NeurIPS OPT 2025: 17th Annual Workshop on Optimization for Machine Learning

Sharpness-Aware Minimization with Z-Score Gradient Filtering

Vincent-Daniel Yun JUYOUNG.YUN@USC.EDU
University of Southern California, USA
Open Neural Network Research Lab, MODULABS, Republic of Korea

Abstract

Deep neural networks achieve high performance across many domains but can still face challenges in generalization when optimization is influenced by small or noisy gradient components. Sharpness-Aware Minimization improves generalization by perturbing parameters toward directions of high curvature, but it uses the entire gradient vector, which means that small or noisy components may affect the ascent step and cause the optimizer to miss optimal solutions. We propose Z-Score Filtered Sharpness-Aware Minimization, which applies Z-score based filtering to gradients in each layer. Instead of using all gradient components, a mask is constructed to retain only the top percentile with the largest absolute Z-scores. The percentile threshold Q_p determines how many components are kept, so that the ascent step focuses on directions that stand out most compared to the average of the layer. This selective perturbation refines the search toward flatter minima while reducing the influence of less significant gradients. Experiments on CIFAR-10, CIFAR-100, and Tiny-ImageNet with architectures including ResNet, VGG, and Vision Transformers show that the proposed method consistently improves test accuracy compared to Sharpness-Aware Minimization and its variants. The code repository is available at: https://github.com/YUNBLAK/Sharpness-Aware-Minimization-with-Z-Score-Gradient-Filtering

1. Introduction

Deep neural networks (DNNs) [10, 37, 38] achieve strong performance in tasks such as image classification [11, 19, 20], speech recognition [1, 12, 28], and natural language understanding [22, 23, 40]. They are typically trained by minimizing empirical loss using optimizers such as SGD and Adam [5, 18, 27, 35].

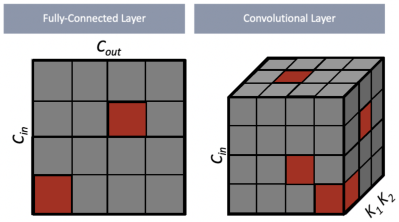

Figure 1: Ascent-step gradients after Z-score filtering.

Despite their success, DNNs often overfit [25, 36, 41], and poor generalization is frequently attributed to convergence toward sharp minima—regions of high curvature where small perturbations cause large increases in loss [9, 13, 16]. This issue becomes more pronounced in large models, where only a small subset of parameter directions meaningfully contributes to the curvature of the loss landscape [6, 15, 30]. Sharpness-Aware Minimization (SAM) addresses this by perturbing parameters in the gradient direction and minimizing the worst-case loss within an ℓ_2 ball [9, 21].

Existing Problem. SAM constructs its ascent direction using the *entire* gradient vector, including many small-magnitude components that provide little curvature information and largely reflect minibatch noise. Because the gradient is normalized before the perturbation is applied, even tiny noisy coordinates influence the ascent direction as much as large, informative ones. This distorts the perturbation, misaligns it with sharp curvature directions, and amplifies stochasticity in high-dimensional parameter spaces. In contrast, large gradient components typically correspond to directions of meaningful curvature, and focusing on them yields a more stable and faithful estimation of sharpness that better reflects the local geometry of the loss.

Our idea. We propose *Z-Score Filtered Sharpness-Aware Minimization (ZSharp)*, which applies layer-wise Z-score normalization [44] to identify statistically significant gradient components and retains only the top percentile (e.g., top 5%) for constructing the ascent direction. The percentile threshold Q_p controls the fraction $(1 - Q_p)$ of coordinates kept. Unlike ASAM [21] and Friendly-SAM [26], ZSharp requires only a single hyperparameter and remains compatible with various optimizers and architectures.

We evaluate ZSharp on CIFAR-10 [19], CIFAR-100 [19], and Tiny-ImageNet [24] with models including ResNet [11], VGG [39], and Vision Transformers [8]. Across all settings, ZSharp consistently matches or outperforms SAM and its variants, demonstrating that selectively filtering out small gradients leads to more stable ascent directions and better generalization.

2. Methodology

We propose Z-Score Filtered Sharpness-Aware Minimization *(ZSharp)*, a method that improves neural network training by using Z-score normalization [44] and filtering in a Sharpness-Aware framework [9]. When using the full gradient in the ascent step, small and noisy gradient components can weaken important curvature directions and may cause the optimizer to miss optimal convergence points. ZSharp mitigates this by retaining only the larger gradient components in each layer during the ascent step, which reduces the influence of noise.

Preliminaries. We consider a supervised learning framework with a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^m$ denotes input features and $y_i \in \mathcal{Y}$ represents labels. The neural network, parameterized by weights $w \in \mathbb{R}^d$, defines a mapping $f : \mathbb{R}^m \times \mathbb{R}^d \to \mathcal{Y}$. For L layers, the ℓ-th layer's parameters are $w^{(\ell)} \in \mathbb{R}^{d_\ell}$, with $\sum_{\ell=1}^L d_\ell = d$. The empirical loss $L(w) = \frac{1}{N}\sum_{i=1}^N \ell(f(\mathbf{x}_i; w), y_i)$ has gradient $\nabla L(w) \in \mathbb{R}^d$, with ℓ_2-norm $\|\nabla L(w)\|_2$. The percentile threshold Q_p controls the proportion of gradient components retained after filtering, where $(1 - Q_p)$ denotes the fraction of components with the largest magnitudes that are kept for the ascent step.

2.1. Sharpness-Aware Minimization

In standard SAM [9, 17, 32], the procedure consists of: (i) *Ascent:* perturbing parameters in the direction that increases the loss the most.

$$\epsilon_{\text{SAM}} = \rho \cdot \frac{\nabla L(w)}{\|\nabla L(w)\|_2 + \delta}, \quad \tilde{w} = w + \epsilon_{\text{SAM}}, \tag{1}$$

(ii) *Minimization:* compute the gradient at the perturbed point, $g = \nabla L(\tilde{w})$, (iii) *Weight update:* apply a base optimizer O (e.g., SGD, Adam) to update parameters,

$$w \leftarrow w - \eta\, O(g). \tag{2}$$

ZSharp keeps steps (ii) and (iii) identical to SAM, but replaces $\nabla L(w)$ in the ascent step with the filtered gradient $\nabla L(w)_\Omega$, focusing the perturbation on statistically significant directions.

2.2. Z-Score Filtered Ascent Step

ZSharp modifies the ascent step of SAM [9, 17, 32] by applying Layer-wise Z-Score Normalization [44] and retaining only the top $(1 - Q_p)$ fraction of components by absolute Z-score.

Let $\{g^{(\ell)}\}_{\ell=1}^{L}$ denote the gradient vectors for each layer ℓ, with $g^{(\ell)} \in \mathbb{R}^{d_\ell}$. For each layer, we define the Z-score normalized gradient as $\Omega(g^{(\ell)})_i = \frac{g_i^{(\ell)} - \mu(g^{(\ell)})}{\sigma(g^{(\ell)})}, \quad i = 1, \dots, d_\ell$ where $\mu(g^{(\ell)}) = \frac{1}{d_\ell} \sum_{i=1}^{d_\ell} g_i^{(\ell)}$ and $\sigma(g^{(\ell)}) = \sqrt{\frac{1}{d_\ell} \sum_{i=1}^{d_\ell} \left(g_i^{(\ell)} - \mu(g^{(\ell)})\right)^2}$ This ensures that each layer's gradient is centered and rescaled independently, emphasizing components that deviate most relative to the layer statistics.

Then, we have $\Omega(\nabla L(w))$, layer-wise Z-score normalized gradients of the loss L at parameters w. We then define a binary mask $\mathbf{m} \in \{0, 1\}^d$ as

$$m_j = \begin{cases} 1 & \text{if } |\Omega(\nabla L(w))_j| > q_{Q_p}, \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$

where q_{Q_p} is the Q_p-th percentile of $|\Omega(\nabla L(w))|$. The filtered gradient is $\nabla L(w)_\Omega = \nabla L(w) \odot \mathbf{m}$, and the perturbation is computed as:

$$\epsilon = \begin{cases} \rho \cdot \frac{\nabla L(w)_\Omega}{\|\nabla L(w)_\Omega\|_2 + \delta} & \|\nabla L(w)_\Omega\|_2 > 0, \\ \rho \cdot \frac{\nabla L(w)}{\|\nabla L(w)\|_2 + \delta} & \text{otherwise}, \end{cases} \tag{4}$$

where $\rho > 0$ is the perturbation radius and $\delta = 10^{-8}$ ensures numerical stability. The ascent update is $\tilde{w} = w + \epsilon$.

3. Theoretical Analysis

We adapt the standard SAM convergence proof [2, 9, 17] to the ZSharp setting, where the ascent step uses the Z-score filtered gradient. Theorem 4 establishes convergence under the same smoothness and bounded-variance assumptions as SAM [2, 9, 17]. Detailed proofs are in Appendix C.

Assumption 1 (β-smoothness) *The loss function $L : \mathbb{R}^d \to \mathbb{R}$ is β-smooth, meaning its gradient is β-Lipschitz continuous, $\|\nabla L(u) - \nabla L(v)\| \leq \beta \|u - v\|, \quad \forall\, u, v \in \mathbb{R}^d$. Equivalently, for any u, v, $L(u) \leq L(v) + \langle \nabla L(v),\, u - v \rangle + \frac{\beta}{2} \|u - v\|^2$.*

Assumption 2 (Unbiased stochastic gradient) *At each iteration t, the stochastic gradient provides an unbiased estimate of the true gradient: $\mathbb{E}[\nabla L_t(w_t)] = \nabla L(w_t)$.*

Lemma 1 *Given a β-smooth loss function $L(x)$, the following bound holds:*

$$\langle \nabla L(u) - \nabla L(v), u - v \rangle \geq -\beta \|u - v\|^2. \tag{5}$$

Lemma 2 *Let L be a β-smooth loss function. At iteration t, let $\nabla L_t(w_t)_\Omega$ be an unbiased stochastic estimator of $\nabla L(w_t)_\Omega$ with bounded variance $\mathbb{E}\left[\|\nabla L_t(w_t)_\Omega - \nabla L(w_t)_\Omega\|^2\right] \leq \frac{\sigma_\Omega^2}{b}$ Then, for any $r > 0$,*

$$\mathbb{E}[\langle \nabla L(w_t + r\, \nabla L_t(w_t)_\Omega), \nabla L(w_t) \rangle] \geq \frac{1}{2}\, \mathbb{E}\left[\|\nabla L(w_t)\|^2\right] - \frac{\beta^2 r^2}{2}\, \mathbb{E}\left[\|\nabla L(w_t)_\Omega\|^2\right] - \frac{\beta^2 r^2 \sigma_\Omega^2}{2b}. \tag{6}$$

Lemma 3 *Assume that L is β-smooth and the filtered stochastic gradient $\nabla L_t(w_t)_\Omega$ satisfies the variance bound $\mathbb{E}\big[\|\nabla L_t(w_t)_\Omega - \nabla L(w_t)_\Omega\|^2\big] \leq \frac{\sigma_\Omega^2}{b}$. Consider the ZSharp-SAM updates $w_{t+1/2} = w_t + r\,\nabla L_t(w_t)_\Omega$, and $w_{t+1} = w_{t+1/2} - \eta\,\nabla L_t(w_{t+1/2})$. If the step size satisfies $\eta \leq \frac{1}{4\beta}$ and the ascent radius satisfies $\beta^2 r^2 \leq \frac{1}{4}$, then*

$$\mathbb{E}[L(w_{t+1})] \leq \mathbb{E}[L(w_t)] - \frac{\eta}{4}\,\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] + \frac{2\eta\beta^2 r^2}{b}\,\sigma_\Omega^2 + \frac{\eta^2\beta}{b}\,\sigma_\Omega^2. \tag{7}$$

Theorem 4 *Assume that L is β-smooth and that the filtered stochastic gradient $\nabla L_t(w_t)_\Omega$ satisfies the variance bound $\mathbb{E}\big[\|\nabla L_t(w_t)_\Omega - \nabla L(w_t)_\Omega\|^2\big]\, le\frac{\sigma_\Omega^2}{b}$. All expectations are taken over the mini-batch at iteration t. Suppose the ZSharp-SAM updates $w_{t+1/2} = w_t + r\,\nabla L_t(w_t)_\Omega and w_{t+1} = w_{t+1/2} - \eta\,\nabla L_t(w_{t+1/2})$ use step size η and ascent radius r satisfying $\eta \leq \frac{1}{4\beta}$ and $\beta^2 r^2 \leq \frac{1}{4}$. Then ZSharp-SAM satisfies*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] \leq \frac{4}{T\eta}\big(L(w_0) - \mathbb{E}[L(w_T)]\big) + \frac{8\beta^2 r^2}{b}\,\sigma_\Omega^2 + \frac{4\eta\beta}{b}\,\sigma_\Omega^2. \tag{8}$$

Corollary 5 (Convergence with diminishing step sizes) *Assume that L is β-smooth, bounded below by $L_{\inf}$, and $\nabla L_t(w_t)_\Omega$ satisfies $\mathbb{E}\big[\|\nabla L_t(w_t)_\Omega - \nabla L(w_t)_\Omega\|^2\big]\leq \frac{\sigma_\Omega^2}{b}$. Consider the ZSharp-SAM updates $w_{t+1/2} = w_t + r_t\,\nabla L_t(w_t)_\Omega$, and $w_{t+1} = w_{t+1/2} - \eta_t\,\nabla L_t(w_{t+1/2})$, with step sizes $\{\eta_t\}_{t\geq 0}$ and ascent radii $\{r_t\}_{t\geq 0}$ satisfying $\eta_t \leq \frac{1}{4\beta}\beta^2 r_t^2 \leq \frac{1}{4}$for all t, $\sum_{t=0}^\infty \eta_t = \infty$,$\sum_{t=0}^\infty \eta_t^2 < \infty$,$\sum_{t=0}^\infty \eta_t r_t^2 < \infty$. Then*

$$\sum_{t=0}^{\infty}\eta_t\,\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] < \infty, \quad \textit{and in particular} \quad \liminf_{t\to\infty}\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] = 0. \tag{9}$$

This corollary establishes that ZSharp retains the convergence guarantees of standard SAM: under the same smoothness and bounded-variance assumptions, Z-score filtering in the ascent step still ensures convergence to stationary points. In other words, applying Z-score gradient filtering does not hinder the convergence behavior of SAM.

4. Experimental Results

To evaluate the effectiveness of ZSharp, we compare it with the standard SAM [9] and its variants, ASAM [21] and Friendly-SAM [26], as well as the baseline optimizer.

Experimental Settings. We evaluate ZSharp on CIFAR-10/100 [19] and Tiny-ImageNet [24] using ResNet-56/110 [11], VGG16_BN [39], and ViT models [8]. All models are trained for 200 epochs with batch size 256 using AdamW [18, 27] (lr=0.001, weight decay 5×10^{-5}) and step decay (0.75 every 10 epochs). For SAM [9], ASAM [21], Friendly-SAM [26], and ZSharp, we set $\rho = 0.05$ following prior work. ZSharp applies Z-score filtering ($Q_p = 0.95$) in the ascent step, keeping the top 5% of gradient components. All experiments run on a single RTX 4090 GPU and results are averaged over 3 seeds. We used the publicly available implementations of ViT training [33], ASAM [21], and FSAM [26] from open github repositories.

Results. Table 1 presents the Top-1 test accuracy and train loss across five architectures (ResNet-56, ResNet-110, VGG-16/BN, ViT-7/8/8-384, and ViT-7/8/12-768) on CIFAR-10, CIFAR-100, and Tiny-ImageNet. Overall, ZSharp consistently achieves the highest or comparable test accuracy among all methods, with gains observed across both convolution-based and transformer-based architectures.

		CIFAR-10 [19]		**CIFAR-100 [19]**		**Tiny-ImageNet [24]**	
Network	Method	Top-1 Test Acc.	Train Loss.	Top-1 Test Acc.	Train Loss.	Top-1 Test Acc.	Train Loss.
ResNet-56 [11]	AdamW (Baseline) [27]	0.9108 ± 0.0045	0.0057 ± 0.0013	0.6420 ± 0.0031	0.0452 ± 0.0629	0.4747 ± 0.0031	0.0121 ± 0.0105
	SAM [9]	0.9160 ± 0.0021	0.0221 ± 0.0051	0.6527 ± 0.0025	0.1097 ± 0.0584	0.4938 ± 0.0026	0.0453 ± 0.0127
	ASAM [21]	0.9228 ± 0.0034	0.0366 ± 0.0093	0.6646 ± 0.0017	0.1952 ± 0.0419	0.5072 ± 0.0012	0.0564 ± 0.0091
	Friendly-SAM [26]	0.9179 ± 0.0025	0.0219 ± 0.0076	0.6549 ± 0.0024	0.1051 ± 0.0417	0.4948 ± 0.0023	0.0444 ± 0.0102
	ZSharp (Ours)	**0.9264 ± 0.0032**	0.0630 ± 0.0064	**0.6679 ± 0.0015**	0.2510 ± 0.0438	**0.5073 ± 0.0014**	0.0828 ± 0.0129
ResNet-110 [11]	AdamW (Baseline) [27]	0.9140 ± 0.0031	0.0056 ± 0.0023	0.6650 ± 0.0025	0.0149 ± 0.0059	0.4878 ± 0.0045	0.0556 ± 0.0114
	SAM [9]	0.9233 ± 0.0025	0.0188 ± 0.0037	0.6815 ± 0.0019	0.0531 ± 0.0121	0.5005 ± 0.0045	0.1417 ± 0.0216
	ASAM [21]	0.9261 ± 0.0023	0.0288 ± 0.0056	0.6796 ± 0.0036	0.0915 ± 0.0123	0.5105 ± 0.0045	0.2894 ± 0.0241
	Friendly-SAM [26]	0.9193 ± 0.0013	0.0190 ± 0.0036	0.6762 ± 0.0021	0.0524 ± 0.0113	0.5027 ± 0.0045	0.1402 ± 0.0091
	ZSharp (Ours)	**0.9293 ± 0.0017**	0.0618 ± 0.0097	**0.6844 ± 0.0023**	0.1656 ± 0.0213	**0.5207 ± 0.0045**	0.4137 ± 0.0311
VGG-16/BN [39]	AdamW (Baseline) [27]	0.9247 ± 0.0013	0.0058 ± 0.0031	0.6999 ± 0.0102	0.0092 ± 0.0051	0.5507 ± 0.0093	0.0043 ± 0.0071
	SAM [9]	0.9337 ± 0.0018	0.0171 ± 0.0093	0.7092 ± 0.0093	0.0139 ± 0.0073	0.5587 ± 0.0103	0.0363 ± 0.0183
	ASAM [21]	**0.9355 ± 0.0012**	0.0237 ± 0.0047	0.7170 ± 0.0121	0.0375 ± 0.0118	0.5647 ± 0.0191	0.0644 ± 0.0237
	Friendly-SAM [26]	0.9290 ± 0.0017	0.0163 ± 0.0093	0.7099 ± 0.0083	0.0495 ± 0.0125	0.5544 ± 0.0204	0.0349 ± 0.0153
	ZSharp (Ours)	0.9327 ± 0.0020	0.0351 ± 0.0144	**0.7207 ± 0.0071**	0.0375 ± 0.0137	**0.5673 ± 0.0231**	0.1248 ± 0.0351
ViT-7/8/8-384 [8]	AdamW (Baseline) [27]	0.8398 ± 0.0028	0.0087 ± 0.0092	0.5479 ± 0.0011	0.0042 ± 0.0031	0.2843 ± 0.0012	0.0056 ± 0.0014
	SAM [9]	0.8432 ± 0.0032	0.0273 ± 0.0101	0.5557 ± 0.0013	0.0255 ± 0.0141	0.2897 ± 0.0009	0.0363 ± 0.0098
	ASAM [21]	0.8302 ± 0.0034	0.0367 ± 0.0138	0.5566 ± 0.0031	0.0349 ± 0.0193	0.2522 ± 0.0032	0.0644 ± 0.0137
	Friendly-SAM [26]	0.8476 ± 0.0044	0.0273 ± 0.0093	0.5608 ± 0.0023	0.0228 ± 0.0138	0.3000 ± 0.0012	0.0349 ± 0.0083
	ZSharp (Ours)	**0.8543 ± 0.0029**	0.0647 ± 0.0216	**0.5748 ± 0.0051**	0.0730 ± 0.0212	**0.3057 ± 0.0021**	0.1248 ± 0.0413
ViT-7/8/12-768 [8]	AdamW (Baseline) [27]	0.8438 ± 0.0021	0.0087 ± 0.0031	0.5615 ± 0.0013	0.0040 ± 0.0045	0.2991 ± 0.0010	0.0065 ± 0.0032
	SAM [9]	0.8486 ± 0.0018	0.0293 ± 0.0098	0.5691 ± 0.0014	0.0234 ± 0.0076	0.3014 ± 0.0015	0.0297 ± 0.0098
	ASAM [21]	0.8395 ± 0.0020	0.0371 ± 0.0101	0.5649 ± 0.0027	0.0347 ± 0.0116	0.3023 ± 0.0008	0.0512 ± 0.0161
	Friendly-SAM [26]	0.8525 ± 0.0021	0.0283 ± 0.0084	0.5655 ± 0.0021	0.0246 ± 0.0122	0.3034 ± 0.0013	0.0441 ± 0.0141
	ZSharp (Ours)	**0.8586 ± 0.0023**	0.0635 ± 0.0196	**0.5777 ± 0.0031**	0.0709 ± 0.0178	**0.3104 ± 0.0019**	0.1341 ± 0.0211

Table 1: Top-1 Test Accuracy and Train Loss for ResNet-56 [11], ResNet-110 [11], VGG-16/BN [39], ViT-7/8/8-384 [8], and ViT-7/8/12-768 [8] on CIFAR-10 [19], CIFAR-100 [19], and Tiny-ImageNet datasets [24] across different SAM variants such as AdamW (Baseline) [27], SAM [9], Friendly-SAM [26], ASAM [21], and ZSharp (Ours). For ViT models, ViT-7/8/8-384 and ViT-7/8/12-768 denote Vision Transformers with 7 layers, 8 attention heads, patch sizes of 8, and MLP dimensions of 384 and 768, respectively.

Network	Method	Q_p	Top-1 Test Acc.	Train Loss.	Network	Method	Q_p	Top-1 Test Acc.	Train Loss.
ResNet-56 [11]	AdamW [27]	N/A	0.9108 ± 0.0045	0.0057 ± 0.0013	ViT-7/8/8-384 [8]	AdamW [27]	N/A	0.8398 ± 0.0028	0.0087 ± 0.0092
	SAM [9]	N/A	0.9160 ± 0.0021	0.0221 ± 0.0051		SAM [9]	N/A	0.8432 ± 0.0032	0.0273 ± 0.0101
	ZSharp (Ours)	0.95	**0.9264 ± 0.0032**	0.0630 ± 0.0064		ZSharp (Ours)	0.95	**0.8543 ± 0.0029**	0.0647 ± 0.0216
	ZSharp (Ours)	0.90	0.9212 ± 0.0015	0.0710 ± 0.0061		ZSharp (Ours)	0.90	0.8482 ± 0.0031	0.0748 ± 0.0081
	ZSharp (Ours)	0.85	0.9189 ± 0.0023	0.0679 ± 0.0067		ZSharp (Ours)	0.85	0.8424 ± 0.0043	0.0825 ± 0.0086
	ZSharp (Ours)	0.80	0.9153 ± 0.0027	0.0731 ± 0.0053		ZSharp (Ours)	0.80	0.8421 ± 0.0038	0.0863 ± 0.0119
	ZSharp (Ours)	0.75	0.9132 ± 0.0017	0.0789 ± 0.0079		ZSharp (Ours)	0.75	0.8378 ± 0.0027	0.0999 ± 0.0102

Table 2: Training hyperparameters and results for all experiments, with settings identical to those in the Experimental Settings section except for varying Q_p values.

Hyperparameter Tuning We analyze the percentile threshold Q_p in ZSharp, which retains the top $(1 - Q_p)\%$ of gradients after Z-score filtering. Higher values (e.g., 0.95) keep fewer but more significant components, while $Q_p = 0.0$ recovers SAM. Table 2 shows that $Q_p = 0.95$ yields the best

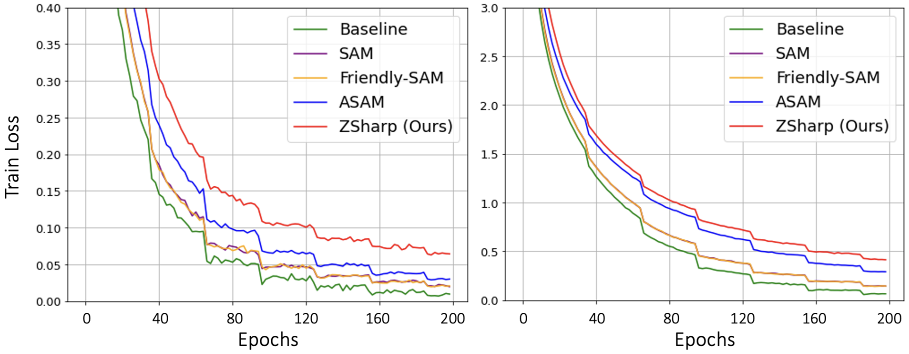

Figure 2: Train Loss comparison on CIFAR-10 for ResNet-56 (left) and ResNet-110 (right) across different SAM variants: Baseline, SAM, Friendly-SAM, ASAM, and ZSharp (Ours).

accuracy on both ResNet-56 and ViT-7/8/8-384, with performance gradually approaching SAM as Q_p decreases. We therefore use $Q_p = 0.95$ for all subsequent experiments.

Generalization Effect. Figure 2 shows that ZSharp has a higher train loss but better test accuracy than other methods. Similar patterns have been observed in prior work [4, 16, 31, 45], where models with slightly higher training loss can generalize better when they converge to flatter or wider minima. This suggests that ZSharp's selective focus on high-magnitude gradient components not only reduces overfitting but also helps the model find solutions with improved generalization performance on unseen data.

5. Conclusion

We proposed ZSharp, a sharpness-aware optimization method that applies z-score gradient filtering to the ascent step of SAM, focusing updates on statistically significant gradient components. ZSharp preserves SAM's convergence guarantees and consistently improves test accuracy over SAM and its variants across CIFAR-10, CIFAR-100, and Tiny-ImageNet on diverse architectures. With only one additional hyperparameter and no architectural changes, ZSharp offers an effective way to enhance generalization in deep neural network training.

6. Acknowledgement

This research was supported by Brian Impact Foundation, a non-profit organization dedicated to the advancement of science and technology for all.

References

[1] Harsh Ahlawat, Naveen Aggarwal, and Deepti Gupta. Automatic speech recognition: A survey of deep learning techniques and approaches. *International Journal of Cognitive Computing in Engineering*, 6:201–237, 2025. ISSN 2666-3074. doi: https://doi.org/10.1016/j.ijcce.2024.12.007.

[2] Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *International Conference on Machine Learning (ICML)*, 2022. doi: 10.48550/arXiv.2206.06232. Camera-ready version.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

[5] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[6] Satrajit Chatterjee and Piotr Zielinski. On the generalization mystery in deep learning. *arXiv preprint arXiv:2203.10036*, 2022. doi: 10.48550/arXiv.2203.10036.

[7] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. *International Conference on Learning Representations*, 2022.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

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

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[11] 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*, pages 770–778, 2016.

[12] Geoffrey E. Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 29(6), 2012.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015. doi: 10.48550/arXiv.1502.03167.

[15] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. In *Mathematical Aspects of Deep Learning*. Cambridge University Press, 2022. doi: 10.1017/9781009025096.003. Also available as arXiv preprint arXiv:1710.05468.

[16] Nitish Shirish Keskar, Jorge Nocedal, et al. On large-batch training for deep learning: Generalization gap and sharp minima. *International Conference on Learning Representations*, 2017.

[17] Pham Duy Khanh, Hoang-Chau Luong, Boris Mordukhovich, and Dat Ba Tran. Fundamental convergence analysis of sharpness-aware minimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[19] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. Technical Report.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[21] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. *arXiv preprint arXiv:2102.11600*, 2021.

[22] Piotr Kłosowski. Deep learning for natural language processing and language modelling. In *2018 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, pages 223–228, 2018. doi: 10.23919/SPA.2018.8563389.

[23] Ivano Lauriola, Alberto Lavelli, and Fabio Aiolli. An introduction to deep learning in natural language processing: Models, techniques, and tools. *Neurocomputing*, 470:443–456, 2022. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2021.05.103.

[24] Ya Le and Xun Yang. Tiny imagenet visual recognition challenge, 2015.

[25] Haidong Li, Jiongcheng Li, Xiaoming Guan, Binghao Liang, Yuting Lai, and Xinglong Luo. Research on overfitting of deep learning. In *2019 15th International Conference on Computational Intelligence and Security (CIS)*, pages 78–81, 2019. doi: 10.1109/CIS.2019.00025.

[26] 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)*, 2024.

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[28] Ambuj Mehrish, Navonil Majumder, Rishabh Bharadwaj, Rada Mihalcea, and Soujanya Poria. A review of deep learning techniques for speech processing. *Information Fusion*, 99:101869, 2023. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2023.101869.

[29] Peng Mi, Li Shen, Tianhe Ren, Yiyi Zhou, Xiaoshuai Sun, Rongrong Ji, and Dacheng Tao. Make sharpness-aware minimization stronger: A sparsified perturbation approach. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 30950–30962. Curran Associates, Inc., 2022.

[30] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5949–5958, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[31] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

[32] Dimitris Oikonomou and Nicolas Loizou. Sharpness-aware minimization: General analysis and improved rates. In *The Thirteenth International Conference on Learning Representations*, 2025.

[33] OmiHub777. ViT-CIFAR: PyTorch implementation for Vision Transformer on CIFAR datasets. https://github.com/omihub777/ViT-CIFAR, 2021. Accessed: 2025-08-15.

[34] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. *International Conference on Learning Representations*, 2013.

[35] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[36] Shaeke Salman and Xiuwen Liu. Overfitting mechanism and avoidance in deep neural networks. *arXiv preprint arXiv:1901.06566*, 2019. doi: 10.48550/arXiv.1901.06566.

[37] Ihsan Hameed Sarker. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2(6):420, 2021. doi: 10.1007/s42979-021-00815-1.

[38] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2014.09.003.

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[41] Xue Ying. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168(2):022022, 2019. doi: 10.1088/1742-6596/1168/2/022022.

[42] Hongyang Yong, Jiancheng Huang, Xinyu Hua, and Lei Zhang. Gradient centralization: A new optimization technique for deep neural networks. *European Conference on Computer Visio*, 2020.

[43] Juyoung Yun. Stochastic gradient sampling for enhancing neural networks training. *Neural Computing and Applications*, 37:14005–14028, July 2025. doi: 10.1007/s00521-025-11242-1.

[44] Juyoung Yun. Znorm: Z-score gradient normalization accelerating skip-connected network training without architectural modification. In Qingyun Wang, Wenpeng Yin, Abhishek Aich, Yumin Suh, and Kuan-Chuan Peng, editors, *AI for Research and Scalable, Efficient Systems*, pages 240–254, Singapore, 2025. Springer Nature Singapore.

[45] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. doi: 10.48550/arXiv.1611.03530.

[46] Zhiyuan Zhang, Ruixuan Luo, Qi Su, and Xu Sun. Ga-sam: Gradient-strength based adaptive sharpness-aware minimization for improved generalization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022. doi: 10.48550/arXiv.2210.06895.

Appendix A. Overview

Figure 3 illustrates the overall process of ZSharp, highlighting how Z-score gradient filtering is integrated into the Sharpness-Aware Minimization (SAM) framework.

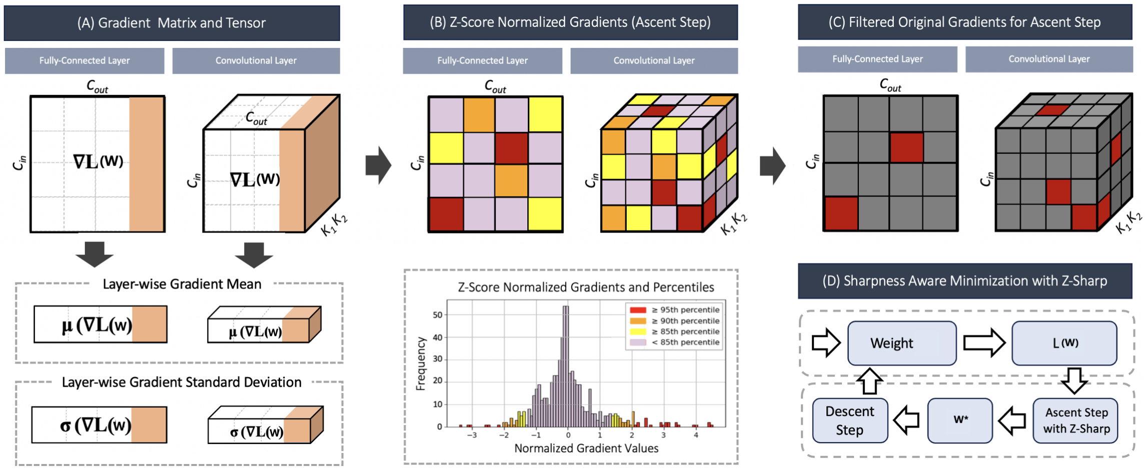
(A) Gradient Matrix and Tensor
Fully-Connected Layer
Convolutional Layer
C_{out}
C_{in}
∇L(W)
K_1K_2
Layer-wise Gradient Mean
μ (∇L(w))
Layer-wise Gradient Standard Deviation
σ (∇L(w))
(B) Z-Score Normalized Gradients (Ascent Step)
Z-Score Normalized Gradients and Percentiles
Frequency
Normalized Gradient Values
0
10
20
30
40
50
−3
−2
−1
0
1
2
3
4
≥ 95th percentile
≥ 90th percentile
≥ 85th percentile
< 85th percentile
(C) Filtered Original Gradients for Ascent Step
(D) Sharpness Aware Minimization with Z-Sharp
Weight
L (W)
Ascent Step with Z-Sharp
W*
Descent Step

Figure 3: Overview of ZSharp: Z-Score Filtered Sharpness-Aware Minimization. (A) Gradients from fully-connected and convolutional layers are used to compute layer-wise statistics. (B) Z-score normalization is applied to standardize gradients, followed by percentile-based filtering to select statistically significant components. (C) A binary mask retains only the top Z-score entries (e.g., top 5%), filtering the gradient for the ascent step. (D) The filtered gradient is then used in the SAM ascent phase to refine the perturbation direction, enhancing generalization by focusing updates on curvature-sensitive directions.

(A) Gradient Matrix and Tensor. For each fully-connected and convolutional layer, we obtain the gradient tensor $\nabla L(w)$ during the ascent step of SAM. The layer-wise gradient mean $\mu(\nabla L(w))$ and standard deviation $\sigma(\nabla L(w))$ are computed to capture the statistical distribution of gradient values.

(B-C) Z-Score Normalization and Gradient Filtering Layer-wise Z-score normalization is applied to standardize the gradient values, producing normalized gradients $\Omega(\nabla L(w))$. A percentile-based ranking is then computed, where components are categorized (e.g., $\geq$ 95th, $\geq$ 90th, $\geq$ 85th percentile, or below). A binary mask is generated to retain only the top $(1 - Q_p)\%$ of components with the largest absolute Z-scores. This mask is applied to the *original* gradient (not the normalized one), resulting in a filtered gradient $\nabla L(w)_\Omega$ that emphasizes statistically significant components.

(D) Integration into SAM. The filtered gradient is used in place of the original gradient in SAM's ascent step to compute the perturbation ϵ. The perturbed parameters w^* are then used in the descent step with the base optimizer.

Appendix B. Related Works

Improving generalization in deep neural networks (DNNs) [10, 37, 38] has motivated many optimization strategies. Among these, normalization-based approaches and sharpness-aware optimization have been widely explored.

Normalization techniques are effective in enhancing generalization performance. Batch Normalization [14] and Layer Normalization [3] act on activations, alleviating gradient vanishing while improving generalization. At the gradient level, gradient clipping [34] limits gradient magnitude, and gradient centralization [42] subtracts mean values to improve convergence. Stochastic Gradient Sampling, as in StochGradAdam [43], selects subsets of gradients during training, leading to stronger generalization particularly in ResNet-based CNNs. More recently, ZNorm [44] applies layer-wise Z-score normalization to gradients, providing consistent scaling and yielding enhanced generalization on benchmarks such as CIFAR-10 and in medical imaging tasks.

Beyond normalization, sharpness-aware optimization has emerged as a key framework, aiming to locate flatter minima that empirically correlate with stronger generalization. Sharpness-Aware Minimization (SAM) [2, 9, 17, 32] perturbs parameters in the gradient direction and minimizes the maximum loss within a local ℓ_2 neighborhood. This approach improves generalization [7] compared to standard optimizers such as SGD [5] and Adam [27]. However, SAM constructs perturbations using the full gradient vector, including noisy or weak components, which can reduce precision in identifying sharpness-sensitive directions [29, 46].

Several extensions have been proposed. Adaptive SAM (ASAM) [21] rescales perturbations by curvature, improving robustness to parameter scaling. Friendly-SAM [26] approximates the sharpness objective to reduce computational cost, though sometimes at the expense of accuracy in architectures such as Vision Transformers (ViTs) [8]. GSAM [46] aligns gradients to stabilize updates but requires additional hyperparameters.

ZSharp builds on SAM [9] and ZNorm [44] by introducing statistical filtering into the perturbation step. During the ascent phase, gradients are first standardized within each layer using Z-score normalization, and these standardized values are used to compute a binary mask that identifies components above a given percentile threshold. This mask is then applied to the original gradients, retaining only the top $(1 - Q_p)\%$ of components with the largest deviations from the mean. In this way, ZSharp reduces the influence of noise and small gradients in the ascent step, yielding sparse but targeted perturbations that better capture sharpness-related directions.

ZSharp introduces only one additional hyperparameter, the percentile threshold, without architectural or training modifications. It remains compatible with SAM implementations and base optimizers. Experiments on CIFAR-10 [19], CIFAR-100 [19], and Tiny-ImageNet [24] show improved generalization across ResNet [11], VGG [39], and ViT [8], suggesting that statistically guided filtering is an effective strategy for enhancing sharpness aware optimization in high dimensional or noisy gradient regimes.

Appendix C. Theoretical Analysis: Proof of Convergence

We first present a couple of useful lemmas here. Our analysis borrows the proof structure used in [2, 17], but we adapt it to the ZSharp setting, where the ascent step uses the Z-score filtered gradient $\nabla L(\cdot)_\Omega$ defined in Section 2.2, while the descent step still uses the original gradient $\nabla L(\cdot)$.

Lemma 1 *Given a β-smooth loss function $L(x)$, the following bound holds:*

$$\langle \nabla L(u) - \nabla L(v), u - v \rangle \geq -\beta \|u - v\|^2. \tag{10}$$

Proof By β-smoothness of L, we have

$$\|\nabla L(u) - \nabla L(v)\| \leq \beta \|u - v\|, \quad \forall\, u, v \in \mathbb{R}^d. \tag{11}$$

Multiplying both sides of (11) by $\|v - u\|$ and using $\|u - v\| = \|v - u\|$, we get

$$\|\nabla L(u) - \nabla L(v)\| \cdot \|v - u\| \leq \beta \|u - v\| \cdot \|v - u\| = \beta \|u - v\|^2.$$

By the Cauchy–Schwarz inequality,

$$\langle \nabla L(u) - \nabla L(v), v - u \rangle \leq \|\nabla L(u) - \nabla L(v)\| \cdot \|v - u\| \leq \beta \|u - v\|^2.$$

Finally, multiplying above equation by -1 yields

$$\langle \nabla L(u) - \nabla L(v), u - v \rangle \geq -\beta \|u - v\|^2. \tag{12}$$

■

Lemma 2 *Let L be a β-smooth loss function. At iteration t, let $\nabla L_t(w_t)_\Omega$ be an unbiased stochastic estimator of $\nabla L(w_t)_\Omega$ with bounded variance $\mathbb{E}\Big[\|\nabla L_t(w_t)_\Omega - \nabla L(w_t)_\Omega\|^2\Big] \leq \frac{\sigma_\Omega^2}{b}$ Then, for any $r > 0$,*

$$\mathbb{E}[\langle \nabla L(w_t + r\, \nabla L_t(w_t)_\Omega), \nabla L(w_t)\rangle] \geq \frac{1}{2}\, \mathbb{E}\big[\|\nabla L(w_t)\|^2\big] - \frac{\beta^2 r^2}{2}\, \mathbb{E}\big[\|\nabla L(w_t)_\Omega\|^2\big] - \frac{\beta^2 r^2 \sigma_\Omega^2}{2b}. \tag{13}$$

Proof Define

$$\Delta_t = \nabla L(w_t + r\, \nabla L_t(w_t)_\Omega) - \nabla L(w_t). \tag{14}$$

Then

$$\langle \nabla L(w_t + r\, \nabla L_t(w_t)_\Omega)\,,\ \nabla L(w_t)\rangle = \|\nabla L(w_t)\|^2 + \langle \Delta_t, \nabla L(w_t)\rangle. \tag{15}$$

By Cauchy–Schwarz and Young's inequality,

$$\langle \Delta_t, \nabla L(w_t)\rangle \geq -\frac{1}{2}\|\Delta_t\|^2 - \frac{1}{2}\|\nabla L(w_t)\|^2. \tag{16}$$

Combining the two displays and taking expectations,

$$\mathbb{E}[\langle \nabla L(w_t + r\,\nabla L_t(w_t)_\Omega)\,,\ \nabla L(w_t)\rangle] \geq \frac{1}{2}\,\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] - \frac{1}{2}\,\mathbb{E}\big[\|\Delta_t\|^2\big]\,. \tag{17}$$

By β-smoothness of L,

$$\begin{aligned}\|\Delta_t\| &= \big\|\nabla L(w_t + r\,\nabla L_t(w_t)_\Omega) - \nabla L(w_t)\big\| \\ &\leq \beta r\,\|\nabla L_t(w_t)_\Omega\|.\end{aligned} \tag{18}$$

Squaring and taking expectations,

$$\mathbb{E}\big[\|\Delta_t\|^2\big] \leq \beta^2 r^2\,\mathbb{E}\big[\|\nabla L_t(w_t)_\Omega\|^2\big]\,. \tag{19}$$

By variance decomposition,

$$\mathbb{E}\big[\|\nabla L_t(w_t)_\Omega\|^2\big] = \mathbb{E}\big[\|\nabla L(w_t)_\Omega\|^2\big] + \frac{\sigma_\Omega^2}{b}. \tag{20}$$

Therefore,

$$\mathbb{E}\big[\|\Delta_t\|^2\big] \leq \beta^2 r^2\,\mathbb{E}\big[\|\nabla L(w_t)_\Omega\|^2\big] + \frac{\beta^2 r^2 \sigma_\Omega^2}{b}. \tag{21}$$

Substituting this bound into the earlier inequality gives

$$\mathbb{E}[\langle \nabla L(w_t + r\,\nabla L_t(w_t)_\Omega)\,,\ \nabla L(w_t)\rangle] \geq \frac{1}{2}\,\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] - \frac{\beta^2 r^2}{2}\,\mathbb{E}\big[\|\nabla L(w_t)_\Omega\|^2\big] - \frac{\beta^2 r^2 \sigma_\Omega^2}{2b}, \tag{22}$$

which proves the claim. ■

Lemma 3 *Assume that L is β-smooth and the filtered stochastic gradient $\nabla L_t(w_t)_\Omega$ satisfies the variance bound $\mathbb{E}\big[\|\nabla L_t(w_t)_\Omega - \nabla L(w_t)_\Omega\|^2\big] \leq \frac{\sigma_\Omega^2}{b}$. Consider the ZSharp-SAM updates $w_{t+1/2} = w_t + r\,\nabla L_t(w_t)_\Omega$, and $w_{t+1} = w_{t+1/2} - \eta\,\nabla L_t(w_{t+1/2})$. If the step size satisfies $\eta \leq \frac{1}{4\beta}$ and the ascent radius satisfies $\beta^2 r^2 \leq \frac{1}{4}$, then*

$$\mathbb{E}[L(w_{t+1})] \leq \mathbb{E}[L(w_t)] - \frac{\eta}{4}\,\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] + \frac{2\eta\beta^2 r^2}{b}\,\sigma_\Omega^2 + \frac{\eta^2\beta}{b}\,\sigma_\Omega^2. \tag{23}$$

Proof Define the ascent-step parameter

$$w_{t+1/2} = w_t + r\,\nabla L_t(w_t)_\Omega. \tag{24}$$

By β-smoothness,

$$L(w_{t+1}) \leq L(w_t) - \eta\,\langle \nabla L_t(w_{t+1/2}), \nabla L(w_t)\rangle + \frac{\eta^2\beta}{2}\|\nabla L_t(w_{t+1/2})\|^2. \tag{25}$$

Taking expectations and using $\langle p, q\rangle = \frac{1}{2}(\|p\|^2 + \|q\|^2 - \|p-q\|^2)$ with $p = \nabla L(w_{t+1/2})$ and $q = \nabla L(w_t)$ (and inserting/subtracting the population gradient where needed), we obtain

$$\mathbb{E}[L(w_{t+1})] \leq \mathbb{E}[L(w_t)] - \frac{\eta}{2}\Big(\mathbb{E}\|\nabla L(w_{t+1/2})\|^2 + \mathbb{E}\|\nabla L(w_t)\|^2 - E_1\Big) + \frac{\eta^2\beta}{2}E_2, \tag{26}$$

where

$$E_1 = \mathbb{E}\big\|\nabla L(w_{t+1/2}) - \nabla L(w_t)\big\|^2, \tag{27}$$

$$E_2 = \mathbb{E}\big\|\nabla L_t(w_{t+1/2})\big\|^2. \tag{28}$$

Bounding E_1. By β-smoothness,

$$\|\nabla L(w_{t+1/2}) - \nabla L(w_t)\|^2 \le \beta^2 \|w_{t+1/2} - w_t\|^2 = \beta^2 r^2 \|\nabla L_t(w_t)_\Omega\|^2. \tag{29}$$

Hence

$$\begin{aligned}
E_1 &\le \beta^2 r^2\, \mathbb{E}\|\nabla L_t(w_t)_\Omega\|^2 && (30)\\
&= \beta^2 r^2\, \mathbb{E}\big\|\nabla L_t(w_t)_\Omega - \nabla L(w_t)_\Omega + \nabla L(w_t)_\Omega\big\|^2 && (31)\\
&\le 2\beta^2 r^2\, \mathbb{E}\|\nabla L_t(w_t)_\Omega - \nabla L(w_t)_\Omega\|^2 + 2\beta^2 r^2\, \mathbb{E}\|\nabla L(w_t)_\Omega\|^2 && (32)\\
&\le \frac{2\beta^2 r^2}{b}\sigma_\Omega^2 + 2\beta^2 r^2\, \mathbb{E}\|\nabla L(w_t)\|^2, && (33)
\end{aligned}$$

where we used $\|\nabla L(w_t)_\Omega\|^2 \le \|\nabla L(w_t)\|^2$ since filtering only removes coordinates.

Bounding E_2. We have

$$\begin{aligned}
E_2 &= \mathbb{E}\|\nabla L_t(w_{t+1/2})\|^2 && (34)\\
&\le 2\,\mathbb{E}\|\nabla L_t(w_{t+1/2}) - \nabla L(w_{t+1/2})\|^2 + 2\,\mathbb{E}\|\nabla L(w_{t+1/2})\|^2 && (35)\\
&\le \frac{2\sigma_\Omega^2}{b} + 2\,\mathbb{E}\|\nabla L(w_{t+1/2})\|^2. && (36)
\end{aligned}$$

Using smoothness again with $u = w_t + r\nabla L(w_t)_\Omega$,

$$\|\nabla L(w_{t+1/2}) - \nabla L(u)\| \le \beta r\, \|\nabla L_t(w_t)_\Omega - \nabla L(w_t)_\Omega\|, \tag{37}$$

which implies

$$\|\nabla L(w_{t+1/2})\|^2 \le 2\|\nabla L(u)\|^2 + 2\beta^2 r^2\, \|\nabla L_t(w_t)_\Omega - \nabla L(w_t)_\Omega\|^2. \tag{38}$$

Taking expectations and applying the variance bound,

$$\mathbb{E}\|\nabla L(w_{t+1/2})\|^2 \le 2\|\nabla L(w_t + r\nabla L(w_t)_\Omega)\|^2 + \frac{2\beta^2 r^2}{b}\sigma_\Omega^2. \tag{39}$$

Thus

$$E_2 \le \frac{2\sigma_\Omega^2}{b} + 2\|\nabla L(w_t + r\nabla L(w_t)_\Omega)\|^2 + \frac{2\beta^2 r^2}{b}\sigma_\Omega^2. \tag{40}$$

Putting the bounds together. Substituting the bounds on E_1 and E_2 into the main inequality gives

$$\begin{aligned}
\mathbb{E}[L(w_{t+1})] &\le \mathbb{E}[L(w_t)] - \frac{\eta}{2}\mathbb{E}\|\nabla L(w_{t+1/2})\|^2 - \frac{\eta}{2}\mathbb{E}\|\nabla L(w_t)\|^2 + \frac{\eta}{2}E_1 + \frac{\eta^2\beta}{2}E_2 && (41)\\
&\le \mathbb{E}[L(w_t)] - \frac{\eta}{2}\mathbb{E}\|\nabla L(w_{t+1/2})\|^2 - \eta\Big(\tfrac{1}{2} - \beta^2 r^2\Big)\mathbb{E}\|\nabla L(w_t)\|^2 && (42)\\
&\quad + \frac{\eta\beta^2 r^2}{b}\sigma_\Omega^2 + \frac{\eta^2\beta}{b}\sigma_\Omega^2 + \eta^2\beta\,\mathbb{E}\|\nabla L(w_{t+1/2})\|^2. && (43)
\end{aligned}$$

The coefficient of $\mathbb{E}\|\nabla L(w_{t+1/2})\|^2$ is

$$-\frac{\eta}{2}+\eta^2\beta. \tag{44}$$

For $\eta \leq \frac{1}{4\beta}$ this coefficient is non-positive, so we can drop this term. Moreover, if $\beta^2 r^2 \leq \frac{1}{4}$, then $\frac{1}{2}-\beta^2 r^2 \geq \frac{1}{4}$, and hence

$$\mathbb{E}[L(w_{t+1})] \leq \mathbb{E}[L(w_t)] - \frac{\eta}{4}\,\mathbb{E}\|\nabla L(w_t)\|^2 + \frac{2\eta\beta^2 r^2}{b}\sigma_\Omega^2 + \frac{\eta^2\beta}{b}\sigma_\Omega^2, \tag{45}$$

which proves the claim. ■

Theorem 4 *Assume that L is β-smooth and that the filtered stochastic gradient $\nabla L_t(w_t)_\Omega$ satisfies the variance bound*

$$\mathbb{E}\big[\|\nabla L_t(w_t)_\Omega - \nabla L(w_t)_\Omega\|^2\big] \leq \frac{\sigma_\Omega^2}{b}. \tag{46}$$

All expectations are taken over the mini-batch at iteration t. Suppose the ZSharp-SAM updates

$$w_{t+1/2} = w_t + r\,\nabla L_t(w_t)_\Omega, \tag{47}$$

$$w_{t+1} = w_{t+1/2} - \eta\,\nabla L_t(w_{t+1/2}) \tag{48}$$

use step size η and ascent radius r satisfying $\eta \leq \frac{1}{4\beta}$ and $\beta^2 r^2 \leq \frac{1}{4}$. Then ZSharp-SAM satisfies

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] \leq \frac{4}{T\eta}\big(L(w_0) - \mathbb{E}[L(w_T)]\big) + \frac{8\beta^2 r^2}{b}\,\sigma_\Omega^2 + \frac{4\eta\beta}{b}\,\sigma_\Omega^2. \tag{49}$$

Proof From Lemma 4 (ZSharp-SAM one-step descent bound), for each t we have

$$\mathbb{E}[L(w_{t+1})] \leq \mathbb{E}[L(w_t)] - \frac{\eta}{4}\,\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] + \frac{2\eta\beta^2 r^2}{b}\,\sigma_\Omega^2 + \frac{\eta^2\beta}{b}\,\sigma_\Omega^2. \tag{50}$$

Averaging (50) over $t = 0, \ldots, T-1$ yields

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[L(w_{t+1})] \leq \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[L(w_t)] - \frac{\eta}{4T}\sum_{t=0}^{T-1}\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] + \frac{2\eta\beta^2 r^2}{b}\,\sigma_\Omega^2 + \frac{\eta^2\beta}{b}\,\sigma_\Omega^2. \tag{51}$$

Using the telescoping identity

$$\frac{1}{T}\sum_{t=0}^{T-1}\big(\mathbb{E}[L(w_t)] - \mathbb{E}[L(w_{t+1})]\big) = \frac{1}{T}\big(L(w_0) - \mathbb{E}[L(w_T)]\big), \tag{52}$$

inequality (51) can be rewritten as

$$\frac{\eta}{4T}\sum_{t=0}^{T-1}\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] \leq \frac{1}{T}\big(L(w_0) - \mathbb{E}[L(w_T)]\big) + \frac{2\eta\beta^2 r^2}{b}\,\sigma_\Omega^2 + \frac{\eta^2\beta}{b}\,\sigma_\Omega^2. \tag{53}$$

Dividing both sides of (53) by $\eta/4$ gives

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] \le \frac{4}{T\eta}\big(L(w_0)-\mathbb{E}[L(w_T)]\big)+\frac{8\beta^2 r^2}{b}\,\sigma_\Omega^2+\frac{4\eta\beta}{b}\,\sigma_\Omega^2, \tag{54}$$

which is the claimed bound. ■

Corollary 5 (Convergence with diminishing step sizes) *Assume that L is β-smooth, bounded below by $L_{\inf}$, and $\nabla L_t(w_t)_\Omega$ satisfies $\mathbb{E}\big[\|\nabla L_t(w_t)_\Omega - \nabla L(w_t)_\Omega\|^2\big]\le \frac{\sigma_\Omega^2}{b}$. Consider the ZSharp-SAM updates $w_{t+1/2} = w_t + r_t\,\nabla L_t(w_t)_\Omega$, and $w_{t+1} = w_{t+1/2} - \eta_t\,\nabla L_t(w_{t+1/2})$, with step sizes $\{\eta_t\}_{t\ge 0}$ and ascent radii $\{r_t\}_{t\ge 0}$ satisfying $\eta_t \le \frac{1}{4\beta}\beta^2 r_t^2 \le \frac{1}{4}$for all t, $\sum_{t=0}^{\infty}\eta_t = \infty$,$\sum_{t=0}^{\infty}\eta_t^2 < \infty$,$\sum_{t=0}^{\infty}\eta_t r_t^2 < \infty$. Then*

$$\sum_{t=0}^{\infty}\eta_t\,\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] < \infty, \tag{55}$$

and in particular

$$\liminf_{t\to\infty}\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] = 0. \tag{56}$$

Proof By the same argument as in Lemma 4, but allowing time-varying (η_t, r_t), for each t we obtain

$$\mathbb{E}[L(w_{t+1})] \le \mathbb{E}[L(w_t)] - \frac{\eta_t}{4}\,\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] + \frac{2\eta_t\beta^2 r_t^2}{b}\,\sigma_\Omega^2 + \frac{\eta_t^2\beta}{b}\,\sigma_\Omega^2. \tag{57}$$

Rearranging,

$$\frac{\eta_t}{4}\,\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] \le \mathbb{E}[L(w_t)] - \mathbb{E}[L(w_{t+1})] + \frac{2\eta_t\beta^2 r_t^2}{b}\,\sigma_\Omega^2 + \frac{\eta_t^2\beta}{b}\,\sigma_\Omega^2. \tag{58}$$

Summing from $t = 0$ to $T-1$ gives

$$\frac{1}{4}\sum_{t=0}^{T-1}\eta_t\,\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] \le \mathbb{E}[L(w_0)] - \mathbb{E}[L(w_T)] + \frac{2\beta^2\sigma_\Omega^2}{b}\sum_{t=0}^{T-1}\eta_t r_t^2 + \frac{\beta\sigma_\Omega^2}{b}\sum_{t=0}^{T-1}\eta_t^2. \tag{59}$$

Using $L(w_T) \ge L_{\inf}$ and letting $T\to\infty$, the right-hand side is bounded by

$$\mathbb{E}[L(w_0)] - L_{\inf} + \frac{2\beta^2\sigma_\Omega^2}{b}\sum_{t=0}^{\infty}\eta_t r_t^2 + \frac{\beta\sigma_\Omega^2}{b}\sum_{t=0}^{\infty}\eta_t^2 < \infty, \tag{60}$$

by the assumptions on $\{\eta_t\}$ and $\{r_t\}$. Hence

$$\sum_{t=0}^{\infty}\eta_t\,\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] < \infty. \tag{61}$$

Because $\sum_{t=0}^{\infty}\eta_t = \infty$, this implies

$$\liminf_{t\to\infty}\mathbb{E}\big[\|\nabla L(w_t)\|^2\big] = 0, \tag{62}$$

which completes the proof. ■