REVISITING SHARPNESS-AWARE MINIMIZATION: A MORE FAITHFUL AND EFFECTIVE IMPLEMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sharpness-Aware Minimization (SAM) enhances generalization by minimizing the maximum training loss within a predefined neighborhood around the parameters. However, its practical implementation approximates this as gradient ascent(s) followed by applying the gradient at the ascent point to update the current parameters. Although this practice is justified as approximately optimizing the objective by neglecting the (full) derivative of the ascent point with respect to the current parameters, a direct and intuitive understanding of why using the gradient at the ascent point to update the current parameters works superiorly (despite a shift in location) is still lacking. Our work bridges this gap by proposing and justifying a novel, intuitive interpretation: the gradient at the single-step ascent point, when applied to the current parameters, provides a better approximation of the direction *from the current parameters towards the maximum within the local neighborhood* than the local gradient, thereby enabling a more direct escape from the maximum within the local neighborhood. Nevertheless, our analysis further reveals that: i) the approximation by the gradient at the single-step ascent point is often inaccurate; and ii) the approximation quality may degrade as the number of ascent steps increases (explaining the unexpectedly inferior performance of multi-step SAM). To address these limitations, we propose in this paper eXplicit Sharpness-Aware Minimization (XSAM), which addresses the first limitation by explicitly estimating the direction of the maximum during training (and then updates parameters along the opposite direction), and the second by crafting a search space that can effectively leverage the information provided by the gradient at the multi-step ascent point. XSAM features a unified formulation that applies to both single-step and multi-step settings and only incurs negligible additional computational overhead. Extensive experiments demonstrate the consistent superiority of XSAM against existing counterparts across various models, datasets, and settings.

1 INTRODUCTION

The success of modern machine learning relies heavily on overparameterization. This necessitates strong regularization, either implicit or explicit, from the training procedures (Srivastava et al., 2014; Gidel et al., 2019; Karakida et al., 2023) to ensure generalization beyond the training set (Zhang et al., 2021). In recent years, Sharpness-Aware Minimization (SAM) (Foret et al., 2020; Kwon et al., 2021; Liu et al., 2022b; Kim et al., 2023; Mordido et al., 2024) has attained significant attention for its potential to enhance the generalization of machine learning models, *in a direct optimization manner*.

SAM seeks to minimize the maximum training loss within a predefined neighborhood around the parameters, thereby promoting flatter minima and better generalization. Its effectiveness is evidenced by empirical successes across various domains (Bahri et al., 2021; Rangwani et al., 2022b;a; Fan et al., 2025). However, its practical implementation approximates this as: carry out one or a few steps of gradient ascent, and then apply the gradient from the ascent point to update the current parameters.

Though being justified as approximately optimizing the objective by neglecting the Jacobian matrix of the ascent point with respect to the current parameters (Foret et al., 2020) and a body of research (Wen et al., 2023; Bartlett et al., 2023; Andriushchenko et al., 2023a; Andriushchenko & Flammarion, 2022; Andriushchenko et al., 2023b) have sought to demystify the underlying mechanism of SAM after such approximations, a direct and intuitive understanding of why applying the *nonlocal* gradient at

the ascent point to update the current parameter works superiorly is still lacking. This gap necessitates a deeper investigation into SAM's fundamental mechanisms, which motivates our work.

Common misinterpretation. A prevalent misunderstanding must be clarified before we proceed: applying the gradient at the estimated maximum point DOES NOT necessarily lead to the minimization of the maximum loss within the local neighborhood. *The key here is that there is a shift in location: the gradient is computed at the estimated maximum point, but applied to the current parameters.* The nuisance can be clear on considering the extreme case: the gradient at a point arbitrarily distant from the current parameters provides vanishingly little information about the local loss geometry.

To unravel the mystery of the SAM update, we commence by visualizing the local loss surface during SAM training. As shown in Figure 1a and further illustrated in Appendix A, our visualization analysis reveals that the gradient at the single-step ascent point, when applied to the current parameters, generally provides a better approximation of the direction from the current parameters toward the maximum within the local neighborhood than the gradient at the current parameters. This indicates that applying the gradient at the single-step ascent point to the current parameters enables a more direct escape from the maximum within the local neighborhood, thereby more effectively reducing the worst-case loss in the neighborhood and ultimately leading to better generalization.

The above interpretation rationalizes the application of the gradient at the single-step ascent point to the current parameters. Nevertheless, our visualizations simultaneously reveal that the approximation by the gradient at the single-step ascent point is often inaccurate (as exemplified in Figure 1a), and the approximation quality is unstable, exhibiting large variations during training as the local loss landscape evolves (evidenced by further visualizations in Appendix A). Moreover, as illustrated by Figure 1b (and Figure 10 in Appendix A), the approximation quality may get worse as the number of ascent steps increases, explaining the unexpectedly inferior performance of multi-step SAM.

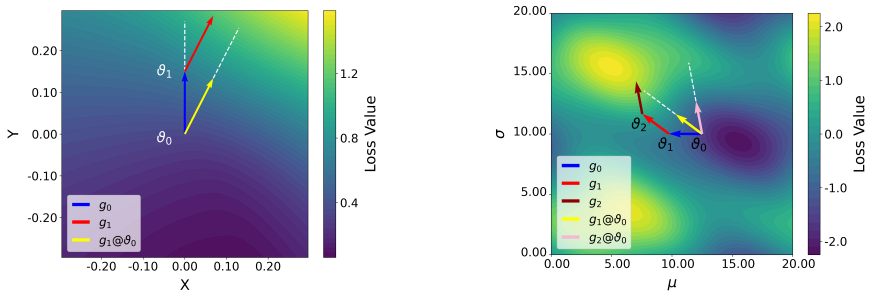
Motivated by these observations, we propose in this paper eXplicit Sharpness-Aware Minimization (XSAM), which addresses the approximation inaccuracy issue of the SAM gradient fundamentally by explicitly estimating the direction from the current parameters toward the maximum via probing the loss values in different directions at the neighborhood boundary, while ensuring its high quality throughout the training by updating the estimation dynamically during training.

Probing the entire high-dimensional neighborhood for estimating the direction can be computationally intractable. We therefore constrain the probe to a two-dimensional hyperplane spanned by the gradient at the final ascent point (i.e., the point reached after $k \geq 1$ ascent steps) and the vector from the current parameters to that point. This definition is crucial. It ensures that the point with the highest known loss, i.e., the one pointed to by the gradient at the final ascent point, lies within the hyperplane. Such a definition also simultaneously addresses the inaccuracy issue of directly applying the gradient at the multi-step ascent point to the current parameters, while fully leveraging its informational value.

We express the estimated direction in terms of the spherical interpolation factor of the two spanning vectors, which, according to our experiments, changes slowly during training, therefore requiring only infrequent updates and incurring negligible additional computational overhead. With this improved estimate of the direction toward the maximum, XSAM escapes the nearby high-loss regions more effectively, thereby achieving better generalization. Extensive experiments demonstrate that XSAM consistently outperforms existing counterparts across various models, datasets, and settings.

The primary contributions of this work are threefold:

- We provide a novel, intuitive interpretation of the fundamental mechanism of SAM, demonstrating that the gradient at the (single-step) ascent point offers a superior approximation of the direction from the current parameter toward the maximum within the local neighborhood than the local gradient, thereby enabling a more direct escape from the maximum within the local neighborhood.
- Our analysis further reveals that the approximation by the gradient at the single-step ascent point is often inaccurate, and its quality varies largely during training. Moreover, the approximation quality may degrade as the number of ascent steps increases, explaining the inferior performance of multi-step SAM. These collectively demonstrate the sub-optimality of the SAM gradient.
- We propose XSAM, which simultaneously addresses all these limitations of SAM by explicitly estimating the direction from the current parameter toward the maximum, within a novel, principled search space during training, leading to a more faithful and effective implementation of sharpness-aware minimization. Extensive experiments demonstrate the consistent superiority of XSAM.



(a) Visualization of single-step SAM (b) Simulation of multi-step SAM

Figure 1: (a) Visualization of the local loss surface of single-step SAM[1] on the hyperplane spanned by the gradient g_0 at the current parameter ϑ_0 and the gradient g_1 at the single-step ascent point ϑ_1. ϑ_0 is set as the origin, the Y-axis is defined along the direction of g_0, and the X-axis is aligned with the component of g_1 perpendicular to g_0. The visualized arrows of gradients are set to have length ρ. **We see that $g_1@\vartheta_0$ (i.e., g_1 applied to ϑ_0) points clearly closer to the direction from ϑ_0 toward the maximum within the local neighborhood (which is roughly from the origin to the upper-right corner) than g_0.** The loss along $g_1@\vartheta_0$ (i.e., $L(\vartheta_0 + \rho_m \cdot g_1/\|g_1\|)$) is higher than that along g_0 (i.e., $L(\vartheta_0 + \rho_m \cdot g_0/\|g_0\|)$), for sufficiently large ρ_m. (b) A simulation of multi-step SAM on a 2D test function. The approximation quality by the SAM gradient may get worse as the number of ascent steps increases: $g_2@\vartheta_0$ inferiorly identifies the direction from ϑ_0 toward the maximum within the local neighborhood (the upper-left high-loss region in yellow) than $g_1@\vartheta_0$.

2 Revisiting Sharpness-Aware Minimization

This section reviews the objective of Sharpness-Aware Minimization (SAM) and its classical approximate optimization method, followed by our novel interpretation of its underlying mechanism.

2.1 The Objective and Classical Approximation of SAM

SAM (Foret et al., 2020) aims to find parameters that minimize the maximum training loss (i.e., worst-case loss) over a predefined ρ-neighborhood around the parameters. The formal objective is:

$$\min_{\theta} \max_{\|\delta\| \le \rho} L(\theta + \delta), \tag{1}$$

where L is the training loss, $\theta \in \mathbb{R}^n$ is the model parameters, and $\delta \in \mathbb{R}^n$ is the perturbation vector.[2]

Since exactly solving the inner maximization in Equation (1) is computationally expensive, SAM approximates it by performing one or a few steps of gradient ascent from the current parameters.

Assuming the procedure involves $k \ge 1$ successive gradient ascent steps, it proceeds as follows: initialize $\vartheta_0 = \theta$, and then for each step $i = 0, 1, \ldots, k-1$:

1) Compute the gradient at the current point ϑ_i: $g_i = \nabla_{\vartheta_i} L(\vartheta_i)$;
2) Ascend along the direction of g_i by a distance of ρ_i: $\vartheta_{i+1} = \vartheta_i + \rho_i \frac{g_i}{\|g_i\|}$.

This formulation unifies the single-step ($k = 1$) and multi-step ($k > 1$) settings, with the constraint $\sum_{i=0}^{k-1} \rho_i \le \rho$ ensuring the total perturbation remains within the ρ-ball. The procedure yields the final perturbed parameters directly as ϑ_k, while approximating the best perturbation δ^* as $\vartheta_k - \vartheta_0$.

After such approximation of the best perturbation, the SAM objective in Equation (1) reduces to:

$$\min_{\theta} L(\theta + \delta^*), \qquad \text{or equivalently,} \qquad \min_{\theta} L(\vartheta_k). \tag{2}$$

To optimize this objective efficiently, SAM employs a key approximation. It assumes $\nabla_\theta \, \delta^* = \mathbf{0}$, or equivalently, $\nabla_\theta \, \vartheta_k = I$, thereby avoiding involving expensive higher-order derivatives. Formally,

$$\nabla_\theta L(\theta + \delta^*) = \nabla_\theta L(\vartheta_k) = \nabla_{\vartheta_k} L(\vartheta_k) \cdot \underbrace{\nabla_\theta(\vartheta_k)}_{\text{Approximated as identity matrix } I} \approx \nabla_{\vartheta_k} L(\vartheta_k). \tag{3}$$

[1] Data is collected at the first iteration of the 150th epoch in training ResNet-18 on CIFAR-100.

[2] For simplicity, we default all norms to ℓ_2.

The resulting algorithm essentially applies the gradient at the final ascent point ϑ_k to θ:

$$\theta_{t+1} = \theta_t - \eta_t \cdot \nabla_{\vartheta_k} L(\vartheta_k). \tag{4}$$

2.2 A Novel Interpretation of SAM's Underlying Mechanism

Despite the key approximation in the classical SAM algorithm being justified as assuming $\nabla_\theta \vartheta_k = I$, it leads to an unusual gradient operation, applying the gradient at another point (ϑ_k) to the current parameters (θ). It is apparent that applying the gradient at an arbitrarily distant point to the current parameters makes no sense, since it brings vanishingly little information about the local loss geometry around the current parameters. This contradiction raises a fundamental question: How is ϑ_k special? Why does applying this nonlocal gradient tend to outperform the local gradient in practice?

While a body of literature has sought to explain how SAM works after such approximation (Wen et al., 2023; Bartlett et al., 2023; Andriushchenko et al., 2023a;b), they often attribute it to implicit bias or regularization. None of them directly addresses our core inquiry: the underlying mechanism that enables this specific nonlocal gradient operation to be effective, which is the focus of this work.

2.2.1 Empirical Analysis through Visualizations.

To unravel the underlying mechanism, we start by visualizing the gradients at the ascent point on the local loss surface during SAM training. For a tractable analysis and a clear comparison between the gradient at the ascent point and the gradient at the current parameters, we focus on the loss surface over the hyperplane spanned by these two gradient vectors and begin with the single-step setting.

Better Approximation. As depicted in Figure 1a (and more visualizations in Appendix A), the gradient at the single-step ascent point, when applied to the current parameters, can better approximate the direction from the current parameters toward the maximum within the local neighborhood than the gradient at the current parameters (i.e., the local gradient). More specifically, we see in the figure that $g_1@\vartheta_0$ points clearly closer to the high-loss region around the upper-right corner than g_0, and the loss value along $g_1@\vartheta_0$ is also literally higher. This phenomenon is consistently observed in practice.

Inaccuracy and Instability. Although $g_1@\vartheta_0$ provides a better approximation than g_0, we can clearly see in Figure 1a (and more in Appendix A) that the approximation by $g_1@\vartheta_0$ can be rough and inaccurate. In fact, according to the additional visualizations in Appendix A, the approximation quality by $g_1@\vartheta_0$ is unstable during training, exhibiting large variations, which suggests that such an approximation by $g_1@\vartheta_0$ can not well adapt to the evolving local loss landscape.

Multi-Step Degradation. We further extend the visualization analysis to multi-step settings. To approximate the complexity of high-dimensional landscapes, where multi-step ascent gradients deviate from a 2D plane, we simulate the process on a suitably complex 2D test function. As shown in Figure 1b, the gradient at the multi-step ascent point, when applied to the current parameters, may act as an unexpectedly poorer approximation (toward the maximum within the local neighborhood) compared to the gradient at the single-step ascent point: $g_2@\vartheta_0$ inferiorly indicates the nearby high-loss region for ϑ_0 than $g_1@\vartheta_0$. Notably, g_2 at its original position ϑ_2 indeed points toward the nearby high-loss region; however, when it is applied to ϑ_0, the resulting vector $g_2@\vartheta_0$ points toward a relatively flat region. This offers a visual explanation for why multi-step SAM does not work as well as expected (Foret et al., 2020; Andriushchenko & Flammarion, 2022). Additional simulation results supporting this finding are included in Appendix A.

2.2.2 Theoretical Confirmation under Second-Order Approximation.

In this section, we substantiate our core empirical observations with the following results:

Proposition 1. *Let $L : \mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable function that admits a second-order approximation at ϑ_0 with:*

- $\nabla L(\vartheta_0) = g_0$*, which does not equal to 0;*
- $\nabla L\left(\vartheta_0 + \rho \frac{g_0}{\|g_0\|}\right) = g_1$*, which is not parallel to g_0;*
- *Hessian $H = \nabla^2 L(\vartheta_0)$ positive definite.*

Then there exists $\rho_0 > 0$ such that for all $\rho_m > \rho_0$:

1) *SAM better approximates the direction toward the maximum in the vicinity than SGD*

$$L\left(\vartheta_0+\rho_m\frac{g_1}{\|g_1\|}\right)>L\left(\vartheta_0+\rho_m\frac{g_0}{\|g_0\|}\right);$$

2) *There exist better approximations than SAM* *there exists* $\alpha\in\mathbb{R}$ *such that*

$$L\left(\vartheta_0+\rho_m\frac{g_\alpha}{\|g_\alpha\|}\right)>L\left(\vartheta_0+\rho_m\frac{g_1}{\|g_1\|}\right),\quad g_\alpha=\alpha g_1+(1-\alpha)g_0.$$

The first result in the proposition delivers that for any fixed distance that is sufficiently large, the loss along the direction of the gradient at the single-step ascent point is higher than that along the gradient at the current parameters, which confirms from the loss-value perspective that the gradient of single-step SAM better approximates the direction toward the maximum within its local neighborhood than that of SGD. Note that being sufficiently large is necessarily required since for a distance that is too small, g_0 is by definition the steepest ascent direction. A detailed proof is provided in Appendix B.

The second result in the proposition implies that there exist better approximations than the gradient of single-step SAM even in the two-dimensional hyperplane spanned by g_0 and g_1, confirming our observation that the approximation by the gradient at the single-step ascent point is inaccurate.

2.2.3 Heuristic Explanation and Deductive Analysis

To help establish a more intuitive understanding of why $g_1@\vartheta_0$ provides a better approximation for the maximum within the local neighborhood, we further provide the following heuristic explanation: Assuming the Hessian matrix of the loss function exhibits sufficiently slow variation within the local neighborhood, i.e., the gradient field evolves smoothly. Then, if g_1 is not parallel to g_0, the directional change from g_0 to g_1 reveals how the gradient field evolves in the surroundings. Then, considering additional *virtual* ascent steps within the local region, e.g., ϑ_2 and g_2. The directional change from g_1 to g_2 will tend to follow a similar trend as that from g_0 to g_1. The same pattern persists for all subsequent virtual ascent steps, i.e., the virtual ascent trajectory will tend to curve in a consistent manner. Therefore, the high-loss region within the local neighborhood identified by the virtual ascent trajectory will likely be located at a position that is further shifted from the one-step ascent point ϑ_1 along the direction g_1 but curves further in the evolving direction of the gradient. Its direction relative to ϑ_0 can thus be better captured by $g_1@\vartheta_0$ when compared with g_0. Nevertheless, such an approximation is inherently inaccurate.

In multi-step settings, a crucial observation is that each adjacent pair of steps $(i, i+1)$ recapitulates the configuration of single-step SAM. Consequently, the conclusion from the single-step analysis holds inductively for each step. That is, $g_{i+1}@\vartheta_i$ better approximates the direction toward the maximum than $g_i@\vartheta_i$, for $i\in[0,\dots,k-1]$. However, a critical discrepancy arises in multi-step SAM: it directly applies g_k to ϑ_0, but it remains unclear whether $g_k@\vartheta_0$ stands as a better approximation of the direction from ϑ_0 toward the maximum than $g_1@\vartheta_0$ (or even g_0). The core difference here is that g_1 is evaluated along the ray defined by g_0 and ϑ_0, whereas g_k may substantially deviate from the ray defined by g_0 and ϑ_0. Because the entire multi-step trajectory can curve significantly. This renders the direct application of g_k to ϑ_0 potentially suboptimal or unjustified.

As a final remark, a simple deduction reveals the inherent inaccuracy of the SAM gradient approximation: Consider SAM operating on a fixed loss surface. Regardless of how accurately $g_k@\vartheta_0$ currently approximates the direction, as long as we continuously decrease $\{\rho_i\}$ (for all $i\in[0,k-1]$) toward 0, g_k will reduce to g_0. Consequently, the approximation quality of $g_k@\vartheta_0$ will get reduced arbitrarily close to that of the original gradient g_0. This sensitivity to the choice of $\{\rho_i\}$ also implies that, for an arbitrary $\{\rho_i\}$, it is typically suboptimal (even for a certain fixed loss surface). ▷ On the other hand, we can also tune $\{\rho_i\}$ to make it the best possible approximation, which could have played a role in the practical effectiveness of SAM. Nevertheless, given the evolving local loss landscape during training, the approximation with any fixed $\{\rho_i\}$ can hardly remain relatively accurate throughout.

3 Explicit Sharpness-Aware Minimization

As shown in the above section, the approximation (of the direction from the current parameters toward the maximum within the local neighborhood) by the SAM gradient is often inaccurate and lacks

Algorithm 1 XSAM

Input: Initial parameters θ_0, number of iterations T, number of ascent steps $k \geq 1$, perturbation radius $\{\rho_i\}$, neighborhood radius ρ_m, α^* update frequency T_α, learning rate $\{\eta_t\}$

Output: Final parameters θ_T

for $t = 0$ **to** $T-1$ **do**
  $\vartheta_0 = \theta_t$
  for $i = 0$ **to** $k-1$ **do** ▷ *Single-step:* $k = 1$
    $g_i = \nabla_{\vartheta_i} L(\vartheta_i)$
    $\vartheta_{i+1} = \vartheta_i + \rho_i \frac{g_i}{\|g_i\|}$
  end for
  $g_k = \nabla_{\vartheta_k} L(\vartheta_k)$
  $v_0 = \frac{\vartheta_k - \vartheta_0}{\|\vartheta_k - \vartheta_0\|}, \quad v_1 = \frac{g_k}{\|g_k\|}$
  $\psi = \arccos(v_0 \cdot v_1)$
  if $t \bmod T_\alpha = 0$ **then**
    $\alpha_t^* = \arg\max_\alpha L(\vartheta_0 + \rho_m \cdot v(\alpha))$
  else
    $\alpha_t^* = \alpha_{t-1}^*$
  end if
  $v(\alpha_t^*) = \frac{\sin((1-\alpha_t^*)\psi)}{\sin(\psi)} v_0 + \frac{\sin(\alpha_t^*\psi)}{\sin(\psi)} v_1$
  $\theta_{t+1} = \theta_t - \eta_t \cdot v(\alpha^*) \cdot \|g_k\|$
end for

Table 1: Training time comparison. Values are presented as hours/200 epochs, SAM / XSAM.

	CIFAR-10	CIFAR-100	Tiny-ImageNet
VGG-11	0.93 / 0.96	0.98 / 1.03	2.18 / 2.22
ResNet-18	2.35 / 2.39	2.40 / 2.43	4.95 / 4.98
DenseNet-121	8.02 / 8.08	8.05 / 8.07	16.50 / 16.55

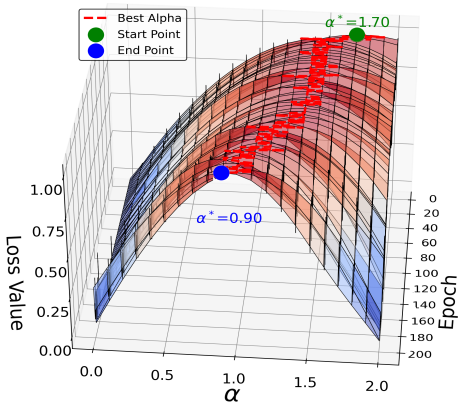


Figure 2: Slow variation of α^* during training.

adaptivity to the evolving local loss landscape. Moreover, the approximation quality may degrade as the number of ascent steps increases. To provide an integrated solution that simultaneously addresses all these limitations, we propose in this section eXplicit Sharpness-Aware Minimization (XSAM).

XSAM addresses the inaccuracy issue through explicitly probing the location of the maximum within the local neighborhood (which thereby gives the direction) and enhances adaptivity to the evolving local loss landscape by dynamically performing this probe during training.

Since probing the maximum within the entire high-dimensional neighborhood can be computationally intractable. We therefore assume the maximum is located at the neighborhood boundary, while further constraining the probe to a two-dimensional hyperplane spanned by the gradient at the final ascent point (i.e., the point reached after $k \geq 1$ ascent steps) and the vector from the current parameters to that point. Formally, the two spanning vectors are defined as:

$$v_0 = \frac{\vartheta_k - \vartheta_0}{\|\vartheta_k - \vartheta_0\|}, \qquad v_1 = \frac{g_k}{\|g_k\|}. \tag{5}$$

This definition of the two-dimensional hyperplane is crucial. It ensures that the point with the highest known loss (the one pointed to by g_k, standing at ϑ_k) lies within the hyperplane. It also simultaneously addresses the inaccuracy issue of directly applying the gradient at the multi-step ascent point to the current parameters, while fully leveraging its informational value: we use ϑ_k and g_k to define a search space that encompasses all the information they contain, instead of directly applying g_k to ϑ_0. This definition further offers a unified formulation for both single-step and multi-step settings. Note that when $k = 1$, v_0 and v_1 correspond to the directions of g_0 and g_1, respectively. Normalization is applied to separate direction from magnitude, as we intend to manage them independently.

To probe within the two-dimensional hyperplane, we generate new directions as the spherical linear interpolation between v_0 and v_1:

$$v(\alpha) = \frac{\sin((1-\alpha)\psi)}{\sin(\psi)} v_0 + \frac{\sin(\alpha\psi)}{\sin(\psi)} v_1, \tag{6}$$

where $\psi = \arccos(v_0 \cdot v_1)$ and α is the interpolation factor. It has $\|v(\alpha)\| = 1$ for any α, $v(0) = v_0$, $v(1) = v_1$, and more generally, $v(\alpha)$ is a unit vector that rotates from v_0 by an angle of $\alpha \cdot \psi$ along the direction toward v_1. It can span all possible directions in the search space.

We then determine the direction, parametrized by α^*, that maximizes the loss at a predefined distance:

$$\alpha^* = \arg\max_{\alpha \in [0,a]} L(\vartheta_0 + \rho_m \cdot v(\alpha)), \tag{7}$$

where ρ_m is a hyperparameter specifying the radius of the *true* (in contrast to the perturbation radius) sharpness-aware neighborhood. In each dynamic search, we uniformly sample α values from $[0, a]$. In practice, setting a to 2 or 4 and sampling 20–40 samples is typically sufficient.

Once α^* is identified, the model parameters are updated using $-v(\alpha^*)$ as the descent direction. The gradient scale, by default, is set to $\|g_k\|$ to make it consistent with SAM[3]. Formally,

$$\theta_{t+1} = \theta_t - \eta_t \cdot v(\alpha^*) \cdot \|g_k\|, \tag{8}$$

by which $v(\alpha^*)$ steers the parameters away from the estimated maximum within the neighborhood.

Faithfulness and Effectiveness. Since we use $L(\vartheta_0 + \rho_m \cdot v(\alpha))$ as a proxy[4], the method explicitly identifies the maximum within a neighborhood of radius ρ_m. Although restricted to a hyperplane, this approximation relies only on the boundary assumption. It thus more faithfully identifies the maximum in the local neighborhood (in contrast to directly regarding ϑ_k as the maximum or approximating its direction by $g_k @ \vartheta_0$) and consequently more effectively realizes the sharpness-aware minimization.

The Cost of Explicit Estimation. The evaluation of each α requires a forward pass. Thus, the cost of explicit estimation scales with the number of sampled α values times the cost of a forward pass. If performed at every iteration, this would introduce substantial overhead. Fortunately, frequent updates of α^* are unnecessary. Our experiments show that α^* remains relatively stable and varies smoothly during training (Figure 2). By default, we adopt an epoch-wise update strategy: α^* is updated at the first iteration of each epoch and then fixed for the remainder. Runtime comparison is shown in Table 1, indicating the additional overhead is negligible. Further details are provided in Appendix C.

4 Related Work

SAM has been extended in several distinct directions. One line of work focuses on improving the gradient ascent (i.e., perturbation) step, addressing issues such as parameter scale dependence (ASAM (Kwon et al., 2021); Fisher SAM (Kim et al., 2022)), approximation quality (RSAM (Liu et al., 2022b); CR-SAM (Wu et al., 2024)), and perturbation stability (VaSSO (Li & Giannakis, 2024); FSAM (Li et al., 2024)). These approaches are largely complementary to ours; for instance, Appendix E.1 demonstrates that integrating XSAM with ASAM yields additional performance gains.

Another line of research targets the parameter update step. WSAM (Yue et al., 2023) and Zhao et al. (2022a) derive their update rules as a linear combination of g_0 and g_1 through weighted sharpness regularization and gradient-norm penalization, respectively. While their superior performance over SAM is readily explained by our interpretation, this very perspective reveals a critical weakness: their dependence on a fixed combination weight, treated as a hyperparameter, is inherently suboptimal. In contrast, XSAM explicitly estimates the optimal interpolation factor dynamically during training and naturally extends this principle to multi-step settings. More fundamentally, our approach is derived from a reformulation of the sharpness-aware objective itself, rather than introducing an auxiliary regularization term, thereby offering a more general and principled solution.

Multi-step SAM variants are discussed in Section 5.3, while additional related work on topics such as flatness and efficiency is deferred to Appendix G.

5 Empirical Results

In this section, we empirically compare SAM and its related variants with the proposed XSAM. Due to space limitations, detailed experimental settings are deferred to Appendix D.

5.1 2D Test Function

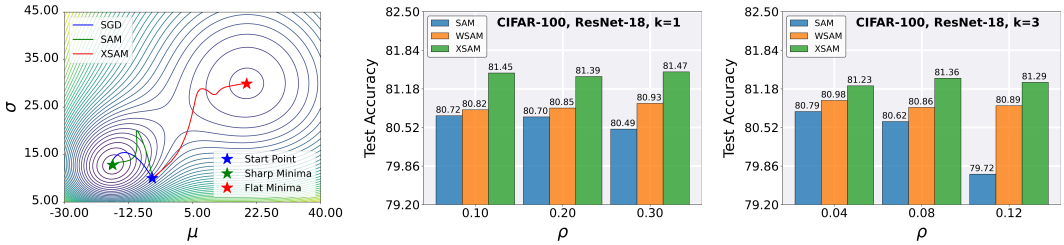
Following (Yue et al., 2023; Kim et al., 2022), we first evaluate methods on a 2D function featuring a sharp and a flat minimum within a certain distance, serving as an ideal testbed for sharpness-aware minimization. We compare SGD, SAM, and XSAM across different initial points and hyperparameters. XSAM consistently converges to the flat minima when ρ_m is sufficiently large, whereas SAM

[3]Alternative gradient scaling strategies are examined in Appendix F.

[4]Our implementation uses only the current batch, consistent with the standard SAM procedure.

Table 2: Test accuracies on classification tasks in the single-step setting.

Dataset	CIFAR-10			CIFAR-100			Tiny-ImageNet		
Model	VGG-11	ResNet-18	DenseNet-121	VGG-11	ResNet-18	DenseNet-121	VGG-11	ResNet-18	DenseNet-121
SGD	$93.19_{\pm 0.11}$	$96.15_{\pm 0.05}$	$96.34_{\pm 0.11}$	$71.46_{\pm 0.17}$	$78.55_{\pm 0.20}$	$81.78_{\pm 0.06}$	$47.44_{\pm 0.33}$	$57.02_{\pm 0.42}$	$61.93_{\pm 0.10}$
SAM	$93.83_{\pm 0.06}$	$96.59_{\pm 0.06}$	$96.97_{\pm 0.02}$	$74.01_{\pm 0.05}$	$80.93_{\pm 0.11}$	$83.81_{\pm 0.02}$	$51.96_{\pm 0.26}$	$62.81_{\pm 0.09}$	$66.31_{\pm 0.09}$
XSAM	$\mathbf{94.25}_{\pm 0.14}$	$\mathbf{96.74}_{\pm 0.04}$	$\mathbf{97.15}_{\pm 0.03}$	$\mathbf{74.21}_{\pm 0.14}$	$\mathbf{81.24}_{\pm 0.07}$	$\mathbf{83.96}_{\pm 0.10}$	$\mathbf{52.58}_{\pm 0.38}$	$\mathbf{63.82}_{\pm 0.23}$	$\mathbf{66.81}_{\pm 0.08}$



(a) (b) (c)

Figure 3: (a) Training trajectory comparisons on 2D test function. (b)-(c) Test accuracy comparisons of ResNet-18 trained on CIFAR-100 in single-step and multi-step ($k = 3$) settings with varying ρ.

and SGD are more prone to get trapped in the sharp minima. Representative training trajectories for each method are shown in Figure 3a. Both SAM and XSAM are evaluated in their single-step form.

5.2 Evaluation under the Single-Step Setting

In this section, we evaluate the methods under the single-step setting across a variety of classification datasets and model architectures. To stress-test the methods, we first tune SAM's learning rate, weight decay, and ρ to achieve its optimal performance on each dataset. Other methods are then tuned using the same hyperparameters whenever feasible. To isolate the effect of different gradient directions and eliminate the influence of gradient scaling, all methods adopt SAM's gradient scale, i.e., $\|g_k\|$.

We evaluate the methods across diverse neural network architectures and datasets to ensure broad applicability. The experiments cover architectures ranging from VGG-11 (Simonyan & Zisserman, 2014) and ResNet-18 (He et al., 2016) to DenseNet-121 (Huang et al., 2017), encompassing classic models of increasing capacity, as well as datasets including CIFAR-10, CIFAR-100, and Tiny-ImageNet, which span increasing complexities. As shown in Table 2, SAM consistently outperforms SGD, confirming the superiority of the gradient direction of g_1 compared to g_0. Meanwhile, XSAM consistently outperforms SAM, highlighting the benefit of explicitly estimating the direction.

To provide a more thorough comparison, we evaluate performance under varying ρ on CIFAR-100 using ResNet-18. For this experiment, we further include a WSAM-like baseline, which implements our method with a fixed but tunable α, to highlight the benefit of dynamically estimating α compared to a static choice. The best fixed α for the WSAM is determined via grid search over $[-1.0, 3.0]$ with a step size of 0.25. As shown in Figure 3b, the WSAM improves over SAM, while XSAM consistently achieves further and significant improvements over the WSAM.

Having established XSAM's potent performance under varying ρ, we further assess XSAM's generality on larger-scale and more diverse tasks. We conduct experiments on ImageNet with ResNet-50, a neural machine translation task with a Transformer (Vaswani et al., 2017), and CIFAR-100 with ViT-Ti (Dosovitskiy et al., 2020). The results in Table 3 show that XSAM consistently outperforms SAM, demonstrating its broad applicability across diverse tasks and models.

5.3 Evaluation under the Multi-Step Setting

We proceed to evaluate and compare methods in a multi-step setting. We use a constant perturbation magnitude ρ for all steps (i.e., $\rho_i = \rho$ for all i), therefore omitting the subscript i for clarity. All experiments in this section are conducted on CIFAR-100 using a ResNet-18.

Table 3: Comparison of SAM and XSAM on larger-scale and more diverse tasks.

	ImageNet ResNet-50 (Accuracy)	Transformer IWSLT2014 (BLEU)	ViT-Ti CIFAR-100 (Accuracy)
SAM	77.04 ± 0.09	35.30 ± 0.04	67.80 ± 0.22
XSAM	$\mathbf{77.22 \pm 0.07}$	$\mathbf{35.63 \pm 0.12}$	$\mathbf{68.32 \pm 0.18}$

Table 4: Multi-step results on CIFAR-100 with ResNet-18. $\rho = \rho^*/k$ with ρ^* for single-step.

Methods	$k=1$	$k=2$	$k=4$
SAM	80.93 ± 0.11	80.91 ± 0.10	80.65 ± 0.26
LSAM	80.93 ± 0.11	80.94 ± 0.09	80.74 ± 0.18
LSAM+	80.61 ± 0.20	80.83 ± 0.11	80.41 ± 0.03
MSAM	80.93 ± 0.11	81.18 ± 0.06	81.01 ± 0.09
MSAM+	80.83 ± 0.05	80.86 ± 0.34	80.77 ± 0.08
XSAM	$\mathbf{81.27 \pm 0.07}$	$\mathbf{81.44 \pm 0.09}$	$\mathbf{81.37 \pm 0.24}$

As the first experiment, we compare XSAM with multi-step SAM variants across different values of k. The considered methods include: MSAM (Kim et al., 2023), which updates parameters with $\sum_{i=1}^{k} g_i$, and LSAM (Mordido et al., 2024), which employs $\sum_{i=1}^{k} g_i/\|g_i\|$. To ensure a thorough comparison, we further introduce two augmented variants that incorporate the initial gradient g_0: MSAM+ ($\sum_{i=0}^{k} g_i$) and LSAM+ ($\sum_{i=0}^{k} g_i/\|g_i\|$). Consistent with our previous protocol, we isolate the effect of gradient direction by readjusting all gradients to have the norm $\|g_k\|$. The perturbation radius is set to $\rho = \rho^*/k$, where ρ^* is the optimal value for single-step SAM, as suggested by Kim et al. (2023); all other hyperparameters remain unchanged from the single-step setup.

As shown in Table 4, the performance of SAM tends to decline as k increases. This phenomenon can be attributed to the growing deviation of g_k from the original ascent direction $g_0@\vartheta_0$ as the single ascent step is subdivided, leading to a poorer approximation of the direction toward the maximum in the vicinity when applied to ϑ_0. In contrast, XSAM is not affected by this issue and typically benefits from more steps, demonstrating its superior ability to leverage multi-step ascent.

LSAM and MSAM, which incorporate intermediate ascent gradients (g_i for $0 \leq i \leq k$), generally surpass SAM. The decline in SAM's performance with large k suggests substantial deviation of g_k from the ideal direction, which makes earlier, less-deviated gradients g_i valuable. Notably, LSAM+, which essentially moves away directly from the identified maximum point by multi-step ascent, even underperforms SAM, highlighting the value of an extra explicit estimation of the direction toward the maximum. Nevertheless, XSAM consistently outperforms all these methods across all settings.

We further evaluate SAM and XSAM under a multi-step setting ($k = 3$) with a varying perturbation radius ρ. A multi-step extension of WSAM, which combines the gradients g_k and g_0 with a fixed interpolation factor, is also compared. The results in Figure 3c indicate that while the WSAM variant outperforms SAM, XSAM consistently outperforms WSAM.

Figure 4 shows the robustness of XSAM to the α^* update frequencies. We observe no consistent pattern in performance when varying the update frequency of α^*. Additional ablation results are presented in Appendix E.3. Appendix E.4 further visualizes the loss surface at convergence, illustrating that XSAM finds **flatter minima** than SAM.

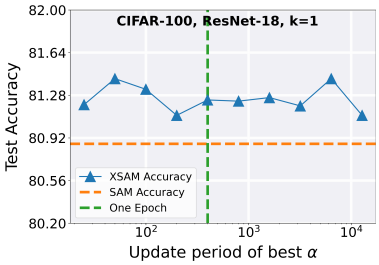


Figure 4: XSAM robustness to the α^* update frequency.

6 Conclusion

We have studied in this paper the underlying mechanism of SAM, providing a novel, intuitive explanation of why it is valid and effective to apply the gradient at the ascent point to the current parameters. We have shown that the SAM gradient in its single-step version can provably better approximate the direction from the current parameters toward the maximum within the local neighborhood than that of SGD; however, such an approximation can be inaccurate, lacks adaptivity to the evolving local loss landscape during training, and may degrade as the number of ascent steps increases. We have proposed XSAM that explicitly estimates the direction (from the current parameters) toward the maximum within the local neighborhood dynamically during training, thereby more faithfully and effectively moving the current parameters away from it. Extensive experiments across various models, datasets, tasks, and settings have demonstrated the effectiveness of XSAM.

Reproducibility Statement

We have provided the code as supplementary material, along with detailed instructions for reproducing our experiments. The experimental settings and hyperparameters are described in the Appendix D. The datasets used in this paper are publicly available and can be downloaded online. Detailed proofs of the proposed proposition are included in the Appendix B.

References

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

Maksym Andriushchenko, Dara Bahri, Hossein Mobahi, and Nicolas Flammarion. Sharpness-aware minimization leads to low-rank features. *Advances in Neural Information Processing Systems*, 36: 47032–47051, 2023a.

Maksym Andriushchenko, Francesco Croce, Maximilian Müller, Matthias Hein, and Nicolas Flammarion. A modern look at the relationship between sharpness and generalization, 2023b. URL https://arxiv.org/abs/2302.07011.

Dara Bahri, Hossein Mobahi, and Yi Tay. Sharpness-aware minimization improves language model generalization. *arXiv preprint arXiv:2110.08529*, 2021.

Peter L Bartlett, Philip M Long, and Olivier Bousquet. The dynamics of sharpness-aware minimization: Bouncing across ravines and drifting towards wide minima. *Journal of Machine Learning Research*, 24(316):1–36, 2023.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. Report on the 11th iwslt evaluation campaign. In *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*, pp. 2–17, 2014.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys, 2017. URL https://arxiv.org/abs/1611.01838.

Terrance DeVries. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

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 YF Tan. Efficient sharpness-aware minimization for improved training of neural networks. *arXiv preprint arXiv:2110.03141*, 2021.

Chongyu Fan, Jinghan Jia, Yihua Zhang, Anil Ramakrishna, Mingyi Hong, and Sijia Liu. Towards llm unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond, 2025. URL https://arxiv.org/abs/2502.05374.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2020.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pp. 2232–2241. PMLR, 2019.

Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. *Advances in neural information processing systems*, 7, 1994.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho, and Krzysztof Geras. The break-even point on optimization trajectories of deep neural networks, 2020. URL https://arxiv.org/abs/2002.09572.

Weisen Jiang, Hansi Yang, Yu Zhang, and James Kwok. An adaptive policy to employ sharpness-aware minimization. *arXiv preprint arXiv:2304.14647*, 2023.

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

Ryo Karakida, Tomoumi Takase, Tomohiro Hayase, and Kazuki Osawa. Understanding gradient regularization in deep learning: Efficient finite-difference computation and implicit bias. In *International Conference on Machine Learning*, pp. 15809–15827. PMLR, 2023.

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

Hoki Kim, Jinseong Park, Yujin Choi, Woojin Lee, and Jaewook Lee. Exploring the effect of multi-step ascent in sharpness-aware minimization. *arXiv preprint arXiv:2302.10181*, 2023.

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

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

Bingcong Li and Georgios Giannakis. Enhancing sharpness-aware optimization through variance suppression. *Advances in Neural Information Processing Systems*, 36, 2024.

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

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*, pp. 5631–5640, 2024.

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, 2022a.

Yong Liu, Siqi Mai, Minhao Cheng, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Random sharpness-aware minimization. *Advances in Neural Information Processing Systems*, 35:24543–24556, 2022b.

Chao Ma and Lexing Ying. On linear stability of sgd and input-smoothness of neural networks. *Advances in Neural Information Processing Systems*, 34:16805–16817, 2021.

David A McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pp. 164–170, 1999.

Peng Mi, Li Shen, Tianhe Ren, Yiyi Zhou, Xiaoshuai Sun, Rongrong Ji, and Dacheng Tao. Make sharpness-aware minimization stronger: A sparsified perturbation approach. *Advances in Neural Information Processing Systems*, 35:30950–30962, 2022.

Gonçalo Mordido, Pranshu Malviya, Aristide Baratin, and Sarath Chandar. Lookbehind-sam: k steps back, 1 step forward, 2024. URL https://arxiv.org/abs/2307.16704.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

Harsh Rangwani, Sumukh K Aithal, Mayank Mishra, Arihant Jain, and Venkatesh Babu Radhakrishnan. A closer look at smoothness in domain adversarial training. In *International conference on machine learning*, pp. 18378–18399. PMLR, 2022a.

Harsh Rangwani, Sumukh K Aithal, Mayank Mishra, et al. Escaping saddle points for effective generalization on class-imbalanced data. *Advances in Neural Information Processing Systems*, 35: 22791–22805, 2022b.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Behrooz Tahmasebi, Ashkan Soleymani, Dara Bahri, Stefanie Jegelka, and Patrick Jaillet. A universal class of sharpness-aware minimization algorithms. *arXiv preprint arXiv:2406.03682*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

Tao Wu, Tie Luo, and Donald C Wunsch II. Cr-sam: Curvature regularized sharpness-aware minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 6144–6152, 2024.

Yun Yue, Jiadi Jiang, Zhiling Ye, Ning Gao, Yongchao Liu, and Ke Zhang. Sharpness-aware minimization revisited: Weighted sharpness as a regularization term. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3185–3194, 2023.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Yang Zhao, Hao Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. In *International conference on machine learning*, pp. 26982–26992. PMLR, 2022a.

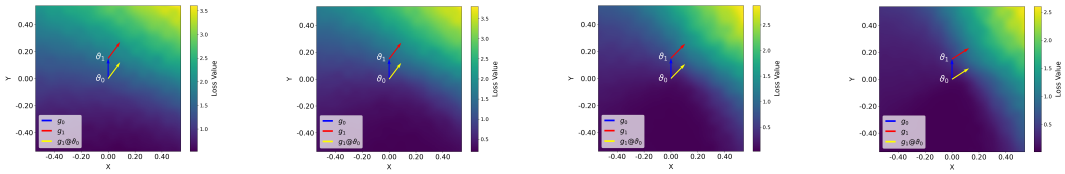
Yang Zhao, Hao Zhang, and Xiuyuan Hu. Randomized sharpness-aware training for boosting computational efficiency in deep learning. *arXiv preprint arXiv:2203.09962*, 2022b.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 13001–13008, 2020.

Zhanpeng Zhou, Mingze Wang, Yuchen Mao, Bingrui Li, and Junchi Yan. Sharpness-aware minimization efficiently selects flatter minima late in training, 2025. URL https://arxiv.org/abs/2410.10373.

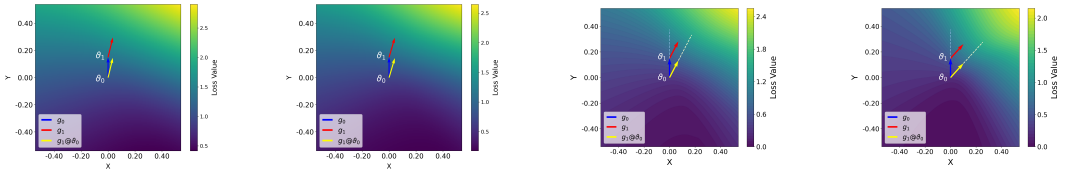
APPENDIX

A VISUALIZATION OF LOSS SURFACE DURING TRAINING



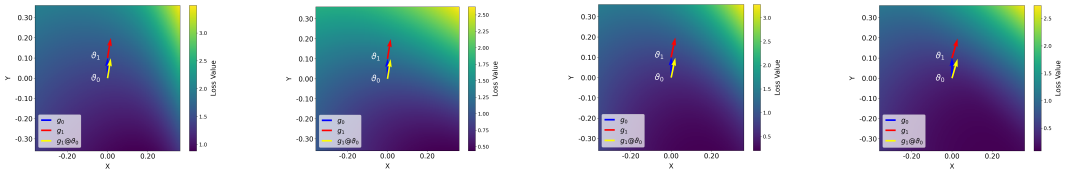
(a) Epoch 50 (b) Epoch 100 (c) Epoch 150 (d) Epoch 200

Figure 5: Visualization of loss surface during training: VGG-11 trained on CIFAR-100.



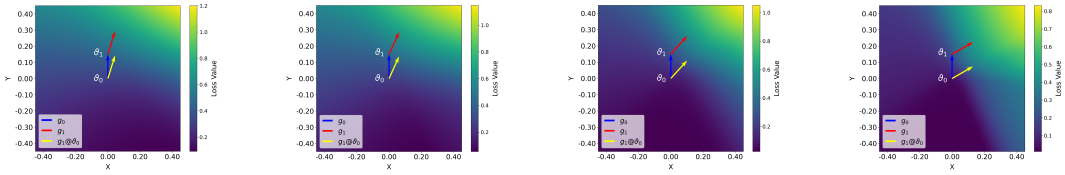
(a) Epoch 50 (b) Epoch 100 (c) Epoch 150 (d) Epoch 200

Figure 6: Visualization of loss surface during training: ResNet-18 trained on CIFAR-100.



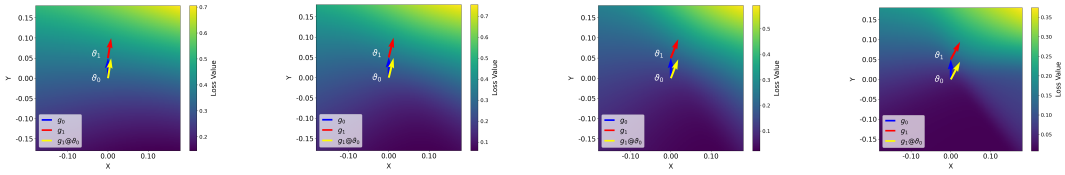
(a) Epoch 50 (b) Epoch 100 (c) Epoch 150 (d) Epoch 200

Figure 7: Visualization of loss surface during training: ViT-Ti trained on CIFAR-100.



(a) Epoch 50 (b) Epoch 100 (c) Epoch 150 (d) Epoch 200

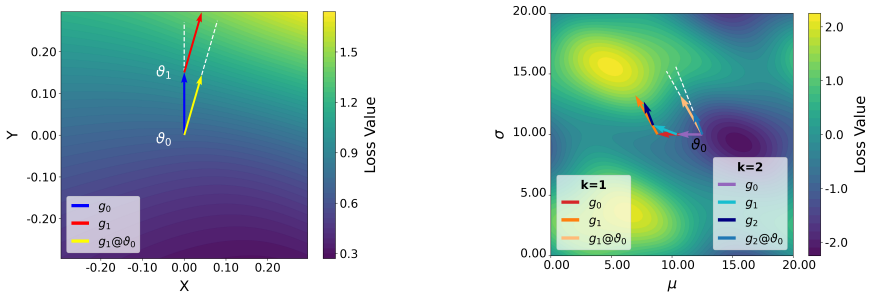
Figure 8: Visualization of loss surface during training: ResNet-18 trained on CIFAR-10.



(a) Epoch 50 (b) Epoch 100 (c) Epoch 150 (d) Epoch 200

Figure 9: Visualization of loss surface during training: DenseNet-121 trained on CIFAR-10.

In this section, we provide more visualizations of the loss surfaces of different datasets and models during SAM training. The results are shown in Figure 5, 6, 7, 8, 9, and 10. The gradient of the ascent



(a) Visualization of single-step SAM

(b) Simulation of multi-step SAM

Figure 10: (a) Visualization of the local loss surface of single-step SAM. The visualization procedure follows the same steps as in Figure 1a. Data is collected at the first iteration of the 100th epoch in training ResNet-18 on CIFAR-100. **We see that $g_1@\vartheta_0$ (i.e., g_1 applied to ϑ_0) points clearly closer to the direction from ϑ_0 toward the maximum within the local neighborhood (which is roughly from the origin to the upper-right corner) than g_0.** The loss along $g_1@\vartheta_0$ (i.e., $L(\vartheta_0 + \rho_m \cdot g_1/\|g_1\|)$) is higher than that along g_0 (i.e., $L(\vartheta_0 + \rho_m \cdot g_0/\|g_0\|)$), for sufficiently large ρ_m. (b) A simulation of multi-step SAM on a test function. The gradient at the multi-step ascent point, when applied to the current parameters, may be an inferior approximation of the direction toward the maximum.

point better approximates the direction toward the maximum within the neighborhood than the local gradient. However, the approximation can often be inaccurate and unstable during training.

B Proofs

Proposition 1. *Let $L : \mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable function that admits a second-order approximation at ϑ_0 with:*

- $\nabla L(\vartheta_0) = g_0$, *which does not equal to 0;*
- $\nabla L\left(\vartheta_0 + \rho \frac{g_0}{\|g_0\|}\right) = g_1$, *which is not parallel to g_0;*
- *Hessian $H = \nabla^2 L(\vartheta_0)$ positive definite.*

Then there exists $\rho_0 > 0$ such that for all $\rho_m > \rho_0$:

1) *SAM better approximates the direction toward the maximum in the vicinity than SGD*

$$L\left(\vartheta_0 + \rho_m \frac{g_1}{\|g_1\|}\right) > L\left(\vartheta_0 + \rho_m \frac{g_0}{\|g_0\|}\right);$$

2) *There exist better approximations than SAM* *there exists $\alpha \in \mathbb{R}$ such that*

$$L\left(\vartheta_0 + \rho_m \frac{g_\alpha}{\|g_\alpha\|}\right) > L\left(\vartheta_0 + \rho_m \frac{g_1}{\|g_1\|}\right), \quad g_\alpha = \alpha g_1 + (1-\alpha) g_0.$$

B.1 Proof of the First Conclusion

Proof.

1. Since L admits a second-order approximation at θ_0:

$$L\left(\vartheta_0 + \rho_m \frac{g_1}{\|g_1\|}\right) = L(\vartheta_0) + \rho_m \frac{g_0^\top g_1}{\|g_1\|} + \frac{\rho_m^2}{2} \frac{g_1^\top H g_1}{\|g_1\|^2} + o(\rho_m^2),$$

$$L\left(\vartheta_0 + \rho_m \frac{g_0}{\|g_0\|}\right) = L(\vartheta_0) + \rho_m \|g_0\| + \frac{\rho_m^2}{2} \frac{g_0^\top H g_0}{\|g_0\|^2} + o(\rho_m^2).$$

2. For sufficiently large ρ_m, the ρ_m^2 term dominates. Thus, we need to show:

$$\frac{g_1^\top H g_1}{\|g_1\|^2} > \frac{g_0^\top H g_0}{\|g_0\|^2}.$$

3. Expand g_1 as the gradient of L (which admits a second-order approximation) at $\vartheta_0 + \rho \frac{g_0}{\|g_0\|}$:

$$g_1 = g_0 + \rho H \frac{g_0}{\|g_0\|} + o(\rho).$$

4. Compute the numerator and denominator to the second order:

$$\begin{aligned} g_1^\top H g_1 &= \left(g_0 + \rho H \frac{g_0}{\|g_0\|} + o(\rho)\right)^\top H \left(g_0 + \rho H \frac{g_0}{\|g_0\|} + o(\rho)\right) \\ &= g_0^\top H g_0 + 2\rho \frac{g_0^\top H^2 g_0}{\|g_0\|} + \rho^2 \frac{g_0^\top H^3 g_0}{\|g_0\|^2} + o(\rho^2), \end{aligned}$$

$$\|g_1\|^2 = \left\| g_0 + \rho H \frac{g_0}{\|g_0\|} + o(\rho) \right\|^2 = \|g_0\|^2 + 2\rho \frac{g_0^\top H g_0}{\|g_0\|} + \rho^2 \frac{g_0^\top H^2 g_0}{\|g_0\|^2} + o(\rho^2).$$

4. Ignoring higher-order terms $o(\rho^2)$, the inequality becomes:

$$\frac{g_0^\top H g_0 + 2\rho \frac{g_0^\top H^2 g_0}{\|g_0\|} + \rho^2 \frac{g_0^\top H^3 g_0}{\|g_0\|^2}}{\|g_0\|^2 + 2\rho \frac{g_0^\top H g_0}{\|g_0\|} + \rho^2 \frac{g_0^\top H^2 g_0}{\|g_0\|^2}} > \frac{g_0^\top H g_0}{\|g_0\|^2}.$$

5. Multiply both sides by the positive denominators (since H is positive definite):

$$\left(g_0^\top H g_0 + 2\rho \frac{g_0^\top H^2 g_0}{\|g_0\|} + \rho^2 \frac{g_0^\top H^3 g_0}{\|g_0\|^2}\right) \|g_0\|^2 > g_0^\top H g_0 \left(\|g_0\|^2 + 2\rho \frac{g_0^\top H g_0}{\|g_0\|} + \rho^2 \frac{g_0^\top H^2 g_0}{\|g_0\|^2}\right).$$

6. Cancel common terms and divide by $\rho > 0$:

$$2\left(\|g_0\| g_0^\top H^2 g_0 - \frac{(g_0^\top H g_0)^2}{\|g_0\|}\right) + \rho \left(g_0^\top H^3 g_0 - \frac{g_0^\top H g_0 g_0^\top H^2 g_0}{\|g_0\|^2}\right) > 0.$$

7. Term verification:

- First term:
$$\|g_0\|^2 g_0^\top H^2 g_0 - (g_0^\top H g_0)^2 > 0.$$
This follows from the strict Cauchy-Schwarz inequality for the inner product, since g_0 and $H g_0$ are not parallel by assumption.

- Second term:
$$\|g_0\|^2 g_0^\top H^3 g_0 - g_0^\top H g_0 g_0^\top H^2 g_0 \geq 0.$$
Let $H = \sum_{i=1}^n \lambda_i v_i v_i^\top$ be the spectral decomposition with $\lambda_i > 0$. Expressing $g_0 = \sum_{i=1}^n \alpha_i v_i$:
$$\|g_0\|^2 g_0^\top H^3 g_0 - g_0^\top H g_0 g_0^\top H^2 g_0 = \left(\sum \alpha_i^2\right)\left(\sum \lambda_i^3 \alpha_i^2\right) - \left(\sum \lambda_i \alpha_i^2\right)\left(\sum \lambda_i^2 \alpha_i^2\right).$$
The nonnegativity follows from Chebyshev's sum inequality applied to the series $\{\lambda_i\}$ and $\{\lambda_i^2\}$.

8. Conclusion:

Since both terms are non-negative and the first is strictly positive, the inequality holds.

For sufficiently large ρ_m, the ρ_m^2 term dominates the Taylor expansion.

That is, $\exists \rho_0 > 0$ such that $\forall \rho_m > \rho_0$:

$$L\left(\vartheta_0 + \rho_m \frac{g_1}{\|g_1\|}\right) > L\left(\vartheta_0 + \rho_m \frac{g_0}{\|g_0\|}\right).$$

□

Remark. *If ρ_m is too small, the first-order term will dominate. The first-order term has*

$$\frac{g_0^\top g_1}{\|g_1\|} = \|g_0\| \cos(\phi) < \|g_0\|,$$

where $\cos\phi = \frac{g_0^T g_1}{\|g_0\|\|g_1\|} < 1$ since g_0 and g_1 do not parallel. Thus, if ρ_m is too small, it will have

$$L\left(\vartheta_0 + \rho_m \frac{g_1}{\|g_1\|}\right) < L\left(\vartheta_0 + \rho_m \frac{g_0}{\|g_0\|}\right).$$

From another perspective, this must hold because g_0 indicates the steepest ascent direction at ϑ_0.

Remark. *ρ_m needs to be large only to ensure that the difference in the second-order term outweighs the first-order term, not intended to be too large to become impractical in real-world applications.*

B.2 Proof of the Second Conclusion

Proof.

1. Since L admits a second-order approximation at θ_0:

$$L\left(\vartheta_0 + \rho_m \frac{g_\alpha}{\|g_\alpha\|}\right) = L(\vartheta_0) + \rho_m \frac{g_0^\top g_\alpha}{\|g_\alpha\|} + \frac{\rho_m^2}{2} \frac{g_\alpha^\top H g_\alpha}{\|g_\alpha\|^2} + o(\rho_m^2),$$
$$L\left(\vartheta_0 + \rho_m \frac{g_1}{\|g_1\|}\right) = L(\vartheta_0) + \rho_m \frac{g_0^\top g_1}{\|g_1\|} + \frac{\rho_m^2}{2} \frac{g_1^\top H g_1}{\|g_1\|^2} + o(\rho_m^2).$$

2. Define the quadratic ratio:

$$f(\alpha) = \frac{g_\alpha^\top H g_\alpha}{\|g_\alpha\|^2}.$$

At boundary points:

$$f(1) = \frac{g_1^\top H g_1}{\|g_1\|^2}, \quad f(0) = \frac{g_0^\top H g_0}{\|g_0\|^2}.$$

3. The derivative is:

$$f'(\alpha) = \frac{2(g_1 - g_0)^\top H g_\alpha \cdot \|g_\alpha\|^2 - 2(g_\alpha^\top H g_\alpha)(g_1 - g_0)^\top g_\alpha}{\|g_\alpha\|^4}.$$

At $\alpha = 1$:

$$f'(1) = \frac{2}{\|g_1\|^4}\left[(g_1 - g_0)^\top H g_1 \cdot \|g_1\|^2 - (g_1^\top H g_1)(g_1 - g_0)^\top g_1\right].$$

4. Using $g_1 = g_0 + \rho H \frac{g_0}{\|g_0\|} + o(\rho)$:

$$g_1 - g_0 = \rho H \frac{g_0}{\|g_0\|} + o(\rho).$$

Substituting into $f'(1)$:

$$f'(1) = \frac{2\rho}{\|g_1\|^4 \|g_0\|}\left[g_0^\top H^2 g_1 \cdot \|g_1\|^2 - (g_1^\top H g_1)(g_0^\top H g_1)\right] + o(\rho).$$

5. Further substituting $g_1 = g_0 + \rho H \frac{g_0}{\|g_0\|} + o(\rho)$ in:

$$\|g_1\|^2 = \left\| g_0 + \rho H \frac{g_0}{\|g_0\|} + o(\rho) \right\|^2 = \|g_0\|^2 + 2\rho \frac{g_0^\top H g_0}{\|g_0\|} + \rho^2 \frac{g_0^\top H^2 g_0}{\|g_0\|^2} + o(\rho^2),$$

$$\begin{aligned}
&g_0^\top H^2 g_1 \|g_1\|^2 \\
&= \left(g_0^\top H^2 g_0 + \rho \frac{g_0^\top H^3 g_0}{\|g_0\|} + o(\rho) \right) \left(\|g_0\|^2 + 2\rho \frac{g_0^\top H g_0}{\|g_0\|} + \rho^2 \frac{g_0^\top H^2 g_0}{\|g_0\|^2} + o(\rho^2) \right) \\
&= g_0^\top H^2 g_0 \|g_0\|^2 + \rho \left(2 \frac{g_0^\top H^2 g_0 \cdot g_0^\top H g_0}{\|g_0\|} + \|g_0\| g_0^\top H^3 g_0 \right) \\
&\quad + \rho^2 \left(\frac{g_0^\top H^2 g_0 \cdot g_0^\top H^2 g_0}{\|g_0\|^2} + 2 \frac{g_0^\top H^3 g_0 \cdot g_0^\top H g_0}{\|g_0\|^2} \right) + o(\rho^2),
\end{aligned}$$

$$\begin{aligned}
&(g_1^\top H g_1)(g_0^\top H g_1) \\
&= \left(g_0^\top H g_0 + 2\rho \frac{g_0^\top H^2 g_0}{\|g_0\|} + \rho^2 \frac{g_0^\top H^3 g_0}{\|g_0\|^2} + o(\rho^2) \right) \left(g_0^\top H g_0 + \rho \frac{g_0^\top H^2 g_0}{\|g_0\|} + o(\rho) \right) \\
&= (g_0^\top H g_0)^2 + \rho \left(3 \frac{g_0^\top H g_0 \cdot g_0^\top H^2 g_0}{\|g_0\|} \right) + \rho^2 \left(\frac{g_0^\top H^3 g_0 \cdot g_0^\top H g_0}{\|g_0\|^2} + 2 \frac{(g_0^\top H^2 g_0)^2}{\|g_0\|^2} \right) + o(\rho^2).
\end{aligned}$$

6. Combining terms:

$$\begin{aligned}
g_0^\top H^2 g_1 \|g_1\|^2 - (g_1^\top H g_1)(g_0^\top H g_1) = &\left(g_0^\top H^2 g_0 \|g_0\|^2 - (g_0^\top H g_0)^2 \right) \\
&+ \rho \left(\|g_0\| g_0^\top H^3 g_0 - \frac{g_0^\top H g_0 \cdot g_0^\top H^2 g_0}{\|g_0\|} \right) \\
&+ \rho^2 \left(\frac{g_0^\top H^3 g_0 \cdot g_0^\top H g_0 - (g_0^\top H^2 g_0)^2}{\|g_0\|^2} \right) + o(\rho^2).
\end{aligned}$$

7. Sign analysis:

- Zero-order term $g_0^\top H^2 g_0 \|g_0\|^2 - (g_0^\top H g_0)^2$: Strictly positive by Cauchy-Schwarz inequality since H is positive definite and g_0 and Hg_0 are not parallel.
- First-order term $\|g_0\|^2 g_0^\top H^3 g_0 - g_0^\top H g_0 \cdot g_0^\top H^2 g_0$: Non-negative by Chebyshev's sum inequality for the sequences $\{\lambda_i\}$ and $\{\lambda_i^2\}$ where $H = \sum \lambda_i v_i v_i^\top$.
- Second-order term $g_0^\top H^3 g_0 \cdot g_0^\top H g_0 - (g_0^\top H^2 g_0)^2$: Non-negative by Chebyshev's sum inequality.

8. Conclusion:

The term is strictly positive, which means $f'(1) > 0$. So, there exists $\alpha > 1$ such that $f(\alpha) > f(1)$. For sufficiently large ρ_m, where the second-order term dominates, this further implies:

$$L\left(\vartheta_0 + \rho_m \frac{g_\alpha}{\|g_\alpha\|} \right) > L\left(\vartheta_0 + \rho_m \frac{g_1}{\|g_1\|} \right).$$

□

Remark. *The ρ_m threshold exists only to ensure the second-order term dominates the first-order term. In practice, moderate values suffice to observe XSAM's advantage over SAM.*

Remark. *Practically, the loss surface may not admit a second-order approximation, and the maximum does not necessarily lie around $\alpha = 1$. So we search a relatively large range of α, e.g., in $[0, 2]$, to make it more generally applicable. Additionally, we use spherical linear combination instead, for a more uniform distribution of searched directions and better coverage.*

C Computational Overhead

The evaluation of each α will involve a forward pass of the neural network for calculating $L(\vartheta_0 + v(\alpha) \cdot \rho_m)$. So, the cost of the dynamic search of α^* roughly equals the number of samples of α times the cost of a forward pass. Typically, we use $20 \sim 40$ samples to search for α^*. If this were required at every iteration, it would incur a considerable computational burden. Fortunately, frequent updates of α^* are unnecessary. According to our experiments, α^* is fairly stable and changes smoothly during training, as depicted in Figure 2 and Figure 11. In experiments, we by default adopt an epoch-wise update strategy: α^* is updated at the first iteration of each epoch and then kept fixed for the rest. Each epoch typically contains over 400 iterations. SAM requires $k+1$ forward and $k+1$ backward passes per iteration. So, the computational overhead of XSAM is roughly $40/(400 \cdot 2 \cdot (k+1)) \leq 0.025$, i.e., the increased cost is typically no more than 2.5% when compared to SAM, which is negligible. A straightforward comparison of runtimes is presented in Table 1. The runtime of XSAM is nearly identical to that of SAM, indicating that the additional computational overhead is negligible.

D Additional Experimental Details for Results in Sections 5

D.1 Details about the 2D Test Function

The test function used is defined by:

$$L(\theta) = L(\mu, \sigma) = -\log\left(0.7e^{-K_1(\mu,\sigma)/1.8^2} + 0.3e^{-K_2(\mu,\sigma)/1.2^2}\right), \tag{9}$$

where $K_i(\mu, \sigma)$ is the KL divergence between two univariate Gaussian distributions,

$$K_i(\mu, \sigma) = \log\frac{\sigma_i}{\sigma} + \frac{\sigma^2 + (\mu - \mu_i)^2}{2\sigma_i^2} - \frac{1}{2}. \tag{10}$$

with $(\mu_1, \sigma_1) = (20, 30)$ and $(\mu_2, \sigma_2) = (-20, 10)$. It features a sharp minimum at around $(-16.8, 12.8)$ with a value of 0.28 and a flat minimum at around $(19.8, 29.9)$ with a value of 0.36.

The visualized training trajectories in Figure 3a share the same start point $(-6.0, 10.0)$ and run for 400 steps. The learning rate is 5 (the gradient scale is small), momentum is 0.9, ρ is 6.0, and ρ_m is 18.0. The points passed at each step were recorded to plot the trajectories.

D.2 Experiment Setup

CIFAR-10, CIFAR-100, and Tint-ImageNet. We use RandomCrop and CutOut (DeVries, 2017) augmentations for CIFAR-10 and CIFAR-100 while using RandomResizedCrop and RandomErasing (Zhong et al., 2020) augmentations for Tiny-ImageNet since we believe improvements over strong augmentations can be more valuable. We use a batch size of 125 for all the datasets, such that the sample size of each dataset is divisible by the batch size, while near the typical choice of 128. We adopt the typical choice, SGD with a momentum of 0.9, as the base optimizer, which carries the true gradient descent to θ. All models are trained for 200 epochs, while the cosine annealing learning rate schedule is adopted in all settings.

We run each experiment 5 times with different random seeds and calculate the mean and standard deviation. Each experiment was conducted using a single NVIDIA Tesla V100 GPU.

ResNet50 on ImageNet. We evaluate our method on the larger dataset, ImageNet. Standard data augmentation techniques are applied, including resizing, cropping, random horizontal flipping, and normalization. We take SGD as base optimizer with a cosine learning rate decay.

IWSLT2014. We conduct experiments on the Neural Machine Translation (NMT) task, specifically German–English translation on the IWSLT2014 dataset (Cettolo et al., 2014), using the Transformer architecture following the FAIRSEQ (Ott et al., 2019). We use AdamW as the base optimizer due to its better performance on the transformer.

ViT-Ti. We further use a lightweight Vision Transformer (ViT-Ti) model on CIFAR-100 to evaluate our method. Note that following (Zhao et al., 2022a), we do not use Cutout augmentation for CIFAR-100 when trained by ViT-Ti. We use AdamW as the base optimizer.

D.3 HYPERPARAMETER DETAILS

Table 5: Hyperparameter details for Results in Table 2.

	CIFAR-10				CIFAR-100				Tiny-ImageNet			
VGG-11	SGD	SAM	WSAM	XSAM	SGD	SAM	WSAM	XSAM	SGD	SAM	WSAM	XSAM
Epoch	200				200				200			
Batch size	125				125				125			
Initial learning rate	0.05				0.05				0.05			
Momentum	0.9				0.9				0.9			
Weight decay	1×10^{-3}				1×10^{-3}				1×10^{-3}			
ρ	–	0.15	0.15	0.15	–	0.15	0.15	0.15	–	0.20	0.20	0.20
ρ_m	–	–	–	0.30	–	–	–	0.30	–	–	–	1.20
α	0.0	1.0	0.75	–	0.0	1.0	1.0	–	0.0	1.0	1.0	–
ResNet-18	SGD	SAM	WSAM	XSAM	SGD	SAM	WSAM	XSAM	SGD	SAM	WSAM	XSAM
Epoch	200				200				200			
Batch size	125				125				125			
Initial learning rate	0.05				0.05				0.05			
Momentum	0.9				0.9				0.9			
Weight decay	1×10^{-3}				1×10^{-3}				1×10^{-3}			
ρ	–	0.15	0.15	0.15	–	0.15	0.15	0.15	–	0.20	0.20	0.20
ρ_m	–	–	–	0.25	–	–	–	0.30	–	–	–	0.25
α	0.0	1.0	0.5	–	0.0	1.0	1.25	–	0.0	1.0	1.0	–
DenseNet-121	SGD	SAM	WSAM	XSAM	SGD	SAM	WSAM	XSAM	SGD	SAM	WSAM	XSAM
Epoch	200				200				200			
Batch size	125				125				125			
Initial learning rate	0.05				0.05				0.05			
Momentum	0.9				0.9				0.9			
Weight decay	1×10^{-3}				1×10^{-3}				1×10^{-3}			
ρ	–	0.05	0.05	0.05	–	0.10	0.10	0.10	–	0.20	0.20	0.20
ρ_m	–	–	–	0.10	–	–	–	0.20	–	–	–	0.20
α	0.00	1.0	1.25	–	0.0	1.0	0.75	–	0.0	1.0	0.75	–

Table 6: Hyperparameter details for Results in Figure 3b. Note that, in this experiment, α for WSAM adopts the average value of the dynamic α^* in the corresponding XSAM. We see from the results that such WSAM already clearly outperforms SAM.

	ρ=0.10			ρ=0.20			ρ=0.30		
	SAM	WSAM	XSAM	SAM	WSAM	XSAM	SAM	WSAM	XSAM
Epoch	200			200			200		
Batch size	125			125			125		
Initial learning rate	0.05			0.05			0.05		
Momentum	0.9			0.9			0.9		
Weight decay	1×10^{-3}			1×10^{-3}			1×10^{-3}		
ρ_m	–	–	0.30	–	–	0.30	–	–	0.30
α	1.0	1.57	–	1.0	1.15	–	1.0	0.92	–

Table 7: Hyperparameter details for Results in Figure 4 and 12a. Note that the basic hyperparameters are provided here, while the other hyperparameters are clearly illustrated in the respective figures.

	Figure 4		Figure 12a	
	SAM	XSAM	SAM	XSAM
Epoch	200		200	
Batch size	125		125	
Initial learning rate	0.05		0.05	
Momentum	0.9		0.9	
Weight decay	1×10^{-3}		1×10^{-3}	
ρ	0.15		0.15	

Table 8: Hyperparameter details for Results in Figure 3c. Note that, in this experiment, α for WSAM adopts the average value of the dynamic α^* in the corresponding XSAM. We see from the results that such WSAM already clearly outperforms SAM.

	ρ=0.04			ρ=0.08			ρ=0.12		
	SAM	WSAM	XSAM	SAM	WSAM	XSAM	SAM	WSAM	XSAM
Epoch		200			200			200	
Batch size		125			125			125	
Initial learning rate		0.05			0.05			0.05	
Momentum		0.9			0.9			0.9	
Weight decay		1×10^{-3}			1×10^{-3}			1×10^{-3}	
ρ_m	–	–	0.30	–	–	0.25	–	–	0.20
α	1.0	1.72	–	1.0	1.15	–	1.0	0.41	–

Table 9: Hyperparameters for SAM and XSAM on ImageNet/ResNet-50, Transformer/IWSLT2014, and ViT-Ti/CIFAR-100 in Table 3.

	ImageNet/ResNet-50		Transformer/IWSLT2014		CIFAR-100/ViT-Ti	
	SAM	XSAM	SAM	XSAM	SAM	XSAM
Epoch	90		300		300	
Batch size / Max Token	512		4096		256	
Initial learning rate	0.2		5×10^{-4}		0.001	
Momentum	0.9		(0.9,0.98)		(0.9,0.999)	
Weight decay	1×10^{-4}		0.3		0.3	
Label smooth	0.0		0.1		0.1	
ρ	0.05		0.15		0.9	
ρ_m	–	0.3	–	0.45	–	0.9
α	1.0	–	1.0	–	1.0	–

E Additional Experimental Results and Analysis

E.1 Evaluation of XSAM Combined with Other SAM Variants

In this section, we further evaluate the performance of SAM variants and their combinations with XSAM. As discussed in Section 4, some SAM variants, such as ASAM, FSAM, and VaSSO, target aspects of SAM that are largely orthogonal to those addressed by our method, making them potentially compatible for integration. Given the large number of such orthogonal approaches, we focus here on combining XSAM with ASAM and evaluating their performance on CIFAR-100 using ResNet-18. The results in Table 10 indicate that XSAM outperforms both SAM and ASAM individually, and that integrating XSAM with ASAM leads to further improvement, demonstrating the effectiveness of XSAM in combination with other SAM variants.

Table 10: Test accuracy of SAM variants and their combinations with XSAM.

	SAM	ASAM	XSAM	XSAM+ASAM
Test Accuracy	80.93 ± 0.11	81.11 ± 0.06	81.24 ± 0.07	$\mathbf{81.68 \pm 0.11}$

E.2 Additional Experiments of Multi-step SAM

In this experiment, we focus on the performance of multi-step SAM, its variants, and XSAM under varying ρ.

As we see in Table 11, all of these variants, especially LSAM and MSAM+, which involve intermediate gradients rather than merely using the last gradient g_k, managed to get consistently superior results than SAM. The performance of SAM constantly decreases as ρ gets large, which, from our perspective, suggests the deviations of g_k from ϑ_0 are too large. Under such circumstances, the earlier

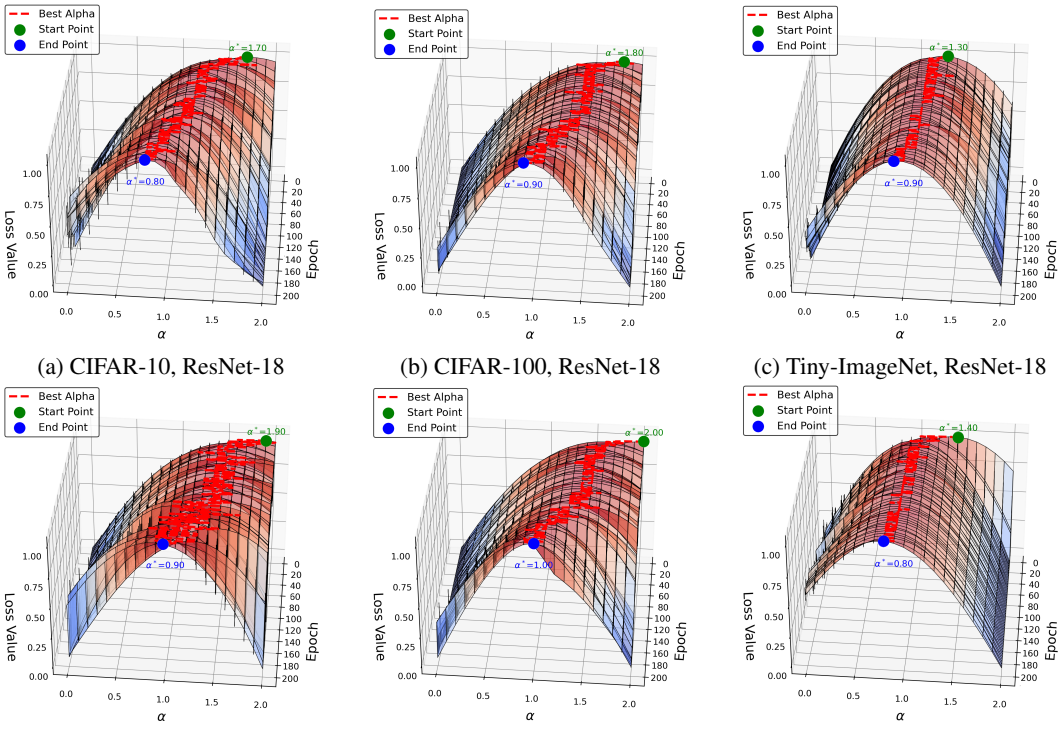
g_i must have less deviation, so combining it with earlier gradients would help. Besides, we see no clear trend for LSAM, LSAM+, MSAM, and MSAM+ as ρ gets large. Although MSAM+ can be viewed as LSAM+ with weights of gradients changed from $1/\|g_i\|$ to simply 1, the performance gap between them is obvious. This demonstrates that the weighting of gradients at different steps affects performance, and a more appropriate weighting scheme can lead to higher accuracy. Regardless, XSAM consistently outperforms all these methods in all cases.

Table 11: Results on CIFAR-100 using ResNet-18 in multi-step ($k = 3$) setting.

Method	$\rho = 0.04$	$\rho = 0.08$	$\rho = 0.12$
SAM	$80.79_{\pm 0.41}$	$80.75_{\pm 0.27}$	$79.72_{\pm 0.33}$
LSAM	$81.00_{\pm 0.21}$	$81.20_{\pm 0.24}$	$81.16_{\pm 0.04}$
LSAM+	$80.56_{\pm 0.20}$	$80.77_{\pm 0.04}$	$80.21_{\pm 0.27}$
MSAM	$81.04_{\pm 0.06}$	$81.12_{\pm 0.17}$	$80.93_{\pm 0.11}$
MSAM+	$80.72_{\pm 0.16}$	$81.16_{\pm 0.05}$	$81.16_{\pm 0.05}$
XSAM	$\mathbf{81.23}_{\pm 0.06}$	$\mathbf{81.36}_{\pm 0.08}$	$\mathbf{81.29}_{\pm 0.06}$

E.3 Inner Properties of XSAM

In this section, we present investigations into the internal properties of XSAM.



(a) CIFAR-10, ResNet-18 (b) CIFAR-100, ResNet-18 (c) Tiny-ImageNet, ResNet-18

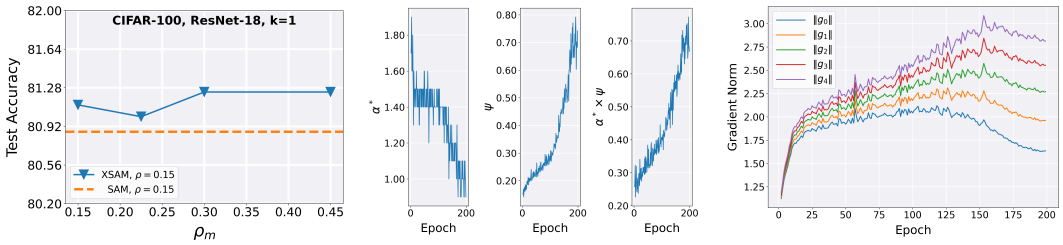
(d) CIFAR-10, DenseNet-121 (e) CIFAR-100, DenseNet-121 (f) Tiny-ImageNet, DenseNet-121

Figure 11: More visualizations of the dynamic estimations of α.

We first visualize the dynamic evaluations of α in a training instance in Figure 2 and in Figure 11, where loss values are normalized for better visibility. As we can see, for every dynamic evaluation of α, there is a clear optimal α. With the epoch-wise evaluation of α, we still see that the change of α^* during training is very smooth, which supports our choice of less frequently updating α^* for reducing computational overhead. On the other hand, we do see that α^* is changing during training, which validates our argument that a fixed α may not be optimum.

We further study how ρ_m influences the final performance. As results presented in Figure 12a, while ρ_m does impact performance to some extent, XSAM is able to outperform SAM in a fairly large range of ρ_m, from ρ to 3ρ. So, we consider that XSAM is not sensitive to ρ_m. The counterpart, as to how ρ influences when fixing the ρ_m, is actually demonstrated in Figure 3b, where we have used a fixed $\rho_m = 0.3$ by intention. It seems fairly robust to ρ.

In our experiments, we also see that α^* has a decreasing tendency during training. In fact, the angle ψ between v_0 and v_1 has an increasing tendency during training. We visualize such changes along with the offset angle $\alpha^* \cdot \psi$ from v_0 to the direction of the local maximum in Figure 12b. We see that the offset angle $\alpha^* \cdot \psi$ tends to increase. This may be because it converges to a lower position in a minima region as the learning rate decreases. Nevertheless, XSAM is able to help it away from the maximum within the local neighborhood in any case, as evident by the test accuracy results.



(a) (b) (c)

Figure 12: (a) Robustness analysis of XSAM with respect to ρ_m. (b)Training statistics of XSAM. (c) The norms of g_i during training.

We show in Figure 12c an instance of norm change of g_i during training in multi-step settings.

E.4 The Flatness/Sharpness of Resulting Models

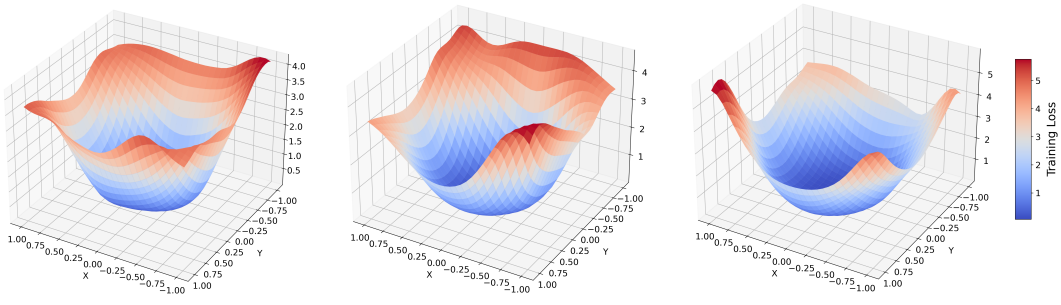
Hessian spectrum. To demonstrate that XSAM converges to flatter minima (more precisely, successfully shifts to a region where the maximum within the local neighborhood is lower), we calculate the Hessian eigenvalues of ResNet-18 trained for 200 epochs on CIFAR-10 with SGD, SAM, and XSAM. Following (Foret et al., 2020; Jastrzebski et al., 2020; Mi et al., 2022), we adopt two metrics: the largest eigenvalue (i.e., λ_1) and the ratio of the largest eigenvalue to the fifth largest one (i.e., λ_1/λ_5). To avoid the expensive computation cost of exact Hessian spectrum calculation, we approximate eigenvalues using the Lanczos algorithm (Ghorbani et al., 2019). The results, shown in **Table 12**, indicate that XSAM yields the smallest hessian spectrum, suggesting that it converges to flatter minima than SAM and SGD.

Table 12: Hessian spectrum of ResNet-18 using SGD, SAM, and XSAM on CIFAR-10.

	SGD	SAM	XSAM
λ_1	78.79	36.15	33.92
λ_1/λ_5	2.26	1.89	1.59

Visualization of loss landscape. We visualize the loss landscape of ResNet-18 trained on CIFAR-10 with SGD, SAM, and XSAM to further compare the flatness of the minimum. Using the visualization procedure in (Li et al., 2018), we randomly choose orthogonal normalization directions (i.e., X axis and Y axis) and then sample 50×50 points in the range of [-1,1] from these two directions. As shown in Figure 13, XSAM has a flatter loss landscape than SAM and SGD.

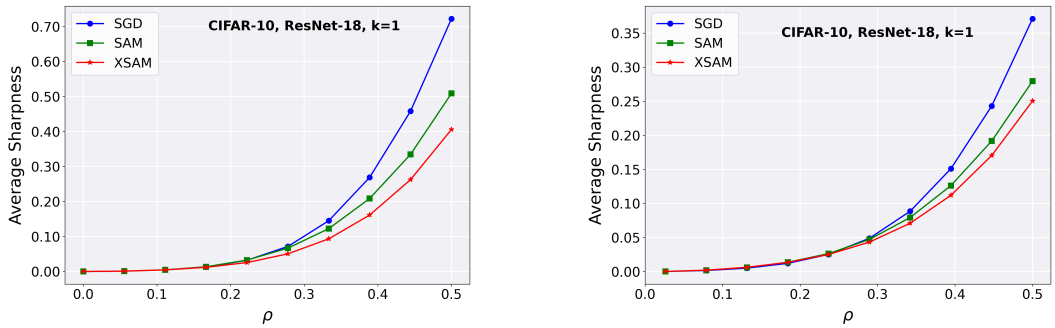
Average sharpness. We further visualize the average sharpness of the loss landscape at the convergence point. Specifically, following (Foret et al., 2020), we define the sharpness as the difference between the loss of the perturbation point and the loss of the convergence point. The average sharpness is then computed as the mean sharpness over multiple perturbations under the same perturbation radius. Then, we sample multiple random directions (e.g., 10, 50, 250, 1250) and continue this



(a) SGD (b) SAM (c) XSAM

Figure 13: Loss landscape visualizations of ResNet-18 on CIFAR-10 with SGD, SAM, and XSAM.

process until the average sharpness loss curve stabilizes, which provides a more representative characterization of the loss behavior around the convergence point. Based on our experiments, sampling 250 random directions is sufficient to achieve stable results. In addition, for the perturbation method, we adopt filter-wise and element-wise perturbation following (Li et al., 2018) to ensure a fair comparison between different optimizers (i.e., SGD, SAM, and XSAM). As shown in Figure 14, SAM exhibits smaller average sharpness compared to SGD, while XSAM further reduces the average sharpness.



(a) Element-wise perturbation (b) Filter-wise perturbation

Figure 14: Visualization of the average sharpness of the loss landscape at the convergence point.

F Strategies for Gradient Scale

Our default gradient scale strategy is using $\|g_k\|$ to match the scale with SAM. In this section, we empirically study a set of different ways for setting the gradient scale, which includes: typical choices like $\|g_k\|$ and $\|g_0\|$, simple extensions like $\sum_{i=0}^{k} \|g_i\|/(k+1)$ and $\max_{i=0}^{k} \|g_i\|$. Besides, we further explored two slope-based strategies:

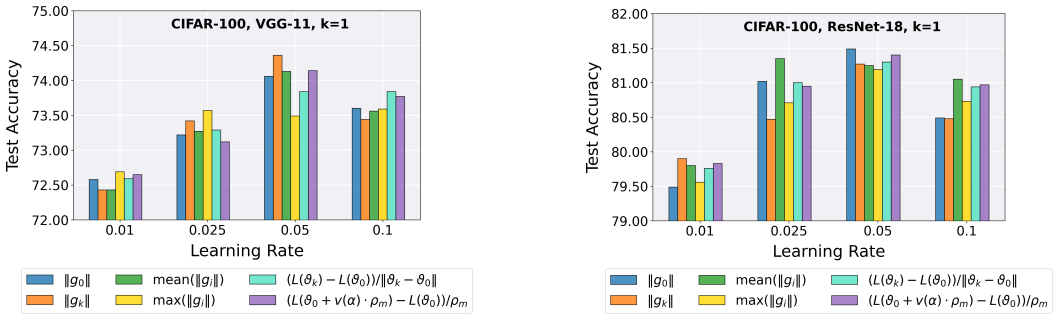
$$\text{slope}_k := \frac{L(\vartheta_k) - L(\vartheta_0)}{\|\vartheta_k - \vartheta_0\|},$$

$$\text{slope}_m := \frac{L(\vartheta_0 + v(\alpha) \cdot \rho_m) - L(\vartheta_0)}{\rho_m},$$

which is the averaged slope from ϑ_0 to ϑ_k and from ϑ_0 to the approximated maximum, respectively.

Note that since our direction is away from the approximated maximum, it can be an interesting combination when using the slope from ϑ_0 to the approximated maximum as the gradient scale, which shares the same intrinsic core as stochastic gradient descent. However, it would require an extra forward pass to evaluate $L(\vartheta_0 + v(\alpha) \cdot \rho_m)$.

The results are shown in Figure 15. As we can see, the gradient scale seems to be something that is even more mysterious than the gradient direction. It is hard to draw a direct conclusion on which



(a) (b)

Figure 15: Comparison of various gradient scale strategies.

might be the best choice among such a reasonably large group. Nevertheless, some choices appear to be good in most circumstances, which may include $\|g_0\|$, $\|g_k\|$, and slope$_m$. These primary results are included for completeness. We would leave further investigation into this as future work.

G Additional Related Work

The connection between flatness/sharpness and generalization was realized early on (Hochreiter & Schmidhuber, 1994) and further explored in subsequent works (Hochreiter & Schmidhuber, 1997; McAllester, 1999; Neyshabur et al., 2017; Jiang et al., 2019), motivating efforts toward finding flatter solutions. While SGD is believed to favor flat minima implicitly (Keskar et al., 2016; Ma & Ying, 2021), more explicit methods are preferred and developed. Typical instances include Entropy-SGD (Chaudhari et al., 2017) that employs entropy regularization, SWA (Izmailov et al., 2018) that seeks flatness by averaging model parameters, and SAM (Foret et al., 2020) that optimizes sharpness.

There are some variants that focus on improving the performance of multi-step SAM. Vanilla multi-step SAM (Foret et al., 2020) updates the model using the gradient at the last step. MSAM (Kim et al., 2023) suggests averaging all gradients except the first gradient at the original location. Lookbehind-SAM (LSAM) (Mordido et al., 2024) suggests another way that utilizes all gradients but excludes the first. In comparison, in multi-step settings, our method leverages all gradients ($\{g_i\}_{i=0}^{k-1}$ in v_0, and g_k in v_1) in a dynamic interpolation manner and explicitly approximates the direction of the maximum.

There are also some works that seek to reduce the computational overhead of SAM. For instance, ESAM (Du et al., 2021) achieves this via stochastic weight perturbation and sharpness-sensitive data selection. SSAM (Mi et al., 2022) accelerates SAM with a sparse perturbation. LookSAM (Liu et al., 2022a) reduces computational overhead by computing SAM's gradient only periodically and relying on an approximate gradient for most of the training time. RST (Zhao et al., 2022b) and AE-SAM (Jiang et al., 2023) suggest alternating between SGD and SAM in randomized and adaptive ways, respectively.

Another important line of research on SAM focuses on understanding its underlying mechanism. For instance, (Wen et al., 2023) finds that the gradient of SAM aligns with the top eigenvector of the Hessian in the late phase of training. This phenomenon is also concurrently found by (Bartlett et al., 2023). (Andriushchenko et al., 2023a) argues that SAM leads to low-rank features. In addition, an interesting fact observed by (Andriushchenko & Flammarion, 2022) is that training with SAM only in the late phase of training can achieve an improvement similar to that of full training with SAM. A recent work (Zhou et al., 2025) further analyzes and theoretically shows the learning dynamics of applying SAM late in training. (Tahmasebi et al., 2024) introduces a universal class of sharpness measures, in which SAM, known for its bias toward minimizing the maximum eigenvalue of the Hessian matrix, can be regarded as a special case. Our work is orthogonal to these works, providing a new perspective for understanding a fundamental question of why applying the gradient from the ascent point to the current parameters is valid, while at the same time, proposing XSAM as a better alternative.

H USE OF LARGE LANGUAGE MODELS

We used a large language model (LLM) only for language polishing (grammar, wording, and clarity) of drafts written by the authors. The model did not generate research ideas, methods, analyses, results, or figures, and it did not write any sections from scratch.