ORTHOSAM: ORTHOGONAL PERTURBATIONS FOR SHARPNESS-AWARE MINIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sharpness-Aware Minimization (SAM) improves generalization by minimizing the worst-case loss within a local neighborhood, encouraging the optimizer to find flatter minima. However, standard SAM treats neural network weights as flattened vectors and applies unconstrained perturbations in the Euclidean space, ignoring the intrinsic matrix structure of modern neural networks. As a result, the adversarial perturbation can become dominated by a few large singular directions, while other important directions receive much less attention. In this work, we propose **OrthoSAM**, a matrix-aware optimization method based on the idea that flatness should be explored more evenly across different directions. Instead of using unrestricted perturbations, OrthoSAM constrains perturbations to a structured manifold with orthogonality properties, reducing the imbalance caused by dominant singular values. This encourages a more uniform exploration of the local loss landscape across matrix directions. To make the method efficient in practice, we use the Newton–Schulz iteration to obtain approximately orthogonal perturbations without requiring expensive singular value decomposition (SVD). Extensive experiments on multiple architectures and datasets show that OrthoSAM consistently improves generalization and provides stronger robustness to label noise.

1 INTRODUCTION

Deep neural networks have emerged as the dominant paradigm for solving complex tasks across diverse domains, including computer vision and natural language processing. Equipped with millions or even billions of parameters, these highly overparameterized models possess sufficient capacity to perfectly fit complex, large-scale datasets (Zhang et al., 2016). In practice, however, models are optimized on a finite empirical dataset $\mathcal{S}$ sampled from an unknown underlying data distribution $\mathcal{D}$. While this immense capacity enables the learning of intricate representations, it simultaneously renders models highly susceptible to overfitting (Arpit et al., 2017; Liu et al., 2020). Rather than capturing the true underlying semantic patterns, networks can easily memorize noisy labels, spurious correlations, or random fluctuations in the training data (Zhang et al., 2016; Arpit et al., 2017). Consequently, a model might achieve near-zero empirical loss on the training set, yet suffer severe performance degradation when evaluated on unseen test data or under domain shifts. Measuring, controlling, and mitigating this generalization gap, often formalized as the discrepancy between the empirical error on $\mathcal{S}$ and the expected population error on $\mathcal{D}$ (Vapnik, 1998; Bartlett & Mendelson, 2003; Dziugaite & Roy, 2017), remains one of the most fundamental and enduring challenges in modern machine learning.

Extensive research in statistical learning theory and non-convex optimization has established a profound connection between the geometry of the loss landscape and a model's generalization capability. Specifically, both theoretical and empirical studies demonstrate that models converging to wider, flatter local minima are inherently more robust to data shifts between the training and testing sets than those trapped in sharp minima (Hochreiter & Schmidhuber, 1994; Keskar et al., 2017; Neyshabur et al., 2017; Jiang et al., 2020; Petzka et al., 2021). Motivated by this geometric insight and grounded in the PAC-Bayes theorem, Sharpness-Aware Minimization (SAM) (Foret et al., 2021) has emerged as a prominent and highly effective optimization framework. SAM proactively regularizes the curvature of the loss landscape by seeking parameters whose entire surrounding neighborhood has uniformly low loss. It achieves this by approximating a worst-case adversarial perturbation within an ℓ_2 Euclidean ball around the current weights, subsequently performing a descent step from this

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.

perturbed point. The success of SAM has inspired a large family of variants aiming to refine its perturbation geometry or reduce computational overhead (Kwon et al., 2021; Kim et al., 2022).

Although SAM is theoretically well motivated and empirically effective, it has an important geometric limitation: it treats all network parameters as a single flattened vector, ignoring the intrinsic matrix structure of layers such as linear projections, convolutions, and attention modules. When the gradient is viewed in matrix form through singular value decomposition (SVD), the SAM perturbation becomes heavily biased toward directions with large singular values. Since deep networks typically exhibit highly skewed singular spectra, the perturbation is dominated by only a few sharp directions, while many structural directions receive little exploration. As a result, SAM explores the local neighborhood unevenly rather than uniformly searching for flat, robust regions; as we show empirically, this imbalance can also make late-stage training less stable. Importantly, although many SAM variants modify the perturbation rule or neighborhood geometry, they still operate in the same fully vectorized parameter space and do not explicitly exploit the underlying matrix structure of neural network weights.

To address this limitation, we propose **OrthoSAM**, a matrix-aware optimization method that adds structure to the adversarial perturbation. Rather than using an unconstrained Euclidean ball, OrthoSAM keeps the weight matrix form and requires the perturbation to have orthonormal columns, so that it lies on a structured orthogonal neighborhood. This constraint helps balance the update across directions and reduces the influence of a few large singular values. As a result, no single sharp direction can dominate the perturbation, and the optimizer explores the local geometry in a more even way. To obtain such a perturbation efficiently, we take inspiration from the Muon optimizer (Jordan et al., 2024) and use the Newton–Schulz iteration. This gives an approximately orthonormal matrix without requiring an exact SVD at every step, while keeping the update close to the original ascent direction. OrthoSAM applies the same orthogonalization to the descent step: the gradient at the perturbed point is replaced by its orthogonal factor, so each step keeps the singular vectors of its gradient and sets all singular values equal. Our ablation (Section 5.4) shows that the two steps are complementary: neither the orthogonal perturbation nor the orthogonalized descent alone reaches the accuracy of the full method. Our main contributions are as follows:

- We identify and formalize a key limitation of standard SAM: after vectorizing the parameters, the adversarial perturbation becomes dominated by a few directions associated with large singular values, leading to unbalanced exploration and potentially unstable training.
- We propose **OrthoSAM**, a matrix-aware optimization method that constrains adversarial perturbations to an orthogonal neighborhood and orthogonalizes the descent step. Using the Newton–Schulz iteration to compute the orthogonal factor, OrthoSAM reduces the dominance of directions associated with large singular values, leading to a more balanced exploration of the loss landscape across different structural directions.
- We validate OrthoSAM through comprehensive experiments across vision and language benchmarks, demonstrating consistent gains over standard SAM and recent variants, flatter learned minima, and strong robustness under severe label noise.

2 Related Work

Sharpness-Aware Minimization and Variants. The observation that flat minima tend to generalize better dates back to Hochreiter & Schmidhuber (1994) and has been supported by both theoretical and empirical studies linking loss curvature to generalization (Keskar et al., 2017; Neyshabur et al., 2017; Jiang et al., 2020; Petzka et al., 2021). Sharpness-Aware Minimization (SAM) (Foret et al., 2021) translates this insight into a practical algorithm by minimizing the worst-case loss within an ℓ_2 ball around the current parameters, justified through a PAC-Bayes generalization bound. A large family of variants has since emerged to refine SAM's perturbation design or to reduce its computational overhead. ASAM (Kwon et al., 2021) makes the perturbation adaptive to parameter scales, and FisherSAM (Kim et al., 2022) replaces the Euclidean ball with a Fisher information neighborhood to account for the geometry of the predictive distribution. GSAM (Zhuang et al., 2022) augments the descent direction with a surrogate gap term that steers optimization away from sharp regions, while GAM (Zhang et al., 2023) penalizes the gradient norm to promote first-order flatness. VaSSO (Li & Giannakis, 2023) reduces variance in the perturbation direction across iterations, and FriendlySAM (Li et al., 2024) improves alignment between the adversarial perturbation and the

descent step. MongeSAM (Jacobsen & Arvanitidis, 2025) constructs a reparameterization-invariant perturbation grounded in the geometry of the loss surface. On the efficiency side, ESAM (Du et al., 2022) halves SAM's cost through stochastic weight perturbation and sharpness-sensitive data selection, LookSAM (Liu et al., 2022) reuses the perturbation direction over multiple consecutive iterations, and MSAM (Becker et al., 2025) amortizes the ascent cost via gradient momentum. Multi-step extensions include Lookahead-SAM (Yu et al., 2024), which integrates Lookahead-style extrapolation–interpolation with SAM to reach wider minima, and Lookbehind-SAM (Mordido et al., 2024), which prepends backward steps before the SAM update to better navigate sharp regions. Riemannian SAM (Yun & Yang, 2023) generalizes SAM to Riemannian manifolds; however, it targets the manifold structure of the parameters rather than perturbation. While several of these variants refine the geometry of the perturbation neighborhood, they all operate on a *vectorized* parameter space and do not exploit the intrinsic matrix structure of weight layers. Our work departs from this paradigm by retaining the matrix form of each weight and constraining perturbations to have orthonormal columns on Stiefel manifold.

Understanding SAM. A complementary line of work studies SAM's optimization dynamics and implicit biases. Andriushchenko & Flammarion (2022); Wen et al. (2023b) show that SAM implicitly regularizes leading Hessian eigenvalues and reduces sharpness along training, explaining its preference for flatter solutions. Chen et al. (2023) prove that SAM avoids sharp minima in two-layer networks and yields tighter generalization bounds than SGD, while Monzio Compagnoni et al. (2023) characterize its dynamics via continuous-time SDEs. Beyond curvature, Wen et al. (2023a) argue that SAM also induces favorable biases beyond sharpness minimization, and Luo et al. (2024) propose explicit eigenvalue regularization to directly control curvature.

Matrix-Aware Optimization. Standard SAM **flattens all parameters into a single vector** and applies a uniform ℓ_2 constraint, thereby ignoring the matrix structure of weight tensors in projections, convolutions, and attention layers. Muon (Jordan et al., 2024; Liu et al., 2025) orthogonalizes hidden-layer descent updates with Newton–Schulz iterations, approximating steepest descent under the spectral norm. Muon has no ascent step; OrthoSAM uses the same iteration to compute an orthogonal adversarial perturbation and applies it again to the descent step. Our ablation (Section 5.4) shows that the two steps interact rather than add, so OrthoSAM is not equivalent to applying SAM on top of Muon.

3 Sharpness-Aware Minimization (SAM)

Let $S = \{(x_i, y_i)\}_{i=1}^N$ be a training set drawn i.i.d. from a data distribution $\mathcal{D}$. For a model parameterized by a vector $\theta \in \mathbb{R}^k$ (all weight matrices are flattened and concatenated), we define the per-data-point loss $\ell(\theta; x, y) \in [0, L]$, together with the empirical and population losses, as follows:

$$\mathcal{L}_S(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(\theta; x_i, y_i), \qquad \mathcal{L}_\mathcal{D}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(\theta; x, y)].$$

Standard training minimizes $\mathcal{L}_S(\theta)$, but for overparameterized models this often yields sharp minima that generalize poorly. SAM (Foret et al., 2021) addresses this by seeking parameters whose *entire neighborhood* has uniformly low loss. The key insight comes from a PAC-Bayes bound:

PAC-Bayes Bound in SAM. *Under the condition $\mathcal{L}_\mathcal{D}(\theta) \le \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\rho I)}\mathcal{L}_\mathcal{D}(\theta+\epsilon)$, with probability $1-\delta$ over the choice of the training set:*

$$\mathcal{L}_\mathcal{D}(\theta) \le \max_{\theta' \in \mathcal{B}_\rho(\theta)} \mathcal{L}_S(\theta') + \sqrt{\frac{k \log\Big(1 + \frac{\|\theta\|_2^2}{\rho^2}\big(1+\sqrt{\frac{\log N}{k}}\big)^2\Big) + 4\log\frac{N}{\delta} + \tilde{\mathcal{O}}(1)}{N-1}}, \tag{1}$$

where $\mathcal{B}_\rho(\theta) = \{\theta' : \|\theta' - \theta\|_2 \le \rho\}$ is the Euclidean ball of radius ρ.

The first term measures the worst-case loss in a neighborhood, i.e., the *sharpness*, while the second term penalizes large weights, grows slowly with k, and vanishes as $N \to \infty$. To turn this bound into an algorithm, SAM proposes to minimize the sharpness-aware loss: $\mathcal{L}_S^{\text{SAM}}(\theta) = \max_{\theta' \in \mathcal{B}_\rho(\theta)} \mathcal{L}_S(\theta')$. The inner maximization is solved approximately by a first-order Taylor expansion of $\mathcal{L}_S$ around θ:

$$\theta' - \theta \approx \hat{\epsilon}(\theta) = \rho \frac{\nabla_\theta \mathcal{L}_S(\theta)}{\|\nabla_\theta \mathcal{L}_S(\theta)\|_2}, \tag{2}$$

which is the vector that maximizes the linearized loss inside the Euclidean ball. SAM then computes the gradient at the perturbed point $\theta + \hat{\epsilon}(\theta)$ and performs a standard descent step:

$$\theta = \theta - \eta \nabla_\theta \mathcal{L}_\mathcal{S}(\theta + \hat{\epsilon}(\theta)). \tag{3}$$

Note that, in this vectorized formulation, the perturbation $\Delta\theta = \theta' - \theta$ has no additional constraint beyond its Euclidean norm, i.e., $\|\Delta\theta\|_2 \leq \rho$.

Limitation of the SAM Perturbation. Standard SAM vectorizes the parameters, ignoring the intrinsic matrix structure of neural network layers. Consider a single weight matrix $W \in \mathbb{R}^{m\times n}$ with $m \geq n$, where m and n denote the output and input dimensions of the layer (a convolutional kernel is reshaped into this form by keeping its output channels as rows and flattening the input-channel and spatial dimensions into the columns). Reshaping its gradient into $G = \nabla_W \mathcal{L}_\mathcal{S}(\theta) \in \mathbb{R}^{m\times n}$, the per-matrix SAM perturbation is $\Delta\theta_{\text{SAM}} = \rho \frac{G}{\|G\|_F}$. Applying the SVD yields $G = \sum_{i=1}^{n} \sigma_i A_i$, where $\sigma_i \geq 0$ are the singular values and $A_i = u_i v_i^\top$ form an orthonormal basis ($\langle A_i, A_j \rangle = \delta_{ij}$). Consequently, the perturbation can be rewritten as:

$$\Delta\theta_{\text{SAM}} = \frac{\rho}{\|G\|_F} \sum_{i=1}^{n} \sigma_i A_i. \tag{4}$$

Equation (4) shows that the perturbation along each direction A_i is proportional to its singular value σ_i. Since gradient spectra in deep networks are typically skewed, most of the perturbation budget ρ is spent on the few leading directions, while the many trailing directions are barely perturbed. Their curvature therefore has little influence on the adversarial loss $\mathcal{L}_\mathcal{S}(\theta + \hat{\epsilon})$ that drives the subsequent descent step, so SAM probes the neighborhood anisotropically rather than searching for a region that is uniformly flat across all matrix directions. This concentration has two practical consequences. **We verify both empirically in Appendix C.2.** First, concentrating the budget on the leading directions biases the ascent toward the sharpest directions instead of probing curvature evenly, so the iterate tends to settle in less flat regions. Second, because the dominant directions of a mini-batch gradient fluctuate across batches, aligning the perturbation with them perturbs the subsequent descent direction more and yields noisier late-training dynamics. These observations motivate an orthogonality-aware perturbation that spreads budget evenly across directions and thereby limits the influence of the few dominant singular values.

4 METHODOLOGY

As discussed in Section 3, standard SAM operates on a vectorized parameter space, thereby *ignoring the intrinsic matrix structure* of many layers (e.g., linear projections and attention layers), which carries information that a flattened vector discards. Moreover, its unconstrained perturbation can be dominated by a small number of sharp directions, particularly those associated with large singular values, which may bias the optimization away from more uniformly flat regions. Motivated by Muon (Jordan et al., 2024), we instead retain the matrix form $\theta \in \mathbb{R}^{m\times n}$ and constrain the perturbation $\Delta\theta = \theta' - \theta$ to have orthonormal columns. Geometrically, this places the perturbation in the set of scaled semi-orthogonal matrices: $\mathbb{R}_\perp^{m\times n} = \left\{X \in \mathbb{R}^{m\times n} : X^\top X = \alpha \mathbb{I}_n \text{ for some } \alpha \geq 0\right\}$. Such a constraint prevents *any single direction from dominating the update and promotes a more balanced exploration of the local neighborhood*, aligning more closely with the notion of robust flatness (see Eq. (11) for further explanation). Formally, we define the orthogonal neighborhood:

$$\mathcal{N}_\rho(\theta) = \mathcal{B}_\rho(\theta) \cap \left(\theta + \mathbb{R}_\perp^{m\times n}\right) = \left\{\theta' \in \mathbb{R}^{m\times n} : \|\theta' - \theta\|_F \leq \rho \ \wedge \ \theta' - \theta \in \mathbb{R}_\perp^{m\times n}\right\}, \tag{5}$$

where $\theta \in \mathbb{R}^{m\times n}$ and $\|\cdot\|_F$ is the Frobenius norm. Here we note that without loss of generality, we assume that $m \geq n \geq 1$.

4.1 THEORY DEVELOPMENT

We generalize the PAC-Bayes perspective of SAM from the Euclidean ball $\mathcal{B}_\rho(\theta)$ to the structured neighborhood $\mathcal{N}_\rho(\theta)$, which captures orthogonality-aware perturbations in the matrix parameter space. This formulation retains the sharpness-minimization principle of SAM while enforcing a more balanced exploration of the local geometry. The full proof is provided in Appendix A, and the resulting generalization bound is stated below.

Theorem 4.1. *Assume the per-sample loss is measurable and bounded, $\ell(x, y; \theta) \in [0, L]$ with $L > 0$, and both $\mathcal{L}_\mathcal{D}$ and $\mathcal{L}_\mathcal{S}$ are globally γ-Lipschitz with respect to the Frobenius norm, for a fixed $\gamma \geq 0$, with the empirical-loss assumption holding almost surely. Let $S \sim \mathcal{D}^N$ be a training set of size $N \geq 1$, fix $\rho > 0$ independently of S, and define $\mathcal{N}_\rho(\theta)$ as in* (5)*. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over S, simultaneously for all $\theta \in \mathbb{R}^{m\times n}$, with $k = mn$, we have:*

$$\mathcal{L}_\mathcal{D}(\theta) \leq \max_{\theta' \in \mathcal{N}_\rho(\theta)} \mathcal{L}_\mathcal{S}(\theta') + \frac{4L}{\sqrt{N}} \left[\sqrt{k \log \left(1 + \frac{\|\theta\|_F^2}{\rho^2} \left(1 + \sqrt{\frac{\log N}{k}} \right)^2 \right)} + 2\sqrt{\log \frac{N+k}{\delta}} + 1 \right] + \gamma\rho \tag{6}$$

Motivated by Theorem 4.1, we seek perturbations that maximize the worst-case increase of the empirical loss over the structured neighborhood $\mathcal{N}_\rho(\theta)$. For the current parameter matrix θ, this leads to the inner problem:

$$\tilde{\theta} = \text{argmax}_{\theta' \in \mathcal{N}_\rho(\theta)} \{\mathcal{L}_\mathcal{S}(\theta') - \mathcal{L}_\mathcal{S}(\theta)\}.$$

Using a first-order Taylor expansion around θ, we approximate:

$$\mathcal{L}_\mathcal{S}(\theta') - \mathcal{L}_\mathcal{S}(\theta) \approx \langle \nabla \mathcal{L}_\mathcal{S}(\theta), \theta' - \theta \rangle.$$

To obtain a tractable objective while explicitly enforcing both a norm constraint and orthogonality, we relax the feasible set to the Euclidean ball and introduce an orthogonality indicator. This yields the surrogate problem:

$$\tilde{\theta} = \text{argmax}_{\theta' \in B_\rho(\theta)} \{\langle \nabla \mathcal{L}_\mathcal{S}(\theta), \theta' - \theta \rangle + \mathbf{I}_{\text{ortho}}(\theta' - \theta)\}, \tag{7}$$

where $\mathbf{I}_{\text{ortho}}(X) = 0$ if $X \in \mathbb{R}^{m\times n}_\perp$ and $-\infty$ otherwise.

The following theorem provides the closed-form solution for the optimization problem in (7).

Theorem 4.2. *Assume that the gradient has the SVD $G = \nabla \mathcal{L}_\mathcal{S}(\theta) = U\Sigma V^\top \in \mathbb{R}^{m\times n}$. The optimal solution of the optimization problem in (7) is as follows:*

$$\Delta\theta = \theta' - \theta = \rho' UV^\top, \tag{8}$$

where $\rho' = \rho\, n^{-1/2}$, so that $\|\Delta\theta\|_F = \rho' \|UV^\top\|_F = \rho$ attains the constraint with equality.

Theorem 4.2, proved in Appendix A, gives the perturbed model in closed form:

$$\tilde{\theta} = \theta + \rho' UV^\top, \qquad \rho' = \rho\, n^{-1/2}. \tag{9}$$

Furthermore, similar to Muon (Jordan et al., 2024), we apply the Newton–Schulz (NS) iteration (Algorithm 1) to $G = \nabla \mathcal{L}_\mathcal{S}(\theta) = U\Sigma V^\top$ to obtain the orthogonal factor $O = \text{NS}(G) = UV^\top$.

Update the New Model. Given the perturbed model $\tilde{\theta}$ in Eq. (9), we compute the gradient $\tilde{G} = \nabla \mathcal{L}_\mathcal{S}(\tilde{\theta})$. The SVD of this gradient reads: $\tilde{G} = \tilde{U}\tilde{\Sigma}\tilde{V}^\top$, which gives us

$$\tilde{G} = \tilde{U}\tilde{\Sigma}\tilde{V}^\top = \sum_i \tilde{\sigma}_i \tilde{u}_i \tilde{v}_i^\top = \sum_i \tilde{\sigma}_i \tilde{A}_i, \tag{10}$$

where $\tilde{A}_i = \tilde{u}_i \tilde{v}_i^\top$ for all i. It is evident that $(\tilde{A}_i)_{i=1}^n$ are orthonormal because $\left\langle \tilde{A}_i, \tilde{A}_j \right\rangle = \delta_{ij}$, where δ_{ij} is the Kronecker delta. We investigate the meaning of each direction $\tilde{A}_i$. We apply the first-order Taylor expansion to obtain:

$$\mathcal{L}_\mathcal{S}\left(\tilde{\theta} - \eta \tilde{A}_i\right) - \mathcal{L}_\mathcal{S}\left(\tilde{\theta}\right) \approx -\eta \left\langle \nabla \mathcal{L}_\mathcal{S}\left(\tilde{\theta}\right), \tilde{A}_i \right\rangle = -\eta \left\langle \sum_j \tilde{\sigma}_j \tilde{A}_j, \tilde{A}_i \right\rangle = -\eta \tilde{\sigma}_i. \tag{11}$$

It is clear from Eq. (11) that the directions $\tilde{A}_i$ with high singular values $\tilde{\sigma}_i$ cause stronger decreases in the loss. Therefore, if we apply the SAM update:

$$\theta = \theta - \eta \nabla \mathcal{L}_\mathcal{S}\left(\tilde{\theta}\right) = \theta - \eta \sum_i \tilde{\sigma}_i \tilde{A}_i,$$

the directions with very high singular values $\tilde{\sigma}_i$ might ruin the training. Therefore, to reduce the negative effects of these directions, inspired by Muon, we update the new model as:

$$\theta = \theta - \eta \sum_i 1 \times \tilde{A}_i = \theta - \eta \sum_i \tilde{u}_i \tilde{v}_i^\top = \theta - \eta \tilde{U}\tilde{V}^\top. \tag{12}$$

4.2 Convergence Analysis

We now establish the convergence rate of OrthoSAM under standard smoothness and variance assumptions. Following the methodology detailed in Section 4, at iteration l, let the stochastic mini-batch gradient at the perturbed point be $g_l = \nabla \mathcal{L}_{B_l}(\theta_l + \Delta_l)$. OrthoSAM updates the parameters by projecting this gradient onto the Stiefel manifold using the NS iteration:

$$\tilde{O}_l = \mathrm{NS}(g_l) \approx \tilde{U}_l \tilde{V}_l^\top, \qquad \theta_{l+1} = \theta_l - \eta \tilde{O}_l, \tag{13}$$

where $\tilde{U}_l \tilde{\Sigma}_l \tilde{V}_l^\top$ is the SVD of g_l. For the convergence analysis, we use the idealized polar update $\tilde{O}_l = \tilde{U}_l \tilde{V}_l^\top$ when $g_l \neq 0$, with $\mathrm{NS}(0) = 0$. Consequently, $\|\tilde{O}_l\|_F^2 \leq n$ and $\langle g_l, \tilde{O}_l \rangle = \|g_l\|_*$. At each iteration, a fresh mini-batch B_l is sampled independently of all preceding iterations. The same mini-batch may be used to construct Δ_l and evaluate g_l, and we assume $\|\Delta_l\|_F \leq \rho$ almost surely. Here ρ denotes the Frobenius perturbation radius. Expectations below are over the sampled mini-batches, with training set fixed. Proofs of Theorem 4.3 and Corollary 4.4 below are in Appendix A.

Assumptions. To establish the convergence guarantees for OrthoSAM, we use the following assumptions (Si & Yun, 2023):

(A1) **β-Smoothness.** The empirical loss $\mathcal{L}_\mathcal{S}$ and every mini-batch loss $\mathcal{L}_B$ are differentiable and β-smooth with respect to the Frobenius norm, for a common $\beta > 0$.

(A2) **Bounded Variance.** For every fixed θ, with $\mathbb{E}_B$ denoting expectation over the mini-batch sampling distribution, the mini-batch gradient is unbiased and has variance bounded by σ^2:

$$\mathbb{E}_B[\nabla \mathcal{L}_B(\theta)] = \nabla \mathcal{L}_\mathcal{S}(\theta), \qquad \mathbb{E}_B\big[\|\nabla \mathcal{L}_B(\theta) - \nabla \mathcal{L}_\mathcal{S}(\theta)\|_F^2\big] \leq \sigma^2. \tag{14}$$

Theorem 4.3 (Convergence of OrthoSAM)**.** *Under Assumptions (A1) and (A2) and the idealized NS update described above, suppose $\|\Delta_l\|_F \leq \rho$ almost surely. Assume $\mathcal{L}_\mathcal{S}$ is bounded below, let θ_0 be fixed, and define $D_0 = \mathcal{L}_\mathcal{S}(\theta_0) - \inf_\theta \mathcal{L}_\mathcal{S}(\theta) < \infty$. The iterates of the OrthoSAM algorithm with learning rate $\eta > 0$ satisfy, for every integer $T \geq 1$:*

$$\frac{1}{T} \sum_{l=0}^{T-1} \mathbb{E}\big[\|\nabla \mathcal{L}_\mathcal{S}(\theta_l)\|_F\big] \leq \frac{D_0}{\eta T} + \frac{\beta \eta n}{2} + (\sqrt{n} + 1)(\sigma + \beta\rho). \tag{15}$$

Corollary 4.4. *Under the conditions of Theorem 4.3, suppose $D_0 > 0$. To express the bound relative to the matrix dimension, we scale the Frobenius perturbation radius as $\rho = \tilde{\rho} n^{-1/2}$ and write $T = \tau n$ for $\tau = T/n > 0$. By setting the learning rate $\eta = \frac{1}{n}\sqrt{\frac{2D_0}{\beta\tau}}$, which minimizes the first two terms of* (15)*, the minimum expected gradient norm satisfies:*

$$\min_{0 \leq l < T} \mathbb{E}\big[\|\nabla \mathcal{L}_\mathcal{S}(\theta_l)\|_F\big] \leq \sqrt{\frac{2\beta D_0}{\tau}} + (\sqrt{n} + 1)\sigma + \beta\tilde{\rho}\left(1 + \frac{1}{\sqrt{n}}\right). \tag{16}$$

In particular, for a run of length $T = \tau n$, if $\rho = \bar{\rho}/\sqrt{n\tau}$ and (A2) holds with $\sigma \leq \bar{\sigma}/\sqrt{n\tau}$, where $\bar{\rho}, \bar{\sigma} \geq 0$ are constants independent of τ, then:

$$\min_{0 \leq l < T} \mathbb{E}\big[\|\nabla \mathcal{L}_\mathcal{S}(\theta_l)\|_F\big] \leq \frac{1}{\sqrt{\tau}}\left[\sqrt{2\beta D_0} + \left(1 + \frac{1}{\sqrt{n}}\right)(\bar{\sigma} + \beta\bar{\rho})\right]. \tag{17}$$

For fixed n, this gives an $\mathcal{O}(T^{-1/2})$ bound tending to zero as $T \to \infty$.

5 Experiments

5.1 Experimental Setup

We evaluate **OrthoSAM** against SGD and a chronological set of SAM-family methods: SAM (Foret et al., 2021), ASAM (Kwon et al., 2021), ESAM (Du et al., 2022), GSAM (Zhuang et al., 2022), FisherSAM (Kim et al., 2022), MSAM (Becker et al., 2025), FriendlySAM (Li et al., 2024), and MongeSAM (Jacobsen & Arvanitidis, 2025). We exclude Riemannian SAM (Yun & Yang, 2023), which targets a different model setting and lacks public code. On CIFAR-10 and CIFAR-100 (Krizhevsky, 2009), we use WRN-16-4, WRN-28-10 (Zagoruyko & Komodakis, 2016), ResNet-50

Table 1: Top-1 test accuracy (%) on CIFAR-10 (CNNs and Tiny-ViT). Best per backbone in **bold**.

Method	ResNet-50	WRN-16-4	PyramidNet-110	WRN-28-10	DenseNet-121	Tiny-ViT
SGD	96.44 ±0.05	95.56 ±0.10	95.19 ±0.08	96.94 ±0.06	95.60 ±0.19	78.06 ±0.19
SAM	96.97 ±0.08	96.02 ±0.02	95.87 ±0.05	97.00 ±0.72	95.62 ±0.15	77.61 ±0.22
ASAM	96.56 ±0.10	95.65 ±0.12	95.62 ±0.04	97.16 ±0.08	95.60 ±0.15	79.61 ±0.28
ESAM	96.32 ±0.17	95.52 ±0.08	95.61 ±0.08	97.28 ±0.09	94.87 ±0.23	69.37 ±0.13
GSAM	96.79 ±0.06	95.88 ±0.05	95.91 ±0.10	97.23 ±0.12	95.56 ±0.13	75.51 ±0.13
FisherSAM	96.68 ±0.18	95.88 ±0.04	96.20 ±0.05	**97.48** ±0.05	95.63 ±0.06	75.47 ±0.05
MSAM	96.54 ±0.06	95.60 ±0.19	95.48 ±0.10	96.98 ±0.10	95.51 ±0.23	78.88 ±0.01
FriendlySAM	96.89 ±0.03	95.92 ±0.05	95.90 ±0.13	97.40 ±0.04	95.59 ±0.11	75.82 ±0.26
MongeSAM	96.89 ±0.06	96.02 ±0.03	96.03 ±0.12	97.43 ±0.04	95.77 ±0.10	77.55 ±0.04
OrthoSAM	**97.34** ±0.05	**96.25** ±0.09	**96.46** ±0.09	97.24 ±0.09	**95.85** ±0.12	**87.36** ±0.20

Table 2: Top-1 test accuracy (%) on CIFAR-100 (CNNs and Tiny-ViT). Best per backbone in **bold**.

Method	ResNet-50	WRN-16-4	PyramidNet-110	WRN-28-10	DenseNet-121	Tiny-ViT
SGD	81.84 ±0.13	76.84 ±0.09	81.19 ±0.13	81.85 ±0.05	77.95 ±0.04	52.67 ±0.09
SAM	83.75 ±0.05	78.48 ±0.03	84.02 ±0.01	84.36 ±0.03	78.81 ±0.05	52.72 ±0.16
ASAM	82.40 ±0.02	77.11 ±0.15	81.50 ±0.12	82.90 ±0.14	78.11 ±0.03	52.99 ±0.03
ESAM	82.51 ±0.11	77.00 ±0.03	80.73 ±0.10	84.52 ±0.03	75.89 ±0.04	43.23 ±0.11
GSAM	83.71 ±0.05	78.87 ±0.05	84.05 ±0.03	84.37 ±0.02	78.98 ±0.01	51.28 ±0.06
FisherSAM	83.92 ±0.06	78.84 ±0.02	84.21 ±0.02	84.40 ±0.02	**79.30** ±0.36	51.36 ±0.04
MSAM	82.66 ±0.17	76.99 ±0.04	81.46 ±0.13	82.42 ±0.04	78.18 ±0.18	53.61 ±0.02
FriendlySAM	83.62 ±0.03	78.53 ±0.06	84.21 ±0.00	84.62 ±0.02	79.21 ±0.02	51.75 ±0.06
MongeSAM	83.18 ±0.12	77.64 ±0.07	83.64 ±0.04	84.39 ±0.07	79.15 ±0.01	52.77 ±0.05
OrthoSAM	**84.48** ±0.06	**79.47** ±0.02	**85.12** ±0.03	**84.95** ±0.04	79.07 ±0.11	**64.17** ±0.09

(He et al., 2016), PyramidNet-110 (Han et al., 2017), DenseNet-121 (Huang et al., 2017), and Tiny-ViT; on Tiny-ImageNet, the 200-class ImageNet subset (Deng et al., 2009), we use ResNet-50 and ResNet-101. For each dataset–backbone pair, all optimizers share the same base training recipe for 200 epochs; only the optimizer and its sharpness-specific hyperparameters differ. Each baseline's perturbation radius ρ is set following the configuration recommended in its original paper, and OrthoSAM's is chosen from a small candidate set by validation accuracy. The per-method radius sweep, including an extended range for the baselines, is in Appendix C.5. We report the top-1 test accuracy of the best validation checkpoint as the mean and standard deviation over three seeds. OrthoSAM applies five Newton–Schulz iterations to matrix-shaped hidden weights and the standard SAM branch to one-dimensional parameters and classifier heads (Appendix D); varying the iteration count is studied in Appendix C.4. Equal-wall-clock comparisons and additional training details are provided in Appendices C.6 and B.

5.2 Image Classification

Tables 1 and 2 report the main CIFAR results. Compared with SGD and prior SAM variants, OrthoSAM is consistently competitive across the evaluated convolutional backbones and gives the best result in most settings. The improvement is most visible on Tiny-ViT, where the gap to the SGD-based SAM variants is substantially larger than on the convolutional networks. Since ViT-style architectures are commonly trained with adaptive optimizers such as Adam in standard transformer training recipes rather than plain SGD (Dosovitskiy et al., 2021), we further evaluate Tiny-ViT with Adam-based baselines in Table 3. Adam-based training substantially improves the Tiny-ViT baseline over the corresponding SGD-based SAM variants, but OrthoSAM still achieves higher accuracy on both CIFAR-10 and CIFAR-100. OrthoSAM uses no adaptive moment estimates, yet it surpasses all Adam-based SAM variants in this setting; since its orthogonalized descent (Eq. (12)) is itself a matrix-level preconditioner, Appendix C.8 isolates the perturbation under a shared AdamW descent. Table 4 reports the Tiny-ImageNet results. While existing SAM variants outperform SGD, their accuracy plateaus around 63–64% on both ResNet backbones. In contrast, OrthoSAM breaks this bottleneck, consistently surpassing 67% across both architectures. This indicates that the orthogonalized perturbation remains effective when the classification task is scaled from CIFAR to a

Table 3: Top-1 test accuracy (%) of Tiny-ViT against Adam-based baselines. Best per dataset in **bold**.

Dataset	Adam	Adam+SAM	Adam+ASAM	Adam+GSAM	Adam+FriendlySAM	OrthoSAM
CIFAR-10	85.26 ±0.09	85.39 ±0.04	85.21 ±0.12	85.40 ±0.02	85.38 ±0.09	**87.36** ±0.20
CIFAR-100	62.63 ±0.16	62.84 ±0.03	62.54 ±0.18	62.71 ±0.13	62.50 ±0.11	**64.17** ±0.09

Table 4: Top-1 test accuracy (%) on Tiny-ImageNet. Best per backbone in **bold**.

Method	ResNet-50	ResNet-101
SGD	60.53 ±0.23	60.49 ±0.15
SAM	63.92 ±0.03	63.42 ±0.33
ASAM	60.70 ±0.11	61.00 ±0.15
ESAM	62.97 ±0.10	62.72 ±0.02
GSAM	63.61 ±0.08	64.28 ±0.08
FisherSAM	63.84 ±0.06	64.38 ±0.15
MSAM	61.01 ±0.16	61.47 ±0.09
FriendlySAM	63.95 ±0.09	63.55 ±0.15
MongeSAM	61.51 ±0.04	61.41 ±0.04
OrthoSAM	**67.55** ±0.08	**67.20** ±0.15

Table 5: Label-noise top-1 accuracy (%). Best per setting and noise rate in **bold**.

Setting	Noise	SGD	SAM	OrthoSAM
CIFAR-10 WRN-16-4	20%	88.87	91.06	**91.18**
	40%	82.10	86.30	**86.44**
	80%	21.38	21.94	**25.75**
CIFAR-100 WRN-16-4	20%	66.38	69.95	**70.22**
	40%	57.73	**62.06**	59.42
	80%	10.00	21.98	**25.85**
CIFAR-100 ResNet-50	20%	71.32	72.81	**73.00**
	40%	54.86	55.47	**58.95**
	80%	8.81	12.55	**19.90**

higher-class-count benchmark. The same trend holds at full **ImageNet-1k** scale: OrthoSAM stays ahead of SGD and SAM trained with the same recipe; the results are in Appendix C.10. **Training and test accuracy curves are in Appendix C.1**. The curves show that OrthoSAM improves test performance and enhances generalization of the trained model across different training regimes.

5.3 NLP Fine-Tuning

We further evaluate OrthoSAM on NLP fine-tuning with BERT-base (Devlin et al., 2019) on the GLUE benchmark (Wang et al., 2018), using the same perturbation radius for all SAM-family methods on each task (Table 6). OrthoSAM leads both SGD and SAM on SST-2, QQP, MNLI, and QNLI, the larger classification tasks in the suite. An AdamW-based comparison that isolates the perturbation geometry, together with an autoregressive language-modeling experiment, is in Appendices C.8 and C.9. In both settings AdamW + OrthoSAM improves over AdamW + SAM and AdamW, indicating that orthogonal perturbation remains useful on top of a strong adaptive optimizer.

5.4 Disentangling the Orthogonal Perturbation and Orthogonalized Descent

OrthoSAM changes SAM in two places: it orthogonalizes both the ascent perturbation and the descent update (Eq. (12)). To separate the two, we vary the perturbation (none, SAM, or orthogonal) and the descent rule (SGD with momentum, orthogonalized, or Muon-style), and evaluate seven combinations under a shared recipe across several dataset–backbone pairs (Table 7).

Contribution of Each Component. OrthoSAM is the most accurate variant in every setting, and neither of its components reaches this accuracy alone. Paired with SGD, the orthogonal perturbation stays close to the SGD baseline; paired with the SAM perturbation, the orthogonalized descent recovers most of the gain on Tiny-ViT but only a minor part, or none, on the convolutional backbones. Muon-style descent, with or without the SAM perturbation, also remains below OrthoSAM in every setting, so the improvement is not explained by applying SAM on top of an existing orthogonalized optimizer. The gain of OrthoSAM therefore requires both components.

Coupling of Perturbation and Descent. The two components are coupled rather than additive. Under SGD with momentum, the orthogonal perturbation is no better than the SAM perturbation; under the orthogonalized descent, it is better in every setting. If the components contributed independently, the ranking of the two perturbations would not depend on the descent rule. This interaction follows from the design in Section 4: both steps keep the singular vectors of the gradient and give all of them equal weight. The orthogonalized descent moves the weights by the same amount along

Table 6: Test results of BERT-base fine-tuned on GLUE tasks. Best per reported score in **bold**.

Method	CoLA	SST-2	MRPC	QQP	MNLI	QNLI	RTE	WNLI	STS-B
	Mcc.	*Acc.*	*Acc./F1*	*Acc./F1*	*Mean Acc.*	*Acc.*	*Acc.*	*Acc.*	*Pear./Spea.*
SGD	**1.82**	54.13	**68.38 / 81.22**	73.93 / 70.91	50.47	72.31	**54.15**	**56.34**	38.49 / 61.07
SAM	-2.07	54.70	**68.38 / 81.22**	73.88 / 70.64	49.86	71.57	47.29	**56.34**	**43.53** / 65.45
OrthoSAM	0.00	**90.25**	**68.38 / 81.22**	**85.88 / 81.66**	**77.69**	**86.45**	47.29	**56.34**	25.93 / **72.84**

Table 7: Component ablation of OrthoSAM (top-1 test accuracy, %). The SAM perturbation is $\rho\, G/\|G\|_F$; the orthogonal perturbation is $\rho\, \mathrm{NS}(G)/\|\mathrm{NS}(G)\|_F$ per matrix. *Orthogonalized* descent is OrthoSAM's Newton–Schulz update; *Muon-style* descent is the reference Muon update. Each entry is the best result over the evaluated radii ρ; the full sweep is in Appendix C.5. Best per dataset–backbone in **bold**.

Method	Descent	CIFAR-100 WRN-16-4	CIFAR-100 Tiny-ViT	Tiny-ImageNet R-50	Tiny-ImageNet R-101
SGD	SGD (momentum)	$76.84_{\pm 0.09}$	$52.67_{\pm 0.09}$	$60.53_{\pm 0.23}$	$60.49_{\pm 0.15}$
SAM	SGD (momentum)	$78.48_{\pm 0.03}$	$52.72_{\pm 0.16}$	$63.92_{\pm 0.03}$	$63.42_{\pm 0.33}$
Orthogonal perturbation only	SGD (momentum)	$76.41_{\pm 0.16}$	$52.55_{\pm 0.25}$	$60.91_{\pm 0.07}$	$61.22_{\pm 0.20}$
Muon-style	Muon-style	$77.28_{\pm 0.05}$	$62.09_{\pm 0.02}$	$59.23_{\pm 0.14}$	$62.24_{\pm 0.26}$
SAM + Muon-style	Muon-style	$77.45_{\pm 0.18}$	$63.22_{\pm 0.08}$	$59.96_{\pm 0.11}$	$63.57_{\pm 0.05}$
SAM + orthogonalized descent	orthogonalized	$77.73_{\pm 0.11}$	$62.65_{\pm 0.08}$	$63.18_{\pm 0.28}$	$64.78_{\pm 0.07}$
OrthoSAM	orthogonalized	$\mathbf{79.47}_{\pm 0.02}$	$\mathbf{64.17}_{\pm 0.09}$	$\mathbf{67.55}_{\pm 0.08}$	$\mathbf{67.20}_{\pm 0.15}$

every singular direction, so sharpness along the trailing (small-singular-value) directions matters as much as along the leading ones; the orthogonal perturbation probes all these directions, whereas the SAM perturbation concentrates its budget on the leading ones (Eq. (4)). Under SGD, whose step is dominated by the leading directions, spreading the budget to the trailing ones brings no benefit. Pairing the two steps thus aligns the directions in which OrthoSAM measures sharpness with the directions in which it updates the weights. The two sharpness objectives are compared exactly in Appendix C.3, and the Hessian spectra of the same variants are reported in Appendix C.7.

5.5 Robustness to Label Noise

We evaluate label-noise robustness by corrupting training labels uniformly at random (20%, 40%, or 80%) while testing on clean data (Table 5). Both sharpness-aware methods improve over SGD at every noise rate, consistent with the known robustness of SAM to label noise. OrthoSAM is the most accurate method in all but one case, and its advantage over SAM is largest at 80% noise, where most training labels are wrong: there it leads SAM by a clear margin in every setting. The benefit of OrthoSAM is thus concentrated where regularization matters most, suggesting that controlling sharpness along all singular directions of the gradient, rather than only the dominant ones, yields a stronger regularizer when fitting corrupted labels dominates training.

6 Conclusion

We identified a limitation of standard SAM: by flattening weight matrices into vectors, it weights the adversarial perturbation by the singular values of the gradient, so a few dominant directions absorb most of the budget. We proposed **OrthoSAM**, which constrains each matrix perturbation to a scaled semi-orthogonal matrix, weighting all singular directions equally, and orthogonalizes the descent step as well, both via Newton–Schulz iterations. Our ablation shows that the two steps are coupled: neither alone matches the full method. We extended the PAC-Bayes bound of SAM to this orthogonal neighborhood and proved convergence of the idealized update. Across convolutional and transformer architectures, OrthoSAM improves generalization over SAM and its recent variants in most settings, finds flatter minima, and is the most robust method under high label noise.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

We have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proofs, assist in writing proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean or reformat datasets, support qualitative data analysis, or interpret results. Additionally, we used large language models only to aid and polish writing: editing sentence wording and grammar, drafting figure and table captions, and designing and aligning tables. We have reviewed all AI-assisted text. We take responsibility for the final content of this work, including text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

The OrthoSAM update is derived in Section 4. Appendix D gives the full procedure (Algorithms 1 and 2), including the Newton–Schulz orthogonalization used for both the ascent perturbation and the descent step, and the fallback to standard SAM on scalar parameters and classifier heads. Assumptions (A1)–(A2), the PAC-Bayes generalization bound (Theorem 4.1), the closed-form orthogonal perturbation (Theorem 4.2), and the convergence guarantees (Theorem 4.3, Corollary 4.4) are stated in the main text; complete proofs are in Appendix A. Section 5.1 and Appendix B record the shared training recipes, the choice of ρ, hardware, and the Lanczos protocol used for Hessian spectra. Sensitivity studies for ρ and for the number of Newton–Schulz iterations, the equal-wall-clock comparison, the ImageNet-1k experiment, and the AdamW-based comparison on GLUE are in Appendices C.5, C.4, C.6, C.10, and C.8. Source code for OrthoSAM and for the experiments reported in this paper is provided as supplementary material.

REFERENCES

Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *International Conference on Machine Learning*, pp. 639–668, 2022.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pp. 233–242, 2017.

Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2003.

Marlon Becker, Frederick Altrock, and Benjamin Risse. Momentum-SAM: Sharpness aware minimization without computational overhead. In *Advances in Neural Information Processing Systems*, 2025.

Zixiang Chen, Junkai Zhang, Yiwen Kou, Xiangning Chen, Cho-Jui Hsieh, and Quanquan Gu. Why does sharpness-aware minimization generalize better than SGD? In *Advances in Neural Information Processing Systems*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

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.

Jiawei Du, Hanshu Yan, Jiashi Feng, Joey Tianyi Zhou, Liangli Zhen, Rick Siow Mong Goh, and Vincent Y. F. Tan. Efficient sharpness-aware minimization for improved training of neural networks. In *International Conference on Learning Representations*, 2022.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

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

Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. In *Advances in Neural Information Processing Systems*, pp. 529–536, 1994.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

Albert Kjøller Jacobsen and Georgios Arvanitidis. Monge SAM: Robust reparameterization-invariant sharpness-aware minimization based on loss geometry. *arXiv preprint arXiv:2502.08448*, 2025.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020.

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

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. In *International Conference on Learning Representations*, 2017.

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

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

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, 2021.

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

Tao Li, Pan Zhou, Zhengbao He, Xinwen Cheng, and Xiaolin Huang. Friendly sharpness-aware minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for LLM training, 2025. Technical Report, Moonshot AI.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *Advances in Neural Information Processing Systems*, volume 33, pp. 20331–20342, 2020.

Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12360–12370, 2022.

Haocheng Luo, Tuan Truong, Tung Pham, Mehrtash Harandi, Dinh Phung, and Trung Le. Explicit eigenvalue regularization improves sharpness-aware minimization. In *Advances in Neural Information Processing Systems*, 2024.

Enea Monzio Compagnoni, Luca Biggio, Antonio Orvieto, Frank Norbert Proske, Hans Kersting, and Aurelien Lucchi. An SDE for modeling SAM: Theory and insights. In *International Conference on Machine Learning*, pp. 25209–25253, 2023.

Gonçalo Mordido, Pranshu Malviya, Aristide Baratin, and Sarath Chandar. Lookbehind-SAM: k steps back, 1 step forward. In *International Conference on Machine Learning*, 2024.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Henning Petzka, Michael Kamp, Linara Adilova, Cristian Sminchisescu, and Mario Boley. Relative flatness and generalization. In *Advances in Neural Information Processing Systems*, pp. 18420–18432, 2021.

Dongkuk Si and Chulhee Yun. Practical sharpness-aware minimization cannot converge all the way to optima. In *Advances in Neural Information Processing Systems*, 2023.

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, 2018.

Kaiyue Wen, Zhiyuan Li, and Tengyu Ma. Sharpness minimization algorithms do not only minimize sharpness to achieve better generalization. In *Advances in Neural Information Processing Systems*, 2023a.

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. How sharpness-aware minimization minimizes sharpness? In *International Conference on Learning Representations*, 2023b.

Runsheng Yu, Youzhi Zhang, and James Kwok. Improving sharpness-aware minimization by lookahead. In *International Conference on Machine Learning*, 2024.

Jihun Yun and Eunho Yang. Riemannian SAM: Sharpness-aware minimization on Riemannian manifolds. In *Advances in Neural Information Processing Systems*, 2023.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

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.

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.

A ALL PROOFS

A.1 PROOF OF THEOREM 4.1

Proof. Let θ be an arbitrary parameter point. We identify $\mathbb{R}^{m\times n}$ with $\mathbb{R}^k$ ($k = mn$) by vectorization, which preserves the Frobenius norm. We have the set $\mathbb{R}_{\perp}^{m\times n}$ is closed and contains the zero matrix. Consequently, $\mathcal{N}_\rho(\theta)$ is compact and contains θ. In particular, the maximum in (6) exists whenever the empirical loss is Lipschitz. Consider the Gaussian posterior $Q = \mathcal{N}(\theta, \sigma^2\mathbb{I}_k)$, where

$$\sigma = \frac{\rho}{2(\sqrt{k} + \sqrt{\log N})}. \tag{18}$$

For any distribution R on the parameter space, write $\mathcal{L}_{\mathcal{D}}(R) = \mathbb{E}_{Z\sim R}[\mathcal{L}_{\mathcal{D}}(Z)]$, and define $\mathcal{L}_{\mathcal{S}}(R)$ analogously. For a Gaussian prior $P = \mathcal{N}(\mathbf{0}, \sigma_P^2\mathbb{I}_k)$ chosen independently of S, the PAC-Bayes-kl inequality applied to ℓ/L, followed by Pinsker's inequality, gives, for any fixed $\eta \in (0, 1)$, with probability at least $1 - \eta$:

$$\mathcal{L}_{\mathcal{D}}(Q) \le \mathcal{L}_{\mathcal{S}}(Q) + L\sqrt{\frac{\mathrm{KL}(Q\|P) + \log(2\sqrt{N}/\eta)}{2N}}. \tag{19}$$

This inequality holds simultaneously for all posteriors with finite KL divergence, including all choices of the posterior mean θ. For Gaussians,

$$\mathrm{KL}(Q\|P) = \frac{1}{2}\Big[\frac{k\sigma^2 + \|\theta\|_F^2}{\sigma_P^2} - k + k\log\frac{\sigma_P^2}{\sigma^2}\Big].$$

The variance minimizing this expression is $\sigma_P^2 = \sigma^2 + \|\theta\|_F^2/k$, at which the KL divergence equals $\frac{1}{2}k\log(1 + \|\theta\|_F^2/(k\sigma^2))$. We first consider the case

$$\|\theta\|_F^2 < k\sigma^2\big(e^{4N/k} - 1\big). \tag{20}$$

The complementary case will be treated at the end of the proof. Since the optimal σ_P depends on θ, we employ a grid of priors. Set:

$$\begin{aligned} c &= \sigma^2\big(1 + \exp\{4N/k\}\big),\\ P_j &= \mathcal{N}\big(\mathbf{0}, c\exp\{(1-j)/k\}\mathbb{I}_k\big),\\ \mathfrak{P} &= \{P_j : j = 1, 2, \dots\}. \end{aligned} \tag{21}$$

Each prior in $\mathfrak{P}$ is independent of both S and θ. With $\delta_j = 6\delta/(\pi^2 j^2)$ and $\sum_{j=1}^{\infty}\delta_j = \delta$, a union bound applied to (19) implies that, with probability at least $1 - \delta$, simultaneously for all j and all θ,

$$\mathcal{L}_{\mathcal{D}}(Q) \le \mathcal{L}_{\mathcal{S}}(Q) + L\sqrt{\frac{\mathrm{KL}(Q\|P_j) + \log(2\sqrt{N}/\delta_j)}{2N}}. \tag{22}$$

Intersect this event with the probability-one event on which $\mathcal{L}_{\mathcal{S}}$ is γ-Lipschitz, and work on the resulting event throughout the proof. We pick:

$$j^* = \Big\lfloor 1 - k\log\frac{\sigma^2 + \|\theta\|_F^2/k}{c}\Big\rfloor. \tag{23}$$

By (20), $\sigma^2 + \|\theta\|_F^2/k < \sigma^2 e^{4N/k} < c$, so $j^* \ge 1$. Writing $\sigma_{P_j}^2 = c\exp\{(1-j)/k\}$, we have

$$\begin{aligned} & 1 - j^* = \Big\lceil k\log\frac{\sigma^2 + \|\theta\|_F^2/k}{c}\Big\rceil \\ \Rightarrow\quad & \log\frac{\sigma^2 + \|\theta\|_F^2/k}{c} \le \frac{1-j^*}{k} \le \log\frac{\sigma^2 + \|\theta\|_F^2/k}{c} + \frac{1}{k} \\ \Rightarrow\quad & \sigma^2 + \frac{\|\theta\|_F^2}{k} \le \sigma_{P_{j^*}}^2 \le \exp(1/k)\Big[\sigma^2 + \frac{\|\theta\|_F^2}{k}\Big]. \end{aligned} \tag{24}$$

Now the KL term can be bounded as:

$$\begin{aligned}\mathrm{KL}(Q\|P_{j^*}) &= \frac{1}{2}\Big[\frac{k\sigma^2+\|\theta\|_F^2}{\sigma^2_{P_{j^*}}} - k + k\log\frac{\sigma^2_{P_{j^*}}}{\sigma^2}\Big] \\ &\le \frac{1}{2}\Big[\frac{k(\sigma^2+\|\theta\|_F^2/k)}{\sigma^2+\|\theta\|_F^2/k} - k + k\log\frac{\exp(1/k)(\sigma^2+\|\theta\|_F^2/k)}{\sigma^2}\Big] \\ &= \frac{1}{2}\Big[1 + k\log\Big(1+\frac{\|\theta\|_F^2}{k\sigma^2}\Big)\Big]. \end{aligned} \tag{25}$$

For the log-term, recalling $c = \sigma^2(1+\exp(4N/k))$, we have:

$$\begin{aligned}\log\frac{1}{\delta_{j^*}} &= \log\frac{1}{\delta} + \log\Big(\frac{\pi^2}{6}\Big) + 2\log(j^*) \\ &\le \log\frac{1}{\delta} + \log\frac{\pi^2}{6} + 2\log\Big(1 + k\log\frac{\sigma^2(1+\exp(4N/k))}{\sigma^2+\|\theta\|_F^2/k}\Big) \\ &\le \log\frac{1}{\delta} + \log\frac{\pi^2}{6} + 2\log\Big(1 + k\log(1+\exp(4N/k))\Big) \\ &\le \log\frac{1}{\delta} + \log\frac{\pi^2}{6} + 2\log(1+k+4N). \end{aligned} \tag{26}$$

Plugging (25) and (26) into (22) gives:

$$\mathcal{L}_{\mathcal{D}}(Q) \le \mathcal{L}_{\mathcal{S}}(Q) + \frac{L}{2\sqrt{N}}\sqrt{H_\theta}, \tag{27}$$

where

$$H_\theta = 1 + k\log\Big(1+\frac{\|\theta\|_F^2}{k\sigma^2}\Big) + 2\log\frac{\pi^2\sqrt{N}}{3\delta} + 4\log(1+k+4N). \tag{28}$$

Bounding $\mathcal{L}_{\mathcal{S}}(Q)$. Recall that $Q = \mathcal{N}(\theta, \sigma^2\mathbb{I}_k)$. For each z, let $\pi_\theta(z)$ be a closest point to z in $\mathcal{N}_\rho(\theta)$, choosing any minimizer if the closest point is not unique. Such a point exists because $\mathcal{N}_\rho(\theta)$ is compact. Since $\theta \in \mathcal{N}_\rho(\theta)$, we have

$$\|z - \pi_\theta(z)\|_F \le \|z-\theta\|_F. \tag{29}$$

We use this projection only to compare losses pointwise. For $Z \sim Q$, the squared Frobenius distance $\|Z-\theta\|_F^2$ is distributed as σ^2 times a χ^2 random variable with k degrees of freedom. Therefore, for any $t > 0$,

$$\mathbb{P}\big(\|Z-\theta\|_F^2 - k\sigma^2 \ge 2\sigma^2\sqrt{kt} + 2t\sigma^2\big) \le e^{-t}. \tag{30}$$

For $N > 1$, setting $t = \frac{1}{2}\log N$ gives, with probability at least $1 - N^{-1/2}$,

$$\begin{aligned}\|Z-\theta\|_F^2 &\le k\sigma^2 + \sigma^2\sqrt{2k\log N} + \sigma^2\log N \\ &\le k\sigma^2\Big(1+\sqrt{\frac{\log N}{k}}\Big)^2 = \frac{\rho^2}{4}, \end{aligned} \tag{31}$$

where the last equality follows from (18). Hence

$$\mathbb{P}\big(\|Z-\theta\|_F \le \rho/2\big) \ge 1 - \frac{1}{\sqrt{N}}. \tag{32}$$

For $N = 1$, this inequality holds trivially. Denote $\mathcal{B}_{\rho/2}(\theta) = \{z : \|z-\theta\|_F \le \rho/2\}$. For every $z \in \mathcal{B}_{\rho/2}(\theta)$, the γ-Lipschitz continuity of $\mathcal{L}_{\mathcal{S}}$ and (29) imply

$$\begin{aligned}\mathcal{L}_{\mathcal{S}}(z) &\le \mathcal{L}_{\mathcal{S}}(\pi_\theta(z)) + \gamma\|z-\pi_\theta(z)\|_F \\ &\le \max_{\theta'\in\mathcal{N}_\rho(\theta)} \mathcal{L}_{\mathcal{S}}(\theta') + \frac{\gamma\rho}{2}. \end{aligned} \tag{33}$$

Now we split the expectation into points inside $\mathcal{B}_{\rho/2}(\theta)$ and points outside:

$$\begin{aligned}\mathcal{L}_{\mathcal{S}}(Q) &= \int_{\mathcal{B}_{\rho/2}(\theta)} \mathcal{L}_{\mathcal{S}}(z)\, dQ(z) + \int_{\mathbb{R}^k \setminus \mathcal{B}_{\rho/2}(\theta)} \mathcal{L}_{\mathcal{S}}(z)\, dQ(z) \\ &\le \Big[\max_{\theta' \in \mathcal{N}_\rho(\theta)} \mathcal{L}_{\mathcal{S}}(\theta') + \frac{\gamma\rho}{2} \Big] Q(\mathcal{B}_{\rho/2}(\theta)) + L Q(\mathbb{R}^k \setminus \mathcal{B}_{\rho/2}(\theta)) \\ &\le \max_{\theta' \in \mathcal{N}_\rho(\theta)} \mathcal{L}_{\mathcal{S}}(\theta') + \frac{\gamma\rho}{2} + \frac{L}{\sqrt{N}}. \end{aligned} \tag{34}$$

The first inequality uses (33) inside the ball and the global bound L outside. The last inequality follows from (32) and the nonnegativity of the loss.

Bounding the Population Loss $\mathcal{L}_{\mathcal{D}}(\theta)$. To connect the expected population loss $\mathcal{L}_{\mathcal{D}}(Q)$ to the loss at θ, we use the γ-Lipschitz continuity of $\mathcal{L}_{\mathcal{D}}$. For $Z \sim Q$,

$$\mathcal{L}_{\mathcal{D}}(\theta) \le \mathcal{L}_{\mathcal{D}}(Z) + \gamma \|Z - \theta\|_F.$$

Moreover, by Cauchy–Schwarz and (18),

$$\mathbb{E}_{Z\sim Q} \|Z - \theta\|_F \le \Big(\mathbb{E}_{Z\sim Q} \|Z - \theta\|_F^2 \Big)^{1/2} = \sigma\sqrt{k} \le \frac{\rho}{2}.$$

Taking expectations therefore yields

$$\mathcal{L}_{\mathcal{D}}(\theta) \le \mathcal{L}_{\mathcal{D}}(Q) + \frac{\gamma\rho}{2}. \tag{35}$$

Completing the Proof. Insert (34) and (35) into (27):

$$\begin{aligned}\mathcal{L}_{\mathcal{D}}(\theta) &\le \mathcal{L}_{\mathcal{D}}(Q) + \frac{\gamma\rho}{2} \\ &\le \max_{\theta' \in \mathcal{N}_\rho(\theta)} \mathcal{L}_{\mathcal{S}}(\theta') + \frac{L}{\sqrt{N}} + \frac{L}{2\sqrt{N}} \sqrt{H_\theta} + \gamma\rho. \end{aligned} \tag{36}$$

If (20) does not hold, then

$$k \log\Big(1 + \frac{\|\theta\|_F^2}{k\sigma^2}\Big) \ge 4N.$$

All remaining terms in H_θ are nonnegative, so $\frac{L}{2\sqrt{N}}\sqrt{H_\theta} \ge L$. Since $\mathcal{L}_{\mathcal{D}}(\theta) \le L$ and the other terms on the right-hand side of (36) are nonnegative, this bound also holds in the complementary case. Thus (36) holds simultaneously for all θ on the same event. To express this bound in the form of the theorem, set

$$u_\theta = \frac{\|\theta\|_F^2}{\rho^2} \Big(1 + \sqrt{\frac{\log N}{k}}\Big)^2, \qquad A_\theta = k \log(1 + u_\theta), \qquad D = \log \frac{N + k}{\delta}.$$

Our choice of σ gives

$$k \log\Big(1 + \frac{\|\theta\|_F^2}{k\sigma^2}\Big) = k \log(1 + 4u_\theta) \le 4k \log(1 + u_\theta) = 4A_\theta,$$

where the inequality follows from $1 + 4u \le (1 + u)^4$ for $u \ge 0$. Using $1 + k + 4N \le 4(N + k)$ and $\log N \le \log(N + k)$, we obtain

$$\begin{aligned} H_\theta &\le 4A_\theta + 5\log(N + k) + 2\log(1/\delta) + 1 + 2\log(\pi^2/3) + 4\log 4 \\ &\le 4A_\theta + 5D + 9, \end{aligned} \tag{37}$$

where $1 + 2\log(\pi^2/3) + 4\log 4 < 9$. Consequently,

$$\begin{aligned} \frac{L}{\sqrt{N}} + \frac{L}{2\sqrt{N}}\sqrt{H_\theta} &\le \frac{L}{\sqrt{N}} \Big[\sqrt{A_\theta} + \frac{\sqrt{5}}{2}\sqrt{D} + \frac{5}{2} \Big] \\ &\le \frac{4L}{\sqrt{N}} \Big[\sqrt{A_\theta} + 2\sqrt{D} + 1 \Big]. \end{aligned} \tag{38}$$

Substituting this into (36), we arrive at the final bound:

$$\mathcal{L}_{\mathcal{D}}(\theta) \leq \max_{\theta' \in \mathcal{N}_\rho(\theta)} \mathcal{L}_{\mathcal{S}}(\theta') + \frac{4L}{\sqrt{N}} \left[\sqrt{k \log \left(1 + \frac{\|\theta\|_F^2}{\rho^2} \left(1 + \sqrt{\frac{\log N}{k}}\right)^2\right)} + 2\sqrt{\log \frac{N+k}{\delta}} + 1 \right] + \gamma\rho.$$

The bound holds simultaneously for all θ with probability at least $1-\delta$ over S. This completes the proof. □

A.2 Proof of Theorem 4.2

Proof. We need to find $\Delta \in \mathbb{R}_{\perp}^{m\times n} \cap \mathcal{B}_\rho(\theta)$ to maximize $\mathrm{Tr}(G^\top \Delta)$. We have $\Delta^\top \Delta = \alpha \mathbb{I}_n$, which leads to:

$$\|\Delta\|_F = \mathrm{tr}\left(\Delta^\top \Delta\right)^{1/2} = (n\alpha)^{1/2} \leq \rho,$$
$$\alpha \leq \frac{\rho^2}{n}.$$

Moreover, we have:

$$\mathrm{Tr}\left(G^\top \Delta\right) = \mathrm{Tr}\left(V \Sigma U^\top \Delta\right) = \mathrm{Tr}\left(\Sigma U^\top \Delta V\right) = \mathrm{Tr}\left(\Sigma Z\right),$$

where we denote $Z = U^\top \Delta V \in \mathbb{R}^{n\times n}$ and write Z_i^c for its i-th column. Since U has orthonormal columns and $\Delta^\top \Delta = \alpha \mathbb{I}_n$, each column of Z obeys:

$$\left(Z_i^c\right)^\top Z_i^c = \|U^\top \Delta v_i\|_2^2 \leq \|\Delta v_i\|_2^2 = v_i^\top \Delta^\top \Delta v_i = \alpha,$$

so that $|Z_{ii}| \leq \left[\left(Z_i^c\right)^\top Z_i^c\right]^{1/2} \leq \sqrt{\alpha}$. Therefore, we obtain:

$$\mathrm{Tr}\left(\Sigma Z\right) = \sum_i \sigma_i Z_{ii} \leq \sum_i \sigma_i |Z_{ii}| \leq \sqrt{\alpha} \sum_i \sigma_i \leq \frac{\rho}{n^{1/2}} \sum_i \sigma_i,$$

where the last step uses $\alpha \leq \rho^2/n$. The maximum is attained when:

$$\sqrt{\alpha} = \frac{\rho}{n^{1/2}} \quad \left(\text{i.e., } \alpha = \frac{\rho^2}{n}\right) \text{ and } Z = \frac{\rho}{n^{1/2}} \mathbb{I}_n,$$

which is feasible ($\Delta^\top \Delta = \alpha \mathbb{I}_n$ and $\|\Delta\|_F = \rho$) and means that:

$$U^\top \Delta V = Z = \frac{\rho}{n^{1/2}} \mathbb{I}_n, \qquad \Delta = \frac{\rho}{n^{1/2}} U V^\top.$$

□

A.3 Proof of Theorem 4.3

Proof. Let $G_l = \nabla \mathcal{L}_{\mathcal{S}}(\theta_l)$ and $g_l = \nabla \mathcal{L}_{B_l}(\theta_l + \Delta_l)$. By the β-smoothness of the loss function, we have:

$$\mathcal{L}_{\mathcal{S}}(\theta_{l+1}) \leq \mathcal{L}_{\mathcal{S}}(\theta_l) - \eta \langle G_l, \tilde{O}_l \rangle + \frac{\beta \eta^2}{2} \|\tilde{O}_l\|_F^2. \tag{39}$$

Under the idealized NS update, $\|\tilde{O}_l\|_F^2 \leq n$ and $\langle g_l, \tilde{O}_l \rangle = \|g_l\|_*$, including the case $g_l = 0$. Taking the expectation, the bound simplifies to:

$$\mathbb{E}[\mathcal{L}_{\mathcal{S}}(\theta_{l+1})] \leq \mathbb{E}[\mathcal{L}_{\mathcal{S}}(\theta_l)] - \eta \mathbb{E}[\langle G_l, \tilde{O}_l \rangle] + \frac{\beta \eta^2 n}{2}. \tag{40}$$

We now bound the inner product term $-\langle G_l, \tilde{O}_l \rangle$. By adding and subtracting g_l, we get:

$$\begin{aligned} -\langle G_l, \tilde{O}_l \rangle &= -\langle g_l, \tilde{O}_l \rangle + \langle g_l - G_l, \tilde{O}_l \rangle \\ &= -\|g_l\|_* + \langle g_l - G_l, \tilde{O}_l \rangle. \end{aligned} \tag{41}$$

Applying the Cauchy–Schwarz inequality to the second term and using $\|\tilde{O}_l\|_F \le \sqrt{n}$:

$$\langle g_l - G_l, \tilde{O}_l \rangle \le \|g_l - G_l\|_F \|\tilde{O}_l\|_F \le \sqrt{n}\|g_l - G_l\|_F. \tag{42}$$

For the nuclear norm term, we apply the lower bound $\|g_l\|_* \ge \|g_l\|_F$, and then use the reverse triangle inequality $\|g_l\|_F \ge \|G_l\|_F - \|g_l - G_l\|_F$. This yields:

$$-\|g_l\|_* \le -\|g_l\|_F \le -\|G_l\|_F + \|g_l - G_l\|_F. \tag{43}$$

Combining these inequalities and taking the expectation, we obtain:

$$-\mathbb{E}[\langle G_l, \tilde{O}_l \rangle] \le -\mathbb{E}[\|G_l\|_F] + (\sqrt{n}+1)\mathbb{E}[\|g_l - G_l\|_F]. \tag{44}$$

Next, we bound the gradient deviation $\mathbb{E}[\|g_l - G_l\|_F]$. Let $\hat{g}_l = \nabla \mathcal{L}_{B_l}(\theta_l)$ be the mini-batch gradient at the unperturbed iterate. Using the triangle inequality, the mini-batch smoothness in (A1), and the variance bound in (A2):

$$\begin{aligned}
\mathbb{E}[\|g_l - G_l\|_F] &\le \mathbb{E}[\|g_l - \hat{g}_l\|_F] + \mathbb{E}[\|\hat{g}_l - G_l\|_F] \\
&\le \beta \mathbb{E}[\|\Delta_l\|_F] + \left(\mathbb{E}[\|\hat{g}_l - G_l\|_F^2]\right)^{1/2} \\
&\le \sigma + \beta\rho.
\end{aligned} \tag{45}$$

Since θ_l is determined by the preceding iterations and B_l is sampled independently, the uniform variance bound in (A2) gives $\mathbb{E}[\|\hat{g}_l - G_l\|_F^2] \le \sigma^2$. The perturbation Δ_l may depend on the same mini-batch.

Substituting (45) into (44), and then into (40), we obtain:

$$\begin{aligned}
\mathbb{E}[\mathcal{L}_{\mathcal{S}}(\theta_{l+1})] \le \mathbb{E}[\mathcal{L}_{\mathcal{S}}(\theta_l)] &- \eta \mathbb{E}[\|G_l\|_F] \\
&+ \eta(\sqrt{n}+1)(\sigma + \beta\rho) + \frac{\beta\eta^2 n}{2}.
\end{aligned} \tag{46}$$

Rearranging and summing over $l = 0, \dots, T-1$ yields a telescoping sum for the loss:

$$\begin{aligned}
\eta \sum_{l=0}^{T-1} \mathbb{E}[\|G_l\|_F] &\le \mathcal{L}_{\mathcal{S}}(\theta_0) - \mathbb{E}[\mathcal{L}_{\mathcal{S}}(\theta_T)] \\
&\quad + \eta T(\sqrt{n}+1)(\sigma + \beta\rho) + \frac{\beta\eta^2 nT}{2} \\
&\le D_0 + \eta T(\sqrt{n}+1)(\sigma + \beta\rho) + \frac{\beta\eta^2 nT}{2}.
\end{aligned} \tag{47}$$

Dividing by ηT produces the final bound:

$$\frac{1}{T}\sum_{l=0}^{T-1} \mathbb{E}[\|G_l\|_F] \le \frac{D_0}{\eta T} + \frac{\beta\eta n}{2} + (\sqrt{n}+1)(\sigma + \beta\rho). \tag{48}$$

This completes the proof. □

A.4 Proof of Corollary 4.4

Proof. Applying the fundamental property of averages $\min_{0 \le l < T} a_l \le \frac{1}{T}\sum_{l=0}^{T-1} a_l$ to Theorem 4.3, we substitute $T = \tau n$ and the learning rate

$$\eta = \sqrt{\frac{2D_0}{\beta n T}} = \frac{1}{n}\sqrt{\frac{2D_0}{\beta\tau}},$$

which minimizes the first two terms of the bound for $D_0 > 0$. The corresponding optimization terms reduce to:

$$\begin{aligned}
\frac{D_0}{\eta T} + \frac{\beta\eta n}{2} &= \frac{D_0}{\left(\frac{1}{n}\sqrt{\frac{2D_0}{\beta\tau}}\right)\tau n} + \frac{\beta n}{2}\left(\frac{1}{n}\sqrt{\frac{2D_0}{\beta\tau}}\right) \\
&= \sqrt{\frac{\beta D_0}{2\tau}} + \sqrt{\frac{\beta D_0}{2\tau}} = \sqrt{\frac{2\beta D_0}{\tau}}.
\end{aligned} \tag{49}$$

Next, we substitute the scaled perturbation radius $\rho = \tilde{\rho} n^{-1/2}$ into the residual term:

$$(\sqrt{n}+1)(\sigma+\beta\rho) = (\sqrt{n}+1)\left(\sigma + \frac{\beta\tilde{\rho}}{\sqrt{n}}\right)$$
$$= (\sqrt{n}+1)\sigma + \beta\tilde{\rho}\left(1+\frac{1}{\sqrt{n}}\right). \tag{50}$$

Combining these components yields:

$$\min_{0\le l<T} \mathbb{E}\big[\|\nabla \mathcal{L}_{\mathcal{S}}(\theta_l)\|_F\big] \le \sqrt{\frac{2\beta D_0}{\tau}} + (\sqrt{n}+1)\sigma + \beta\tilde{\rho}\left(1+\frac{1}{\sqrt{n}}\right). \tag{51}$$

Finally, if $\rho = \bar{\rho}/\sqrt{n\tau}$ and $\sigma \le \bar{\sigma}/\sqrt{n\tau}$, then

$$(\sqrt{n}+1)(\sigma+\beta\rho) \le \frac{1}{\sqrt{\tau}}\left(1+\frac{1}{\sqrt{n}}\right)(\bar{\sigma}+\beta\bar{\rho}). \tag{52}$$

Substituting this inequality into the optimized bound proves (17). This completes the proof. □

B Additional Experimental Details

B.1 Training Details

Main training runs (image classification and NLP fine-tuning) use four NVIDIA A100 GPUs (40 GB). Wall-clock measurements—the Newton–Schulz iteration sweep and the equal-wall-clock comparison—are timed on a single NVIDIA A100 (80 GB), so that per-epoch cost is not confounded by multi-GPU communication. Every image-classification run trains for 200 epochs; we select the checkpoint with the highest validation accuracy and report that checkpoint's test accuracy, as mean and standard deviation over three independent seeds.

For each dataset–backbone pair the base recipe (learning rate, batch size, weight decay, and augmentation) is shared across all evaluated methods; only the update rule and its sharpness-specific hyperparameters change. Each baseline's perturbation radius ρ is set following the configuration recommended in its original paper, and OrthoSAM's radius is chosen from $\{0.005, 0.01, 0.045, 0.05, 0.1, 0.5\}$ by validation accuracy. Where the per-method sweep over an extended range (Appendix C.5) finds a better radius for any method, baselines included, we report the result at that radius instead. OrthoSAM applies the Newton–Schulz orthogonalization branch (five iterations) to all matrix-shaped hidden weights ($\dim \ge 2$, excluding embeddings and classifier heads); one-dimensional parameters follow the standard SAM branch.

CIFAR-10 and CIFAR-100 (Convolutional Backbones). We use SGD with momentum 0.9 (without Nesterov acceleration), cosine learning-rate decay after a one-epoch linear warmup, random crop with four-pixel padding, horizontal flipping, Cutout with probability 0.5, per-channel normalization, and label smoothing with coefficient 0.1. Backbone-specific hyperparameters are listed in Table 8.

CIFAR-10 and CIFAR-100 (Tiny-ViT). For the main comparison (Tables 1 and 2), Tiny-ViT is trained with SGD under the same momentum and learning-rate schedule as the convolutional backbones (cosine decay with a ten-epoch warmup, learning rate 10^{-3}, weight decay 5×10^{-4}, batch size 256). Cutout and label smoothing are disabled for this architecture.

Tiny-ViT with Adam Baselines. Since ViT-style architectures are typically trained with adaptive optimizers, we additionally compare OrthoSAM against Adam-based baselines (Adam, Adam + SAM, Adam + ASAM, Adam + GSAM, and Adam + FriendlySAM) on CIFAR-10 and CIFAR-100. These baselines use Adam with learning rate 10^{-3}, weight decay 5×10^{-4}, batch size 256, cosine decay with a ten-epoch warmup, and $\rho = 0.01$ for SAM variants. OrthoSAM keeps the update used in Tables 1 and 2 (Algorithm 2) and uses no Adam-style moment estimates.

Tiny-ImageNet (ResNet-50 and ResNet-101). We use SGD with momentum 0.9, initial learning rate 0.05, weight decay 10^{-3}, batch size 125, cosine decay after a one-epoch warmup, and RandAugment with magnitude 10. Label smoothing is set to 0.1.

Table 8: Shared training hyperparameters for all image-classification experiments. Every configuration trains for 200 epochs. All backbones use SGD with momentum 0.9 unless noted otherwise.

Dataset	Backbone	Batch size	Initial learning rate	Momentum	Weight decay
CIFAR-10	ResNet-50	256	0.10	0.9	1.0×10^{-3}
	WRN-16-4	256	0.50	0.9	5.0×10^{-4}
	PyramidNet-110	256	0.50	0.9	5.0×10^{-4}
	WRN-28-10	128	0.05	0.9	5.0×10^{-4}
	DenseNet-121	256	0.10	0.9	5.0×10^{-5}
	Tiny-ViT	256	10^{-3}	0.9	5.0×10^{-4}
CIFAR-100	ResNet-50	256	0.10	0.9	1.0×10^{-3}
	WRN-16-4	256	0.50	0.9	5.0×10^{-4}
	PyramidNet-110	256	0.50	0.9	5.0×10^{-4}
	WRN-28-10	128	0.05	0.9	5.0×10^{-4}
	DenseNet-121	256	0.10	0.9	5.0×10^{-5}
	Tiny-ViT	256	10^{-3}	0.9	5.0×10^{-4}
Tiny-ImageNet	ResNet-50	125	0.05	0.9	1.0×10^{-3}
	ResNet-101	125	0.05	0.9	1.0×10^{-3}

NLP Fine-Tuning (BERT-base on GLUE). BERT-base is fine-tuned for 3 epochs with a linear learning-rate schedule (peak 2×10^{-5}), batch size 32, and gradient clipping at global norm 1. Task-specific perturbation radii are: $\rho=0.002$ for CoLA; $\rho=0.004$ for RTE, STS-B, and WNLI; $\rho=0.005$ for the remaining tasks.

Hessian Spectrum Computation. To evaluate the curvature of the learned minima, we approximate the top-k eigenvalues of the full-training-loss Hessian using the Lanczos algorithm, following the protocol adopted in prior sharpness-aware studies (Foret et al., 2021; Li & Giannakis, 2023; Becker et al., 2025). Concretely, we load the final checkpoint, freeze all parameters, and compute Hessian–vector products via two back-to-back automatic-differentiation passes (forward→first-order gradient with `create_graph=True`→dot product with the probe vector→second-order gradient). Each Hessian–vector product is averaged over 20 mini-batches of size 128 drawn from the training set with all data augmentation disabled. The Lanczos iteration runs for 100 steps with full re-orthogonalization, and we report the five largest eigenvalues, averaged over five random seeds (mean and standard deviation in Appendix C.7).

C Additional Experiments and Analyses

Unless otherwise stated, experiments in this section follow the recipes in Appendix B.1.

C.1 Training Dynamics

Figures 1 and 2 show training and test accuracy curves for the runs in Section 5.2. On ResNet-50, OrthoSAM improves rapidly in the early stage, but its final training accuracy is lower. Its test accuracy, however, stays above the other methods throughout training. This behavior suggests that the gain comes from improved generalization rather than stronger fitting of the training set. On PyramidNet-110, both training and test curves fluctuate during optimization, but OrthoSAM remains consistently higher over most of training. Overall, the curves show that OrthoSAM improves test performance and enhances generalization of the trained model across different training regimes.

C.2 Motivation: Empirical and Mechanistic Evidence

This section provides the empirical support for the two claims in Section 3: that SAM's gradient-aligned perturbation (i) steers optimization toward sharper regions and (ii) destabilizes late training. We use a controlled protocol across six model–dataset pairs. Starting from a shared 20-epoch SGD

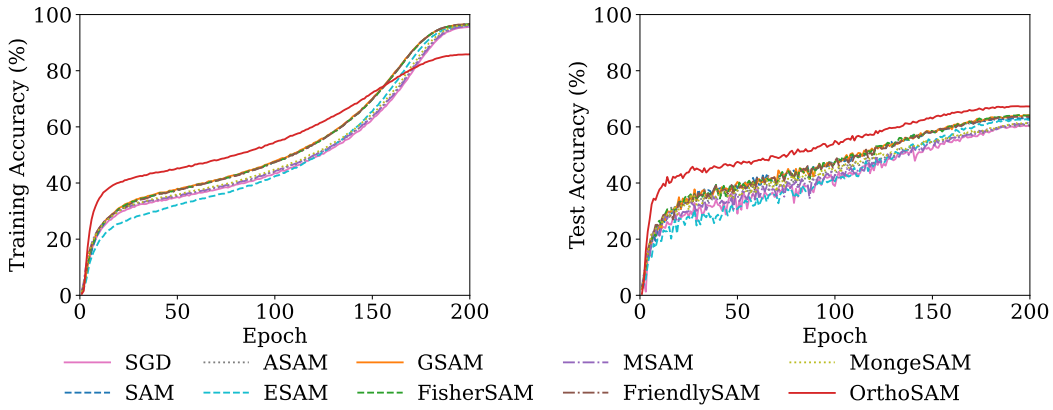

Figure 1: Training and test accuracy curves on Tiny-ImageNet with ResNet-50.

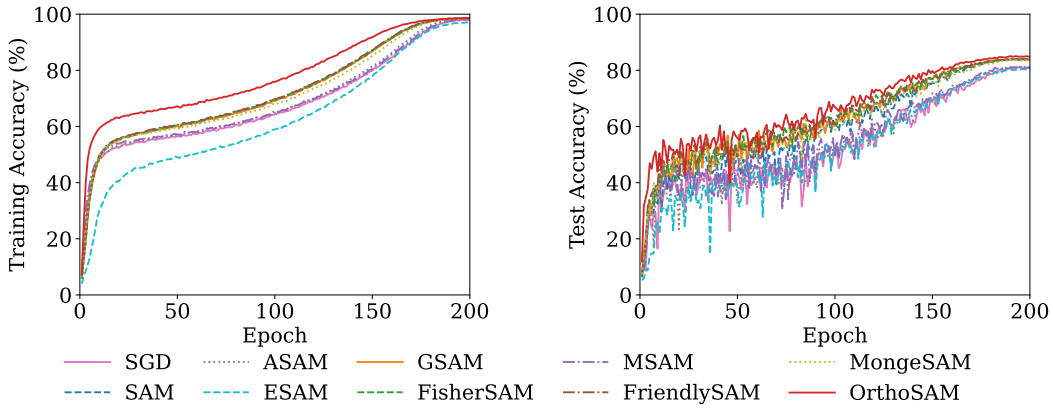

Figure 2: Training and test accuracy curves on CIFAR-100 with PyramidNet-110.

checkpoint, we continue training one copy with SAM and one with OrthoSAM for 80 additional epochs under identical settings, so that any difference is attributable to the update rule alone.

Property 1: Spectral Concentration and Sharper Minima. We first verify the mechanism of Eq. (4) by measuring the fraction of the ascent perturbation energy that lies in the top-3 singular subspace of the gradient, $\text{frac} = \|U_3 U_3^\top \varepsilon V_3 V_3^\top\|_F^2 / \|\varepsilon\|_F^2$, averaged over all logged epochs. We then measure adversarial sharpness $S_{\text{rel}} = (\mathcal{L}(\tilde{\theta} + \varepsilon_{\text{SAM}}) - \mathcal{L}(\tilde{\theta}))/\mathcal{L}(\tilde{\theta})$ at each fork's final checkpoint, using the *same* SAM-ascent probe for both methods so that neither is graded on its own rule; lower S_{rel} indicates a flatter region. Table 9 shows that SAM concentrates roughly 4–10$\times$ more energy on the top-3 directions in every setting, confirming Eq. (4) in practice, and that OrthoSAM consistently reaches a flatter region under the shared probe.

Property 2: Late-Training Stability. We assess stability by the standard deviation of test accuracy and training loss over the last 50 epochs, where the learning rate is already small so the remaining oscillations mainly reflect the optimizer. We also apply both ascent rules to the same held-out mini-batches from the same θ_0 and measure the post-ascent gradient alignment $\cos(\nabla\mathcal{L}(\theta + \varepsilon), \nabla\mathcal{L}(\theta))$ and relative shift $\|\nabla\mathcal{L}(\theta + \varepsilon) - \nabla\mathcal{L}(\theta)\|/\|\nabla\mathcal{L}(\theta)\|$. As reported in Table 10, OrthoSAM shows lower late-training variance of both test accuracy and training loss in every setting, preserves the descent direction markedly better after the ascent step, and induces a smaller gradient shift—at no accuracy cost.

Mechanistic View. The two perturbations admit an exact comparison in the same orthonormal basis. Writing the gradient SVD as $G = \sum_i \sigma_i A_i$ with $A_i = u_i v_i^\top$, SAM's perturbation is

Table 9: Property 1. Fraction of ascent energy in the gradient's top-3 singular subspace and adversarial sharpness S_{rel} under a shared SAM probe (lower is better on both). Lowest in each SAM–OrthoSAM pair in **bold**.

	ε top-3 SVD energy ↓		sharpness S_{rel} ↓	
Setting	SAM	OrthoSAM	SAM	OrthoSAM
CIFAR-10 / WRN-16-4	0.641	**0.142**	1.325	**0.636**
CIFAR-100 / WRN-16-4	0.657	**0.152**	0.476	**0.184**
CIFAR-10 / ResNet-50	0.729	**0.084**	12.344	**4.227**
CIFAR-100 / ResNet-50	0.640	**0.077**	3.637	**0.860**
CIFAR-10 / DenseNet-121	0.559	**0.082**	2.121	**0.244**
Tiny-ImageNet / ResNet-50	0.611	**0.058**	0.926	**0.196**

Table 10: Property 2. Late-training stability over the last 50 epochs: mean test accuracy, standard deviation of test accuracy, standard deviation of training loss, post-ascent gradient alignment, and relative gradient shift. Better value in each SAM–OrthoSAM pair in **bold** (higher mean accuracy / cosine; lower std. / gradient shift).

Setting	Method	Mean test accuracy ↑	Std. of test accuracy ↓	Std. of training loss ↓	Post-ascent cosine ↑	Relative gradient shift ↓
CIFAR-10 / WRN-16-4	SAM	90.03	5.57	0.0857	0.719	3.712
	OrthoSAM	**91.82**	**3.71**	**0.0726**	**0.955**	**0.601**
CIFAR-100 / WRN-16-4	SAM	66.67	9.78	0.2929	0.820	1.633
	OrthoSAM	**70.57**	**6.73**	**0.2504**	**0.982**	**0.289**
CIFAR-10 / ResNet-50	SAM	93.74	2.34	0.0558	0.596	4.652
	OrthoSAM	**95.67**	**1.31**	**0.0333**	**0.914**	**0.634**
CIFAR-100 / ResNet-50	SAM	76.78	4.98	0.2110	0.655	3.132
	OrthoSAM	**80.09**	**2.82**	**0.1331**	**0.956**	**0.458**
CIFAR-10 / DenseNet-121	SAM	94.68	0.41	0.0117	0.664	3.652
	OrthoSAM	**94.97**	**0.40**	**0.0112**	**0.894**	**0.771**
Tiny-ImageNet / ResNet-50	SAM	56.66	6.26	0.4255	0.647	1.766
	OrthoSAM	**57.43**	**4.02**	**0.2061**	**0.963**	**0.310**

$\Delta\theta_{\mathrm{SAM}} = \frac{\rho}{\|G\|_F}\sum_i \sigma_i A_i$, weighting each direction by its singular value, whereas the orthogonalized perturbation follows the identity $UV^\top = \sum_i u_i v_i^\top = \sum_i A_i$, i.e., $\Delta\theta_{\mathrm{Ortho}}/\rho' = \sum_i 1 \cdot A_i$, assigning every direction the same coefficient. OrthoSAM thus spends the perturbation budget uniformly across directions rather than concentrating it in the few with the largest σ_i. This directly explains the energy fractions in Table 9; the flatness and stability gains reflect the full update, whose descent step applies the same orthogonalization.

C.3 Orthogonal-Neighborhood (Nuclear-Norm) and Frobenius Sharpness: An Exact Comparison

For a weight matrix with gradient G and radius ρ, SAM measures sharpness over the full ball $\mathcal{B}_\rho$, whereas OrthoSAM measures it over the structured subset $\mathcal{N}_\rho = \mathcal{B}_\rho \cap \mathbb{R}^{m\times n}_{\perp}$, whose elements weight every singular direction equally. To first order, the two sharpness values are:

$$S_F = \max_{\|\Delta\|_F \le \rho} \langle G, \Delta\rangle = \rho\,\|G\|_F, \qquad S_* = \max_{\Delta\in\mathcal{N}_\rho} \langle G, \Delta\rangle = \rho'\,\|G\|_* = \tfrac{\rho}{\sqrt{n}}\,\|G\|_*, \tag{53}$$

where the second identity uses Theorem 4.2 together with $\langle G, UV^\top\rangle = \|G\|_*$; for full-rank G, the maximizer $\rho' UV^\top$ is exactly the perturbation realized by Algorithm 2. Their ratio is closed-form and depends only on the gradient spectrum:

$$\frac{S_*}{S_F} = \frac{\|G\|_*}{\sqrt{n}\,\|G\|_F} = \sqrt{\frac{\mathrm{erank}(G)}{n}} =: r(G), \qquad \mathrm{erank}(G) := \frac{\|G\|_*^2}{\|G\|_F^2}, \tag{54}$$

with $r(G) \in [n^{-1/2}, 1]$: the two objectives coincide ($r{=}1$) only when the spectrum is flat, and diverge most ($r{=}n^{-1/2}$) when G is rank-one. OrthoSAM therefore does not minimize a rescaled Euclidean

Table 11: Median effective-rank ratio $r(G) = \|G\|_*/(\sqrt{n}\,\|G\|_F)$ per 2-D layer at the final checkpoint. Lower $r(G)$ means a more skewed spectrum and a larger gap between the orthogonal-neighborhood (nuclear-norm) and Frobenius objectives.

	CIFAR-10 WRN-16-4	CIFAR-10 DenseNet-121	CIFAR-10 ResNet-50	Tiny-ImageNet ResNet-50	CIFAR-100 Tiny-ViT
$r(G)$ (median)	0.741	0.641	0.577	0.544	0.336

Table 12: Sensitivity to the number of Newton–Schulz iterations K on CIFAR-100. Residual $r = \|O^\top O - I\|_F/\sqrt{\min(m,n)}$, averaged over 2-D hidden layers. Highest accuracy and lowest residual in **bold** (time not ranked).

Backbone	Metric	K=1	K=2	K=3	K=5
ResNet-50	Accuracy (%)	82.40	83.32	83.70	**84.48**
	Time (minutes/epoch)	1.26	1.44	1.52	1.55
	Residual r	0.96	0.85	0.64	**0.44**
WRN-16-4	Accuracy (%)	78.61	78.70	79.04	**79.47**
	Time (minutes/epoch)	0.36	0.43	0.44	0.45
	Residual r	0.91	0.74	0.49	**0.39**

sharpness; it minimizes a genuinely different, nuclear-norm objective that departs from the Frobenius one exactly in the skewed-spectrum regime that our motivation identifies (Section 3).

We measure the effective-rank ratio $r(G)$ per 2-D layer at the final checkpoints (Table 11). The median lies well below one on every backbone and is lowest on the ViT, whose projection and attention gradients are the most spectrally concentrated. The layer-wise distribution is broad: the central 80% of layers spans, for example, $[0.57, 0.84]$ on CIFAR-10/WRN-16-4, $[0.46, 0.84]$ on CIFAR-10/DenseNet-121, and $[0.19, 0.70]$ on CIFAR-10/ResNet-50. Consequently, the orthogonal-neighborhood (nuclear-norm) and Frobenius sharpness objectives differ by roughly 1.3–$3.0\times$ at the median and by more than $5\times$ in the most skewed tenth of ResNet-50 layers.

C.4 Sensitivity to Newton–Schulz Iterations

We study the effect of the number of Newton–Schulz iterations K on CIFAR-100, reporting top-1 accuracy, wall-clock time (minutes per epoch), and the mean orthogonality residual $r = \|O^\top O - I\|_F/\sqrt{\min(m,n)}$ averaged over all 2-D hidden layers (this is exactly the residual logged by the implementation). Table 12 shows that the residual drops sharply from K=1 to K=3 and continues to shrink at K=5, with accuracy improving alongside it. The residual does not vanish at K=5 by design: the tuned coefficients of Algorithm 1 trade exact convergence for speed, pushing the singular values toward a band around one rather than exactly onto it (Jordan et al., 2024). Because the iteration is an odd matrix polynomial, its output keeps the singular vectors of the gradient exactly and only approximately equalizes the singular values; the convergence analysis (Section 4.2) idealizes this output as the exact factor $UV^\top$. Since the projection is applied only to 2-D hidden weights, the added wall-clock cost from K=1 to K=5 is modest (about 25%), so K=5 offers a good trade-off between approximation quality and cost.

C.5 Sensitivity to the Perturbation Radius

Effective Radius. OrthoSAM normalizes the orthogonal factor before scaling it, so the perturbation applied to a matrix layer is $\rho\,\mathrm{NS}(G)/\|\mathrm{NS}(G)\|_F$ and has Frobenius norm exactly ρ, independent of the gradient magnitude and of the layer shape. For a full-rank gradient this is precisely the closed-form solution $\rho\, n^{-1/2} UV^\top$ of Theorem 4.2, so the ρ reported here is the same quantity that appears in Eq. (5) and in the convergence analysis. Because the budget is allocated per matrix, every layer is perturbed at the same scale; SAM's perturbation $\rho\, G/\|G\|_F$ instead shares a single budget across the whole model, so across L matrix layers the aggregate OrthoSAM perturbation has norm $\rho\sqrt{L}$, while the non-matrix branch keeps the global SAM normalization. Nominal radii are therefore

Table 13: Radius sweep on CIFAR-10/Tiny-ViT (top-1 accuracy, %). ASAM uses its recommended larger range. Best per method in **bold**.

Method	ρ	Accuracy (%)
SAM	0.01	75.52
SAM	0.05	**77.61**
SAM	0.1	54.42
ASAM	0.01	78.22
ASAM	0.5	**79.61**
ASAM	1.0	78.29
ASAM	2.0	75.04
OrthoSAM	0.005	**87.36**
OrthoSAM	0.01	86.70
OrthoSAM	0.05	72.97

Table 14: Fine ρ sweep (top-1 accuracy, %). Top: SAM vs. OrthoSAM on CIFAR-10/Tiny-ViT. Bottom: orthogonal perturbation only on CIFAR-100/Tiny-ViT. Best per method in **bold**.

CIFAR-10 / Tiny-ViT						
ρ	0.002	0.005	0.01	0.02	0.05	0.1
SAM	–	76.45	75.52	72.73	**77.61**	54.42
OrthoSAM	86.95	**87.36**	86.70	86.19	72.97	59.04

CIFAR-100 / Tiny-ViT, orthogonal perturbation only					
ρ	0.0002	0.0003	0.0005	0.0007	0.001
Accuracy (%)	**52.46**	51.88	50.41	48.59	45.68

not directly comparable across methods; we tune ρ per method and compare each method at its best radius. At these radii, OrthoSAM's advantage does not stem from a larger perturbation: on CIFAR-10/Tiny-ViT (L=48 matrix layers), its best radius ρ=0.005 gives an aggregate norm of about 0.035, below SAM's best global radius of 0.05.

Radius Sweeps. Table 13 sweeps ρ for SAM, ASAM, and OrthoSAM on CIFAR-10/Tiny-ViT, extending the range for ASAM to the larger values recommended in its original paper. Two points emerge. First, OrthoSAM's advantage is not an artifact of a shared grid: it remains large even after evaluating ASAM over its own favorable range. Second, extending the OrthoSAM sweep to smaller radii (Table 14) shows a wide interior plateau: across $\rho \in [0.002, 0.02]$ accuracy varies by only 1.17 points and every point exceeds SAM's best by at least 8.6 points, so OrthoSAM is the less radius-sensitive method over this range.

The orthogonal-perturbation-only sweep confirms the reading of the ablation in Section 5.4: the collapse at ρ=0.01/0.05 is a radius artifact, not a property of the geometry. Accuracy rises monotonically as ρ decreases and, at the smallest radii ($\rho \leq 0.0002$; Tables 14 and 15), returns to the SGD baseline level (52.67%) rather than improving on it, while the 45.68 and 16.31 readings sit at 10× and 100× the smallest radius tested.

Full Ablation Radius Sweep. Table 15 extends the component ablation of Table 7 to a common radius grid on both benchmarks; SGD and the descent-only Muon-style variant carry no perturbation and are ρ-independent (52.67/60.53 and 62.09/59.23 on Tiny-ViT/ResNet-50). Three observations reinforce the main-text analysis. First, OrthoSAM improves on SAM at *every* matched radius on both benchmarks, and its best entry (64.17 on Tiny-ViT, 67.55 on ResNet-50) is the largest in each panel, so the gain does not depend on a particular grid. Second, the orthogonal-perturbation-only branch collapses only at the large radii tuned for Euclidean perturbations (to 4.64 and 5.47 at ρ=0.05) and then recovers monotonically as ρ decreases, reaching SAM's accuracy at the same small radius

Table 15: Radius sweep of the ablation variants in Table 7 (top-1 accuracy, %). Column headers are perturbation radii ρ. SGD and Muon-style descent (no perturbation) do not depend on ρ and are omitted; their accuracies are 52.67% and 62.09% on CIFAR-100/Tiny-ViT, and 60.53% and 59.23% on Tiny-ImageNet/ResNet-50, respectively. Best per dataset–backbone and ρ in **bold**.

	CIFAR-100 / Tiny-ViT				Tiny-ImageNet / ResNet-50			
Method	0.0001	0.001	0.01	0.05	0.0001	0.001	0.01	0.05
SAM	52.72	52.46	51.35	39.33	60.05	60.52	62.00	63.92
Orthogonal perturbation only	52.55	45.68	16.31	4.64	60.91	60.75	32.30	5.47
SAM + Muon-style	**63.22**	62.96	62.23	**63.22**	58.65	58.97	59.71	59.96
SAM + orthogonalized descent	62.56	62.65	61.53	61.63	**61.79**	61.61	62.91	63.18
OrthoSAM	62.94	**63.43**	**64.17**	59.88	61.40	**62.57**	**64.87**	**67.55**

Table 16: Per-epoch cost and equal-wall-clock comparison on CIFAR-100/Tiny-ViT under a 50-minute budget on a single A100 (80 GB). Each method uses its own tuned ρ. Accuracy is the test accuracy of the validation-selected checkpoint, as in the main tables. Highest accuracy in **bold**.

Method	Time (minutes/epoch)	Epochs completed	Accuracy (%)
OrthoSAM	0.445	112	**65.28**
Muon	0.234	200	63.07
Muon+SAM	0.336	148	62.70
MSAM	0.165	200	53.54
ASAM	0.283	176	52.75
MongeSAM	0.278	179	52.56
FriendlySAM	0.283	176	51.29
SAM	0.278	179	51.23
FisherSAM	0.304	164	50.87
GSAM	0.319	156	50.31
ESAM	0.262	190	43.12

(52.55 vs. 52.72; 60.91 vs. 60.05), so the branch is simply past its optimum at the larger radii. Third, full OrthoSAM does not collapse at the same radii (67.55 vs. 5.47 at ρ=0.05 on ResNet-50, with an identical perturbation): the orthogonalized descent absorbs the step that destabilizes the perturbation alone. On its own the orthogonal perturbation therefore produces at most a marginal change over the baseline; its large benefit is realized only in combination with the matching descent.

C.6 Training Time and Equal-Wall-Clock Comparison

Because OrthoSAM has a higher per-epoch cost, an equal-epoch comparison could overstate its practical value. Table 16 therefore reports the per-epoch cost, the number of epochs completed, and the resulting accuracy together, so that both the raw training-time overhead and its effect under a matched budget are visible in one place. We compare all methods under a shared wall-clock budget of T=50 minutes on a single A100 (80 GB) for CIFAR-100/Tiny-ViT, allocating each method $\min(200, \lfloor T/t_{\mathrm{ep}} \rfloor)$ epochs from its measured per-epoch time t_{ep} and reporting the test accuracy of the checkpoint with the highest validation accuracy within the budget, as in the main tables. The per-epoch column shows that OrthoSAM is the most expensive method per epoch, roughly $1.6\times$ the cost of SAM, as expected from its two Newton–Schulz projections. Even so, OrthoSAM completes only 112 epochs yet still attains the highest accuracy (65.28%). Muon (63.07%) and Muon + SAM (62.70%) both finish the full 200 epochs within the budget but remain below OrthoSAM, and all Euclidean SAM variants trail further behind. OrthoSAM's advantage therefore holds under matched training time, not only under matched epochs.

C.7 Hessian Spectrum of the Learned Minima

We analyze local curvature through the leading Hessian eigenvalues on CIFAR-10, following the protocol of prior sharpness-aware analyses (Foret et al., 2021; Li & Giannakis, 2023; Becker et al.,

Table 17: Leading Hessian eigenvalues on CIFAR-10 (mean ± std over five seeds). Lower is better; lowest mean in each column within each architecture in **bold**.

Architecture	Method	λ_1	λ_2	λ_3	λ_4	λ_5
DenseNet-121	SGD	11.90 ±0.03	5.49 ±0.05	5.12 ±0.07	4.93 ±0.05	4.75 ±0.12
	SAM	**8.39** ±0.07	4.07 ±0.38	3.64 ±0.16	3.40 ±0.04	3.32 ±0.06
	OrthoSAM	8.59 ±0.42	**1.85** ±0.28	**1.49** ±0.22	**1.30** ±0.18	**1.10** ±0.15
WRN-16-4	SGD	80.30 ±2.17	55.64 ±0.64	45.43 ±1.52	41.02 ±1.07	38.10 ±1.43
	SAM	42.31 ±1.19	35.03 ±0.71	29.69 ±1.45	25.87 ±1.09	24.30 ±1.33
	OrthoSAM	**17.03** ±0.07	**8.72** ±0.12	**6.32** ±0.13	**5.85** ±0.13	**5.08** ±0.08

Table 18: Leading Hessian eigenvalues of the ablation variants under a shared Lanczos protocol. Lower is better; lowest in each column and setting in **bold**.

Setting	Method	λ_1	λ_2	λ_3	λ_4	λ_5
Tiny-ImageNet/ResNet-50	SAM	88.06	49.53	49.38	42.63	38.43
	Orthogonal perturbation only	488.68	262.71	64.65	42.68	18.95
	SAM+orthogonalized descent	**34.14**	**22.30**	**20.27**	**17.27**	**15.30**
	OrthoSAM	42.85	34.50	28.34	26.57	26.14
CIFAR-100/Tiny-ViT	SAM	430.38	391.25	368.76	338.79	320.06
	Orthogonal perturbation only	48.37	33.48	32.17	26.53	23.01
	SAM+orthogonalized descent	17.17	**3.45**	**1.30**	**0.88**	**0.81**
	OrthoSAM	**9.36**	7.59	6.86	5.63	4.69

2025) (Lanczos with full re-orthogonalization; see Appendix B.1). Smaller leading eigenvalues indicate flatter solutions. To assess robustness to seed and mini-batch selection, we repeat the computation over *five* random seeds and report mean ± standard deviation in Table 17. The results are stable across seeds. On WRN-16-4, OrthoSAM consistently lowers all five leading eigenvalues (about $2.5\times$ below SAM on λ_1) at the lowest variance of the three methods. On DenseNet-121, λ_1 is statistically tied with SAM (8.59 ± 0.42 vs. 8.39 ± 0.07), while λ_2–λ_5 are substantially smaller, so the slight reversal on λ_1 falls within estimation noise.

Curvature Across the Ablation Grid. To connect the flatness effect to the two components of Section 5.4, we also measure the leading eigenvalues for each ablation cell under a single Lanczos protocol (Table 18). Both variants with the orthogonalized descent reach much flatter minima than SAM, lowering λ_1 by a factor of two or more on Tiny-ImageNet/ResNet-50 and by more than an order of magnitude on CIFAR-100/Tiny-ViT, where OrthoSAM attains the lowest λ_1 of all variants. As in the accuracy ablation, the orthogonalized descent drives most of the curvature reduction, and the further contribution of the orthogonal perturbation depends on the backbone. The orthogonal-perturbation-only rows are measured at the shared nominal radius, where this branch is far past its optimal radius and underfits (Appendix C.5); their eigenvalues reflect a poorly trained model rather than the perturbation geometry, so we do not compare them.

C.8 Extended NLP: AdamW Baselines

Since BERT fine-tuning is most often performed with adaptive optimizers, we complement the SGD-based GLUE comparison of Section 5.3 with an AdamW-based trio that holds the descent rule fixed as AdamW and changes only the perturbation geometry: plain AdamW, AdamW + SAM (Euclidean perturbation), and AdamW + OrthoSAM (orthogonal perturbation). Table 19 reports the results. Under a fixed AdamW descent rule, AdamW + OrthoSAM achieves the best score on seven of the nine tasks, with the largest gains on the fine-tuning-unstable tasks (CoLA $+1.27$ and WNLI $+7.04$ over AdamW + SAM). The same controlled comparison on CIFAR-100/Tiny-ViT gives AdamW 64.15, AdamW + SAM 64.20, and AdamW + OrthoSAM 65.38: the Euclidean perturbation barely moves AdamW ($+0.05$), while the orthogonal perturbation adds $+1.23$. This isolates the

Table 19: BERT-base on GLUE with a shared AdamW descent; only the perturbation geometry varies. Best per reported score in **bold**.

Method	**CoLA**	**SST-2**	**MRPC**	**QQP**	**MNLI**	**QNLI**	**RTE**	**WNLI**	**STS-B**
	Mcc.	*Acc.*	*Acc./F1*	*Acc./F1*	*Mean Acc.*	*Acc.*	*Acc.*	*Acc.*	*Pear./Spea.*
AdamW	54.96	92.20	83.09/88.00	90.54/87.33	84.22/84.33	90.92	**65.34**	38.03	88.05/87.78
AdamW+SAM	58.30	92.20	83.33/88.28	90.61/87.43	**84.45**/84.36	91.10	62.45	49.30	88.03/87.79
AdamW+OrthoSAM	**59.57**	**92.43**	**84.31/88.81**	**90.68/87.52**	84.31/**84.40**	**91.36**	64.62	**56.34**	**88.17/87.89**

Table 20: NanoGPT speedrun on FineWeb: validation cross-entropy (lower is better). The last column is the change relative to AdamW. Lowest cross-entropy in **bold**.

Method	Validation cross-entropy ↓	Change vs. AdamW
Muon	**3.851**	-0.643
AdamW+OrthoSAM (ρ=0.05)	3.947	-0.547
AdamW+SAM (ρ=0.05)	4.019	-0.475
AdamW	4.494	—

perturbation geometry as the source of the improvement, since only the ascent step differs across the trio.

C.9 Autoregressive Language Modeling

Because OrthoSAM is inspired by Muon-style optimization, we additionally evaluate it in an autoregressive language-modeling setting: the NanoGPT speedrun benchmark introduced with Muon (Jordan et al., 2024), which trains a GPT-2-style transformer on the FineWeb corpus under a fixed architecture and token budget and reports validation cross-entropy. Following the Muon setup, Muon (and the orthogonal branch of AdamW + OrthoSAM) is applied only to the 2-D hidden transformer weights, while AdamW handles the embeddings, output head, and scalar parameters. Table 20 reports the results. Muon attains the lowest validation cross-entropy, which is expected since it directly modifies the descent update rather than the perturbation. Nevertheless, under the *same* AdamW descent rule, AdamW + OrthoSAM achieves lower validation cross-entropy than both AdamW + SAM and plain AdamW. We note that this is a next-token prediction task on a fixed pre-training corpus, so validation cross-entropy measures fit to that distribution rather than held-out generalization in the sense of our vision and BERT experiments. We therefore treat this experiment as a supplementary result—evidence that the orthogonal perturbation remains beneficial atop a strong adaptive optimizer in the language-modeling regime—rather than as a primary generalization claim of the paper.

C.10 ImageNet-1k

To verify that the gains persist at large scale, we train ResNet-50 on ImageNet-1k (1.28M images, 1000 classes) (Deng et al., 2009). Even within a reduced 30-epoch budget, OrthoSAM improves over both SGD and SAM under the same recipe (Table 21), reaching 63.14% top-1 accuracy. This indicates that the benefits of OrthoSAM extend to large-scale, high-resolution classification.

D Algorithm

Algorithm 2 summarizes the full OrthoSAM procedure. Each iteration consists of two forward–backward passes. In the first pass, the gradient is orthogonalized via the Newton–Schulz iteration to form the adversarial perturbation; in the second pass, the gradient at the perturbed point is again orthogonalized before the descent step (Eq. (12)). Scalar parameters, biases, and classifier heads fall back to the standard SAM perturbation and plain gradient descent. Because the Newton–Schulz iteration normalizes its input to unit Frobenius norm before iterating (Algorithm 1), the ascent scale is set entirely by ρ and is independent of the gradient magnitude. The semi-orthogonal factor O_W is rescaled to unit Frobenius norm before it is applied, so the realized per-matrix perturbation radius

Table 21: Top-1 accuracy (%) on ImageNet-1k with ResNet-50 under a 30-epoch budget. Highest in **bold**.

Backbone	SGD	SAM	OrthoSAM
ResNet-50	57.83	61.45	**63.14**

Algorithm 1 Newton–Schulz Orthogonalization

Require: $M \in \mathbb{R}^{m \times n}$, iterations K
Ensure: $O \approx UV^\top$ where $M = U\Sigma V^\top$
$(a, b, c) \leftarrow (3.4445,\ -4.7750,\ 2.0315)$
$X \leftarrow M^\top$ if $m > n$, else $X \leftarrow M$ ▷ work with shorter dimension
$X \leftarrow X \,/\, \|X\|_F$
for $k = 1, \ldots, K$ **do**
    $A \leftarrow X X^\top$
    $X \leftarrow a\, X + (b\, A + c\, A^2)\, X$
end for
return $X^\top$ if transposed, else X

is exactly ρ. When G_W has full rank, $\|O_W\|_F = \sqrt{\min(m, n)}$ and the perturbation coincides with the closed-form solution $\rho\, n^{-1/2} UV^\top$ of Eq. (9); the explicit rescaling keeps $\|\widetilde{W} - W\|_F = \rho$ even when G_W is rank-deficient or K is small, so the radius used in the analysis is realized exactly rather than approximately. How this radius relates to the baselines' is discussed in Appendix C.5. The descent step applies $\mathrm{NS}(\widetilde{G}_W)$ directly (each singular value set to one), without the additional $\sqrt{m/n}$ scaling used by canonical Muon; matrix-shaped weights that are embeddings or the output head are excluded from both Newton–Schulz branches.

E Limitations and Future Work

OrthoSAM inherits the two-forward-pass cost of SAM and adds two Newton–Schulz projections per iteration—one for the perturbation and one for the descent update. Although each projection requires only K matrix multiplications (typically $K = 5$) and is substantially cheaper than an explicit SVD, the overhead is non-negligible for very large weight matrices, and practical wall-clock savings relative to exact orthogonalization have not been characterized at scale. More fundamentally, the current formulation applies the Stiefel constraint only to two-dimensional weight matrices; one-dimensional parameters (biases, layer-norm gains) and classifier heads default to the standard SAM branch. The analysis of Section 4 therefore covers the matrix branch of a single layer: it constrains $\|\Delta\|_F \le \rho$ per matrix, whereas the fallback branch keeps SAM's global normalization, so the two branches share a symbol but not a geometry. A unified treatment of the mixed neighborhood induced by the full parameter vector is left to future work. Extending the orthogonality constraint to higher-order tensors—such as the four-dimensional kernels in convolutional layers, which are currently reshaped into matrices—would require a more principled treatment of multi-mode structure (e.g., via tensor decompositions on the Grassmann manifold). Finally, the orthogonal perturbation has been combined with an adaptive optimizer only at small scale: AdamW on BERT-base fine-tuning, the NanoGPT speedrun, and CIFAR-100/Tiny-ViT (Appendices C.8 and C.9). Its interplay with adaptive optimizers in large-scale pre-training, where the gradient spectrum evolves considerably over training, is a direction for future work.

Algorithm 2 OrthoSAM

Require: Training set S, learning rate η, perturbation radius ρ, NS iterations K

Initialize parameters θ_0
for $t = 0, 1, \ldots, T-1$ **do**
  Sample mini-batch $\mathcal{B}_t \subset S$
  $G \leftarrow \nabla L_{\mathcal{B}_t}(\theta_t)$ ▷ first forward–backward
  for each layer weight W with gradient G_W **do**
    if W is matrix-shaped hidden weight **then**
      $O_W \leftarrow \mathrm{NS}(G_W,\, K)$ ▷ semi-orthogonal factor, $O_W \approx UV^\top$
      $\widetilde{W} \leftarrow W + \rho \cdot O_W / \|O_W\|_F$ ▷ orthogonal perturbation, $\|\widetilde{W} - W\|_F = \rho$
    else
      $\widetilde{W} \leftarrow W + \rho \cdot G_W \,/\, \|G\|_F$ ▷ standard SAM perturbation ($\|G\|_F$ is the global gradient norm)
    end if
  end for
  $\widetilde{G} \leftarrow \nabla L_{\mathcal{B}_t}(\widetilde{\theta})$ ▷ second forward–backward
  for each layer weight W with gradient $\widetilde{G}_W$ **do**
    if W is matrix-shaped hidden weight **then**
      $W \leftarrow W - \eta \cdot \mathrm{NS}(\widetilde{G}_W,\, K)$
    else
      $W \leftarrow W - \eta \cdot \widetilde{G}_W$ ▷ standard SGD step
    end if
  end for
end for