SHARPNESS-AWARE MINIMIZATION AND MUON: ROBUSTNESS UNDER THE SPECTRAL NORM

Anonymous authors
Paper under double-blind review

ABSTRACT

Sharpness-Aware Minimization (SAM) aims to improve generalization by encouraging insensitivity to small, worst-case parameter perturbations. However, the notion of a "small" perturbation is inherently geometry-dependent: while existing SAM variants have explored a wide range of choices, a clear perspective on which geometries are most effective in practice remains elusive. Recent work on matrix-aware optimization, particularly the Muon optimizer, suggests that respecting the matrix structure of hidden-layer weights can lead to strong empirical performance. Motivated by this, we study matrix-aware geometry in both stages of SAM: we introduce a layerwise spectral inner perturbation for matrix-valued hidden-layer parameters and combine it with either AdamW/SGDW or Muon in the outer update. Across ImageNet-1K experiments on ViT-Small/16 and ResNet-50, we find that the combination of a spectral inner step with a Muon outer step performs consistently strongly, achieving the best validation accuracy on both models among the evaluated methods.

1 INTRODUCTION

The flat minimum hypothesis proposes that neural network parameter settings in flat regions of the loss landscape yield better generalizing networks (Hochreiter & Schmidhuber, 1997). Inspired by this, Sharpness-Aware Minimization (SAM; Foret et al., 2020) is an optimization algorithm which encourages solutions in flat basins by minimizing the loss subject to a perturbation of the weights, therefore seeking neighbourhoods that have uniformly low loss. Concretely, the SAM algorithm uses a two-step procedure: we first perturb the weights in the uphill direction of the gradient (the *inner step*), and then compute the descent direction from that perturbed point (the *outer step*), applying the actual step to the original unperturbed parameters. This original formulation of SAM defines sharpness in terms of Euclidean geometry, i.e., the inner perturbation step is taken within a fixed Euclidean norm radius, while the outer descent step traditionally uses SGD, whose update direction is the negative Euclidean gradient.

Recently, a separate line of work has proposed that steepest descent under a spectral norm step-size penalty could be more appropriate for neural network optimization, since it takes into account the linear operator structure of weight matrices, instead of just treating them as flattened vectors of parameters (Bernstein & Newhouse, 2025). This has led to a rise in popularity of *matrix-aware* optimizers, in particular Muon (Jordan et al., 2024), which is now an integral component of many frontier large language model training setups (Kimi Team, 2025; GLM-4.5 Team, 2025; Singh et al., 2026; DeepSeek-AI, 2026). While various works have extended SAM to incorporate non-Euclidean geometry (such as Fisher, Riemannian, and adaptive geometries, Kim et al., 2022; Yun & Yang, 2023; Kwon et al., 2021), to our knowledge, no work has studied the effects of spectral geometry in either the inner or outer step of the SAM setting.

In this work, we investigate the use of spectral geometry in the inner and/or outer step. In particular, in the outer step we consider a Muon update, while in the inner step we consider an orthogonalized gradient perturbation[1]. We test these modifications on two ImageNet training tasks (a vision transformer and a ResNet) and find the following:

[1]This is similar to a Muon ascent step, but without momentum.

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.

- Inner and outer geometry interact. Spectral sharpness helps most when the outer update is also spectral and least when it is Euclidean, whereas Euclidean sharpness is comparatively insensitive to the outer optimizer.
- Spectral perturbations with Muon achieve the best validation accuracy on both ViT-S/16 and ResNet-50.

2 PRELIMINARIES

2.1 A NORM-AGNOSTIC TEMPLATE FOR SAM

The flat minimum hypothesis suggests we should judge a parameter choice not only by its training loss, but also by its training loss *under small perturbations*. Following Foret et al. (2020) we focus on robustness under worst-case perturbations: let $\Theta \cong \mathbb{R}^p$ denote the parameter space, $\|\cdot\|_{\text{in}}$ be a norm on Θ, and $L : \Theta \to \mathbb{R}_{\geq 0}$ be the training loss. For perturbation size $\rho \geq 0$ we define the *robust loss* associated with L as

$$L_\rho(\boldsymbol{\theta}) = \sup_{\|\boldsymbol{\varepsilon}\|_{\text{in}} \leq 1} L(\boldsymbol{\theta} + \rho\boldsymbol{\varepsilon}). \tag{1}$$

Our goal here is not to give an implementation-level description of SAM, involving batching etc., but rather to isolate a general first-order template common to SAM-type methods which allows their comparison. Since evaluating equation 1 is generally intractable, the approach taken in Foret et al. (2020) is to replace the inner maximization by its first-order linearization. In what follows, we formulate this linearization using the Fréchet differential, rather than immediately identifying it with a gradient vector. We do this in order to present a more general SAM framework in which the perturbation geometry can be determined by a possibly non-Euclidean norm $\|\cdot\|_{\text{in}}$[2]. Assume L is Fréchet differentiable at $\boldsymbol{\theta}$. We write $DL(\boldsymbol{\theta}) \in \Theta^*$ (where Θ^* is the dual space of Θ) for its differential, which recall in this case is the linear functional satisfying

$$L(\boldsymbol{\theta} + \boldsymbol{\varepsilon}) = L(\boldsymbol{\theta}) + DL(\boldsymbol{\theta})[\boldsymbol{\varepsilon}] + o(\|\boldsymbol{\varepsilon}\|_{\text{in}}) \qquad \text{as } \boldsymbol{\varepsilon} \to 0.$$

Thus, for small ρ,

$$L(\boldsymbol{\theta} + \rho\boldsymbol{\varepsilon}) = L(\boldsymbol{\theta}) + \rho\, DL(\boldsymbol{\theta})[\boldsymbol{\varepsilon}] + o(\rho), \qquad \|\boldsymbol{\varepsilon}\|_{\text{in}} \leq 1.$$

The first-order adversarial perturbation is any steepest-ascent direction for the linear functional $DL(\boldsymbol{\theta})$ over the $\|\cdot\|_{\text{in}}$-unit ball,

$$\boldsymbol{\varepsilon}^*(\boldsymbol{\theta}) \in \arg\max_{\|\boldsymbol{\varepsilon}\|_{\text{in}} \leq 1} DL(\boldsymbol{\theta})[\boldsymbol{\varepsilon}]. \tag{2}$$

Equivalently, using the canonical identification $\Theta \cong \Theta^{**}$, $\boldsymbol{\varepsilon}^*(\boldsymbol{\theta})$ is an element of the subdifferential of the dual norm, $\boldsymbol{\varepsilon}^*(\boldsymbol{\theta}) \in \partial\|DL(\boldsymbol{\theta})\|_{\text{in},*}$. From the definition of the dual norm it therefore follows that [3]

$$L_\rho(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + \rho\,\|DL(\boldsymbol{\theta})\|_{\text{in},*} + o(\rho). \tag{3}$$

This first-order calculation motivates the following SAM surrogate for the robust loss L_ρ, in which one simply evaluates the loss at the first-order adversarially perturbed point,

$$\boldsymbol{\theta} \mapsto L(\boldsymbol{\theta} + \rho\boldsymbol{\varepsilon}^*(\boldsymbol{\theta})). \tag{4}$$

In particular, the surrogate given in equation 4 agrees with L_ρ up to first order in ρ and is convenient to work with. If one is able to differentiate this surrogate exactly, then the dependence of $\boldsymbol{\varepsilon}^*(\boldsymbol{\theta})$ on $\boldsymbol{\theta}$ contributes an additional term: if L is differentiable at $\boldsymbol{\theta} + \rho\boldsymbol{\varepsilon}^*(\boldsymbol{\theta})$ and a differentiable selection of $\boldsymbol{\varepsilon}^*$ exists, then the chain rule gives

$$D\big(L(\boldsymbol{\theta} + \rho\boldsymbol{\varepsilon}^*(\boldsymbol{\theta}))\big)[\Delta] = DL(\boldsymbol{\theta} + \rho\boldsymbol{\varepsilon}^*(\boldsymbol{\theta}))\big[\Delta + \rho\, D\boldsymbol{\varepsilon}^*(\boldsymbol{\theta})[\Delta]\big].$$

However, in general, the maximizer map $\boldsymbol{\varepsilon}^*$ may be non-unique or non-differentiable, as is the case for non-smooth norms (such as the spectral norm we consider shortly), and even when a differentiable selection exists, the term $D\boldsymbol{\varepsilon}^*(\boldsymbol{\theta})$ involves second-order information about L. This is typically too expensive to compute in large-scale training. Standard SAM therefore uses a stop-gradient

[2] If $\Theta \cong \mathbb{R}^p$ is equipped with its standard Euclidean inner product, $DL(\boldsymbol{\theta})$ is typically represented by $\nabla L(\boldsymbol{\theta})$

[3] Note also by Fréchet differentiability the remainder is uniform over the $\|\cdot\|_{\text{in}}$-unit ball

Algorithm 1 Abstract SAM Template

Require: Initial parameters $\boldsymbol{\theta}_0$; inner norm $\|\cdot\|_{\text{in}}$; perturbation radii $(\rho_t)_{t\geq 0}$; step sizes $(\eta_t)_{t\geq 0}$; outer direction maps $(\mathcal{D}_t)_{t\geq 0}$.

1: **for** $t = 0, 1, \ldots, T-1$ **do**
2: Compute an inner steepest ascent direction: $\boldsymbol{\varepsilon}_t \in \partial\|DL(\boldsymbol{\theta}_t)\|_{\text{in},*}$
3: Form the perturbed first-order signal: $\phi_t := DL(\boldsymbol{\theta}_t + \rho_t\boldsymbol{\varepsilon}_t)$.
4: Convert the perturbed signal into an update direction: $\Delta_t = \mathcal{D}_t(\phi_t)$.
5: Update the parameters: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta_t\Delta_t$.
6: **end for**
7: **return** $\boldsymbol{\theta}_T$

approximation: it forms the perturbation $\boldsymbol{\varepsilon}^*(\boldsymbol{\theta})$, evaluates the loss at $\boldsymbol{\theta} + \rho\boldsymbol{\varepsilon}^*(\boldsymbol{\theta})$, and uses only the differential of L at that perturbed point. We denote this stopped-gradient first-order signal by

$$\phi_\rho(\boldsymbol{\theta}) := DL(\boldsymbol{\theta} + \rho\boldsymbol{\varepsilon}^*(\boldsymbol{\theta})) \in \Theta^*.$$

At this point, the SAM perturbation mechanism has produced a covector $\phi_\rho(\boldsymbol{\theta}) \in \Theta^*$ (i.e. a perturbed-loss gradient). An outer algorithm (i.e. an optimizer) must turn this into a parameter-space update direction. We represent this operation by a possibly time and state dependent direction map

$$\mathcal{D}_t : \Theta^* \to \Theta, \qquad \Delta_t = \mathcal{D}_t(\phi_\rho(\boldsymbol{\theta}_t)).$$

The abstraction of SAM given in Algorithm 1 separates two ways in which SAM-type methods can vary. First, they may differ in the geometry used to define the inner adversarial perturbation: classical SAM (Foret et al., 2020) uses the Euclidean norm, ASAM (Kwon et al., 2021) uses a parameter-dependent perturbation geometry which may be written as $\|\boldsymbol{\varepsilon}\|_{\text{in},\boldsymbol{\theta}} = \|T_{\boldsymbol{\theta}}^{-1}\boldsymbol{\varepsilon}\|_2$ with $T_{\boldsymbol{\theta}}$ diagonal, and Fisher / Functional SAM (Kim et al., 2022; Singh et al., 2025) use a norm defined through the model Jacobian so that perturbations are measured by their functional effect on the network output. Second, SAM-type methods may differ in how they convert the perturbed first-order signal ϕ_t into an update direction. For vanilla SGD, $\mathcal{D}_t$ is essentially the negative Euclidean gradient direction, up to scaling and mini-batch noise. Momentum, Adam (Kingma & Ba, 2015), AdamW (Loshchilov & Hutter, 2019), and related optimizers instead apply state-dependent transformations to the same perturbed signal.

2.2 Orthogonalized matrix updates and Muon

A central operation in what follows is the orthogonalization of a matrix-valued update. Let $\boldsymbol{M} \in \mathbb{R}^{m\times n}$ and write its reduced singular value decomposition as $\boldsymbol{M} = \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top$. We define its orthogonalization by

$$\text{Ortho}(\boldsymbol{M}) := \boldsymbol{U}_r\boldsymbol{V}_r^\top.$$

Thus $\text{Ortho}(\boldsymbol{M})$ preserves the left and right singular subspaces of $\boldsymbol{M}$ while replacing all non-zero singular values by 1. This operation admits a natural geometric interpretation: by the duality between the spectral and nuclear norms (Bhatia, 1997),

$$\text{Ortho}(\boldsymbol{M}) \in \operatorname*{arg\,max}_{\|\Delta\|_{2\to 2}\leq 1} \langle \boldsymbol{M}, \Delta\rangle, \tag{5}$$

where $\|\Delta\|_{2\to 2}$ denotes the spectral norm of Δ, i.e. its maximum singular value. Thus $\text{Ortho}(\boldsymbol{M})$ is a steepest-ascent direction for the linear functional $\Delta \mapsto \langle \boldsymbol{M}, \Delta\rangle$ under a spectral-norm constraint. An orthogonalized update changes the action of the layer by at most η in spectral norm:

$$\|(\boldsymbol{W} - \eta\,\text{Ortho}(\boldsymbol{M})) - \boldsymbol{W}\|_{2\to 2} = \eta\|\,\text{Ortho}(\boldsymbol{M})\|_{2\to 2} \leq \eta.$$

Equivalently, for every $\boldsymbol{x} \in \mathbb{R}^n$,

$$\big|\,\|(\boldsymbol{W} - \eta\,\text{Ortho}(\boldsymbol{M}))\boldsymbol{x}\| - \|\boldsymbol{W}\boldsymbol{x}\|\,\big| \leq \eta\|\boldsymbol{x}\|.$$

Orthogonalized updates therefore control the change in activations, which in turn is widely thought to improve the stability of optimization for matrix-valued hidden-layer parameters. Muon (Jordan et al., 2024), or *MomentUm Orthogonalized by Newton-Schulz*, is a practical optimizer built on this principle. Given a matrix-valued parameter $\boldsymbol{W}_t$ and gradient $\boldsymbol{G}_t = \nabla_{\boldsymbol{W}} L(\boldsymbol{W}_t)$, Muon first forms a

momentum matrix $\boldsymbol{M}_{t+1}$, and then updates $\boldsymbol{W}_t$ using an approximate orthogonalization of $\boldsymbol{M}_{t+1}$. In its idealized form the Muon update is

$$\boldsymbol{M}_{t+1} = \beta \boldsymbol{M}_t + (1-\beta)\boldsymbol{G}_t, \qquad \boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta_t \mathrm{Ortho}(\boldsymbol{M}_{t+1}).$$

In practice, computing a singular value decomposition at every iteration would be too expensive, so Muon approximates $\mathrm{Ortho}(\boldsymbol{M}_{t+1})$ using a small number of Newton–Schulz iterations. Later work has focused on making this approximation stable and efficient at scale, including choices of iteration coefficients, normalization, and mixed-precision behaviour (e.g. Amsel et al., 2026; Liu et al., 2025) but the core idea of orthogonalizing the momentum matrix remains the same.

3 Exploring the use of spectral layerwise geometry in SAM

Rather than proposing a single modification to SAM, in this work we study a variety of SAM optimizers obtained by varying the geometry used in the inner perturbation step, the outer update step, or both. In particular, motivated by the recent empirical success of orthogonalized optimization methods such as Muon, we investigate the effect of introducing *layerwise spectral geometry* into the SAM pipeline. Concretely, we compare inner perturbations generated either by the Euclidean norm on a flattened vector space, as in standard SAM, versus a layerwise spectral norm which we describe in Section 3.1. Concurrently, we also vary the outer step optimizer between AdamW, an elementwise optimizer commonly used in SAM's outer step (Chen et al., 2022), and Muon. Although Muon is a natural outer optimizer candidate, its effectiveness for minimizing SAM's robust-loss surrogate is not obvious. Finally, we emphasize that our aim is not to advocate a single algorithm, but rather to gain insight through experiments as to how these different geometric choices in the inner and outer step of SAM interact, and provide a practical reference for the effect of layerwise or matrix-aware optimization in SAM. We hope this will motivate and support further research in this direction.

3.1 Layerwise geometry

Instead of thinking of our parameters as a flattened vector $\Theta \cong \mathbb{R}^p$, we explicitly decompose the parameter space into layerwise blocks:

$$\Theta = \mathbb{R}^{m_1 \times m_2} \times \mathbb{R}^{m_2 \times m_3} \times \cdots \times \mathbb{R}^{m_{L-1} \times m_L} \times \mathbb{R}^q.$$

We write $\boldsymbol{\theta} = (\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(L-1)}, \boldsymbol{v}) \in \Theta$ where each $\boldsymbol{W}^{(\ell)} \in \mathbb{R}^{m_\ell \times m_{\ell+1}}$ is a matrix-valued parameter and $\boldsymbol{v} \in \mathbb{R}^q$ collects all remaining non-matrix parameters, such as biases and normalization parameters. As is standard practice in Muon, convolutions are reshaped as local-patch-operator matrices (Jordan et al., 2024). Given norms $\|\cdot\|_{(\ell)}$ on the matrix-valued layers and a norm $\|\cdot\|_{(L)}$ on $\mathbb{R}^q$, we define, for perturbations

$$\boldsymbol{\varepsilon} = (\boldsymbol{E}^{(1)}, \ldots, \boldsymbol{E}^{(L-1)}, \boldsymbol{e}) \in \Theta,$$

the product norm $\|\cdot\|_{\times,r}$ as the ℓ_r-norm, $1 \le r \le \infty$, over the norms of all blocks:

$$\|\boldsymbol{\varepsilon}\|_{\times,r} = \left\| \left(\|\boldsymbol{E}^{(1)}\|_{(1)}, \ldots, \|\boldsymbol{E}^{(L-1)}\|_{(L-1)}, \|\boldsymbol{e}\|_{(L)} \right) \right\|_r. \tag{6}$$

Equivalently, for $1 \le r < \infty$,

$$\|\boldsymbol{\varepsilon}\|_{\times,r} = \left(\sum_{\ell=1}^{L-1} \|\boldsymbol{E}^{(\ell)}\|_{(\ell)}^r + \|\boldsymbol{e}\|_{(L)}^r \right)^{1/r},$$

while

$$\|\boldsymbol{\varepsilon}\|_{\times,\infty} = \max \left\{ \|\boldsymbol{E}^{(1)}\|_{(1)}, \ldots, \|\boldsymbol{E}^{(L-1)}\|_{(L-1)}, \|\boldsymbol{e}\|_{(L)} \right\}.$$

The block norms $\|\cdot\|_{(\ell)}$ determine how a perturbation's magnitude is measured within each layer, while the outer ℓ_r-aggregation determines how perturbation budgets are distributed across layers.

For notational ease we write $\boldsymbol{G}^{(\ell)}(\boldsymbol{\theta}) := \nabla_{\boldsymbol{W}^{(\ell)}} L(\boldsymbol{\theta})$ and $\boldsymbol{g}(\boldsymbol{\theta}) := \nabla_{\boldsymbol{v}} L(\boldsymbol{\theta})$ for the loss gradients of the blocks. We identify $DL(\boldsymbol{\theta}) \in \Theta^*$ with the block gradients via the canonical Frobenius-Euclidean pairing: letting $\langle \boldsymbol{A}, \boldsymbol{B} \rangle_F = \mathrm{Tr}(\boldsymbol{A}^T \boldsymbol{B})$ denote the Frobenius inner product, then

$$DL(\boldsymbol{\theta})[\boldsymbol{\varepsilon}] = \sum_{\ell=1}^{L-1} \langle \boldsymbol{G}^{(\ell)}(\boldsymbol{\theta}), \boldsymbol{E}^{(\ell)} \rangle_F + \langle \boldsymbol{g}(\boldsymbol{\theta}), \boldsymbol{e} \rangle.$$

Therefore, if r^* is the Hölder conjugate[4] of r, with the conventions $1^* = \infty$ and $\infty^* = 1$, the dual product norm evaluated at the differential is

$$\|DL(\boldsymbol{\theta})\|_{\times,r,*} = \left\| \left(\|\boldsymbol{G}^{(1)}(\boldsymbol{\theta})\|_{(1),*}, \ldots, \|\boldsymbol{G}^{(L-1)}(\boldsymbol{\theta})\|_{(L-1),*}, \|\boldsymbol{g}(\boldsymbol{\theta})\|_{(L),*} \right) \right\|_{r^*}. \tag{7}$$

Observe the flattened Euclidean norm used in classic SAM is recovered from equation 6 by using Frobenius norms $\|\cdot\|_{(\ell)} = \|\cdot\|_F$ for $\ell \in [L-1]$, the Euclidean norm, $\|\cdot\|_{(L)} = \|\cdot\|_2$ on the remaining flattened parameters, and the Euclidean aggregation norm $r = 2$. Motivated by Muon, we pay particular attention to the ℓ_∞ aggregation geometry. When $r = \infty$ we have $r^* = 1$, and therefore equation 7 becomes

$$\|DL(\boldsymbol{\theta})\|_{\times,\infty,*} = \sum_{\ell=1}^{L-1} \|\boldsymbol{G}^{(\ell)}(\boldsymbol{\theta})\|_{(\ell),*} + \|\boldsymbol{g}(\boldsymbol{\theta})\|_{(L),*}.$$

Let $\partial\|\cdot\|_{\times,\infty,*}(DL(\boldsymbol{\theta}))$ denote the subdifferential of the dual norm as a function on Θ^*, evaluated at the covector $DL(\boldsymbol{\theta})$. Using the finite-dimensional identification $(\Theta^*)^* \cong \Theta$, its elements are identified with perturbations in parameter space. Since the dual product norm is a sum of convex functions, each acting on a disjoint parameter block, the standard subdifferential rule for sums gives

$$\partial\|\cdot\|_{\times,\infty,*}\big(DL(\boldsymbol{\theta})\big) = \partial\|\cdot\|_{(1),*}\big(\boldsymbol{G}^{(1)}(\boldsymbol{\theta})\big) \times \cdots \times \partial\|\cdot\|_{(L-1),*}\big(\boldsymbol{G}^{(L-1)}(\boldsymbol{\theta})\big) \times \partial\|\cdot\|_{(L),*}\big(\boldsymbol{g}(\boldsymbol{\theta})\big).$$

Thus, under the $r = \infty$ block-aggregation geometry, the steepest ascent perturbation may be computed independently for each layer by choosing a member of the subdifferential of the corresponding blockwise dual norm. Motivated by Muon, we specialize to the spectral norm, $\|\cdot\|_{(\ell)} = \|\cdot\|_{2\to 2}$ for all $\ell \in [L-1]$, and the Euclidean norm $\|\cdot\|_{(L)} = \|\cdot\|_2$ on the remaining parameters. Recall the dual norm of the spectral norm is the nuclear norm: the nuclear norm is differentiable at a matrix argument if and only if said matrix is full rank. Therefore, in general, the steepest direction need not be unique. As highlighted in Subsection 2.2, a canonical choice of nuclear-norm subgradient is the polar factor $\mathrm{Ortho}(\boldsymbol{G}^{(\ell)}(\boldsymbol{\theta}))$. Therefore, for $\boldsymbol{g}(\boldsymbol{\theta}) \neq 0$, the idealized layerwise spectral unit perturbation is

$$\boldsymbol{\varepsilon}^*(\boldsymbol{\theta}) = \left(\mathrm{Ortho}\big(\boldsymbol{G}^{(1)}(\boldsymbol{\theta})\big), \ldots, \mathrm{Ortho}\big(\boldsymbol{G}^{(L-1)}(\boldsymbol{\theta})\big), \frac{\boldsymbol{g}(\boldsymbol{\theta})}{\|\boldsymbol{g}(\boldsymbol{\theta})\|_2} \right). \tag{8}$$

The spectral norm used for matrix blocks and the Euclidean norm used for non-matrix parameters measure perturbation size in different "units". In principle, this suggests using separate perturbation radii, or equivalently separate radius scaling factors, for matrix and non-matrix parameter groups.

This leads to the practical spectral perturbation used in Section 4:

$$\widetilde{\boldsymbol{\varepsilon}}^*(\boldsymbol{\theta}) = \left(\mathrm{NS}_k\big(\boldsymbol{G}^{(1)}(\boldsymbol{\theta})\big), \ldots, \mathrm{NS}_k\big(\boldsymbol{G}^{(L-1)}(\boldsymbol{\theta})\big), \alpha_g \frac{\boldsymbol{g}(\boldsymbol{\theta})}{\|\boldsymbol{g}(\boldsymbol{\theta})\|_2} \right), \tag{9}$$

where $\mathrm{Ortho}(\cdot)$ is approximated by a k-step Newton–Schulz iteration NS_k (Jordan et al., 2024), and $\alpha_g \geq 0$ sets the perturbation budget of the Euclidean-measured non-matrix block relative to the spectral-measured matrix blocks.

3.2 Experimental variants

We now instantiate Algorithm 1 with concrete choices for the inner perturbation geometry and the outer update rule, with the specific goal of exploring how matrix-aware geometry (in the sense of Bernstein & Newhouse, 2025) may be incorporated into SAM. The variants we explore are summarized in Table 1, including their shortened names used for brevity throughout the paper.

For the inner step, we consider two main choices. The first is standard Euclidean SAM under the global ℓ_2 geometry on the full parameter vector, with inner perturbation $\rho\boldsymbol{\varepsilon}_t = \rho\,\mathbf{g}_t/\|\mathbf{g}_t\|_2$ for the loss gradient $\mathbf{g}_t$ over all parameters. We refer to this variant by the prefix "SAM-". The second is a layerwise variant that combines the $r = \infty$ block aggregation with the spectral norm on each matrix-valued parameter block, giving the perturbation of equation 9. We refer to it by the prefix "SpecSAM-". The spectral geometry applies only to matrix-valued blocks, with all remaining parameters perturbed under the Euclidean geometry. Note that this set of matrices is larger than the

[4]Recall r^* is the Hölder conjugate of r iff $\frac{1}{r} + \frac{1}{r^*} = 1$

Table 1: **A summary of the experimental design space.** Here $\mathbf{g}_t = \nabla L(\boldsymbol{\theta}_t)$ is the gradient and $\phi_{\rho,t} = \nabla L(\boldsymbol{\theta}_t + \rho\boldsymbol{\varepsilon}_t)$ the robust-loss gradient at step t. ADAMW, SGDW, and MUON denote the procedure that converts a (perturbed) gradient signal into a parameter update with the corresponding outer optimizer, i.e. the direction map $\mathcal{D}_t$ of Algorithm 1 followed by the parameter step.

Inner Step	**Outer Step**	
	AdamW/SGDW	**Muon**
None	ADAMW/SGDW $\boldsymbol{\theta}_{t+1} = \mathrm{AdamW/SGDW}(\boldsymbol{\theta}_t, \mathbf{g}_t)$	MUON $\boldsymbol{\theta}_{t+1} = \mathrm{Muon}(\boldsymbol{\theta}_t, \mathbf{g}_t)$
Euclidean	SAM-ADAMW/SGDW $\boldsymbol{\varepsilon}_t = \mathbf{g}_t/\|\mathbf{g}_t\|_2$ $\boldsymbol{\theta}_{t+1} = \mathrm{AdamW/SGDW}(\boldsymbol{\theta}_t, \phi_{\rho,t})$	SAM-MUON $\boldsymbol{\varepsilon}_t = \mathbf{g}_t/\|\mathbf{g}_t\|_2$ $\boldsymbol{\theta}_{t+1} = \mathrm{Muon}(\boldsymbol{\theta}_t, \phi_{\rho,t})$
Spectral	SPECSAM-ADAMW/SGDW $\boldsymbol{\varepsilon}_t = \widetilde{\boldsymbol{\varepsilon}}^*(\boldsymbol{\theta}_t)$ from *equation* 9 $\boldsymbol{\theta}_{t+1} = \mathrm{AdamW/SGDW}(\boldsymbol{\theta}_t, \phi_{\rho,t})$	SPECSAM-MUON $\boldsymbol{\varepsilon}_t = \widetilde{\boldsymbol{\varepsilon}}^*(\boldsymbol{\theta}_t)$ from *equation* 9 $\boldsymbol{\theta}_{t+1} = \mathrm{Muon}(\boldsymbol{\theta}_t, \phi_{\rho,t})$

Muon group used in the outer step; see Appendix A.4 for details. As an ablation, we additionally consider a "LAYERSAM-" control that uses the same $r = \infty$ layerwise aggregation as SPECSAM- but replaces the spectral block norm with the Frobenius norm. This isolates the effect of per-layer budgeting from that of the choice of block norm. We study it in Appendix B.2.

For the outer step, we consider AdamW, SGDW, and Muon. AdamW is used for the vision transformer (ViT) and SGDW for the convolutional network (CNN); throughout, we write ADAMW/SGDW for this architecture-dependent choice. In the ADAMW/SGDW variants, the direction map $\mathcal{D}_t$ of Algorithm 1 is AdamW/SGDW applied to all parameters. In the MUON variants, we follow the usual hybrid routing, in which Muon updates only the matrix-valued hidden-layer parameters,[5] while the remaining parameters are updated by AdamW. With the prefixes introduced above, this yields four methods: SAM-ADAMW/SGDW, SAM-MUON, SPECSAM-ADAMW/SGDW, and SPECSAM-MUON. Together with the two unperturbed baselines, these form the grid of Table 1.

4 EXPERIMENTS

To evaluate the effect of matrix-aware geometry in the inner and outer steps, we evaluate the proposed SAM variants on two common SAM settings: ImageNet-1K (Russakovsky et al., 2015) classification with a vision transformer and a convolutional neural network. Across both settings, we compare the SAM variants against their non-SAM baselines, AdamW/SGDW and Muon. Full experimental details are given in Appendix A. In addition to ImageNet-1K Top-1 validation accuracy, we report the "ReaL" accuracy for ImageNet (Beyer et al., 2020), which replaces the original labels with multi-label sets for each image, marking the model as correct if it predicts any label in the set (a framework designed to address label ambiguity). We also assess the Top-1 accuracy on ImageNet-R (Hendrycks et al., 2021), an out-of-distribution 30k image test set of artistic renditions of ImageNet images. We additionally perform diagnostic experiments on a small ViT trained on CIFAR-100 (Krizhevsky & Hinton, 2009), sweeping the SAM radius and analysing sharpness, effective rank, and generalization gap. The final vision transformer experiments as presented, including tuning and all algorithms, took roughly 4000 GPU-hours to run, while the ResNet experiments took 2000 GPU-hours. Models are trained on 8 H100 GPUs and we use the m-sharpness perturbation for SpecSAM and SAM, i.e. the parameter is perturbed independently on each GPU, and only the outer step gradient is accumulated (Foret et al., 2020). The entire project, including preliminary and diagnostic experiments, used about 10,000 hours of H100 96GB GPU compute.

4.1 IMAGENET-1K RESULTS

We train ViT-Small/16 (see Table 3 for architectural details) for 300 epochs following Beyer et al. (2022); Chen et al. (2022), and ResNet-50 (Table 4) for 120 epochs following the `timm resnet50`

[5] Some matrices are excluded from the Muon group and updated with AdamW; see Appendix A.4.

Table 2: **ImageNet-1K results**, reported as mean$_{\pm\text{std}}$ over three seeds and ordered within each panel by Val Top-1. In each column except Train Top-1, we bold the method with the highest mean together with methods for which a two-sided Welch t-test does not reject equality with the highest-mean method at the 5% level. Bold entries therefore indicate that no statistically significant difference from the highest-mean method was detected.

Name	Inner Step	Outer Step	Val Top-1	Train Top-1	ReaL	ImageNet-R
ViT-Small/16, 300 epochs						
SpecSAM-Muon	Spectral	Muon	$\mathbf{80.23}_{\pm 0.06}$	$89.84_{\pm 0.05}$	$\mathbf{86.26}_{\pm 0.08}$	$\mathbf{42.18}_{\pm 0.24}$
SAM-Muon	Euclidean	Muon	$79.97_{\pm 0.06}$	$90.68_{\pm 0.08}$	$86.00_{\pm 0.03}$	$\mathbf{41.68}_{\pm 0.56}$
SAM-AdamW	Euclidean	AdamW	$79.17_{\pm 0.35}$	$91.02_{\pm 0.52}$	$85.12_{\pm 0.23}$	$\mathbf{39.17}_{\pm 1.41}$
SpecSAM-AdamW	Spectral	AdamW	$78.71_{\pm 0.08}$	$89.34_{\pm 0.23}$	$84.95_{\pm 0.15}$	$38.13_{\pm 0.53}$
Muon	None	Muon	$74.95_{\pm 0.16}$	$93.38_{\pm 1.93}$	$80.79_{\pm 0.11}$	$34.35_{\pm 0.35}$
AdamW	None	AdamW	$74.26_{\pm 0.09}$	$92.41_{\pm 0.61}$	$80.09_{\pm 0.06}$	$32.68_{\pm 0.19}$
ResNet-50, 120 epochs						
SpecSAM-Muon	Spectral	Muon	$\mathbf{78.55}_{\pm 0.06}$	$93.67_{\pm 0.01}$	$\mathbf{84.76}_{\pm 0.10}$	$\mathbf{38.12}_{\pm 0.45}$
SAM-SGDW	Euclidean	SGDW	$77.99_{\pm 0.14}$	$92.69_{\pm 0.08}$	$84.26_{\pm 0.07}$	$\mathbf{37.50}_{\pm 0.34}$
SAM-Muon	Euclidean	Muon	$77.84_{\pm 0.03}$	$95.24_{\pm 0.15}$	$83.90_{\pm 0.05}$	$\mathbf{37.92}_{\pm 0.23}$
SpecSAM-SGDW	Spectral	SGDW	$77.77_{\pm 0.15}$	$93.16_{\pm 0.02}$	$84.05_{\pm 0.06}$	$36.75_{\pm 0.13}$
SGDW	None	SGDW	$77.35_{\pm 0.07}$	$93.33_{\pm 0.11}$	$83.55_{\pm 0.13}$	$36.04_{\pm 0.12}$
Muon	None	Muon	$77.05_{\pm 0.07}$	$96.35_{\pm 0.01}$	$83.17_{\pm 0.12}$	$34.92_{\pm 0.24}$

implementation (He et al., 2015; Wightman, 2019), using AdamW as the non-Muon outer optimizer for ViT-Small/16 and SGDW for ResNet-50. To isolate the effects of inner and outer step geometry from unrelated hyperparameter effects, we tune in two stages: first the learning rate and weight decay of the non-SAM AdamW/SGDW and Muon baselines, then the perturbation radius parameters for each SAM variant with the learning rate and weight decay fixed to that of its outer optimizer. Full configurations are given in Appendix A.5. Throughout the ImageNet-1K experiments, the number of Newton-Schulz iterations for the Muon outer step is 5, while the spectral inner step uses 3 iterations.

The first half of Table 2 contains the results for ViT-Small/16. Without an inner perturbation, Muon improves only modestly over AdamW ($74.26 \to 74.95$). A Euclidean (ℓ_2) inner step then contributes almost the same gain regardless of the outer optimizer (roughly $+4.9$ for AdamW and $+5.0$ for Muon): here the inner and outer effects are essentially *additive*. The spectral inner step behaves differently: it amplifies the advantage of Muon. SpecSAM-AdamW is in fact the weakest SAM method (78.71, below even SAM-AdamW), whereas SpecSAM-Muon attains the best validation accuracy of all methods (80.23). The spectral perturbation improves validation accuracy by 5.28 percentage points with Muon, compared with 4.45 points with AdamW. This suggests that the spectral inner geometry is more effective when paired with a similarly motivated outer step. The out-of-distribution ImageNet-R results point in the same direction (42.18 for SpecSAM-Muon), though at three seeds the top three methods there are not significantly different under the reported test. Interestingly, both spectral inner step variants reach *lower* train accuracy than their Euclidean counterparts, perhaps implying that the spectral inner geometry more harshly penalizes memorizing the training data.

The second half of Table 2 shows analogous results for ResNet-50. Without a SAM inner step, Muon slightly trails the standard SGDW baseline (77.05 vs 77.35). A Euclidean inner step again adds a roughly constant gain across outer optimizers ($+0.64$ for SGDW, $+0.79$ for Muon), leaving SAM-SGDW (77.99) marginally ahead of SAM-Muon (77.84). The spectral inner step once more amplifies Muon's advantage: it lifts Muon by 1.50 to 78.55, the best result overall, but SGDW by only 0.42 to 77.77, flipping the ordering so that the Muon pairing now clearly wins. SpecSAM-Muon also attains the highest ImageNet-ReaL accuracy, and leads on ImageNet-R as well, although as on ViT-Small/16 the ImageNet-R margins are within seed noise. Figure 1 summarizes this interaction across both architectures: the ℓ_2 inner step shifts both outer optimizers by a near-constant amount, whereas the spectral inner step selectively benefits Muon.

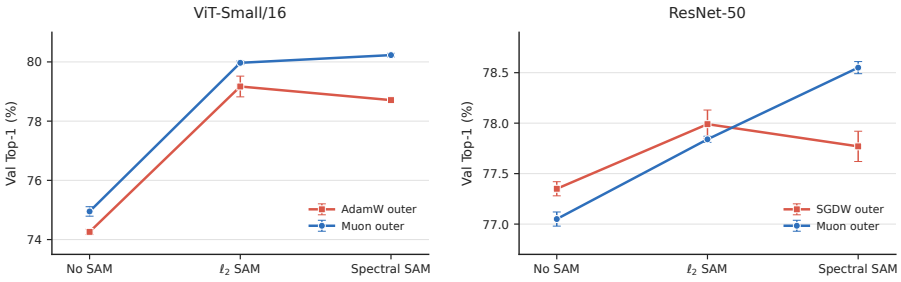


Figure 1: Inner step geometry vs. outer optimizer on ImageNet-1K (left: ViT-Small/16, right: ResNet-50).

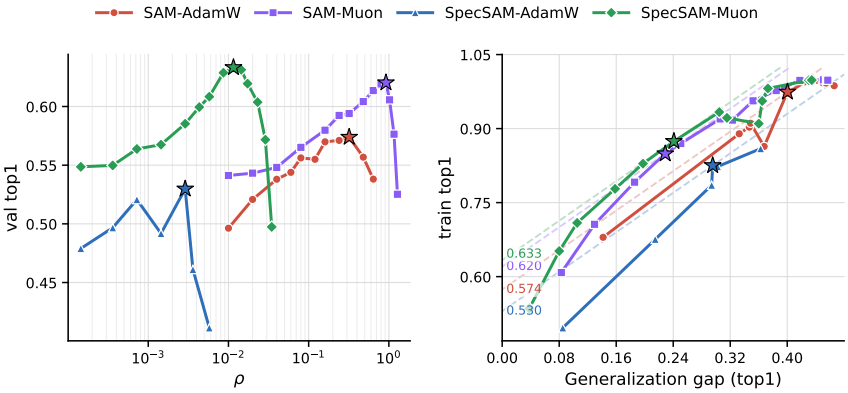


Figure 2: Sensitivity to SAM training radius ρ on CIFAR-100 with ViT-Tiny/4. **Left:** best validation top-1 vs. ρ. **Right:** train top-1 vs. generalization gap, with ρ as an implicit variable along each curve; dashed diagonals are constant val-top-1 contours since $\mathrm{val} = \mathrm{train} - \mathrm{gap}$. Stars mark the best-validation checkpoint per method.

4.2 CIFAR-100 DIAGNOSTICS

The ImageNet tables compare a single tuned perturbation radius per method. To see the interaction across ρ, we sweep ρ for the four SAM variants on CIFAR-100 with ViT-Tiny/4 (Beyer et al., 2022) trained for 120 epochs, with all four methods reusing one learning rate and weight decay selected by an AdamW grid search. Figure 2 summarizes validation top-1 sensitivity to ρ, and the train-accuracy–generalization-gap relationship.

The left panel of Figure 2 shows that the Euclidean inner steps prefer a larger ρ than the spectral inner steps. The right panel plots the relationship between train accuracy and the generalization gap (train minus validation top-1 accuracy) from the same set of results. Because validation top-1 equals train top-1 minus the gap, the dashed diagonals are constant-validation contours, and a method is ”better” when it reaches a higher contour at its optimal ρ. This contour position reflects how efficiently a method converts loss minimization into a reduced generalization gap, which depends on two factors: how well the outer optimizer minimizes the robust loss, and how much regularization the inner perturbation delivers per unit of training accuracy given up. SPECSAM-ADAMW fares worst, sacrificing too much training accuracy for too little regularization in return. SAM-MUON and SPECSAM-MUON both reach the highest contour, retaining substantial train accuracy while keeping the gap moderate, with SPECSAM-MUON handling the trade-off slightly better.

Effective-rank diagnostics on the same sweep, reported in full in Appendix C.3.1, give further insights into how spectral geometry interacts with SAM: in particular, Muon outer updates maintain higher weight, gradient, and feature ranks across the radius range. Inner perturbations at large ρ, by contrast, reduce all three rank quantities; this phenomenon was previously discussed by Andriushchenko et al. (2023), who posit that SAM may lead to low rank features.

5 Related Work

Sharpness-Aware Minimization (SAM) was introduced by Foret et al. (2020) as a two-step procedure for improving generalization by optimizing a local worst-case loss. Subsequent work has studied why SAM improves generalization (e.g. Andriushchenko & Flammarion, 2022; Wen et al., 2023) and proposed variants aimed at improving performance, reducing cost, or adapting SAM to particular settings (e.g. Liu et al., 2022; Zhuang et al., 2022; Li et al., 2024; Becker et al., 2025).

Most relevant to our work are SAM variants that modify the geometry of the perturbation. ASAM (Kwon et al., 2021) rescales the inner perturbation to reduce sensitivity to parameter rescaling; Fisher SAM (Kim et al., 2022) replaces Euclidean perturbation balls with Fisher-induced ellipsoids; and Riemannian SAM (Yun & Yang, 2023) formulates perturbations on Riemannian manifolds. Our work differs in two respects: we study spectral-norm geometry for matrix-valued parameters, and we vary both the inner perturbation geometry and the outer optimizer. Empirically, our strongest results arise when spectral inner perturbations are paired with a matrix-aware outer update. We also distinguish our method from Spectral-SAM (Zagitov et al., 2025), which targets adversarial robustness by projecting weights onto spectral-norm balls, whereas we modify the geometry of the SAM perturbation and update.

A separate line of work argues that hidden-layer weight matrices should be optimized using matrix-aware geometry rather than flattened Euclidean geometry (Bernstein & Newhouse, 2025; Pethick et al., 2025). This perspective motivates Muon (Jordan et al., 2024), which applies momentum followed by approximate orthogonalization and can be interpreted as a spectral-norm steepest-descent method for matrix parameters. Muon has recently been adopted in large-scale language-model training (Kimi Team, 2025; GLM-4.5 Team, 2025). To our knowledge, however, this matrix-aware optimization perspective has not previously been studied in the SAM setting.

6 Conclusion

Summary. We studied Sharpness-Aware Minimization through a norm-agnostic template that separates the geometry of the inner perturbation from that of the outer update, and, motivated by matrix-aware optimization, introduced a layerwise spectral inner perturbation for matrix-valued parameters. Across ImageNet-1K experiments on ViT-Small/16 and ResNet-50, the two geometries interact: a Euclidean inner step shifts both outer optimizers by a near-constant amount, whereas the spectral inner step selectively amplifies Muon's advantage. SpecSAM-Muon attains the best validation accuracy on both architectures (80.23 and 78.55) and the best ImageNet-ReaL accuracy, while the same perturbation paired with AdamW/SGDW is markedly less effective. The ordering on out-of-distribution ImageNet-R is the same, though at three seeds it is not statistically resolvable. Why the two geometries should interact this way is not settled by our experiments. One suggestive observation is that Muon sustains higher weight and feature effective rank across the CIFAR-100 radius sweep, which may be what lets it absorb a spectral perturbation without losing representational capacity; a proper account is left to future work.

Limitations. Compute constraints limited us to three seeds for the main ImageNet results, single runs for some diagnostics, and non-exhaustive hyperparameter searches. Our evidence is confined to image classification with two architectures; whether the same inner/outer interaction holds for language modelling, where Muon is most widely deployed, remains untested. Finally, like standard SAM, our methods require two backward passes per step, and the spectral variants add Newton–Schulz overhead in the inner step; we compare at equal epoch budgets rather than equal wall-clock time.

Future work. Our results suggest that future SAM variants should consider using Muon for matrix-shaped parameters in the outer step, since it is a simple replacement for AdamW/SGDW and performed strongly in our experiments. Matrix-aware inner geometries also deserve further study, especially when paired with a matrix-aware outer optimizer. Beyond the fixed Euclidean and spectral choices studied here, preliminary experiments provided anecdotal evidence that scheduling or interpolating between inner geometries may be promising. Finally, more theoretical work is needed to understand precisely why certain inner geometries pair better with some optimizers than others.

AI USE STATEMENT

In this work, we used generative AI tools for developing theoretical models or conceptual frameworks, formulating mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, assisting with translation, cleaning or reformatting datasets, and interpreting results. We have not used generative AI tools for any other applicable tasks with required disclosure, and generating synthetic data sets, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, and supporting qualitative or thematic data analysis are not applicable to this work.

Additionally, we used generative AI tools for searching for information, summarising or analysing existing literature, creating or editing software code, drafting parts of the research paper, and editing the research paper to improve readability. We have reviewed all AI-assisted work. AI-assisted code was manually reviewed and tested for correctness by the authors, and AI-assisted mathematical content, translations, data-processing outputs, and interpretations were checked for technical accuracy and consistency with the reported experiments. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Full experimental configurations, hyperparameter searches, optimizer routing, dataset preprocessing, and architectural details are provided in the appendix. Source code is included in the supplementary material. In addition, the appendix provides definitions and protocols for the diagnostic experiments.

REFERENCES

Noah Amsel, David Persson, Christopher Musco, and Robert M. Gower. The polar express: Optimal matrix sign methods and their application to the Muon algorithm. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yRtgZ1K8hO.

Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization, 2022. URL https://arxiv.org/abs/2206.06232.

Maksym Andriushchenko, Dara Bahri, Hossein Mobahi, and Nicolas Flammarion. Sharpness-aware minimization leads to low-rank features. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 47032–47051. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/92dd1adab39f362046f99dfe3c39d90f-Paper-Conference.pdf.

Marlon Becker, Frederick Altrock, and Benjamin Risse. Momentum-SAM: Sharpness aware minimization without computational overhead, 2025. URL https://arxiv.org/abs/2401.12033.

Jeremy Bernstein and Laker Newhouse. Modular duality in deep learning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 3920–3930. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/bernstein25a.html.

Lucas Beyer, Olivier J. Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with ImageNet?, 2020. URL https://arxiv.org/abs/2006.07159.

Lucas Beyer, Xiaohua Zhai, and Alexander Kolesnikov. Better plain ViT baselines for ImageNet-1k, 2022. URL https://arxiv.org/abs/2205.01580.

Rajendra Bhatia. *Matrix Analysis*, volume 169. Springer, 1997. ISBN 0387948465.

Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform ResNets without pre-training or strong data augmentations, 2022. URL https://arxiv.org/abs/2106.01548.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. Technical report, DeepSeek-AI, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/resolve/main/DeepSeek_V4.pdf.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Jiawei Du, Hanshu Yan, Jiashi Feng, Joey Tianyi Zhou, Liangli Zhen, Rick Siow Mong Goh, and Vincent Tan. Efficient sharpness-aware minimization for improved training of neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=n0OeTdNRG0Q.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

GLM-4.5 Team. GLM-4.5: Agentic, reasoning, and coding (ARC) foundation models, 2025. URL https://arxiv.org/abs/2508.06471.

Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers, 2022. URL https://arxiv.org/abs/2104.05704.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization, 2021. URL https://arxiv.org/abs/2006.16241.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Comput.*, 9(1):1–42, January 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.1.1. URL https://doi.org/10.1162/neco.1997.9.1.1.

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/.

Minyoung Kim, Da Li, Shell X Hu, and Timothy Hospedales. Fisher SAM: Information geometry and sharpness aware minimisation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 11148–11161. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/kim22f.html.

Kimi Team. Kimi K2: Open agentic intelligence, 2025. URL https://arxiv.org/abs/2507.20534.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *ICLR (Poster)*, 2015. URL http://dblp.uni-trier.de/db/conf/iclr/iclr2015.html#KingmaB14.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. ASAM: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5905–5914. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/kwon21b.html.

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 (CVPR)*, pp. 5631–5640, June 2024.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for LLM training, 2025. URL https://arxiv.org/abs/2502.16982.

Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12350–12360, 2022. doi: 10.1109/CVPR52688.2022.01204.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained LMOs, 2025. URL https://arxiv.org/abs/2502.07529.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European Signal Processing Conference*, pp. 606–610, 2007.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Sidak Pal Singh, Hossein Mobahi, Atish Agarwala, and Yann Dauphin. Avoiding spurious sharpness minimization broadens applicability of SAM. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55702–55719. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/singh25b.html.

Varun Singh, Lucas Krauss, Sami Jaghouar, Matej Sirovatka, Charles Goddard, Fares Obied, Jack Min Ong, Jannik Straube, Fern, Aria Harley, Conner Stewart, Colin Kealty, Maziyar Panahi, Simon Kirsten, Anushka Deshpande, Anneketh Vij, Arthur Bresnu, Pranav Veldurthi, Raghav Ravishankar, Hardik Bishnoi, DatologyAI Team, Arcee AI Team, Prime Intellect Team, Mark McQuade, Johannes Hagemann, and Lucas Atkins. Arcee Trinity large technical report, 2026. URL https://arxiv.org/abs/2602.17004.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.

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

Ross Wightman. PyTorch image models. https://github.com/huggingface/pytorch-image-models, 2019.

Jihun Yun and Eunho Yang. Riemannian SAM: Sharpness-aware minimization on Riemannian manifolds. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 65784–65800. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/cf701db0e3b4d0b8681ca6915ac3e87e-Paper-Conference.pdf.

Artur Zagitov, Roman Garaev, and Bader Rasheed. Sharpness-aware minimization meets spectral norm. Poster at the NeurIPS 2025 Workshop on Constrained Optimization for Machine Learning, 2025. URL https://www.researchgate.net/publication/400079982_Sharpness-Aware_Minimization_Meets_Spectral_Norm.

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

A EXPERIMENT DETAILS

A.1 ARCHITECTURES

The main results in Table 2 are obtained by evaluating the methods listed in Table 1 on a vision transformer model and a convolutional neural network, both at 224×224 resolution. The transformer model is the ViT-Small/16 from the original ViT family (Dosovitskiy et al., 2020) with non-overlapping 16×16 patches, modified following Beyer et al. (2022) with fixed two-dimensional sine–cosine positional embeddings, global average pooling over patch tokens rather than a class-token representation, and an MLP classification head. The convolutional baseline is the standard `timm resnet50` model with no architectural modifications. The model used for the CIFAR-100 diagnostics is ViT-Tiny/4, which follows the same overall design as ViT-Small/16 but uses a smaller architectural configuration and is adapted to 32×32 CIFAR images. Tables 3 and 4 list the main numerical specifications for reference.

Table 3: Architecture specifications of ViT-Small/16 and ViT-Tiny/4.

Specification	ViT-Small/16	ViT-Tiny/4
Input resolution	224×224	32×32
Patch size	16×16	4×4
Patch stride	16	4
Number of patch tokens	196	64
Embedding width	384	192
Depth	12	12
Attention heads	6	3
Head dimension	64	64
MLP ratio	4	4
Hidden expansion	1536	768
Classifier output dim	1000	100

Table 4: Architecture specifications of ResNet-50.

Specification	ResNet-50
Input resolution	224×224
Stem kernel size	7
Stem stride	2
Stem width	64
Total layers	50
Stage/block layout	3/4/6/3
Stage widths	256/512/1024/2048
Classifier output dim	1000

A.2 DATASETS AND PREPROCESSING

The main experiments use ImageNet-1K (Russakovsky et al., 2015), with the official training split for optimization, the validation split for evaluation, and inception-style augmentation (Szegedy et al., 2016). As in other SAM works (Chen et al., 2022; Zhuang et al., 2022; Du et al., 2022), we deliberately do not use Mixup, CutMix, RandAugment, so that the comparison reflects the optimization effect rather than the augmentation recipe. ImageNet-ReaL accuracy (Beyer et al., 2020) is computed on the same validation images using the multi-label ReaL annotations: a prediction is correct if it matches any label in the ReaL set. ImageNet-R (Hendrycks et al., 2021) is an out-of-distribution benchmark of approximately 30k rendition images covering 200 ImageNet classes, evaluated with the same preprocessing as the validation split. The diagnostic experiments use CIFAR-100 (Krizhevsky & Hinton, 2009) with the standard $50\text{k}/10\text{k}$ split, random cropping with 4-pixel padding and horizontal flipping during training.

A.3 TRAINING PROTOCOL

ViT-Small/16 follows the 300-epoch recipe of Beyer et al. (2022); Chen et al. (2022), using a cosine learning-rate schedule with linear warmup. ResNet-50 is trained for 120 epochs with the same family of schedules, matching the standard `timm resnet50` reference setup. CIFAR-100 ViT-Tiny/4 runs use the same family of cosine-with-warmup schedules over 120 epochs. For ResNet-50 and ViT-Small/16, we use a batch size of 4096 and 6000 warmup steps for ResNet and 10000 warmup steps for ViT, while for ViT-Tiny/4 we use a batch size of 512 and 1000 warmup steps. Following Chen et al. (2022), ViT experiments use binary cross-entropy loss with label smoothing, while ResNet-50 uses standard softmax cross-entropy.

For all experiments, the model dropout rate is 0.1 and the loss label smoothing is 0.1. We do not apply drop path to the ViT models. AdamW uses $(\beta_1, \beta_2) = (0.9, 0.999)$, SGDW uses momentum 0.9, and Muon uses momentum 0.95. ViT-Small/16 and ResNet-50 are trained on 8 H100 GPUs, while ViT-Tiny/4 is trained on 1 H100 GPU. All reported metrics are evaluated at the best-validation checkpoint; training top-1 accuracy is evaluated using the same transform as the validation split, without training-time augmentation. The results reported in Table 2 are based on three random seeds.

A.4 OPTIMIZER ROUTING AND IMPLEMENTATION

The methods of Table 1 fix the inner perturbation geometry and the outer optimizer family but leave parameter routing implicit; we make these choices explicit here. For ADAMW/SGDW and the *-ADAMW/SGDW variants, all parameters are updated by AdamW / SGDW respectively. For MUON and the *-MUON variants, parameters are split into a Muon group containing the hidden-layer matrix-valued backbone parameters and an AdamW group containing the rest. On ViT-Small/16, the classification head and the patch-embedding layer are excluded from the Muon group; on ResNet-50, convolutional kernels are reshaped to two-dimensional matrices and routed to Muon with the final classification head and the first convolutional layer again excluded. We use a Moonlight-style update scaling for Muon (Liu et al., 2025) to match update RMS of AdamW, and Newton–Schulz iterations for the outer Muon orthogonalization are fixed at 5. The SPECSAM inner step uses a strictly broader perturbation set than the outer Muon group: all matrix-valued parameters are perturbed, including those routed to AdamW for the outer step (note that this includes embeddings, the classification layer, and the low-dimension convolutions), while non-matrix parameters use the Euclidean SAM variant with a separately tuned radius (see Appendix A.5). The orthogonalization in the inner step uses the same Newton–Schulz iteration as the outer Muon step (Jordan et al., 2024), with 3 steps used in the main reported numbers for ImageNet experiments, and five steps for CIFAR-100 experiments. The spectral perturbation radii ρ for the output layer of ViT-Tiny/4 are $\sqrt{1.92}$ times as large as those for the other parameters.

A.5 HYPERPARAMETER SEARCH

We tune hyperparameters separately for each architecture using a two-stage grid search. To reduce the computational cost, all hyperparameter searches use half of the training budget of the final experiments, namely 150 epochs for ViT-Small/16 and 60 epochs for ResNet-50. We first tune the learning rate parameters and weight decay for the non-SAM baselines. For each sharpness-aware variant, we retain the learning rate and weight decay parameters selected for the corresponding outer optimizer and tune the perturbation radius parameters. This protocol prevents a SAM variant from benefiting solely from a learning-rate choice unavailable to its corresponding non-SAM baseline.

Notation. Throughout this section, LR and WD denote the base learning rate and the weight decay, respectively, and ρ denotes the perturbation radius of the sharpness-aware update. As described in Appendix A.4, when Muon is the outer optimizer the parameters are split into a matrix branch, updated by Muon, and a non-matrix branch, updated by AdamW; these two branches use separate learning rates, which we denote by mLR and nLR respectively. Similarly, the spectral perturbation uses separate radii for the matrix and non-matrix parameter groups, which we denote by ρ_m and ρ_n, i.e., if $\rho = \rho_m$ in Table 1, $\rho_n = \alpha_g \rho_m$ according to equation 9.

Table 5: Hyperparameter search grids for ViT-Small/16. The sharpness-aware variants inherit the learning rate and weight decay selected for their base optimizer, so only the perturbation-radius grids are listed for them.

Method	Hyperparameter	Search grid
AdamW	LR	$10^{-3}\{1, \sqrt{3}, 3, 3\sqrt{3}, 9\}$
	WD	$10^{-1}\{\frac{1}{3},\ 1,\ 3\}$
Muon	$\mathrm{mLR}, \mathrm{nLR}$	$10^{-3}\{1, \sqrt{3}, 3, 3\sqrt{3}, 9\}$
	WD	$10^{-1}\{\frac{1}{3},\ 1,\ 3\}$
SAM-AdamW	ρ	$10^{-1}\{3, \sqrt{21}, 7, \frac{7\sqrt{21}}{3}\}$
SAM-Muon	ρ	$10^{-1}\{3, \sqrt{21}, 7, \frac{7\sqrt{21}}{3}\}$
SpecSAM-AdamW	ρ_m	$5.1 \times 10^{-4}\{\frac{9\sqrt{21}}{49}, \frac{9}{7}, \frac{3\sqrt{21}}{7}, 3\}$
	ρ_n	$10^{-3}\{0, 0.1, 0.1\sqrt{10}, 1, \sqrt{10}, 10\}$
SpecSAM-Muon	ρ_m	$5.1 \times 10^{-4}\{\frac{9\sqrt{21}}{49}, \frac{9}{7}, \frac{3\sqrt{21}}{7}, 3\}$
	ρ_n	$10^{-3}\{0, 0.1, 0.1\sqrt{10}, 1, \sqrt{10}, 10\}$

Table 6: Hyperparameter search grids for ResNet-50. As above, the sharpness-aware variants inherit the learning rate and weight decay from their base optimizer; only the additional perturbation-radius grids are listed for them.

Method	Hyperparameter	Search grid
SGDW	LR	$1.6\{\frac{1}{3}, \frac{1}{\sqrt{3}}, 1, \sqrt{3}, 3\}$
	WD	$10^{-3}\{\frac{1}{3}, 1, 3\}$
Muon	$\mathrm{mLR}, \mathrm{nLR}$	$10^{-2}\{0.2, 0.5, 1, 2, 5\}$
	WD	$10^{-1}\{\frac{1}{3}, 1, 3\}$
SAM-SGDW	ρ	$10^{-1}\{0.125, 0.25, 0.5, 1, 2\}$
SAM-Muon	ρ	$10^{-1}\{0.5, 1, 2, 4, 8\}$
SpecSAM-SGDW	ρ_m	$10^{-3}\{0.2, 0.5, 1, 2, 5\}$
	ρ_n	$10^{-3}\{0, 0.1, 0.1\sqrt{10}, 1, \sqrt{10}, 10\}$
SpecSAM-Muon	ρ_m	$10^{-2}\{0.2, 0.5, 1, 2, 5\}$
	ρ_n	$10^{-3}\{0, 0.1, 0.1\sqrt{10}, 1, \sqrt{10}, 10\}$
LayerSAM-SGDW	ρ_m	$10^{-2}\{0.1, 0.2, 0.5, 1, 2, 5\}$
	ρ_n	$10^{-3}\{0, 0.1, 0.1\sqrt{10}, 1, \sqrt{10}, 10\}$
LayerSAM-Muon	ρ_m	$10^{-1}\{0.1, 0.2, 0.5, 1, 2, 5\}$
	ρ_n	$10^{-4}\{0, 0.1, 0.1\sqrt{10}, 1, \sqrt{10}, 10\}$

The search values for the two architectures are listed in Tables 5 and 6. Final configurations are selected by validation Top-1 accuracy at the reduced training budgets and are reported in Tables 7 and 8.

For the ViT-Tiny/4 experiments in Appendix C.3, we perform a search over learning rates $\{0.0005, 0.001\}$ and weight decay values $\{0.02, 0.05, 0.1\}$ for AdamW. The selected learning rate and weight decay are 0.0005 and 0.05, respectively. We then use the same learning rate and weight decay for all methods when training ViT-Tiny/4.

Table 7: Selected hyperparameters for ViT-Small/16.

Method	LR/mLR	nLR	WD	ρ, ρ_m	ρ_n
AdamW	3×10^{-3}	–	3×10^{-1}	–	–
Muon	$3\sqrt{3} \times 10^{-3}$	3×10^{-3}	3×10^{-1}	–	–
SAM-AdamW	3×10^{-3}	–	3×10^{-1}	$\sqrt{21} \times 10^{-1}$	–
SAM-Muon	$3\sqrt{3} \times 10^{-3}$	3×10^{-3}	3×10^{-1}	7×10^{-1}	–
SpecSAM-AdamW	3×10^{-3}	–	3×10^{-1}	$\frac{9}{7} \times 5.1 \times 10^{-4}$	0
SpecSAM-Muon	$3\sqrt{3} \times 10^{-3}$	3×10^{-3}	3×10^{-1}	$\frac{3\sqrt{21}}{7} \times 5.1 \times 10^{-4}$	$\sqrt{10} \times 10^{-3}$

Table 8: Selected hyperparameters for ResNet-50.

Method	LR/mLR	nLR	WD	ρ, ρ_m	ρ_n
SGDW	1.6	–	10^{-3}	–	–
Muon	2×10^{-2}	2×10^{-2}	10^{-1}	–	–
SAM-SGDW	1.6	–	10^{-3}	5×10^{-2}	–
SAM-Muon	2×10^{-2}	2×10^{-2}	10^{-1}	2×10^{-1}	–
SpecSAM-SGDW	1.6	–	10^{-3}	5×10^{-4}	10^{-3}
SpecSAM-Muon	2×10^{-2}	2×10^{-2}	10^{-1}	2×10^{-2}	$\sqrt{10} \times 10^{-4}$
LayerSAM-SGDW	1.6	–	10^{-3}	2×10^{-3}	$\sqrt{10} \times 10^{-3}$
LayerSAM-Muon	2×10^{-2}	2×10^{-2}	10^{-1}	2×10^{-2}	10^{-5}

A.6 Datasets, models, and licenses.

We use ImageNet-1K/ILSVRC2012 for training and validation under the ImageNet access terms, which restrict use to non-commercial research and educational purposes and note that ImageNet does not own the copyright to the underlying images. We evaluate ImageNet-ReaL annotations from the `google-research/reassessed-imagenet` repository, released under the Apache License 2.0, on the ImageNet validation images and therefore still subject to the ImageNet image terms. We evaluate ImageNet-R, whose repository is released under the MIT License. We use CIFAR-100 from the official Toronto dataset page; we could not identify a separate explicit dataset license, so we treat its license as not clearly specified. Our ViT-style backbone follows `timm` with modifications according to the Google ViT and `big_vision` references, whose code is released under the Apache License 2.0, and our ResNet-50 implementation follows `timm`, whose code is released under the Apache License 2.0. We train models from scratch and do not use third-party pretrained checkpoints.

A.7 Training recipe for ablation

Here we describe the setup for the ablation experiments in Appendix B.1. Inspired by Hassani et al. (2022), we use a lightweight CNN–Transformer hybrid model designed for 32×32 CIFAR images. The model first applies a three-layer convolutional stem, which downsamples the input by a factor of 4 and produces an 8×8 grid of 64 tokens with embedding dimension 192. Fixed two-dimensional sine–cosine positional embeddings are then added, followed by 4 pre-norm Transformer blocks with 2 attention heads and MLP ratio 6. The final prediction is obtained by global average pooling followed by a two-layer MLP classification head.

For training, we train for 150 epochs on CIFAR-100 with batch size 512, 2910 warmup iterations, and the basic data augmentation described in Appendix A.2. For each outer optimizer, Muon or AdamW, we select the learning rate and weight decay by grid search and then keep them fixed when sweeping the perturbation radii. Specifically, for the Muon outer optimizer we use learning rate 0.009 and weight decay 0.5, while for the AdamW outer optimizer we use learning rate 0.006 and weight decay 0.3.

B ABLATION EXPERIMENTS

B.1 ABLATION FOR NON-MATRIX PARAMETERS PERTURBATION

This appendix checks whether adding a separate perturbation radius for the non-matrix parameter group is useful in a smaller CIFAR-100 setting, details of the experiments are described in Appendix A.7. For these ablations, we split the inner perturbation radius into two parts: ρ_m for matrix-valued parameters and ρ_n for the remaining non-matrix parameters, corresponding to the $\boldsymbol{W}^{(\ell)}$ and $\boldsymbol{v}$ blocks in Section 3.1. Tables 9 and 10 report the mean and standard deviation of validation top-1 accuracy. Overall, the results do not show a consistent improvement from taking $\rho_n > 0$.

Table 9: SPECSAM-MUON non-matrix perturbation ablation on CIFAR-100. ρ_m denotes the perturbation radius for matrix-valued parameters, while ρ_n denotes the perturbation radius for the remaining non-matrix parameters. Each entry reports mean validation top-1 accuracy $\pm$ sample standard deviation over three random seeds.

ρ_m	$\rho_n = 0$	$\rho_n = 10^{-5}$	$\rho_n = 10^{-4}$	$\rho_n = 10^{-3}$	$\rho_n = 10^{-2}$	$\rho_n = \rho_m$
0.005	73.39 ± 0.17	73.42 ± 0.30	73.20 ± 0.26	73.37 ± 0.09	73.34 ± 0.15	73.17 ± 0.20
0.02	74.65 ± 0.24	74.64 ± 0.32	74.60 ± 0.32	74.68 ± 0.38	74.72 ± 0.43	74.70 ± 0.36
0.035	74.87 ± 0.23	74.93 ± 0.06	74.94 ± 0.17	74.76 ± 0.12	74.80 ± 0.20	74.80 ± 0.04

Table 10: SPECSAM-ADAMW non-matrix perturbation ablation on CIFAR-100. ρ_m denotes the perturbation radius for matrix-valued parameters, while ρ_n denotes the perturbation radius for the remaining non-matrix parameters. Each entry reports mean validation top-1 accuracy $\pm$ sample standard deviation over three random seeds.

ρ_m	$\rho_n = 0$	$\rho_n = 10^{-5}$	$\rho_n = 10^{-4}$	$\rho_n = 10^{-3}$	$\rho_n = 10^{-2}$	$\rho_n = \rho_m$
0.0005	70.62 ± 0.50	70.59 ± 0.67	70.84 ± 0.56	70.34 ± 0.32	70.48 ± 0.46	70.75 ± 0.59
0.001	70.70 ± 0.30	71.00 ± 0.45	70.81 ± 0.26	70.70 ± 0.46	70.75 ± 0.72	–
0.002	70.94 ± 0.30	70.66 ± 0.43	70.92 ± 0.19	70.80 ± 0.27	70.87 ± 0.13	70.57 ± 0.13

B.2 ABLATION FOR LAYERWISE PERTURBATION

The main experiments contrast a *global* Euclidean inner perturbation (SAM) with a *layerwise spectral* one (SPECSAM). These two choices differ in two respects at once: relative to standard SAM, SPECSAM both (i) allocates a separate perturbation budget to each matrix-valued layer, through the $r = \infty$ block aggregation of equation 6, and (ii) measures each block with the spectral norm instead of the Frobenius norm. To disentangle these two effects, we introduce LAYERSAM, an intermediate variant that keeps the layerwise ($r = \infty$) budgeting of SPECSAM but replaces the per-block spectral norm with the per-block Frobenius (Euclidean) norm, i.e. $\|\cdot\|_{(\ell)} = \|\cdot\|_F$ for all $\ell \in [L-1]$. Since the Frobenius norm is self-dual with subgradient $\boldsymbol{G}^{(\ell)}/\|\boldsymbol{G}^{(\ell)}\|_F$, the resulting inner perturbation normalizes each matrix block independently,

$$\boldsymbol{\varepsilon}^*(\boldsymbol{\theta}) = \left(\frac{\boldsymbol{G}^{(1)}(\boldsymbol{\theta})}{\|\boldsymbol{G}^{(1)}(\boldsymbol{\theta})\|_F}, \ldots, \frac{\boldsymbol{G}^{(L-1)}(\boldsymbol{\theta})}{\|\boldsymbol{G}^{(L-1)}(\boldsymbol{\theta})\|_F}, \alpha_g \frac{\boldsymbol{g}(\boldsymbol{\theta})}{\|\boldsymbol{g}(\boldsymbol{\theta})\|_2} \right), \tag{10}$$

with the non-matrix block handled exactly as in our SPECSAM runs (a separate Euclidean radius ρ_n; see Appendices A.4 and A.5). LAYERSAM therefore isolates the effect of *per-layer budgeting* from that of *spectral geometry*: any gap between SAM and LAYERSAM is attributable to layerwise normalization, while any further gap between LAYERSAM and SPECSAM is attributable to the spectral norm itself.

We evaluate both outer optimizers on ResNet-50/ImageNet-1K under the protocol of Section 4.1, with search grids and selected radii reported in Appendix A.5. Table 11 places LAYERSAM alongside the corresponding non-SAM baselines and the SAM/SPECSAM variants.

Two observations stand out. First, moving from a global ℓ_2 ball to a layerwise ℓ_2 ball does not help: LAYERSAM trails SAM with both outer optimizers (77.67 vs. 77.99 for SGDW, and 77.49 vs.

Table 11: LAYERSAM ablation on ResNet-50/ImageNet-1K. Each block fixes the outer optimizer and varies only the inner-step geometry: no perturbation (None), a global ℓ_2 ball (SAM), a layerwise ℓ_2/Frobenius ball (LAYERSAM, equation 10), and a layerwise spectral ball (SPECSAM). Validation Top-1 is reported as mean$_{\pm\text{std}}$ over 3 seeds; the non-LayerSAM rows are reproduced from Table 2. Bolding follows the convention of Table 2.

Name	Inner Step	Outer Step	Val Top-1
SGDW	None	SGDW	$77.35_{\pm 0.07}$
SAM-SGDW	ℓ_2 (global)	SGDW	$77.99_{\pm 0.14}$
LAYERSAM-SGDW	ℓ_2 (layerwise)	SGDW	$77.67_{\pm 0.06}$
SPECSAM-SGDW	Spectral (layerwise)	SGDW	$77.77_{\pm 0.15}$
MUON	None	Muon	$77.05_{\pm 0.07}$
SAM-MUON	ℓ_2 (global)	Muon	$77.84_{\pm 0.03}$
LAYERSAM-MUON	ℓ_2 (layerwise)	Muon	$77.49_{\pm 0.04}$
SPECSAM-MUON	Spectral (layerwise)	Muon	$\mathbf{78.55}_{\pm 0.06}$

77.84 for Muon). Per-layer budget allocation on its own is therefore not the source of SPECSAM's gains. Second, the large SPECSAM-MUON result (78.55) is recovered only when the layerwise budgeting is combined with the spectral norm: replacing the spectral block norm by the Frobenius block norm (LAYERSAM-MUON) erases the advantage, dropping the method back below even global SAM-MUON. With the SGDW outer step the three inner geometries remain within ≈ 0.3 of one another, consistent with the weaker inner/outer interaction observed for non-matrix-aware optimizers in the main text. Taken together, this ablation indicates that the benefit of SPECSAM-MUON comes specifically from the spectral geometry of the inner perturbation, rather than from layerwise normalization alone, and that it is realized only when paired with a matrix-aware outer step. This also points to a further design for matrix-aware perturbations: coupling the per-layer budgets through a finite aggregation exponent (e.g. $r = 2$) rather than the $r = \infty$ aggregation used here, which we leave to future work.

C ADDITIONAL EXPERIMENTAL RESULTS ON CIFAR-100

C.1 SHARPNESS ESTIMATION

Top Hessian eigenvalues are a single scalar summary of curvature and do not distinguish between different geometric notions of sharpness. Following Section 2.1, we measure sharpness with respect to the same perturbation sets that each method's inner step probes. Fix a parameter $\boldsymbol{\theta} \in \Theta$ and a closed perturbation set $\mathcal{C}_\rho \subset \Theta$. Let S be a fixed subset of the training set, drawn once and reused across all methods, and let L_S denote the empirical loss restricted to S; we take S to be the full training set. We define

$$\mathcal{S}_{\mathcal{C}}(\boldsymbol{\theta}; \rho) \;:=\; \max_{\boldsymbol{\varepsilon} \in \mathcal{C}_\rho} L_S(\boldsymbol{\theta} + \boldsymbol{\varepsilon}) - L_S(\boldsymbol{\theta}).$$

We use two choices of $\mathcal{C}_\rho$, matching the two inner-step geometries in Table 1. ℓ_2 *sharpness* uses the Euclidean ball $\{\boldsymbol{\varepsilon} : \|\boldsymbol{\varepsilon}\|_2 \le \rho\}$ on the flattened parameter vector. *Spectral sharpness* uses a simplified version of the layerwise spectral ball: every matrix block $\boldsymbol{E}^{(\ell)}$ has spectral norm at most ρ, and the non-matrix block is set to zero. The radii are stated in Appendix C.3.2. The inner maximum is approximated by projected gradient ascent with three random restarts. For ℓ_2 sharpness, the iterate is rescaled so its ℓ_2-norm does not exceed ρ. For spectral sharpness, each matrix block is rescaled so its spectral norm does not exceed ρ, with the spectral norm estimated by 8 steps of power iteration, and the non-matrix block is fixed at zero; the per-block ascent direction mirrors the (simplified) SPECSAM inner step, replacing each block gradient by its five-step Newton–Schulz approximation. These approximations introduce small errors, so the reported values should be read as a qualitative comparison rather than exact maxima.

C.2 EFFECTIVE-RANK DIAGNOSTICS

For a non-zero matrix $\boldsymbol{A}$ with non-zero singular values $\sigma_1 \ge \cdots \ge \sigma_r > 0$, let $p_i = \sigma_i / \sum_{j=1}^{r} \sigma_j$. We use the entropy-based effective rank of Roy & Vetterli (2007),

$$\mathrm{erank}(\boldsymbol{A}) \;:=\; \exp\Big(-\sum_{i=1}^{r} p_i \log p_i\Big),$$

with $\mathrm{erank}(\mathbf{0}) = 0$. It lies between 1 and $\mathrm{rank}(\boldsymbol{A})$, attaining the upper bound when the non-zero singular values are equal and decreasing as the spectrum concentrates. We apply this definition to three objects, all evaluated on the entire training dataset.

The *weight effective rank* sums $\mathrm{erank}(\cdot)$ over the attention and MLP weight matrices of a transformer block. The *gradient effective rank* applies erank to the corresponding block gradients $\boldsymbol{G}^{(\ell)}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{W}^{(\ell)}} L_S(\boldsymbol{\theta})$ and sums them within the block. The *feature effective rank* takes the matrix $\boldsymbol{H} \in \mathbb{R}^{|S| \times d}$ of block-output token-pooled features (where d is the embedding width and each row corresponds to one input in S), column-centers it to obtain $\bar{\boldsymbol{H}}$, and reports $\mathrm{erank}(\bar{\boldsymbol{H}}^\top \bar{\boldsymbol{H}} / |S|)$; since this matrix is symmetric positive semidefinite, its singular values coincide with its eigenvalues. For ViT-Tiny/4 we report each quantity at layers 4 and 9 out of 12, chosen as a representative early and late block, and S is taken as the full training set.

C.3 CIFAR-100 RADIUS-SWEEP

The diagnostic experiments in Section 4.2, the rank diagnostics in Appendix C.3.1, and the sharpness diagnostics in Appendix C.3.2 all use a single CIFAR-100 setup, so that the rank and sharpness panels can be read together as different views of the same underlying sweep. The model and data augmentation used for the diagnostic experiments are described in Appendix A.1 and Appendix A.2. Learning rate and weight decay are selected by an AdamW grid search and reused for all four SAM-type methods, so that the sweep isolates the effect of ρ. For each run we record the best validation top-1 checkpoint and evaluate train top-1, the top-1 generalization gap, both sharpness measures, and the effective-rank diagnostics on that checkpoint.

C.3.1 CIFAR-100 RANK DIAGNOSTICS

We use the CIFAR-100 sweep to ask what changes inside the network as the inner and outer geometry varies, computing the effective rank of hidden weight matrices, of loss gradients with respect to them,

and of block-output feature covariances at two depths, with definitions and aggregation given in Appendix C.2. The curves are shown in Figure 3.

Muon-based outer updates keep the weight effective rank near the top of the range over much of the sweep and also maintain higher feature effective rank than the AdamW-based variants around the best-validation checkpoints. This may help explain why the Muon variants can profitably use larger radii: high-rank representations have more fitting capacity, which makes stronger regularization useful. However, when ρ becomes too large, all three rank quantities fall and validation accuracy drops, suggesting a loss of representation capacity rather than improved generalization.

C.3.2 CIFAR-100 SHARPNESS DIAGNOSTICS

On the same CIFAR-100 sweep we measure finite-radius post-training sharpness with projected gradient ascent at measurement radius $\rho = 0.25$ for ℓ_2 sharpness and $\rho = 0.018$ for spectral sharpness. Figure 4 plots the resulting sharpness values against the perturbation radii ρ in the top row, and against the top-1 generalization gap in the second row.

The top row confirms that increasing ρ generally lowers both measured sharpness values, so along the sweep over ρ, measured sharpness and the top-1 gap are tightly entangled: larger radii flatten the loss landscape under the measurement and reduce the gap, which is useful evidence that the perturbation acts as the regularizer it is designed to be but is not a sufficient explanation of which checkpoint generalizes best. It is worth noting that, for fixed generalization gap, the SPECSAM variants almost always find sharper minima in terms of both ℓ_2 sharpness and spectral sharpness, which they penalize explicitly. The sharpness plots should also be read together with the rank plots of Appendix C.3.1: sharpness measures how much local sensitivity has been controlled, while rank measures how much representation capacity remains.

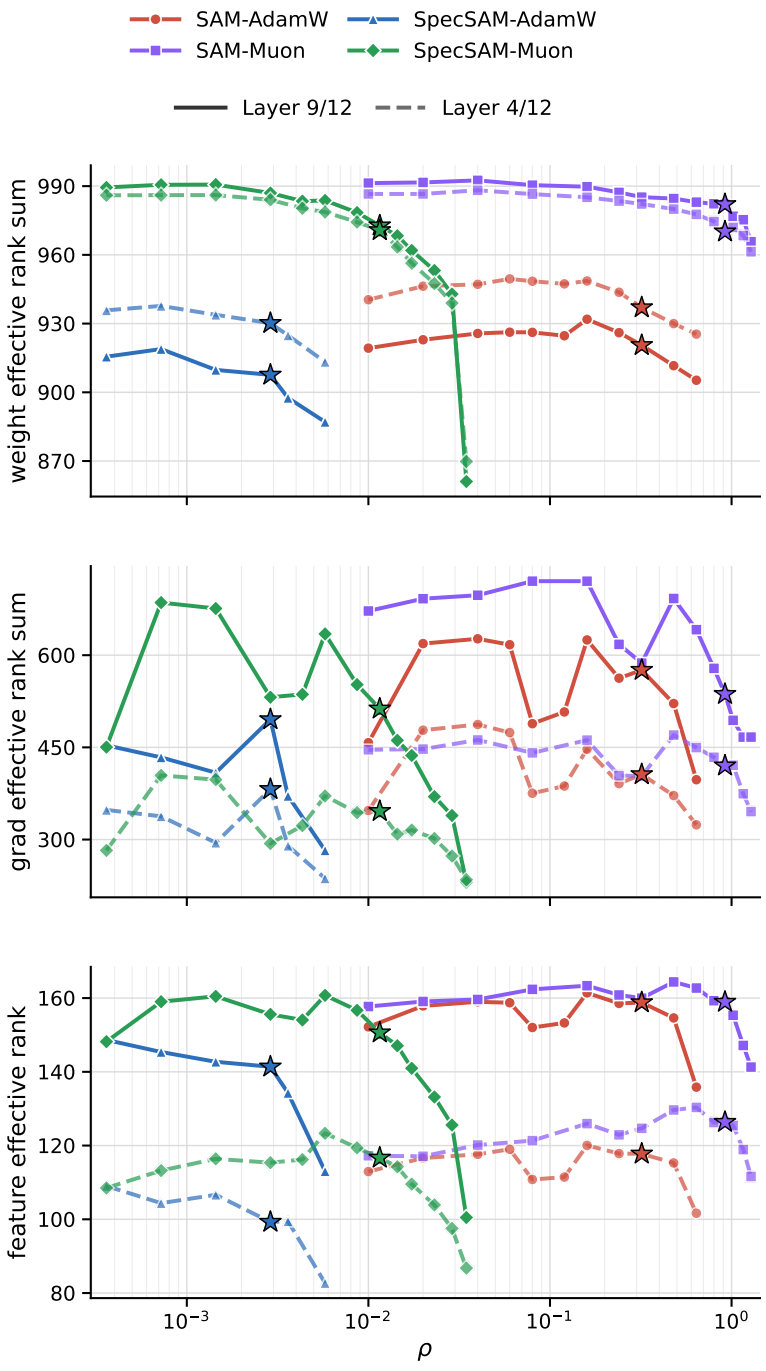


Figure 3: Effective-rank diagnostics for the CIFAR-100 ViT-Tiny/4 radius sweep. Within each transformer block, weight and gradient ranks are summed over the attention and MLP matrices; feature rank is the effective rank of the centered block-output covariance. Solid and dashed curves correspond to layer $9/12$ and $4/12$. Stars mark the best-validation checkpoint per method.

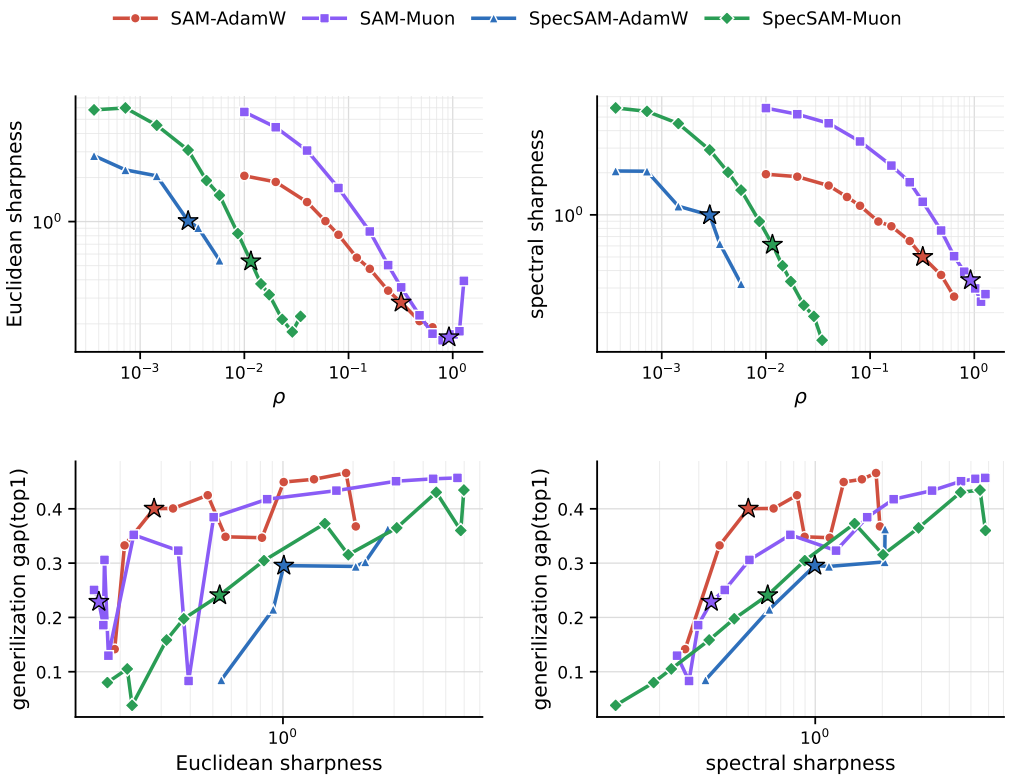


Figure 4: Post-training sharpness analysis on the CIFAR-100 ViT-Tiny/4 radius sweep. The "sharpness" labels are described in Appendix C.1. The left and right columns use ℓ_2 and spectral sharpness, respectively. The top row plots measured sharpness against the training radius ρ; the second row shows the **top-1** generalization gap against the measured sharpness. Stars mark the best-validation checkpoint per method.