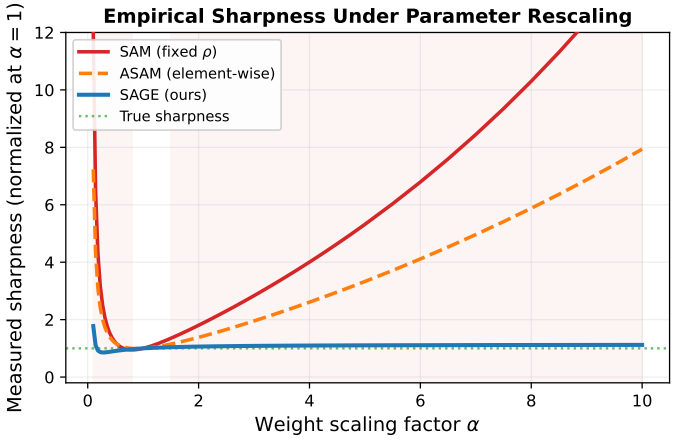


Figure 4: **Empirical scale invariance.** Measured sharpness as a function of weight rescaling factor α for a two-layer MLP (no normalization layers). The network computes the same function at all α, so the true sharpness (green, dotted) is constant. Both SAM (red) and ASAM (orange, dashed) produce scale-dependent sharpness estimates that diverge away from $\alpha = 1$, despite the underlying function being identical. SAGE (blue) is approximately invariant to the rescaling, correctly reflecting the true geometry.

D.1 ILLUSTRATIVE EXAMPLE

To empirically demonstrate the scale-invariance failure, we train a two-layer MLP without normalization layers ($\texttt{Linear}(2, 64) \to \texttt{ReLU} \to \texttt{Linear}(64, 2)$) on a synthetic concentric-circles task until convergence. At the converged parameters θ^*, we apply the rescaling $W_1 \mapsto \alpha W_1$, $b_1 \mapsto \alpha b_1$, $W_2 \mapsto W_2/\alpha$ for $\alpha \in [0.1, 10.0]$. As previously mentioned, the network computes the same function for all α. For this example, we measure the sharpness $\mathcal{L}(\theta + \epsilon) - \mathcal{L}(\theta)$ at each α using SAM (fixed ρ), ASAM (element-wise adaptive), and SAGE (spectral, scale-adaptive). As shown in Figure 4, both SAM and ASAM produce sharpness estimates that diverge away from $\alpha = 1$, while SAGE remains approximately constant, correctly reflecting the invariance of the underlying loss geometry[5].

[5]The effect on SAM's fixed-ρ perturbation depends on which layer dominates. In this architecture, W_2's gradient grows and α causes the perturbation to push harder on W_2, thus increasing sharpness for large α.

E EXPERIMENTAL DETAILS & HYPERPARAMETERS

E.1 DETAILS & HYPERPARAMETERS

Domain Generalization For DG, we follow the protocol of the widely adopted and challenging DomainBed Gulrajani & Lopez-Paz (2021) benchmark, and conduct experiments on five image classification datasets. Specifically, we follow the leave-one-domain-out evaluation protocol for PACS Li et al. (2017) (9,991 images, 4 domains and 7 classes), VLCS Fang et al. (2013) (10,729 images, 4 domains and 5 classes), OfficeHome Venkateswara et al. (2017) (15,588 images, 4 domains and 65 classes), TerraIncognita Beery et al. (2018) (24,788 images, 4 domains and 10 classes), and DomainNet Peng et al. (2019) (586,575 images, 6 domains and 345 classes), and report the average top-1 accuracy over 3 runs based on training-domain split validation. Following previous implementations Li et al. (2026); Zhang et al. (2024), we select a batch size of 12 where images are uniformly sampled across source domains, set an initial learning rate of $1e-6$ and employ the Adam optimizer for finetuning a CLIP Radford et al. (2021) pretrained ViT-B/16 Dosovitskiy et al. (2021) . Regarding SAGE-specific hyperparameters, we select $\rho = 1e-6$ and $\gamma = 1e-6$ across all datasets.

Multi-Task Learning For the MTL setting, we follow the standard protocol in recent literature Ban et al. (2025); Liu et al. (2023); Navon et al. (2022) and evaluate on Cityscapes Cordts et al. (2016), a dataset of urban street scenes containing $5{,}000$ images with pixel-level annotations, and it supports two tasks: 7-class semantic segmentation and depth estimation. In contrast, NYU-v2 Silberman et al. (2012) focuses on indoor environments, providing $1{,}449$ densely annotated images and enabling three tasks: 13-class semantic segmentation, depth estimation, and surface normal prediction. For training, we employ MTAN Liu et al. (2019) as the shared backbone, with task-specific attention modules built on top of a SegNet Badrinarayanan et al. (2017). Following previous implementations Ban et al. (2025); Liu et al. (2023) the model is trained for 200 epochs with a batch size of 8 for Cityscapes and 2 for NYU-v2. The learning rate is set to $1e-4$ for the first 100 epochs and is then halved for the remainder. Regarding SAGE hyperparameters, we set $\rho = 1e-4$ and $\gamma = 1e-5$ for both datasets.

E.2 INFRASTRUCTURE

All experiments were conducted on a cluster containing 4×40 GB NVIDIA A100 GPU cards, split into 8 20GB virtual MIG devices and $1\times$ 24GB NVIDIA RTX A5000 GPU card, via a SLURM workload manager.

F ADDITIONAL ABLATIONS

In Figure 5 we provide training dynamics, including validation accuracy, gradient agreement and approximate Hessian trace curves. For the Hessian trace approximation, Hutchinson's method was applied. Interestingly, as apparent from the curve values the following observations are of note:

1. Gradient agreement is consistently higher under SAGE than under ERM and SAM throughout training. Although similarity declines over epochs for all three methods, SAGE maintains the highest values at every comparable point.
2. The Hessian trace is lower for SAGE than for ERM and SAM in 3 of 4 target domains, with the exception of the Product domain, where SAM attains a slightly lower trace.
3. There is an interesting observation regarding the dynamic of the two components (agreement and curvature), where as that the Hessian trace decreases (indicating lower curvature), so does gradient similarity. This points to the broader point of Remark 1 and motivation of our work, that the two quantities should be targeted jointly during optimization.

G COMPUTATIONAL ANALYSIS

Each SAGE training iteration requires two forward-backward passes through the network, identical in structure to SAM and all SAM-like methods. The first pass computes the aggregate and per-environment gradients g_{e_k} from a single forward call over the concatenated mini-batches from all

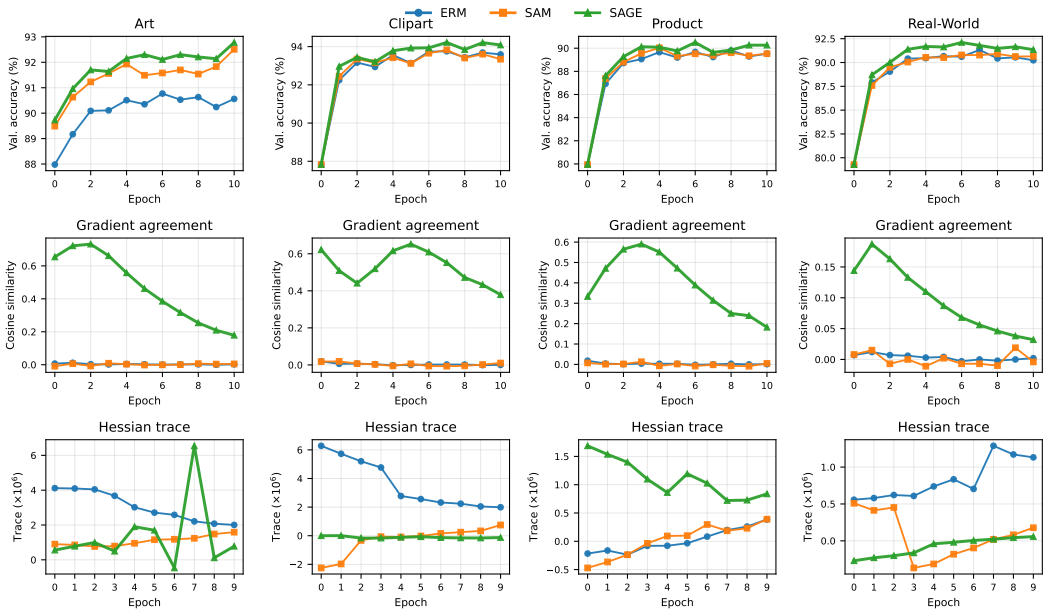


Figure 5: Validation accuracy, average gradient vector cosine similarity and Hessian trace approximation via Hutchinson's method on the OfficeHome dataset.

Table 3: Computational analysis between ERM, SAM and SAGE on the OfficeHome dataset.

Method	Peak Memory	Total Training Time
ERM	4.7 GB	114 min
SAM	8.1 GB	119 min
SAGE	8.2 GB	120 min

K distributions (domains in DG or tasks in MTL), incurring no additional forward passes beyond what ERM already performs with the same batch. From these per-distribution gradients, the pairwise cosine similarity $S(\theta)$ and the noise scale β are computed in $\mathcal{O}(K^2 d)$ time, which is negligible relative to the cost of backpropagation for any practical number of environments ($K \leq 6$ in all of our experiments). The spectral perturbation replaces SAM's L_2-normalization of the gradient with $T = 5$ Newton-Schulz iterations per weight matrix, each consisting of a single matrix multiplication and a matrix subtraction. These operations are fully parallelized on modern GPU hardware and add negligible overhead. The second forward-backward pass at the perturbed point $\theta + \epsilon$ is identical to that of SAM. Finally, the noise injection $\beta\,\xi$ at the descent step amounts to sampling a Gaussian vector and a single multiply-add operation, which is $\mathcal{O}(d)$. In total, SAGE's per-iteration cost is approximately 2×2 that of ERM, the same factor as SAM, with the additional operations (cosine similarities, Newton-Schulz iterations, and noise sampling) contributing an overhead that is negligible in practice compared to the forward-backward passes. Importantly, inference is entirely unaffected by SAGE, as all modifications occur exclusively during the training optimization loop.

To provide further empirical results regarding the computational comparison between ERM, SAM and SAGE, in Table 3 we provide details regarding Peak Memory and Training Time on the OfficeHome dataset. Specifically, SAGE adds 100 MB peak memory and relatively insignificant training time over SAM, i.e. the Newton-Schulz iterations and the per-distribution gradient computation are negligible in practice relative to the two forward-backward passes both methods share.

H Pseudo code of the proposed SAGE algorithm

Algorithm 1 SAGE: Spectral-Aware and Gradient-Aligned Exploration

Require: Learning rate η, perturbation radius ρ, noise scale γ, sub-objectives $\{D_1, \ldots, D_K\}$, feature extractor f_θ, cosine similarity function (cos), Newton-Schulz iterations T, loss function ℓ, base optimizer OPT(), total number of training operations n.

for t $\leftarrow 1$ to n **do**
  Sample mini-batches $\{(x_k, y_k)\}_{k=1}^{K}$ from each sub-objective
  // Phase 1: Sub-objective and total gradients
  Compute logits $\hat{y} = f_\theta(\text{concat}(x_1, \ldots, x_K))$
  for $k = 1, \ldots, K$ **do**
    $\mathcal{L}_k \leftarrow \ell(\hat{y}_k, y_k)$
    $g_k \leftarrow \nabla_\theta \mathcal{L}_k$
  end for
  $\bar{g} \leftarrow \sum_k w_k \, g_k$ {sample-weighted total gradient}
  // Phase 2: Calculate gradient agreement & noise scale
  $S \leftarrow \frac{2}{K(K-1)} \sum_{i<j} \cos(g_i, g_j)$
  $\beta \leftarrow \gamma \, (1 - S)$
  // Phase 3: Compute Spectral Perturbations (Muon Logic)
  for each weight matrix $W \in \theta$ and its corresponding gradient $G \in g$ **do**
    if G is a matrix or higher-order tensor **then**
      $G_{\text{ortho}} \leftarrow \text{NewtonSchulz5}(G)$ {Orthogonalize via Newton-Schulz}
      $\epsilon_W \leftarrow \rho \|W\|_F G_{\text{ortho}}$ {Scaled spectral perturbation}
    else
      $\epsilon_W \leftarrow \rho \frac{G}{\|G\|_2}$
    end if
  end for
  // Phase 4: Apply Perturbation (Ascent Step)
  $\theta \leftarrow \theta + \epsilon$
  // Phase 5: Adversarial Forward & Backward Pass
  Compute perturbed loss $\mathcal{L}_{\text{pert}} \leftarrow \ell(f_\theta(\text{concat}(x_1, \ldots, x_K)))$
  Compute perturbed gradient $g_{\text{pert}} \leftarrow \nabla_\theta \mathcal{L}_{\text{pert}}$
  // Phase 6: Restore Weights & Inject Gradient Noise (Descent Step)
  $\theta \leftarrow \theta - \epsilon$ {Restore clean weights}
  $g_{\text{final}} \leftarrow g_{\text{pert}} + \beta \mathcal{N}(0, \mathbf{I})$ {Inject domain-conflict noise}
  // Phase 7: Optimizer Step
  $\theta \leftarrow \text{OPT}(\theta, \eta, g_{\text{final}})$
end for