Sparsity-Adaptive Sharpness-Aware Minimization

Shiryu Ueno
Faculty of Engineering
Gifu University
Gifu, Japan
ueno@cv.info.gifu-u.ac.jp

Yoshikazu Hayashi
Faculty of Engineering
Gifu University
Gifu, Japan
hayashi.yoshikazu.a8@f.gifu-u.ac.jp

Kunihito Kato
Faculty of Engineering
Gifu University
Gifu, Japan
kato.kunihito.k6@f.gifu-u.ac.jp

***Abstract*—Deploying deep neural networks in real-world settings requires models that are both compact and robust to common corruptions. However, at deployment-relevant high sparsity, standard pruning pipelines often degrade corruption robustness, and existing sharpness-aware training/pruning approaches provide limited robustness gains. We address this issue by introducing Sparsity-Adaptive Sharpness-Aware Minimization (SA-SAM), which derives a sparsity-dependent SAM/ASAM perturbation radius by keeping the mean absolute perturbation (an ℓ_1-based proxy) approximately invariant as sparsity increases. As a simple complementary option, we evaluate Magnitude-Weighted Hessian (MWH), derived from a second-order removal-path analysis, yielding an importance proportional to $\mathrm{Diag}(F)_i\,|w_i|$, where $\mathrm{Diag}(F)$ is the diagonal empirical Fisher used as a curvature proxy in our implementation. Across CIFAR-10-C, CIFAR-100-C, and ImageNet-100-C, our approach achieved stronger corruption robustness than the considered pruning baselines at 80–90% sparsity, while preserving clean accuracy. We additionally quantify the robustness–throughput trade-off by reporting measured inference throughput under sparse execution at deployment-relevant sparsity levels.**

***Index Terms*—Pruning, corruption robustness, sharpness-aware minimization, curvature (Hessian/Fisher)**

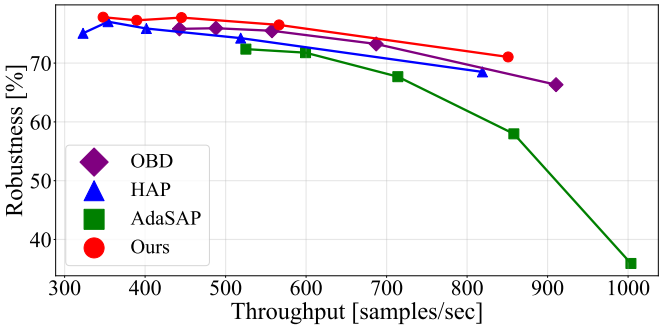

Fig. 1. **Throughput vs. robustness on CIFAR-10-C.** The x-axis shows throughput (samples/sec) and the y-axis shows mean Top-1 accuracy (%) on CIFAR-10-C averaged over all corruption types and severities. Throughput is averaged over 10 runs (full passes over the test set). Our method achieves higher corruption robustness than HAP and AdaSAP at comparable throughput (see legend). Note that at a fixed global sparsity, measured throughput can vary across methods due to differences in layer-wise nonzero distributions induced by different pruning criteria.

I. Introduction

Pruning [1] is a major approach to model compression that removes parameters deemed less important by an importance score. While most pruning methods primarily target preserving clean accuracy, they can compromise robustness to corruptions that frequently arise in deployment (e.g., the CIFAR-C/ImageNet-C setting [2]). Recent efforts therefore incorporate robustness into compression. For example, AdaSAP [3] leverages Adaptive Sharpness-Aware Minimization (ASAM) [4] to bias training toward flatter minima, which are empirically associated with improved robustness to weight or input perturbations [5], [6]. Hessian-Aware Pruning (HAP) [7] uses curvature information to estimate parameter sensitivity. Despite these advances, achieving strong corruption robustness at deployment-relevant high unstructured sparsity (e.g., 80–90%) remains challenging. First, because pruning constrains perturbations to the surviving subspace, a fixed SAM/ASAM radius can implicitly shrink the mean absolute perturbation over the full parameter vector (treating pruned coordinates as zeros) as sparsity increases. Motivated by keeping the mean absolute perturbation (an ℓ_1-based proxy) approximately invariant under sparsity, we derive a simple sparsity-dependent radius schedule that stabilizes sharpness-aware exploration during pruning. Furthermore, second-order pruning methods such as HAP [7] leverage Hessian information to estimate parameter sensitivity; however, at high sparsity the specific coupling between curvature and parameter scale can significantly affect which parameters are preserved. As a simple complementary option, we evaluate Magnitude-Weighted Hessian (MWH), derived from a second-order removal-path analysis, yielding an importance proportional to $\mathrm{Diag}(F)_i\,|w_i|$, where $\mathrm{Diag}(F)$ is the diagonal empirical Fisher used as a curvature proxy in our implementation. These limitations motivate stabilizing sharpness-aware exploration under sparsity, which we focus on via SA-SAM; we also study the impact of a simple curvature–magnitude importance variant (MWH) as an optional component.

In this work, we propose Sparsity-Adaptive Sharpness-Aware Minimization (SA-SAM) as an optimization method for pruning. SA-SAM extends sharpness-aware training to sparse models by dynamically adjusting the perturbation radius according to the model's current sparsity, aiming to keep the effective exploration behavior stable throughout pruning. Furthermore, we evaluate Magnitude-Weighted Hessian (MWH) as an optional importance score that combines curvature with weight magnitude in high-sparsity pruning. We follow a standard gradual pruning pipeline that iteratively updates a pruning

mask using an importance score and retrains the surviving weights; SA-SAM is a drop-in replacement for sharpness-aware optimization during pruning, and MWH is an optional drop-in replacement for the importance score. As illustrated in Fig. 1, our method improves the robustness–throughput trade-off on CIFAR-10-C at 80–90% sparsity. Across CIFAR-10-C, CIFAR-100-C, and ImageNet-100-C, we observe consistent robustness gains at these sparsity levels while maintaining competitive clean accuracy. We further provide Fourier-domain and loss-landscape analyses (primarily on CIFAR-10) that help explain the observed robustness improvements.

Our contributions are summarized as follows:

- **SA-SAM:** We propose a sparsity-adaptive sharpness-aware optimization method that calibrates the perturbation radius according to current sparsity, stabilizing sharpness-aware exploration during pruning.
- **Optional importance-score variant (MWH):** We evaluate a simple curvature–magnitude importance score, proportional to $\mathrm{Diag}(F)_i\,|w_i|$ (diagonal empirical Fisher as a curvature proxy), as an optional variant of second-order pruning criteria for high-sparsity robustness.
- **Robust high-sparsity pruning:** We demonstrate corruption-robustness gains at 80–90% sparsity on CIFAR-10-C, CIFAR-100-C, and ImageNet-100-C, and provide a robustness–throughput analysis highlighting deployment-relevant trade-offs.

Note. The implementation code is publicly available at https://github.com/ia-gu/Sparsity-Adaptive-Sharpness-Aware-Minimization.

II. Related Work

A. Pruning and robustness under common corruptions

Pruning compresses neural networks by removing weights (unstructured) or groups of weights (structured) based on an importance score [1]. Representative criteria include magnitude-based pruning [8] and gradient-based saliency methods such as SNIP [9], as well as second-order criteria derived from local curvature. While many pruning methods focus on preserving clean accuracy, robustness to distribution shifts and common corruptions (e.g., CIFAR-C/ImageNet-C [2]) can degrade substantially at high sparsity. This motivates incorporating robustness-aware objectives or sensitivity estimates into pruning pipelines. Second-order pruning traces back to Optimal Brain Damage (OBD), which derives a saliency proportional to $H_{ii}w_i^2$ under a diagonal Hessian approximation [10]. Hessian-Aware Pruning (HAP) [7] modernizes such ideas via efficient curvature estimation for pruning at scale.

B. Sharpness-aware optimization and its variants

Deep networks are typically optimized with first-order methods such as SGD or Adam, which minimize empirical risk. However, minimizing training loss alone can converge to sharp minima that are sensitive to perturbations and may generalize poorly under distribution shifts. Sharpness-Aware Minimization (SAM) [11] addresses this by optimizing parameters that minimize the worst-case loss within a neighborhood in *weight space*, implemented by (i) applying an ascent perturbation toward the local worst direction and (ii) updating the original weights using gradients computed at the perturbed point. Unlike adversarial training [12], which targets worst-case perturbations in *input space* and can induce a stronger clean-accuracy trade-off, SAM perturbs weights and is frequently observed to retain competitive clean accuracy while improving robustness in practice [13]. Several variants improve SAM's stability and efficiency. ASAM [4] rescales perturbations based on parameter magnitudes to achieve approximate scale invariance, and SSAM [14] stabilizes updates by normalizing the sharpness-aware gradient to match the norm of the corresponding SGD gradient. LookSAM [15] reduce computational overhead via approximation or scheduling. However, these methods are primarily designed for *dense* models with a fixed effective dimension. In gradual pruning, the feasible perturbations are constrained to the surviving subspace; with a fixed global radius, the average per-parameter perturbation magnitude can implicitly shrink as sparsity increases, weakening the sharpness-aware effect at deployment-relevant high sparsity. Our SA-SAM directly targets this sparsity-induced shrinkage by adapting the global SAM/ASAM radius as a function of the current sparsity.

III. Preliminaries

A. Sharpness-Aware Minimization

In image classification, given a training dataset $\mathcal{D}$ with input images $\mathbf{X}$ and labels $\mathbf{Y}$, a model learns by minimizing the loss ℓ with respect to parameters $\mathbf{W} \in \mathbb{R}^{m\times n}$ as follows:

$$\min_{\mathbf{W}} \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim\mathcal{D}}\left[\ell(f(\mathbf{X};\mathbf{W}),\mathbf{Y})\right] \tag{1}$$

SAM [11] formulates learning as a bi-level optimization problem, minimizing the worst-case local loss in the weight space simultaneously:

$$\min_{\mathbf{W}} \max_{\|\boldsymbol{\epsilon}\|_2\le\rho} \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim\mathcal{D}}\big[\ell\big(f(\mathbf{X};\,\mathbf{W}+\boldsymbol{\epsilon}),\mathbf{Y}\big)\big], \tag{2}$$

where $\rho > 0$ is the radius of the perturbation ball (ℓ_2-ball) defined in the weight space, and it is a hyperparameter. Using a first-order Taylor approximation of the inner objective, an approximate maximizer can be written as:

$$\boldsymbol{\epsilon}^\star \;=\; \rho\,\frac{\nabla_{\mathbf{W}}\ell}{\|\nabla_{\mathbf{W}}\ell\|_2} \tag{3}$$

This enables an easy two-step implementation using standard optimizers, "(i) compute perturbation → (ii) update gradients." However, since SAM employs the same radius for all parameters, the sharpness is not invariant to weight re-scaling transformations $\mathbf{W} \mapsto a\mathbf{W}$, leading to scale-dependency issues. ASAM [4] addresses this limitation by introducing a parameter-dependent re-scaling in the constraint. Specifically, the perturbation is constrained by:

$$\|\mathbf{T}_{\mathbf{W}}^{-1}\,\boldsymbol{\epsilon}\|_2 \;\le\; \rho, \qquad \mathbf{T}_{\mathbf{W}} \;=\; \mathrm{diag}\big(|\mathbf{W}|+\eta\big), \tag{4}$$

where $\eta > 0$ is a small constant for numerical stability. The optimal perturbation then becomes:

$$\boldsymbol{\epsilon}^\star \;=\; \rho\,\frac{\mathbf{T}_{\mathbf{W}}^2\,\nabla_{\mathbf{W}}\ell}{\left\|\mathbf{T}_{\mathbf{W}}\,\nabla_{\mathbf{W}}\ell\right\|_2} \tag{5}$$

In this formulation, dimensions with larger parameter values admit proportionally larger perturbations, making the sharpness measure approximately invariant to weight re-scaling, thus alleviating the scale-dependency issue observed in SAM.

B. Pruning

In pruning, since the mask $\mathbf{M} \in \{0,1\}^{m\times n}$ is applied to $\mathbf{W}$, Eq. (1) can be written as:

$$\min_{\mathbf{W},\mathbf{M}} \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim\mathcal{D}}\left[\ell(f(\mathbf{X};\mathbf{W}\odot\mathbf{M}),\mathbf{Y})\right] \tag{6}$$

Here, $\odot$ represents Hadamard product. In pruning, the importance score I is designed to generate a mask satisfying the sparsity constraints. Denoting each element of $\mathbf{W}$ as $w_{i,j}$, the mask matrix is defined by calculating the importance $I(w_{i,j})$ for each parameter:

$$M_{i,j} = \begin{cases} 1, & I(w_{i,j}) \geq \tau \\ 0, & I(w_{i,j}) < \tau \end{cases} \tag{7}$$

We choose τ such that exactly the top $(1-s)mn$ parameters (by I) are kept.

IV. Method

A. Sparsity-Adaptive Sharpness-Aware Minimization

Let $W \in \mathbb{R}^d$ denote the vectorized *prunable* weight vector (convolution and linear weights) and let $m \in \{0,1\}^d$ be the corresponding elementwise pruning mask; we write $M = \mathrm{diag}(m)$ so that masking is $MW = W \odot m$. BatchNorm parameters and biases are kept dense in all experiments and are omitted from W in this derivation for simplicity. Let $S = \{i : m_i = 1\}$ be the support with $d_s = |S| = (1-s)d$, where s denotes the sparsity over these prunable weights, and let $\Pi_S = M$ be the projector onto the coordinate subspace $\mathrm{range}(\Pi_S) \subset \mathbb{R}^d$. The inner maximization must be solved on that subspace. Throughout this subsection, we abuse notation and write $\ell(W)$ as shorthand for $\ell(f(X; W \odot m), Y)$ when the pruning mask m is fixed. To avoid inverting a singular matrix outside S, we use the Moore–Penrose pseudoinverse:

$$\max_{\epsilon_S \in \mathrm{range}(\Pi_S)} \langle \nabla_W \ell(W),\, \epsilon_S\rangle \quad \text{s.t.} \quad \left\|(\,\Pi_S T_W \Pi_S\,)^{+}\epsilon_S\right\|_2 \leq \rho. \tag{8}$$

Here T_W denotes the diagonal ASAM scaling matrix; for SAM, set $T_W = I$. Solving the Lagrangian of the linearized inner problem yields the closed-form perturbation in $\mathrm{range}(\Pi_S)$:

$$\epsilon^\star = \rho\,\frac{\Pi_S T_W^2 \nabla_W \ell}{\|\Pi_S T_W \nabla_W \ell\|_2}, \tag{9}$$

which coincides with the heuristic formula obtained by simply zeroing pruned coordinates, but Eq. (8) and Eq. (9) make the subspace and invertibility explicit.

Pruning reduces the dimension $d \to d_s$ and can shrink the per-parameter step implied by Eq. (9). We quantify the global step size by the mean absolute step $\Phi(\epsilon) = \frac{1}{d}\|\epsilon\|_1$, which measures the average absolute perturbation per parameter in the *original* dense parameterization. This choice makes the effective sharpness-aware exploration comparable across sparsity levels when pruned coordinates are treated as zeros. Let $t = \mathrm{diag}(T_W) \in \mathbb{R}^d_+$ and define the averages $\bar{t} = \frac{1}{d}\sum_i t_i$ and $\bar{t}_S = \frac{1}{d_s}\sum_{i\in S} t_i$. Write $u = \frac{\Pi_S T_W \nabla_W \ell}{\|\Pi_S T_W \nabla_W \ell\|_2} \in \mathbb{S}^{d_s-1}$. Under the standard exchangeability/isotropy approximation for u, $\mathbb{E}[|u_1|] = c(d_s) \approx \sqrt{2/(\pi d_s)}$, and we obtain

$$\mathbb{E}[\Phi(\epsilon^\star) \mid s] = \rho\,(1-s)\,\bar{t}_S\,c(d_s). \tag{10}$$

Relative to the dense case ($s = 0$), the ratio is

$$\frac{\mathbb{E}[\Phi(\epsilon^\star) \mid s]}{\mathbb{E}[\Phi(\epsilon^\star) \mid 0]} = (1-s)\,\frac{\bar{t}_S}{\bar{t}}\,\frac{c(d_s)}{c(d)} \;\approx\; \sqrt{1-s}\cdot\frac{\bar{t}_S}{\bar{t}}. \tag{11}$$

To maintain invariant step size, we set

$$\begin{aligned} \rho^\star(s) &= \rho\cdot\alpha(s), \\ \alpha(s) &:= \frac{c(d)}{(1-s)\,c(d_s)}\cdot\frac{\bar{t}}{\bar{t}_S} \;\approx\; \frac{1}{\sqrt{1-s}}. \end{aligned} \tag{12}$$

In practice we use the simple approximation, which depends only on s (available from the mask) and introduces no extra schedules. For SAM, set $T_W = I$ in Eqs. (8)–(12).

Remarks. The exchangeability/isotropy assumption for u provides a tractable approximation of typical coordinate magnitudes; in practice, we found that using the approximation $\alpha(s) \approx 1/\sqrt{1-s}$ works well. Calibrating ρ with Eq. (12) (or its approximation) compensates for the reduced effective dimension and mitigates the shrinkage of sharpness-aware perturbations at high sparsity.

B. Optional importance-score: Magnitude-Weighted Hessian

In our experiments, we approximate the Hessian diagonal H_{ii} with a non-negative curvature proxy, namely the diagonal empirical Fisher $\mathrm{Diag}(F)_i$, and we use the term "curvature" to refer to this proxy. Magnitude-based methods rank parameters by $|w_i|$, while curvature-based methods use second-order information. Following a second-order removal-path analysis, we consider removing a weight along $w_i(t) = (1-t)w_i,\ t \in [0,1]$, and use the quadratic expansion

$$\Delta\ell(t) \approx -t\,w_i\,\nabla_i\ell \;+\; \tfrac{1}{2}\,t^2\,H_{ii}\,w_i^2. \tag{13}$$

Assuming the first-order term is small (as in OBD), the second-order approximation predicts the loss increase from full removal ($t = 1$) as $\Delta\ell(1) \approx \frac{1}{2}H_{ii}w_i^2$. Normalizing by the removal-path length $|w_i|$ yields the per-unit-removal penalty

$$\frac{1}{|w_i|}\cdot\frac{1}{2}H_{ii}|w_i|^2 = \frac{1}{2}\,H_{ii}\,|w_i| \;\propto\; H_{ii}\,|w_i|. \tag{14}$$

Curvature proxy (diagonal Fisher). In our implementation, we use the diagonal empirical Fisher as a practical, non-negative proxy for curvature. Concretely, when computing the score we replace H_{ii} in Eq. (14) with $\mathrm{Diag}(F)_i$. Let $g(\mathbf{x},\mathbf{y}) = \nabla_W \ell(f(\mathbf{x};W),\mathbf{y})$ denote the gradient on a

sample. Let $g_b = \nabla_W \ell(f(x_b; W), y_b)$ denote the gradient of the per-example loss for a sample b. We estimate the diagonal empirical Fisher using squared gradients computed at mask-update time. (A) If per-example gradients $\{g_b\}_{b\in B}$ are available, we estimate $\mathrm{Diag}(F)$ by averaging the elementwise squared per-example gradients over a single mini-batch B. (B) Otherwise, we use the squared mini-batch gradient as a proxy, i.e., $\mathrm{Diag}(F) := g_B \odot g_B$ where $g_B = \nabla_W \frac{1}{|B|} \sum_{b\in B} \ell_b$. In all experiments reported in this paper, we use option (B), i.e., the squared mini-batch gradient proxy, for efficiency. To keep the curvature proxy consistent across pruning iterations, we use a single fixed mini-batch B (sampled once per run/seed) and reuse it for all mask updates.

Discarding constant factors, our MWH importance score is

$$I_i \;=\; \mathrm{Diag}(F)_i \cdot |w_i|. \tag{15}$$

Given a target sparsity s, we keep the top $(1-s)$ fraction ranked by Eq. (15) and zero the rest:

$$m_i = \mathbf{1}\big[I_i \text{ is among the top } (1-s)d\big], \qquad W \leftarrow W \odot m. \tag{16}$$

Computationally, SA-SAM has the same per-step overhead as ASAM, since it only rescales the global radius as a function of sparsity, and MWH reuses squared-gradient statistics computed at mask-update time.

V. Experiments

A. Settings

Datasets. We evaluate on CIFAR-10, CIFAR-100 [16] and ImageNet-100 [17]; robustness is assessed on CIFAR-10-C, CIFAR-100-C and ImageNet-100-C [2]. ImageNet-100 is a 100-class subset of ImageNet [18] used to reduce computational cost; ImageNet-100-C is constructed by restricting ImageNet-C to the same 100 classes using the synset list of ImageNet-100; concretely, we filter the ImageNet-C validation images by the selected synset IDs and then evaluate using the standard ImageNet-C corruption protocol. Unless otherwise noted, we report Top-1 accuracy on the clean test set and mean Top-1 accuracy over all corruptions and severities. For CIFAR-10-C and CIFAR-100-C we average over 19 corruption types and 5 severity levels, while for ImageNet-100-C we follow the ImageNet-C protocol (15 corruption types, 5 severity levels).

Architectures. ResNet-18 for CIFAR-10 and CIFAR-100, ResNet-50 for ImageNet-100. **Methods compared.** We study two orthogonal design choices in gradual pruning: (i) the sharpness-aware optimizer used during the retraining steps (ASAM vs. SA-SAM), and (ii) the importance score used for mask updates (OBD/HAP/AdaSAP vs. MWH). Unless otherwise stated, **Ours** denotes **SA-SAM + MWH**, i.e., we apply SA-SAM during pruning retraining and use MWH for mask updates. Unless otherwise stated, **Ours** denotes **SA-SAM + MWH**. To isolate the optimizer effect, we additionally report ASAM $\rightarrow$ SA-SAM swaps while keeping the pruning criterion fixed (Table IV). For MWH, we use the diagonal empirical Fisher (approximated by a squared mini-batch gradient; see Sec. IV-B) as a practical curvature proxy and rank parameters by $\mathrm{Diag}(F)_i\,|w_i|$ (Eq. (15)). To isolate the effect of sparsity-adaptive radius scaling, we keep the base optimizer and training schedule fixed across methods; SA-SAM introduces no additional tunable hyperparameters beyond the initial radius ρ_0 used in ASAM.

We emphasize that our experimental focus is corruption robustness under deployment-relevant high unstructured sparsity. We therefore compare against pruning pipelines that explicitly incorporate curvature-aware mechanisms in this context (e.g., AdaSAP and HAP), along with a classical diagonal-Hessian baseline (OBD). A broader comparison against generic saliency criteria (e.g., Fisher/Taylor-style variants) is beyond the scope of this paper and left as future work.

Optimization. All methods use learning rate 0.1, momentum 0.9, weight decay 0.0005, batch size 128, and a base sharpness radius of $\rho_0 = 0.5$. For SA-SAM, we set $\rho(s) = \rho_0\alpha(s)$ with $\alpha(s) \approx 1/\sqrt{1-s}$ following Eq. (12). We use a step learning-rate schedule that multiplies the learning rate by 0.2 at epochs 120 and 160. Dense, HAP and AdaSAP use ASAM; our method uses SA-SAM with the same base optimizer and hyperparameters. For ASAM, we use the standard stabilizer $\eta = 0.01$ in $\mathbf{T_W} = \mathrm{diag}(|\mathbf{W}| + \eta)$.

Pruning. For all pruning methods, we perform gradual pruning for 220 epochs. At each epoch, we update the mask to match a target sparsity that increases according to a fixed schedule until reaching the final sparsity $s \in \{0.8, 0.9\}$. Specifically, at epoch $t \in \{1, \ldots, T\}$ we set $s_t = s_{\text{final}} \cdot (t/T)^3$. At the beginning of each epoch, we compute the importance scores on the current model, update the mask to keep the top $(1-s_t)$ fraction, and then train for one epoch while keeping the updated mask fixed. We prune convolution and linear weights with unstructured masks. Throughout the paper, the reported global sparsity s is computed over the prunable convolution and linear weights only; BatchNorm parameters and biases are excluded. For a fair comparison, we run all methods under the same gradual-pruning schedule, training budget (epochs), data pipeline, and base optimizer settings; only the sharpness-aware optimizer (ASAM vs. SA-SAM) and/or the importance score differ as specified.

Throughput measurement. We measure inference throughput (samples/sec) on an NVIDIA RTX 3090. All throughput numbers are measured under the same COO sparse-execution backend for consistency across sparsity levels; Dense@0.0 is included as a reference under this backend. We report mean $\pm$ std over 10 runs (after warm-up) with batch size 128. Exact software versions and implementation details will be provided in the released code. Throughput numbers are backend- and batch-size-dependent; we report them to provide a controlled robustness–throughput comparison under a single COO sparse-execution backend, rather than as absolute edge-device latency.

Unless otherwise stated, Tables I–II use ASAM for Dense/HAP/AdaSAP and SA-SAM for Ours; Section V-D additionally reports ASAM vs. SA-SAM ablations for each pruning method.

TABLE I

CIFAR-10 / CIFAR-100. TOP-1 (%) ON THE CLEAN TEST SETS AND ON THE CORRUPTION SETS (MEAN OVER 19 CORRUPTION TYPES × 5 SEVERITIES). VALUES ARE MEAN ± STD OVER 3 SEEDS. DENSE DENOTES THE BASELINE WITH SPARSITY s=0.0. THE RIGHTMOST COLUMN REPORTS INFERENCE THROUGHPUT (SAMPLES/SEC). **OURS** USES SA-SAM DURING PRUNING RETRAINING AND MWH FOR MASK UPDATES.

Method@s	CIFAR-10	CIFAR-10-C	CIFAR-100	CIFAR-100-C	Throughput
Dense@0.0	96.12 ± 0.03	78.19 ± 0.46	79.93 ± 0.05	53.37 ± 0.44	200 ± 0
OBD@0.8	93.86 ± 0.10	75.82 ± 0.54	76.50 ± 0.30	48.61 ± 0.41	446 ± 11
HAP@0.8	95.29 ± 0.12	75.08 ± 0.80	78.46 ± 0.21	48.97 ± 0.66	316 ± 6
AdaSAP@0.8	93.59 ± 0.16	72.38 ± 0.96	72.72 ± 0.13	45.64 ± 0.71	689 ± 12
Ours@0.8	**96.00 ± 0.07**	**77.81 ± 0.53**	**79.49 ± 0.05**	**52.63 ± 0.53**	535 ± 7
OBD@0.9	93.64 ± 0.03	75.50 ± 0.53	75.19 ± 0.37	46.32 ± 0.57	1310 ± 90
HAP@0.9	95.76 ± 0.08	75.88 ± 0.81	77.01 ± 0.54	47.22 ± 0.76	1443 ± 118
AdaSAP@0.9	89.62 ± 0.13	67.69 ± 0.81	61.41 ± 0.09	37.70 ± 0.46	1680 ± 167
Ours@0.9	**96.12 ± 0.10**	**77.74 ± 0.53**	**78.80 ± 0.14**	**51.16 ± 0.56**	1423 ± 116

TABLE II

IMAGENET-100. TOP-1 (%) ON THE CLEAN TEST SET AND THE CORRUPTION SET.

Method@s	ImageNet-100	ImageNet-100-C	Throughput
Dense@0.0	82.91 ± 0.30	44.58 ± 0.69	173 ± 3
OBD@0.8	81.81 ± 0.42	40.86 ± 1.18	365 ± 3
HAP@0.8	81.27 ± 0.53	42.13 ± 0.53	381 ± 14
AdaSAP@0.8	77.51±0.57	35.74 ± 0.82	426 ± 17
Ours@0.8	**83.14 ± 0.06**	**44.45 ± 0.50**	387 ± 15
OBD@0.9	80.93 ± 0.56	39.59 ± 1.47	420 ± 17
HAP@0.9	80.63 ± 0.36	39.36 ± 1.21	442 ± 19
AdaSAP@0.9	72.30 ± 0.47	31.27 ± 0.90	512 ± 25
Ours@0.9	**82.80 ± 0.16**	**43.17 ± 0.47**	449 ± 19

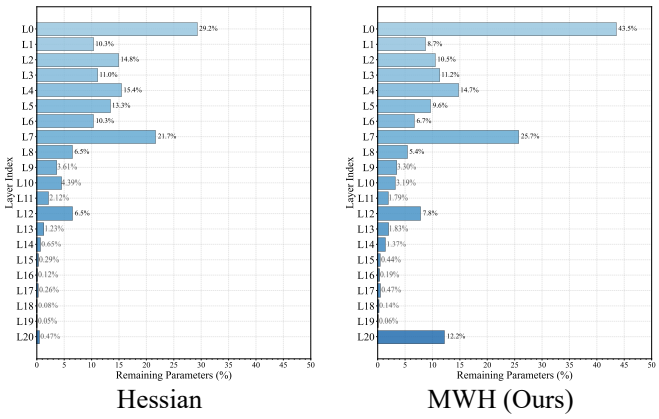

Fig. 2. **Layer-wise remaining parameter ratios at a fixed global sparsity (ResNet-18 on CIFAR-10 at $s = 0.9$).**

B. Experimental Results

Tables I and II summarize clean and corruption accuracy at s=0.8, 0.9. Overall, **Ours (SA-SAM + MWH)** improves corruption robustness over HAP and AdaSAP while preserving clean Top-1. The gains are more pronounced at $s = 0.9$: on CIFAR-10-C, we improve mean corruption accuracy from 75.88 (HAP) / 67.69 (AdaSAP) to 77.74 while matching clean accuracy (96.12), and achieve comparable throughput to HAP (1423 vs. 1443 samples/sec). On CIFAR-100-C at $s = 0.9$, we improve corruption accuracy from 47.22 (HAP) / 37.70 (AdaSAP) to 51.16. On ImageNet-100-C at $s = 0.9$, we improve corruption accuracy from 39.36 (HAP) / 31.27 (AdaSAP) to 43.17 with clean Top-1 of 82.80. We also report measured throughput in Tables I–II; because different pruning criteria induce different layer-wise nonzero patterns, throughput can differ even at the same global sparsity (see Fig. 2). To disentangle the contributions of the optimizer and the pruning criterion, Table IV further reports optimizer-only swaps (ASAM → SA-SAM) on ImageNet-100 and shows consistent gains in corruption robustness, while indicating that MWH can also help under a fixed optimizer.

C. Analysis

Fourier heat map. Fourier heat map analysis examines model robustness from a frequency domain perspective. We perturb test images using single-frequency Fourier-basis noise and measure the classification error. Fourier heat map is evaluated both qualitatively through visualization and quantitatively using the performance degradation rate $D_{i,j}$:

$$D_{i,j} = \frac{E_{\text{pruned}(i,j)} - E_{\text{dense}(i,j)}}{E_{\text{dense}(i,j)}} \times 100 \quad (\%) \qquad (17)$$

Finally, the robustness of each model is obtained by calculating $D_{i,j}$ for all coordinates and then averaging them. Furthermore, to examine the tendency for each frequency band, we define the range satisfying $|i| + |j| \leq 8$ as "Low" frequency and the rest as "High" frequency.

Fig. 3 visualizes the Fourier heat maps, and Table III reports the corresponding degradation rates. The central area indicates low-frequency performance, while the outer area shows high-frequency performance. Blue indicates higher performance, red indicates lower performance. The table shows degradation rates for Low, High, and Average frequencies, where smaller values indicate better maintenance of Dense performance. From the figure and the table, our method improved the model robustness over the baselines. While our Fourier-domain analysis is conducted on CIFAR-10 (32×32), its trend—notably the smaller average degradation in Table III—was consistent with the end-to-end robustness gains observed on ImageNet-100-C in Table II. Many common corruptions in ImageNet-

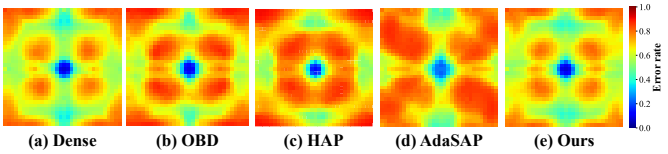

Fig. 3. **Fourier heat maps on CIFAR-10** for (a) Dense, (b) OBD, (c) HAP, (d) AdaSAP, and (e) Ours. Methods (b)-(e) used sparsity $s = 0.9$. Our method shows smaller degradation in the low-frequency region compared to other sparse baselines.

TABLE III
FREQUENCY-WISE PERFORMANCE DEGRADATION FOR THE SPARSE MODELS ($s = 0.9$). EACH VALUE IS THE DEGRADATION RATE DEFINED IN EQ. (17), LOWER IS BETTER.

Method	Low↓	High↓	Average↓
OBD	9.45	3.80	5.02
HAP	10.45	3.52	5.01
AdaSAP	12.46	15.29	14.68
Ours	**1.94**	**0.13**	**0.52**

100-C contain significant high-frequency components, and the improved frequency-wise stability we observe on CIFAR may be related to the end-to-end robustness gains at larger resolution; verifying this with a systematic Fourier analysis on ImageNet-100 is future work.

Loss landscape and Eigenvalue Spectrum Density. Loss landscape [19] visualizes the geometry around a model's solution by applying random perturbations to the trained parameters and plotting the resulting loss (or error rate). We perturb the trained weights along two random directions and plot the resulting error-rate surface; we use the same plotting protocol for all methods. We evaluate the geometry quantitatively via the Hessian Eigenvalue Spectrum Density (ESD) [20]. ESD characterizes the distribution of Hessian eigenvalues around a trained solution. We do not form the full Hessian explicitly; instead, we estimate the spectrum using Hessian–vector products and a Lanczos-based procedure following standard practice. We approximate the top 100 eigenvalues $\{\lambda_i\}_{i=1}^{100}$ via Lanczos iterations and report the ℓ_2 energy $\left(\sum_{i=1}^{100} \lambda_i^2\right)^{1/2}$ as a proxy for curvature magnitude.

Fig. 4 and Fig. 5 show Loss Landscapes and ESDs, respectively. From Fig. 4, our method yielded a flatter loss landscape than existing methods, which means that our model performs well on both the clean and corruption sets, similar to the results in Tables I and II. From Fig. 5, we observed the same tendency. The reported value ($\|\mathbf{H}\|_F$) indicates that our model yields the smallest curvature energy among the sparse models.

D. Ablations

To demonstrate the effectiveness of the proposed optimizer SA-SAM, we compare ASAM with SA-SAM for all pruning methods. For each method, we keep the pruning criterion, pruning schedule, and training budget fixed, and only replace the sharpness-aware optimizer (ASAM → SA-SAM) by scaling the global radius using Eq. (12). Experimental settings are

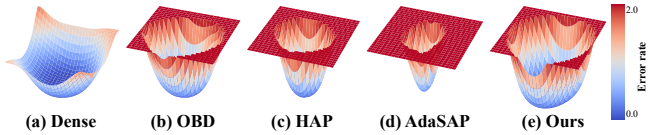

Fig. 4. **Loss landscapes on CIFAR-10** for (a) Dense, (b) OBD, (c) HAP, (d) AdaSAP, and (e) Ours. Methods (b)-(e) use sparsity $s = 0.9$. Each surface is obtained by applying perturbations along two random directions in weight space and plotting the resulting error rate.

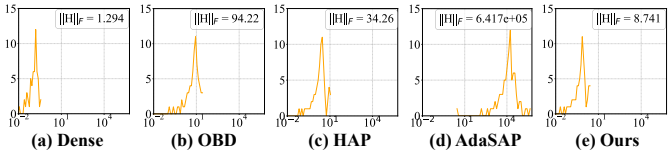

Fig. 5. **ESDs (λ=100) on CIFAR-10** for (a) Dense, (b) OBD, (c) HAP, (d) AdaSAP, and (e) Ours. Methods (b)-(e) use sparsity $s = 0.9$. The x-axis shows eigenvalues and the y-axis shows density. The ℓ_2 energy of the top-100 eigenvalues, $\left(\sum_{i=1}^{100} \lambda_i^2\right)^{1/2}$, is reported in the top-right corner of each plot.

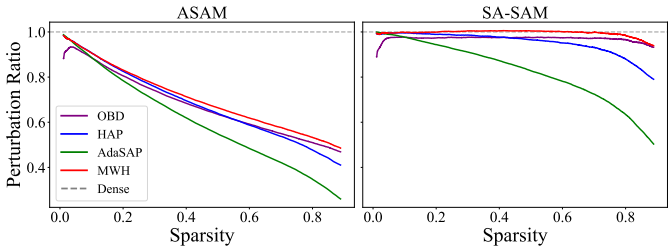

Fig. 6. **Perturbation magnitude ratio** relative to the dense model across different pruning methods. The gray dashed line represents the dense model.

the same as the main settings. We omit throughput because all settings use the same target sparsity and unstructured masks; throughput differences across optimizers are negligible and not the focus of this ablation.

Table IV shows the result of the ablation experiments. As shown in the table, SA-SAM consistently improves (or matches within variance) clean and corruption accuracy across pruning methods, with more pronounced gains in corruption robustness at higher sparsity. Under ASAM, MWH yields the strongest overall performance among the compared pruning criteria. These results support the effectiveness of SA-SAM and the complementary benefit of MWH under high sparsity.

Finally, to directly validate the motivation of SA-SAM, we measure the mean absolute perturbation $\bar{\Phi}(\boldsymbol{\epsilon}^*) = \|\boldsymbol{\epsilon}^*\|_1/d$ and report its ratio to the dense baseline as pruning progresses. Fig. 6, (left) shows that under ASAM with a fixed radius, the perturbation magnitude shrinks substantially as sparsity increases, consistent with the sparsity-induced shrinkage predicted by Eq. (11). In contrast, SA-SAM rescales the radius as Eq. (12), which compensates for the reduced effective dimension and maintains perturbation magnitudes closer to the dense baseline across all pruning criteria (Fig. 6, right). This confirms that SA-SAM preserves the intended sharpness-aware

TABLE IV

COMPARISON OF ASAM WITH SA-SAM FOR IMAGENET-100. TOP-1 (%) ON THE CLEAN TEST SET AND IMAGENET-100-C (AVERAGED OVER 15 CORRUPTION TYPES × 5 SEVERITIES). THROUGHPUT IS OMITTED BECAUSE ALL SETTINGS USE THE SAME TARGET SPARSITY AND UNSTRUCTURED MASKS, AND THROUGHPUT DIFFERENCES ACROSS OPTIMIZERS ARE NEGLIGIBLE IN THIS ABLATION.

		ImageNet-100		ImageNet-100-C	
Sparsity	Method	ASAM	SA-SAM	ASAM	SA-SAM
0.00	Dense	82.91 ± 0.30	82.91 ± 0.30	44.58 ± 0.69	44.58 ± 0.69
0.80	OBD	81.81 ± 0.42	81.37 ± 0.31 (↓ 0.44)	40.86 ± 1.18	40.34 ± 1.13 (↓ 0.52)
	HAP	81.27 ± 0.53	82.08 ± 0.46 (↑ 0.81)	42.13 ± 0.53	42.32 ± 0.62 (↑ 0.19)
	AdaSAP	77.51 ± 0.57	77.82 ± 0.17 (↑ 0.31)	35.74 ± 0.82	36.29 ± 0.73 (↑ 0.55)
	MWH (Ours)	82.85 ± 0.42	**83.14 ± 0.06** (↑ 0.29)	44.04 ± 0.57	**44.45 ± 0.50** (↑ 0.41)
0.90	OBD	80.93 ± 0.56	81.33 ± 0.33 (↑ 0.40)	39.59 ± 1.47	39.84 ± 1.03 (↑ 0.25)
	HAP	80.63 ± 0.36	80.71 ± 0.13 (↑ 0.08)	39.36 ± 1.21	40.16 ± 0.64 (↑ 0.80)
	AdaSAP	72.30 ± 0.47	72.58 ± 0.50 (↑ 0.28)	31.27 ± 0.90	31.83 ± 0.74 (↑ 0.56)
	MWH (Ours)	82.27 ± 0.28	**82.80 ± 0.16** (↑ 0.53)	42.88 ± 0.44	**43.17 ± 0.47** (↑ 0.29)

exploration behavior throughout gradual pruning.

VI. CONCLUSION

In this work, we proposed SA-SAM, which adapts the ASAM optimization method, originally designed for dense models, to sparse models created by pruning. As a complementary option, we also evaluated MWH, a simple curvature–magnitude importance variant, to study its benefit under high sparsity. Through experiments using multiple datasets and models with different parameter counts, we demonstrated the effectiveness of our proposed methods. Furthermore, through detailed analysis on CIFAR-10, we clarified the factors behind the performance improvement over existing methods. Future work includes extending SA-SAM to structured sparsity and validating sparse execution across more optimized kernels and hardware backends.

ACKNOWLEDGMENTS

We used a GPT-based language model (ChatGPT, OpenAI) for minor English language polishing. We verified all technical content, experimental settings, and results.

REFERENCES

[1] H. Cheng, M. Zhang, and J. Q. Shi, "A Survey on Deep Neural Network Pruning: Taxonomy, Comparison, Analysis, and Recommendations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10 558–10 578, 2024.

[2] D. Hendrycks and T. Dietterich, "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations," in *International Conference on Learning Representations*, 2018.

[3] A. Bair, H. Yin, M. Shen, P. Molchanov, and J. M. Alvarez, "Adaptive Sharpness-Aware Pruning for Robust Sparse Networks," in *International Conference on Learning Representations*, 2023.

[4] J. Kwon, J. Kim, H. Park, and I. K. Choi, "ASAM: Adaptive Sharpness-Aware Minimization for Scale-Invariant Learning of Deep Neural Networks," in *International Conference on Machine Learning*, 2021, pp. 5905–5914.

[5] D. Stutz, M. Hein, and B. Schiele, "Relating Adversarially Robust Generalization to Flat Minima," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 7787–7797.

[6] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," in *International Conference on Learning Representations*, Feb. 2017.

[7] S. Yu, Z. Yao, A. Gholami, Z. Dong, S. Kim, M. W. Mahoney, and K. Keutzer, "Hessian-Aware Pruning and Optimal Neural Implant," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022, pp. 3665–3676.

[8] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning Filters for Efficient ConvNets," in *International Conference on Learning Representations*, 2017.

[9] N. Lee, T. Ajanthan, and P. Torr, "SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY," in *International Conference on Learning Representations*, 2018.

[10] Y. LeCun, J. Denker, and S. Solla, "Optimal Brain Damage," in *Advances in Neural Information Processing Systems*, vol. 2, 1989.

[11] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware Minimization for Efficiently Improving Generalization," in *International Conference on Learning Representations*, 2021.

[12] W. Zhao, S. Alwidian, and Q. H. Mahmoud, "Adversarial Training Methods for Deep Learning: A Systematic Review," *Algorithms*, vol. 15, no. 8, p. 283, 2022.

[13] Y. Zhang, H. He, J. Zhu, H. Chen, Y. Wang, and Z. Wei, "On the Duality Between Sharpness-Aware Minimization and Adversarial Training," in *International Conference on Machine Learning*, 2024.

[14] C. Tan, J. Zhang, J. Liu, Y. Wang, and Y. Hao, "Stabilizing Sharpness-aware Minimization Through A Simple Renormalization Strategy," Sep. 2024, arXiv:2401.07250 [cs]. [Online]. Available: http://arxiv.org/abs/2401.07250

[15] Y. Liu, S. Mai, X. Chen, C.-J. Hsieh, and Y. You, "Towards efficient and scalable sharpness-aware minimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 360–12 370.

[16] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," *Master's thesis, University of Toronto*, 2009. [Online]. Available: https://cir.nii.ac.jp/crid/1572824499126417408

[17] A. Shekhar, "ImageNet100," 2021. [Online]. Available: https://www.kaggle.com/datasets/ambityga/imagenet100

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[19] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the Loss Landscape of Neural Nets," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[20] Z. Yao, A. Gholami, K. Keutzer, and M. W. Mahoney, "PyHessian: Neural Networks Through the Lens of the Hessian," in *IEEE International Conference on Big Data (Big Data)*, 2020, pp. 581–590.