A Single-Step, Sharpness-Aware Minimization is All You Need to Achieve Efficient and Accurate Sparse Training

Jie Ji, Gen Li, Jingjing Fu, Fatemeh Afghah, Linke Guo, Xiaoyong Yuan, Xiaolong Ma
Clemson University
jji@g.clemson.edu

Abstract

Sparse training stands as a landmark approach in addressing the considerable training resource demands imposed by the continuously expanding size of Deep Neural Networks (DNNs). However, the training of a sparse DNN encounters great challenges in achieving optimal generalization ability despite the efforts from the state-of-the-art sparse training methodologies. To unravel the mysterious reason behind the difficulty of sparse training, we connect network sparsity with the structure of neural loss functions and identify that the cause of such difficulty lies in a chaotic loss surface. In light of such revelation, we propose S^2-SAM, characterized by a Single-step Sharpness-Aware Minimization that is tailored for Sparse training. For the first time, S^2-SAM innovates the traditional SAM-style optimization by approximating sharpness perturbation through prior gradient information, incurring *zero extra cost*. Therefore, S^2-SAM not only exhibits the capacity to improve generalization but also aligns with the efficiency goal of sparse training. Additionally, we study the generalization result of S^2-SAM and provide theoretical proof for convergence. Through extensive experiments, S^2-SAM demonstrates its universally applicable plug-and-play functionality, enhancing accuracy across various sparse training methods. Code available at https://github.com/jjsrf/SSAM-NEURIPS2024.

1 Introduction

The arrival of the Artificial General Intelligence (AGI) [1] era has urged an ever-expanding realm of artificial intelligence, bringing significant growth in deep neural networks (DNNs) depth and intricacy. The efficient training of DNN has thus emerged as an uttermost imperative, demanding immediate and concerted efforts.

To train an overparameterized, large-scale DNN efficiently while preserving its high accuracy, state-of-the-art literature has introduced sparse training [2–6] as a straightforward solution that reduces both parameter footprint and computation cost. With the presence of a large portion of zeros in the network, both forward and backward computation for training can be saved by skipping those zeros, as well as reducing the memory consumption since the zeros are not necessary to be stored. However, training a sparse neural network is difficult since the optimization under sparse regime readily converges to stationary points with a sub-optimal generalization accuracy (i.e., saddle points) [7]. Due to the difficulty of *directly* (i.e., without the time-consuming linear interpolation) assessing if the sparse solution is in the saddle point, finding workaround approaches for training a better sparse neural network is critical, especially for the practical usage in efficient learning [8–17].

To address the above difficulty, many efforts have been made. For instance, the Static Sparse Training (SST) such as the Lottery Ticket Hypothesis (LTH) [20], SNIP [21], GraSP [18] and SynFlow [22]

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-1405

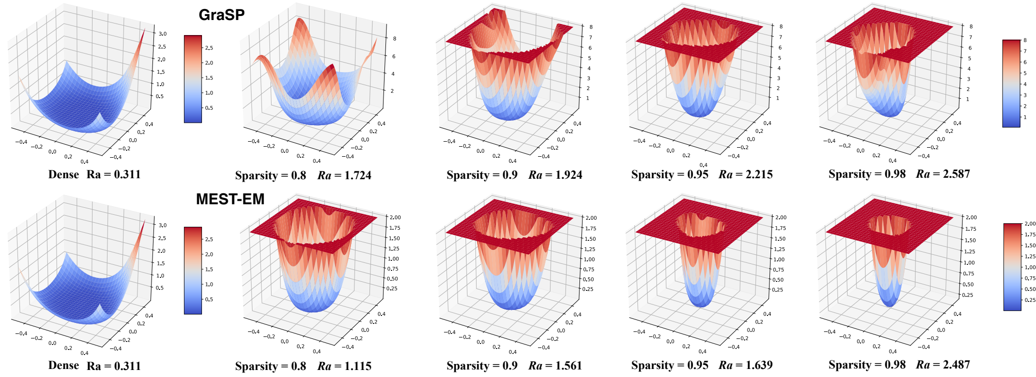

Figure 1: The loss surface visualization for training a sparse neural network using ResNet-32 on CIFAR-10. We select two representative sparse training methods [18, 3] and incorporate different levels of sparsity. We also quantify the loss surface behavior using coefficient Ra [19] to evaluate sharpness. With increased sparsity, Ra becomes larger, indicating sharper and steeper surface.

determine a static sparse pattern at training initialization. On the other hand, Dynamic Sparse Training (DST) such as SET [2], RigL [23], and MEST [3] iteratively updates the sparse topology during training to find a better sparse model. However, those methods either suffer from suboptimal generalization ability due to heuristic nature of their methodology or metric settings, or experience difficulties in parameter setting and dynamic sparsity scheduling. With the increase of the sparsity, these phenomenon become more severe.

We identify that the cause of the sparse network learning difficulties lies in the variable learning dynamics, which is closely related to the network topology. When incorporating sparsity into a neural network, the effective structure of the original network becomes narrower. According to the study of neural loss function structure [24], a wide network resulted in flat minima and wide regions of apparent convexity, which helps prevent the chaotic behavior that occurs during training. Evidently, higher sparsity indicates a narrower structure, which suggests more chaotic behavior is expected during training, thus degrading the accuracy. We perform different types of sparse training with different levels of sparsity and plot the loss surface in Figure 1 to demonstrate the convexity of the training behavior. According to the sharpness of the loss basin, we observe a transition from a smooth and close-to-convex surface to a steep one as sparsity is introduced and increased. Since higher sparsity levels lead to more chaotic behavior, in landscapes that are narrow and sharp with a large Ra value, we might expect to encounter more chaotic training behavior that degrades generalization ability.

It seems that achieving coexistence of sparsity and good generalization ability during training is challenging. Therefore, we raise the following question: *is there a simple, effective method that can improve generalization ability of training a sparse neural network, without sacrificing efficiency (sparsity) and incurring zero extra cost?* We believe the answer to the above question lies in the sharpness of the loss surface. Inspired by the Sharpness-Aware Minimization (SAM) [25] technique that finds flatter minima that have uniformly low loss in nearby regions, we argue that such technique is especially suitable for sparse training since the steep loss surface induced by sparsity can be directly mitigated. However, to leverage sharpness for better generalization, SAM uses an additional full training step (i.e., forward and backward propagation) to quantify and evaluate loss on the summation of current weights and a constrained perturbation, which roughly doubles the computation of training. Such extra costs contradict the goal of sparse training. Although some literature [26, 27] have been proposed to reduce the computation cost for SAM, the computation is still significant, posing a roadblock to the wide application of SAM to the realm of sparse training.

In this paper, we propose a novel approach to achieve sharpness-aware minimization with *zero extra cost*, tailored for the sparse training regime to maintain efficiency while improving generalization ability. It is the *first time* that a **S**ingle-step **S**harpness-**A**ware **M**inimization is proposed for **S**parse training (S^2-SAM). Different from the traditional two-step computation regime of SAM, S^2-SAM uses a unique single-step approach that leverages sharpness and trains weights with one training step. Specifically, S^2-SAM uses the weight gradients from the prior step to approximate the perturbation to the weights, thus solving the sharpness evaluation without performing an extra full training step. Therefore, S^2-SAM incurs zero extra cost to achieve sharpness-aware training, which aligns with the efficiency goal of sparse training. We also study the generalization result of S^2-SAM and provide theoretical proof for convergence. We demonstrate that S^2-SAM provides a straightforward plug-and-play functionality on variety of sparse training methods, significantly boosting their accuracy. Our contributions are summarized as follows:

- We identify that the difficulty of training a sparse neural network lies in the increasingly chaotic and steep loss surface when sparsity is introduced and increased.
- We develop a novel Single-step Sharpness-Aware Minimization technique tailored for Sparse training (S^2-SAM), and it is the *first time* that a SAM-style optimization with *zero* extra computation cost has been proposed.
- We study the generalization result of S^2-SAM and provide theoretical proof to demonstrate that the S^2-SAM is guaranteed for convergence.
- Through systematic evaluations, we show that S^2-SAM provides a plug-and-play functionality applied to a variety of sparse training methods, and consistently improves accuracy on different networks and datasets.

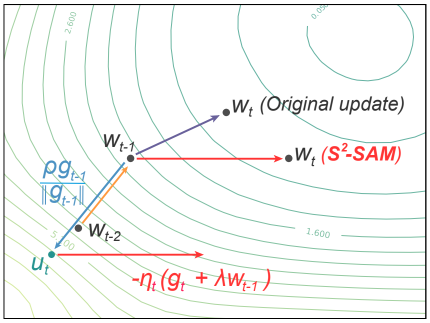

Figure 2: Illustration of the optimization mechanism of S^2-SAM. The perturbation on the current weights is approximated by the weight gradients from prior step. Please see Section 2.2 for detailed discussion.

2 Proposed Method

2.1 Preliminary of Sharpness-Aware Minimization

SAM is an optimization technique designed to enhance neural network generalization and mitigate overfitting. It minimizes the maximum loss in a neighborhood around the current parameters, as opposed to solely focusing on the loss at the current point. This approach identifies flatter minima that have uniformly low loss in nearby regions, ultimately contributing to improved generalization performance.

Specifically, consider a family of models parameterized by $\mathbf{w} \in \mathcal{W} \subseteq \mathbb{R}^d$; L is the loss function, and $\mathcal{S}$ denotes the training dataset. SAM aims to minimize the following upper bound of the PAC-Bayesian generalization error: for any $\rho > 0$,

$$L(\mathbf{w}) \leq \max_{\|\epsilon\|_p \leq \rho} L_{\mathcal{S}}(\mathbf{w} + \epsilon) + \frac{\lambda}{2}\|\mathbf{w}\|^2. \tag{1}$$

To solve the above minimax problem, at each iteration t, SAM updates the following steps:

$$\begin{aligned} \epsilon_t &= \frac{\rho \cdot \operatorname{sign}(\nabla L_{\mathcal{S}}(\mathbf{w}_{t-1})) |\nabla L_{\mathcal{S}}(\mathbf{w}_{t-1})|^{q-1}}{\left(\|\nabla L_{\mathcal{S}}(\mathbf{w}_{t-1})\|_q^q\right)^{1/p}}, \\ \mathbf{w}_t &= \mathbf{w}_{t-1} - \eta_t \left(\nabla L_{\mathcal{S}}\left(\mathbf{w}_{t-1} + \epsilon_t\right) + \lambda \mathbf{w}_{t-1}\right), \end{aligned} \tag{2}$$

where $1/p + 1/q = 1, \rho > 0$ is a hyperparameter, $\lambda > 0$ is the parameter for weight decay, and $\eta_t > 0$ is the learning rate. By setting $p = q = 2$ and introducing an intermediate variable $\mathbf{u}_t$, we have:

$$\mathbf{u}_t = \mathbf{w}_{t-1} + \frac{\rho \nabla L_{\mathcal{S}}\left(\mathbf{w}_{t-1}\right)}{\|\nabla L_{\mathcal{S}}\left(\mathbf{w}_{t-1}\right)\|}, \tag{3}$$

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta_t \left(\nabla L_{\mathcal{S}}\left(\mathbf{u}_t\right) + \lambda \mathbf{w}_{t-1}\right). \tag{4}$$

2.2 The Proposed S^2-SAM Method

As shown in Equation (2), SAM needs to compute the gradient twice at each iteration, involving additional computation costs. Further, the two-step gradient computation is not parallelizable, which presents challenges for deployment in large-scale training scenarios. Thus, we propose a new algorithm that only needs to compute the gradient *once* in each iteration. Different from prior efficient SAM works [26–28] aiming to reduce the computation by introducing periodically SAM steps or data selection, our proposed framework S^2-SAM achieves *zero* extra computation cost while maintaining the improved generalization ability of sparse network training.

In Figure 2, we demonstrate that the perturbation on the current weights is approximated by the weight gradients g_{t-1} from the prior step. The rational behind such design is that the gradient direction (i.e., $\mathbf{w}_{t-2}$ to $\mathbf{w}_{t-1}$) from prior step is optimizing the prior loss to a considerable extent, thus it can be used to represent a sharp direction among all perturbation directions at $t-1$. Since the loss surface of a sparse network is too chaotic, a perturbation with relatively high degree of sharpness (i.e., the prior gradient) can still find a neighborhood with near-maximum loss. The following equations specifying g_{t-1} information are used to substitute the $\nabla L_{\mathcal{S}}\left(\mathbf{w}_{t-1}\right)$ step in Equation (2). Thus, we only need to compute the gradient once in each iteration:

$$\mathbf{u}_t = \mathbf{w}_{t-1} + \frac{\rho \mathbf{g}_{t-1}}{\|\mathbf{g}_{t-1}\|}, \tag{5}$$

$$\mathbf{g}_t = \nabla L_{\mathcal{S}}\left(\mathbf{u}_t\right), \tag{6}$$

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta_t \left(\mathbf{g}_t + \lambda \mathbf{w}_{t-1}\right). \tag{7}$$

Remark 1. The parameter ρ can vary in terms of iteration $t : \rho_t = \sqrt{c/t}$, where $c > 0$ is a constant.

2.3 Generalization Analysis

In this section, we study the generalization result of S^2-SAM. First, we give some notations, where most of them are followed by [29, 30]. Let $\mathbf{w}_{\mathcal{S}} = \mathcal{A}(\mathcal{S})$ be a solution that generated by a random algorithm $\mathcal{A}$ based on dataset $\mathcal{S}$. Recall that problem (8)

$$\min_{\mathbf{w}\in\mathcal{W}} F_{\mathcal{S}}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(\mathbf{y}_i, f\left(\mathbf{w}; \mathbf{x}_i\right)\right) \tag{8}$$

is called empirical risk minimization in the literature, and the true risk minimization is given by

$$\min_{\mathbf{w}\in\mathcal{W}} F(\mathbf{w}) := \mathrm{E}_{(\mathbf{x},\mathbf{y})}[\ell(\mathbf{y}, f(\mathbf{w};\mathbf{x}))]. \tag{9}$$

We define its optimal solution: $\mathbf{w}_* \in \arg\min_{\mathbf{w}\in\mathcal{W}} F(\mathbf{w})$. Then the excess risk bound (ERB) is defined as

$$\mathrm{E}_{\mathcal{A},\mathcal{S}}\left[F\left(\mathbf{w}_{\mathcal{S}}\right)\right] - F\left(\mathbf{w}_*\right). \tag{10}$$

It has been shown that the ERB can be upper bounded by optimization error and generalization error [29, 30]. We notice that there are several works [29, 30] studying the generalization result of SGD for non-convex setting under different conditions, such as bounded stochastic gradient $\|\nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{w};\mathbf{x}))\| \leq G$ and decaying learning rate $\eta_t \leq \frac{c}{t}$ with a constant $c > 0$, where t is the optimization iteration. In this paper, we are not interested in establishing a fast rate in ERB under different conditions, but we want to explore the generalization ability of S^2-SAM with the fewest possible modifications when building a bridge between theory and practice. For example, weight decay is a widely used trick when training deep neural networks. With the use of weight decay, the empirical risk minimization in practice becomes $\min_{\mathbf{w}\in\mathcal{W}}\left\{\widehat{F}_{\mathcal{S}}(\mathbf{w}) := F_{\mathcal{S}}(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|^2\right\}$. Then, we define some notations as follows. Specifically, let

$$\begin{aligned}\widehat{F}_{\mathcal{S}}(\mathbf{w}) &= F_{\mathcal{S}}(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|^2 = \frac{1}{n}\sum_{i=1}^{n}\underbrace{\ell\left(\mathbf{y}_i, f\left(\mathbf{w};\mathbf{x}_i\right)\right) + \frac{\lambda}{2}\|\mathbf{w}\|^2}_{\widehat{\ell}(\mathbf{y}_i, f(\mathbf{w};\mathbf{x}_i))}, \\ \widehat{F}(\mathbf{w}) &= F(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|^2\end{aligned} \tag{11}$$

Followed by [30], we use the following decomposition of testing error:

$$\mathrm{E}_{\mathcal{A},\mathcal{S}}\left[F\left(\mathbf{w}_{\mathcal{S}}\right)\right] - \mathrm{E}_{\mathcal{S}}\left[F_{\mathcal{S}}\left(\mathbf{w}_{\mathcal{S}}^*\right)\right] \leq \mathrm{E}_{\mathcal{S}}\left[\mathrm{E}_{\mathcal{A}}\left[F_{\mathcal{S}}\left(\mathbf{w}_{\mathcal{S}}\right) - F_{\mathcal{S}}\left(\mathbf{w}_{\mathcal{S}}^*\right)\right]\right] + \mathrm{E}_{\mathcal{A},\mathcal{S}}\left[F\left(\mathbf{w}_{\mathcal{S}}\right) - F_{\mathcal{S}}\left(\mathbf{w}_{\mathcal{S}}\right)\right] \tag{12}$$

where the upper bound is the optimization error plus the generalization error.

Next, we present some notations and assumptions that will be used in the convergence analysis. Throughout this paper, we also make the following assumptions for solving the problem (8).

https://doi.org/10.52202/079017-1405

Assumption 1. Assume the following conditions hold: (i) The stochastic gradient of $F_{\mathcal{S}}(\mathbf{w})$ is unbiased, i.e., $\mathrm{E}_{(\mathbf{x},\mathbf{y})}[\nabla\ell(\mathbf{y}, f(\mathbf{w};\mathbf{x}))] = \nabla F_{\mathcal{S}}(\mathbf{w})$, and the variance of stochastic gradient is bounded, i.e., there exists a constant $\sigma^2 > 0$, such that

$$\mathrm{E}_{(\mathbf{x},\mathbf{y})}\left[\|\nabla\ell(\mathbf{y}, f(\mathbf{w};\mathbf{x})) - \nabla F_{\mathcal{S}}(\mathbf{w})\|^2\right] = \sigma^2.$$

(ii) $F_{\mathcal{S}}(\mathbf{w})$ is smooth with an L-Lipchitz continuous gradient, i.e., it is differentiable and there exists a constant $L > 0$ such that $\|\nabla F_{\mathcal{S}}(\mathbf{w}) - \nabla F_{\mathcal{S}}(\mathbf{u})\| \leq L\|\mathbf{w} - \mathbf{u}\|, \forall \mathbf{w}, \mathbf{u} \in \mathcal{W}$.

Assumption 2. There exists a constant $\mu > 0$ such that $2\mu\left(F_{\mathcal{S}}(\mathbf{w}) - F_{\mathcal{S}}\left(\mathbf{w}_{\mathcal{S}}^*\right)\right) \leq \|\nabla F_{\mathcal{S}}(\mathbf{w})\|^2, \forall \mathbf{w} \in \mathcal{W}$, where $\mathbf{w}_{\mathcal{S}}^* \in \min_{\mathbf{w}\in\mathcal{W}} F_{\mathcal{S}}(\mathbf{w})$ is a optimal solution (PL condition [31]).

Now consider that $\mathcal{A} = \mathrm{S}^2$-SAM. We define the gradient update rule $\mathcal{G}_{\widehat{\ell},\eta}$ as follows

$$\mathbf{u} = \mathbf{w} + \frac{\rho\nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{w},\mathbf{x}))}{\|\nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{w},\mathbf{x}))\|}, \tag{13}$$

$$\mathcal{G}_{\widehat{\ell},\eta}(\mathbf{w}) = \mathbf{w} - \eta\underbrace{\left(\nabla_{\mathbf{u}}\ell(\mathbf{y}, f(\mathbf{u},\mathbf{x})) + \lambda\mathbf{w}\right)}_{\nabla_{\mathbf{u}}\widetilde{\ell}(\mathbf{y},f(\mathbf{u},\mathbf{x}))}, \tag{14}$$

Then we have the following lemma, which is similar to Lemma 2.5 and Lemma 4.2 in [29] that use recursive definition through variable $\mathcal{G}'$ and $\mathbf{w}_t'$.

Lemma 1. *Assume that $\ell(\mathbf{y}, f(\mathbf{w},\mathbf{x}))$ is L-smooth and B-Lipschitz. Let $\mathbf{w}_{t+1} = \mathcal{G}\left(\mathbf{w}_t\right)$ and another sequence $\mathbf{w}_{t+1}' = \mathcal{G}'\left(\mathbf{w}_t'\right)$, then*

$$\left\|\mathbf{w}_{t+1} - \mathbf{w}_{t+1}'\right\| = \begin{cases} (1 + \eta L - \eta\lambda)\left\|\mathbf{w}_t - \mathbf{w}_t'\right\| + 2\eta\rho, & \mathcal{G} = \mathcal{G}' \\ (1 - \eta\lambda)\left\|\mathbf{w}_t - \mathbf{w}_t'\right\| + 2\eta B, & \mathcal{G} \neq \mathcal{G}' \end{cases}$$

Please see the proof of Lemma 1 in Appendix B.

Theorem 1. *Under Assumption 1, assume that $\ell(\mathbf{y}, f(\mathbf{w},\mathbf{x}))$ is L-smooth and B-Lipschitz, suppose $\widehat{F}_{\mathcal{S}}(\mathbf{w})$ satisfies Assumption 2 and $\min_{\mathbf{w}\in\mathcal{W}} \widehat{F}_{\mathcal{S}}(\mathbf{w}) \leq F_{\mathcal{S}}\left(\mathbf{w}_{\mathcal{S}}^*\right) + \frac{\lambda}{2}\left\|\mathbf{w}_t\right\|^2$ with $\lambda = 2L$, where $\mathbf{w}_t$ is the intermediate solution of $\mathcal{A}$, then*

$$\mathrm{E}_{R,\mathcal{A},\mathcal{S}}\left[F\left(\mathbf{w}_R\right)\right] - \mathrm{E}_{\mathcal{S}}\left[F\left(\mathbf{w}_*\right)\right] \leq \frac{\widehat{F}_{\mathcal{S}}\left(\mathbf{w}_0\right)}{\mu\eta T} + \frac{\eta(L+\lambda)\sigma^2}{2\mu} + \frac{6B+1}{n}$$

where $\mathcal{A}$ is SGD.

The $\mathbf{w}_{\mathcal{S}}^* \in \min_{\mathbf{w}\in\mathcal{W}} F_{\mathcal{S}}(\mathbf{w})$ is an optimal solution, and we show that the generalization error is bounded. Based on Lemma 1, the proof of Theorem 1 is derived in Appendix C.

3 Experimental Results

In this section, we carry out comprehensive experiments to demonstrate how S^2-SAM improves sparse training performance. We test S^2-SAM on CIFAR-10/100 [34] with ResNet-32 [32] and VGG-19 [35], and we also perform experiments on ImageNet-1K [36] and ImageNet-C [37] based on ResNet-50 [38].

Following the recent developments in sparse training techniques, we apply S^2-SAM on static sparse training such as LTH [20], SNIP [21], GraSP [18], as well as dynamic sparse training methods such as SET [2], DSR [33], RigL [23], MEST [3], CHEX [10] and Chase [39]. We apply S^2-SAM to the official codes or published implementations to show performance gains. All of our experiments are performed on NVIDIA 4× A6000 GPUs. We repeat training experiments for 3 times and report the mean and standard deviation of the accuracy. For training throughput evaluation, we adopt the original settings of each baseline method and record the throughput on 4× A6000 GPUs.

Table 1: Test accuracy (%) of pruned ResNet-32 on CIFAR-10/100.

Datasets	CIFAR-10			CIFAR-100		
Pruning ratio	90%	95%	98%	90%	95%	98%
ResNet-32	94.58 (Dense)			74.89 (Dense)		
LT [20]	92.31	91.06	88.78	68.99	65.02	57.37
LT+ S^2-SAM (ours)	**92.58±0.07** (0.27↑)	**91.47±0.10** (0.41↑)	**89.35±0.11** (0.57↑)	**69.34±0.09** (0.35↑)	**65.45±0.11** (0.43↑)	**57.76±0.13** (0.39↑)
SNIP [21]	92.59±0.10	91.01±0.21	87.51±0.31	68.89±0.45	65.02±0.69	57.37±1.43
SNIP+ S^2-SAM (ours)	**93.17±0.16** (0.58↑)	**91.59±0.22** (0.58↑)	**88.08±0.29** (0.57↑)	**69.33±0.28** (0.44↑)	**65.66±0.49** (0.64↑)	**58.25±0.77** (0.88↑)
GraSP [32]	92.38±0.21	91.39±0.25	88.81±0.14	69.24±0.24	66.50±0.11	58.43±0.43
GraSP+ S^2-SAM (ours)	**92.87±0.14** (0.49↑)	**91.98±0.22** (0.59↑)	**89.66±0.29** (0.85↑)	**69.98±0.22** (0.74↑)	**67.12±0.18** (0.62↑)	**59.45±0.19** (1.02↑)
SET [2]	92.30	90.76	88.29	69.66	67.41	62.25
SET+ S^2-SAM (ours)	**92.92±0.23** (0.62↑)	**91.50±0.19** (0.74↑)	**88.78±0.20** (0.49↑)	**70.23±0.20** (0.57↑)	**68.28±0.15** (0.87↑)	**63.56±0.19** (1.31↑)
DSR [33]	92.97	91.61	88.46	69.63	68.20	61.24
DSR+ S^2-SAM (ours)	**93.49±0.21** (0.52↑)	**92.08±0.22** (0.47↑)	**89.11±0.17** (0.65↑)	**70.11±0.16** (0.48↑)	**68.87±0.16** (0.67↑)	**62.00±0.17** (0.76↑)
RigL [23]	93.07	91.83	89.00	70.34	68.22	64.07
RigL+ S^2-SAM (ours)	**93.55±0.14** (0.48↑)	**92.11±0.21** (0.28↑)	**90.40±0.17** (1.40↑)	**72.38±0.11** (2.04↑)	**70.29±0.14** (2.07↑)	**64.98±0.06** (0.91↑)
RigL (ERK) [23]	93.55	92.39	90.22	70.62	68.47	64.14
RigL (ERK)+ S^2-SAM (ours)	**93.75±0.19** (0.20↑)	**92.81±0.08** (0.42↑)	**91.16±0.11** (0.94↑)	**72.56±0.07** (1.94↑)	**70.33±0.10** (1.86↑)	**65.15±0.12** (1.01↑)
MEST (EM) [3]	92.56±0.07	91.15±0.29	89.22±0.11	70.44±0.26	68.43±0.32	64.59±0.27
MEST (EM) + S^2-SAM (ours)	**93.43±0.12** (0.87↑)	**91.58±0.07** (0.43↑)	**91.22±0.14** (2.00↑)	**71.95±0.13** (1.51x↑)	**70.04±0.10** (1.61↑)	**65.69±0.34** (1.10↑)
MEST (EM&S) [3]	93.27±0.14	92.44±0.13	90.51±0.11	71.30±0.31	70.36±0.05	67.16±0.25
MEST (EM&S) + S^2-SAM (ours)	**93.39±0.17** (0.12↑)	**92.97±0.17** (0.53↑)	**91.32±0.18** (0.81↑)	**72.74±0.08** (1.44↑)	**71.85±0.09** (1.49↑)	**69.13±0.20** (1.97↑)

3.1 Accuracy Enhancement for State-of-the-art Sparse Training Methods

CIFAR-10 and CIFAR-100. We show S^2-SAM is universally applicable on various sparse training methods. The results on ResNet-32 is shown in Table 1. Due to limited space, we demonstrate results of VGG-19 on CIFAR-10 in Table A.1 in the appendix. For each baseline method, we perform training with S^2-SAM at 90%, 95% and 98% sparsity, and compare the accuracy with the original accuracy. We set hyper-parameter $\rho = 0.05$ for all experiments. From the comparison results, we can notice that all sparse training methods trained with S^2-SAM obtain consistent and significant improvements, not only for static sparse training but also for dynamic sparse training. We stress that for all experiments, we only use one setting for S^2-SAM, which indicates an easy implementation that has the potential for wide applicability. We also find that S^2-SAM works better on high sparsity. For all sparse training with a 98% sparsity ratio, S^2-SAM improves the original baseline accuracy by an average of 0.77% on CIFAR-10 and 1.12% on CIFAR-100, which shows S^2-SAM is especially suitable for challenging tasks.

ImageNet-1K. We evaluate S^2-SAM on ImageNet-1K dataset. The results are shown in Table 2. We report the top-1 accuracy of the sparse training results, as well the overall training cost with FLOPs. We set hyper-parameter $\rho = 1.0$ for all experiments. Compared to the baseline methods, our result shows higher test accuracy at the same computational cost. Similarly, S^2-SAM also shows better gain on a higher sparsity ratio for most of the sparse training baselines, which indicates the consistency of using our methods on different scales of datasets. To make a fair comparison with different training recipes, we slightly scale up our training epochs to have the same or less overall training FLOPs to compare with longer training baselines, and we find out that S^2-SAM shows stable improvement on those methods.

3.2 Sparse Training with Structured Sparsity

In this section, we evaluate how S^2-SAM performs on structured sparse training methods. Note that all previous baselines use unstructured sparsity where the zeros are scattered in the network. Such unstructured sparse training shows the potential of training a sparse model, but cannot fully represent training acceleration ability. Therefore, we apply S^2-SAM to two representative structured sparse trainings, CHEX [10] and Chase [39] which dynamically train sparse networks with *channel-level* sparsity. The results are shown in Table 3. From the results, we can see that the average accuracy improvement is around 0.5%. Considering structured (channel-wise) sparsity is one challenging sparsity type to achieve good accuracy, these results indicate S^2-SAM works very well on structured sparse training methods.

3.3 Improvement on the Roughness of the Loss Surface

In this part, we plot the loss surface of the sparse training with and without S^2-SAM. The comparisons are demonstrated in Figure 3, where we perform our experiments with three representative sparse training methods SNIP [21], GraSP [18] and MEST [3] at 90%, 95% and 98% sparsity ratios on

Table 2: Results of ResNet-50 on ImageNet-1K.

Method	Sparsity Distribution	Top-1 Accuracy (%)	Training FLOPs	Inference FLOPs	Top-1 Accuary (%)	Training FLOPs	Inference FLOPs
ResNet-50	dense	76.9	(×e18)	(×e9)	76.9	(×e18)	(×e9)
Sparsity		80%			90%		
LT [20]	non-uniform	72.6	n/a	2.7	70.1	n/a	1.7
LT + S^2-SAM (ours)		**73.11±0.08** (0.51↑)	n/a	2.7	**70.78±0.05** (0.68↑)	n/a	1.7
SNIP [21]		69.7	1.67	2.8	62.0	0.91	1.9
SNIP + S^2-SAM (ours)		**70.55±0.05** (0.85↑)	1.67	2.8	**62.62±0.07** (0.62↑)	0.91	1.9
GraSP [32]		72.1	1.67	2.8	68.1	0.91	1.9
GraSP + S^2-SAM (ours)		**72.66±0.10** (0.56↑)	1.67	2.8	**68.78±0.12** (0.68↑)	0.91	1.9
SET [2]	non-uniform	72.9	0.74	1.7	69.6	0.10	0.1
SET + S^2-SAM (ours)		**73.66±0.11** (0.76↑)	0.74	1.7	**70.41±0.08** (0.81↑)	0.10	0.1
DSR [33]		73.3	1.28	3.3	71.6	0.96	2.5
DSR + S^2-SAM (ours)		**74.08±0.17** (0.78↑)	1.28	3.3	**72.32±0.13** (0.72↑)	0.96	2.5
RigL [23]	uniform	74.6	0.74	1.7	72.0	0.39	0.9
RigL + S^2-SAM (ours)		**75.39±0.12** (0.79↑)	0.74	1.7	**72.44±0.06** (0.44↑)	0.39	0.9
MEST (EM) [3]		75.7	1.10	1.7	73.6	0.48	0.9
MEST (EM) + S^2-SAM (ours)		**76.35±0.02** (0.65↑)	1.10	1.7	**74.58±0.03** (0.98↑)	0.48	0.9
MEST (EM&S) [3]		75.7	1.27	1.7	75.0	0.65	0.9
MEST (EM&S) + S^2-SAM (ours)		**76.44±0.06** (0.74↑)	1.27	1.7	**75.36±0.04** (0.36↑)	0.65	0.9
Top-KAST [40]		-	-	-	73.0	0.63	0.9
Top-KAST + S^2-SAM (ours)		-	-	-	**73.82±0.17** (0.82↑)	0.63	0.9
$\text{MEST}_{1.7\times}$ [3]	uniform	76.7	1.84	1.7	75.9	0.80	0.9
$\text{MEST}_{1.7\times}$ + S^2-SAM (ours)		**77.73±0.11** (1.03↑)	1.84	1.7	**76.82±0.12** (0.92↑)	0.80	0.9
$\text{RigL}_{5\times}$ [23]		76.6	3.71	1.7	75.7	1.95	0.9
$\text{RigL}_{5\times}$ + S^2-SAM (ours)		**77.72±0.05** (1.12↑)	3.71	1.7	**76.88±0.13** (1.18↑)	1.95	0.9

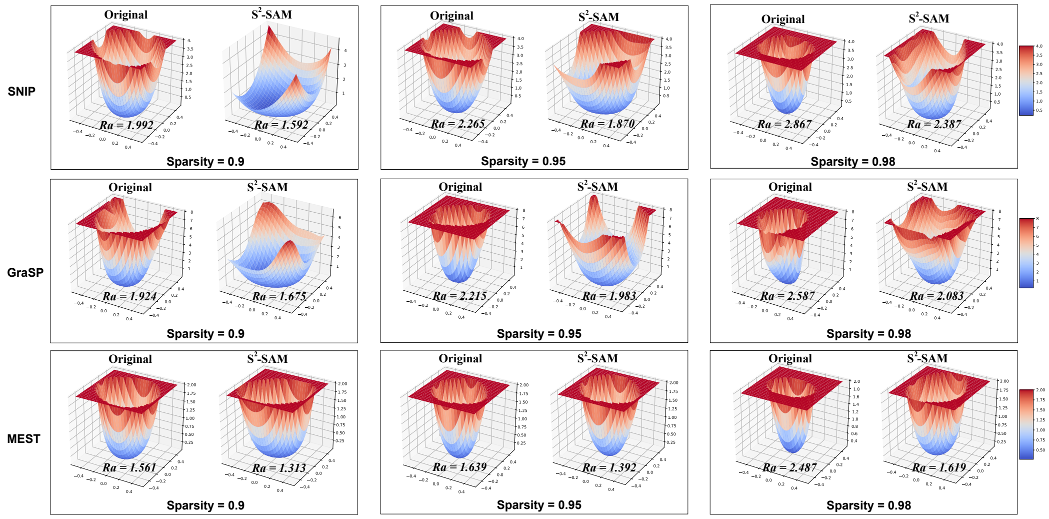

Figure 3: Loss surface sharpness comparison of different sparse training methods with original training and with S^2-SAM. We also quantitatively evaluate the coefficient Ra. Using S^2-SAM compared to the original method results in a smaller Ra, indicating a wider and smoother loss surface, which suggests improved generalization ability.

CIFAR-10 dataset with ResNet-32. We observe that as sparsity increases, suggesting more chaotic training behavior, the coefficient Ra also increases, indicating a steeper loss surface and a narrower basin. While with S^2-SAM, we can see that under the same sparsity, the basin of the loss surface widens and enlarges as Ra values decrease, suggesting a smoother loss trajectory during training of a sparse network.

3.4 Training Speed Comparison on GPU

S^2-SAM is a highly efficient approach to optimizing the sharpness of the loss surface, which incurs zero extra cost for achieving its functionality. Compared to the traditional SAM, S^2-SAM uses fewer computations and has better training performance on GPUs. In Table 4, we obtain the computation cost as well as the training speed in terms of throughput (i.e., imgs/sec) on 4× NVIDIA A6000 GPUs using ResNet-50 on the ImageNet-1K dataset. We record the throughput for different sparse training methods with their original training, with SAM [25], and with S^2-SAM. From the results, we can see that S^2-SAM achieves the same training throughput (with only negligible < 20 imgs/s decrease due to processing) as the baseline methods, where no sharpness optimization is involved. Compared to original SAM, we can see that training throughput with SAM is *less than half* of the ones that train with S^2-SAM, which aligns with the observation that SAM uses *twice the computation* cost of S^2-SAM or original training. Although SAM yields slightly better accuracy, the associated training cost outweighs the benefits, rendering it impractical.

Table 3: Accuracy of S^2-SAM on structured sparse training CHEX [10] and Chase [39].

Methods	Networks	Training	FLOPs	Accuracy (%)
CHEX	ResNet-34	Original	2.0G	73.50
		S^2-SAM	2.0G	**73.94** (0.44$\uparrow$)
	ResNet-50	Original	1.0G	76.00
		S^2-SAM	1.0G	**76.51** (0.51$\uparrow$)
Chase	ResNet-34	Original*	2.1G	72.34
		S^2-SAM	2.1G	**72.77** (0.43$\uparrow$)
	ResNet-50	Original	1.3G	75.62
		S^2-SAM	1.3G	**76.17** (0.55$\uparrow$)

* ResNet-34 original result of Chase is our own implementation.

Table 4: Training speed of SAM [25] and S^2-SAM for different sparse training at 90% sparsity.

Methods	Training	Accuracy (%)	Throughput ($\uparrow$)
GraSP	Original	68.10	**2148** imgs/s
	SAM	**68.95**	1021 imgs/s
	S^2-SAM	68.78	2132 imgs/s
RigL	Original	72.00	**3133** imgs/s
	SAM	**72.75**	1508 imgs/s
	S^2-SAM	72.44	3098 imgs/s
MEST (EM)	Original	73.60	**2981** imgs/s
	SAM	**74.88**	1398 imgs/s
	S^2-SAM	74.58	2977 imgs/s

3.5 Robustness Improvement by S^2-SAM

Since sparse training approaches have potentially high usage in practical scenarios, the system's robustness against perturbation (e.g., inclement weather conditions for image/video tasks) is usually critical. We perform experiments to evaluate the sparse model (80% sparsity) robustness against perturbations in Table 5. Our intuition is that the model trained with its sharpness optimized has a wider loss basin, which indicates higher endurance on perturbations since the loss won't change much when it is located in a wide and flat region. We adopt ImageNet-C [37] which contains a test set with the same images as ImageNet-1K but with nineteen types of corruptions applied with five different levels of severity. We train the sparse networks using different methods on ImageNet-1K, and then test their accuracy on ImageNet-C. We report test accuracy for both datasets. We can see that when the sparse model is trained using the original method, the accuracy of ImageNet-C is significantly lower than the accuracy of the original ImageNet-1K test set (around 35% lower). With S^2-SAM, the test accuracy on ImageNet-C shows promising improvement. The robust accuracy improves by an average of 3.23%.

Table 5: Testing accuracy on ImageNet-C test set. We compare the results with and without S^2-SAM using 80% sparsity.

Methods	ImageNet-1K Accuracy (%)	ImageNet-C Accuracy (%)
SNIP	69.70	31.12
SNIP + S^2-SAM	**70.55** (0.85$\uparrow$)	**34.87** (3.75$\uparrow$)
GraSP	72.10	32.24
GraSP + S^2-SAM	**72.66** (0.56$\uparrow$)	**35.17** (2.93$\uparrow$)
MEST (EM)	75.70	33.87
MEST (EM) + S^2-SAM	**76.35** (0.65$\uparrow$)	**36.98** (3.11$\uparrow$)
RigL	74.60	33.68
RigL + S^2-SAM	**75.39** (0.79$\uparrow$)	**36.80** (3.12$\uparrow$)

Table 6: Testing accuracy on dense model training. We compare original training with S^2-SAM in same settings.

Networks	Params. Count	Original Accuracy (%)	S^2-SAM Accuracy (%)
	CIFAR-10		
ResNet-32	1.86M	94.58	**94.99** (0.41$\uparrow$)
MobileNet-V2	2.30M	94.13	**94.55** (0.42$\uparrow$)
VGG-19	20.03M	94.21	**94.48** (0.27$\uparrow$)
	ImageNet-1K		
EfficientNet-B0	5.30M	76.54	**77.10** (0.56$\uparrow$)
ResNet-34	21.80M	74.09	**74.58** (0.49$\uparrow$)
ResNet-50	25.50M	76.90	**77.32** (0.42$\uparrow$)

3.6 Applying S^2-SAM to Dense Model Training

We also apply S^2-SAM to the dense model training. The results are shown in Table 6. We test on two datasets CIFAR-10 and ImageNet-1K with two additional networks MobileNet-V2 [41] and EfficientNet-B0 [42]. For ResNet-32 and VGG-19 on CIFAR-10, we follow the settings in [3] and train for 160 epochs, and for MobileNet-V2, we use the 1.0 width version and train for 350 epochs for better convergence [43]. For ResNet family on ImageNet-1K, we follow the setting in [3] and train for 150 epochs, and we train EfficientNet-B0 for 350 epochs for better convergence [44]. From the results, we can see that S^2-SAM still improves the accuracy of the dense networks. We find out that the less the parameter count of the network, the better effectiveness of S^2-SAM it achieves. This phenomenon proves our finding that a narrower network is harder to train, and S^2-SAM is an effective solution for better generalization. We must also stress that S^2-SAM is focusing on sparse neural network training, and we will leave the study of S^2-SAM on dense model training for our future research.

4 Related Works

Static Sparse Training Static sparse training determines the structure of the sparse network through the application of a pruning algorithm in the early stages of training. The lottery ticket hypothesis (LTH) [20, 45, 46] uses iterative magnitude-based pruning (IMP) to find a subnetwork that can be trained from scratch without losing accuracy. SNIP [21], GraSP [18], SynFlow [22] determine a static sparse pattern at training initialization by obtaining gradient information with a few iterations of dense training. FISH [47] acquires fixed subnetwork by pre-computing a sparse mask using Fisher information.

Dynamic Sparse Training Dynamic sparse training starts with a randomly selected sparse network structure and adapts it throughout the training process in an effort to find a better sparse structure. Sparse Evolutionary Training (SET) [2] prunes small magnitude weights and grows back randomly at the end of each training epoch. Deep R [48] uses a combination of stochastic parameter updates and dynamic sparse parameterization for training. Dynamic Sparse Reparameterization (DSR) [33] proposes to redistribute parameters between layers during training. Sparse Networks from Scratch (SNFS) [49] creates the sparse momentum algorithm, which finds the layers and weights that effectively reduce the error by using exponentially smoothed gradients (momentum). In RigL [23], the gradients of all the weights are computed when the model needs to be updated to grow new connections. Top-KAST [40] proposes a scalable and performant sparse-to-spars DST framework for maximum efficacy. Powerpropagation [50] suggests a novel neural network weight parameterization that largely preserves low-magnitude parameters from learning. ITOP [51] investigates the underlying DST mechanism and finds that the advantages of DST result from a time-based search for all potential factors. MEST [3] designs a memory-economic sparse training framework targeting accurate and fast execution on edge devices. By co-training dense and sparse models, AD/AC [52] suggests a technique that, at the conclusion of training, produces precise sparse–dense model pairings. Chase [39] dynamically translates the unstructured sparsity into channel-level sparsity to achieve direct speedup on GPU.

Sharpness-Aware Minimization Sharpness-Aware Minimization was first introduced in [25]. This optimization technique is designed to identify flatter minima characterized by consistently low loss in neighboring regions, aiming to enhance the model generalization capability during training. Nevertheless, the computational cost of Sharpness-Aware Minimization is doubled due to its two-step gradient computation regime, presenting challenges for deployment in large-scale training scenarios. To reconcile such, ESAM [26] introduces two novel and efficient training strategies: stochastic weight perturbation and sharpness-sensitive data selection, enhancing the efficiency of the SAM process without compromising its generalization performance. LookSAM [27] only periodically calculates the inner gradient ascent to significantly reduce the additional training cost of SAM. SAF [28] introduces a novel trajectory loss based on KL-divergence to measure the rate of change in training loss along the model update trajectory and replace the SAM sharpness measure. CrAM [53] optimizes over the compression projection applied to the intermediate model at every training step, which results in a compressible models that can be pruned in an on-shot manner after training. However, all those methods need extra computation cost, and not target on sparse training method.

5 Conclusion

In this paper, we propose a novel Single-step Sharpness-Aware Minimization that is tailored for Sparse training (S^2-SAM), which revolutionizes the originally computation-intensive sharpness-aware optimization into a highly efficient tool with zero extra cost. In light of the improved optimization trajectory in the loss surface, S^2-SAM successfully enhances the accuracy of the sparse network

https://doi.org/10.52202/079017-1405

training, as well as the robustness of the sparse model in practical scenarios. S^2-SAM offers seamless plug-and-play functionality, showcasing its potential for widespread applicability in the evolving landscape of efficient training. The research is inherently scientific, and we anticipate no adverse societal impact stemming from its findings.

Acknowledgments

This work is partly supported by the National Science Foundation CCF-2427875, CCF-2312616, CNS-2232048, CNS-2008049, CNS-2431440, National Aeronautics and Space Administration (NASA) 80NSSC23K1393, and the Army Research Office W911NF-24-1-0044. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NSF, NASA, the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] Ben Goertzel and Cassio Pennachin. *Artificial general intelligence*, volume 2. Springer, 2007.

[2] Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications*, 9(1):1–12, 2018.

[3] Geng Yuan, Xiaolong Ma, Wei Niu, Zhengang Li, Zhenglun Kong, Ning Liu, Yifan Gong, Zheng Zhan, Chaoyang He, Qing Jin, Siyue Wang, Minghai Qin, Bin Ren, Yanzhi Wang, Sijia Liu, and Xue Lin. Mest: Accurate and fast memory-economic sparse training framework on the edge. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[4] Lu Yin, Gen Li, Meng Fang, Li Shen, Tianjin Huang, Zhangyang Wang, Vlado Menkovski, Xiaolong Ma, Mykola Pechenizkiy, Shiwei Liu, et al. Dynamic sparsity is channel-level sparsity learner. *Advances in Neural Information Processing Systems*, 36, 2024.

[5] Gen Li, Lu Yin, Jie Ji, Wei Niu, Minghai Qin, Bin Ren, Linke Guo, Shiwei Liu, and Xiaolong Ma. Neurrev: Train better sparse neural network practically via neuron revitalization. In *The Twelfth International Conference on Learning Representations*, 2024.

[6] Jie Ji, Gen Li, Lu Yin, Minghai Qin, Geng Yuan, Linke Guo, Shiwei Liu, and Xiaolong Ma. Advancing dynamic sparse training by exploring optimization opportunities. In *Forty-first International Conference on Machine Learning*, 2024.

[7] Utku Evci, Fabian Pedregosa, Aidan Gomez, and Erich Elsen. The difficulty of training sparse neural networks. *ICML Workshop Deep Phenomena*, 2019.

[8] Alex Bronstein, Pablo Sprechmann, and Guillermo Sapiro. Learning efficient structured sparse models. In *International Conference on Machine Learning*, 2012.

[9] Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research*, 22(241):1–124, 2021.

[10] Zejiang Hou, Minghai Qin, Fei Sun, Xiaolong Ma, Kun Yuan, Yi Xu, Yen-Kuang Chen, Rong Jin, Yuan Xie, and Sun-Yuan Kung. Chex: Channel exploration for cnn model compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12287–12298, 2022.

[11] Xiaolong Ma, Minghai Qin, Fei Sun, Zejiang Hou, Kun Yuan, Yi Xu, Yanzhi Wang, Yen-Kuang Chen, Rong Jin, and Yuan Xie. Effective model sparsification by scheduled grow-and-prune methods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[12] Bo Hui, Da Yan, Xiaolong Ma, and Wei-Shinn Ku. Rethinking graph lottery tickets: Graph sparsity matters. In *The Twelfth International Conference on Learning Representations*, 2023.

https://doi.org/10.52202/079017-1405

[13] Haoyu Ma, Chengming Zhang, Xiaolong Ma, Geng Yuan, Wenkai Zhang, Shiwei Liu, Tianlong Chen, Dingwen Tao, Yanzhi Wang, Zhangyang Wang, et al. Hrbp: Hardware-friendly regrouping towards block-based pruning for sparse cnn training. In *Conference on Parsimony and Learning*, pages 282–301. PMLR, 2024.

[14] Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, PRANAY SHARMA, Sijia Liu, et al. Model sparsity can simplify machine unlearning. *Advances in Neural Information Processing Systems*, 36, 2024.

[15] Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, Ajay Kumar Jaiswal, Mykola Pechenizkiy, Yi Liang, Michael Bendersky, Zhangyang Wang, and Shiwei Liu. Outlier weighed layerwise sparsity (OWL): A missing secret sauce for pruning LLMs to high sparsity. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 57101–57115. PMLR, 2024.

[16] Shuyao Wang, Yongduo Sui, Jiancan Wu, Zhi Zheng, and Hui Xiong. Dynamic sparse learning: A novel paradigm for efficient recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 740–749, 2024.

[17] Haizhong Zheng, Xiaoyan Bai, Beidi Chen, Fan Lai, and Atul Prakash. Learn to be efficient: Build structured sparsity in large language models. *arXiv preprint arXiv:2402.06126*, 2024.

[18] Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations (ICLR)*, 2020.

[19] NO Myers. Characterization of surface roughness. *Wear*, 5(3):182–189, 1962.

[20] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *ICLR*, 2018.

[21] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations (ICLR)*, 2019.

[22] Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

[23] Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *International Conference on Machine Learning (ICML)*, pages 2943–2952. PMLR, 2020.

[24] 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.

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

[26] 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.

[27] 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*, pages 12360–12370, 2022.

[28] Jiawei Du, Daquan Zhou, Jiashi Feng, Vincent Tan, and Joey Tianyi Zhou. Sharpness-aware training for free. *Advances in Neural Information Processing Systems*, 35:23439–23451, 2022.

[29] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234. PMLR, 2016.

https://doi.org/10.52202/079017-1405

[30] Zhuoning Yuan, Yan Yan, Rong Jin, and Tianbao Yang. Stagewise training accelerates convergence of testing error over sgd. *Advances in Neural Information Processing Systems*, 32, 2019.

[31] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I 16*, pages 795–811. Springer, 2016.

[32] Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2020.

[33] Hesham Mostafa and Xin Wang. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *International Conference on Machine Learning (ICML)*, pages 4646–4655. PMLR, 2019.

[34] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

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

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[37] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[39] Lu Yin, Gen Li, Meng Fang, Li Shen, Tianjin Huang, Zhangyang Wang, Vlado Menkovski, Xiaolong Ma, Mykola Pechenizkiy, and Shiwei Liu. Dynamic sparsity is channel-level sparsity learner. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[40] Siddhant Jayakumar, Razvan Pascanu, Jack Rae, Simon Osindero, and Erich Elsen. Top-kast: Top-k always sparse training. *Advances in Neural Information Processing Systems*, 33:20744–20754, 2020.

[41] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[42] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 6105–6114. PMLR, 2019.

[43] Zhuangwei Zhuang, Mingkui Tan, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, and Jinhui Zhu. Discrimination-aware channel pruning for deep neural networks. *Advances in neural information processing systems*, 31, 2018.

[44] Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35:12934–12949, 2022.

[45] Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. In *International Conference on Learning Representations*, 2020. arXiv:2003.02389.

[46] Xiaohan Chen, Yu Cheng, Shuohang Wang, Zhe Gan, Zhangyang Wang, and Jingjing Liu. Earlybert: Efficient bert training via early-bird lottery tickets. *ACL-IJCNLP*, 2021.

[47] Yi-Lin Sung, Varun Nair, and Colin A Raffel. Training neural networks with fixed sparse masks. *Advances in Neural Information Processing Systems*, 34:24193–24205, 2021.

[48] Guillaume Bellec, David Kappel, Wolfgang Maass, and Robert Legenstein. Deep rewiring: Training very sparse deep net-works. In *International Conference on Learning Representations (ICLR)*, 2018.

[49] Tim Dettmers and Luke Zettlemoyer. Sparse networks from scratch: Faster training without losing performance. *arXiv preprint arXiv:1907.04840*, 2019.

[50] Jonathan Schwarz, Siddhant Jayakumar, Razvan Pascanu, Peter E Latham, and Yee Teh. Powerpropagation: A sparsity inducing weight reparameterisation. *Advances in neural information processing systems*, 34:28889–28903, 2021.

[51] Shiwei Liu, Lu Yin, Decebal Constantin Mocanu, and Mykola Pechenizkiy. Do we actually need dense over-parameterization? in-time over-parameterization in sparse training. In *International Conference on Machine Learning*, pages 6989–7000. PMLR, 2021.

[52] Alexandra Peste, Eugenia Iofinova, Adrian Vladu, and Dan Alistarh. Ac/dc: Alternating compressed/decompressed training of deep neural networks. *Advances in neural information processing systems*, 34:8557–8570, 2021.

[53] Alexandra Peste, Adrian Vladu, Eldar Kurtic, Christoph H Lampert, and Dan Alistarh. Cram: A compression-aware minimizer. *arXiv preprint arXiv:2207.14200*, 2022.

https://doi.org/10.52202/079017-1405

Appendix

A Results of VGG-19 on CIFAR-10/100

We test S^2-SAM on VGG-19 using CIFAR-10 and CIFAR-100. The results are shown in Table A.1. We demonstrate results on both uniform and ERK distributions, and achieve SOTA results on CIFAR-10/100.

Table A.1: Test accuracy (%) of pruned VGG-19 on CIFAR-10/100.

Datasets	CIFAR-10			CIFAR-100		
Pruning ratio	90%	95%	98%	90%	95%	98%
VGG-19	94.20			74.17		
LT [20]	93.51	92.92	92.34	72.78	71.44	68.95
LT+ S^2-SAM (ours)	**93.82±0.12** (0.31↑)	**93.41±0.11** (0.49↑)	**92.92±0.14** (0.58↑)	**73.08±0.08** (0.30↑)	**71.69±0.10** (0.25↑)	**69.78±0.17** (0.83↑)
SNIP [21]	93.63±0.06	93.43±0.20	92.05±0.28	72.84±0.22	71.83±0.23	58.46±1.10
SNIP+ S^2-SAM (ours)	**93.91±0.11** (0.28↑)	**93.91±0.21** (0.48↑)	**92.86±0.28** (0.81↑)	**73.33±0.21** (0.49↑)	**72.41±0.35** (0.58↑)	**60.12±0.42** (1.66↑)
GraSP [32]	93.30±0.14	93.43±0.18	92.19±0.12	71.95±0.18	71.23±0.12	68.90±0.41
GraSP+ S^2-SAM (ours)	**93.88±0.11** (0.58↑)	**93.75±0.21** (0.32↑)	**92.95±0.23** (0.76↑)	**72.44±0.27** (0.49↑)	**72.91±0.25** (1.68↑)	**69.98±0.31** (1.08↑)
SET [2]	92.46	91.73	89.18	72.36	69.81	65.94
SET+ S^2-SAM (ours)	**92.97±0.22** (0.51↑)	**92.55±0.20** (0.82↑)	**90.18±0.19** (1.00↑)	**72.65±0.14** (0.29↑)	**70.11±0.14** (0.30↑)	**66.76±0.17** (0.82↑)
DSR [33]	93.75	93.86	93.13	72.31	71.98	70.70
DSR+ S^2-SAM (ours)	**94.24±0.13** (0.49↑)	**94.27±0.15** (0.41↑)	**93.81±0.27** (0.68↑)	**72.78±0.20** (0.47↑)	**72.55±0.22** (0.57↑)	**71.69±0.24** (0.99↑)
RigL [23]	93.12	92.43	90.65	71.14	69.02	64.87
RigL+ S^2-SAM (ours)	**93.61±0.16** (0.49↑)	**92.87±0.21** (0.44↑)	**91.88±0.21** (1.23↑)	**71.98±0.18** (0.84↑)	**70.16±0.18** (1.14↑)	**66.01±0.25** (1.14↑)
RigL (ERK) [23]	93.77	92.75	90.87	71.34	69.21	65.02
RigL (ERK)+ S^2-SAM (ours)	**94.07±0.21** (0.30↑)	**93.33±0.12** (0.58↑)	**91.79±0.12** (0.92↑)	**72.12±0.20** (0.78↑)	**69.95±0.18** (0.74↑)	**66.13±0.19** (1.11↑)
MEST (EM) [3]	93.07±0.36	92.59±0.41	90.55±0.44	71.23±0.37	69.08±0.41	64.92±0.34
MEST (EM) + S^2-SAM (ours)	**93.76±0.13** (0.69↑)	**93.18±0.12** (0.59↑)	**91.86±0.23** (1.31↑)	**71.93±0.11** (0.70↑)	**69.98±0.10** (0.90↑)	**66.08±0.16** (1.16↑)
MEST (EM&S) [3]	93.61±0.36	93.46±0.41	92.30±0.44	72.52±0.37	71.21±0.41	69.02±0.34
MEST (EM&S) + S^2-SAM (ours)	**94.51±0.11** (0.90↑)	**93.98±0.11** (0.52↑)	**91.55±0.14** (0.75↑)	**72.90±0.17** (0.38↑)	**72.42±0.09** (1.21↑)	**71.01±0.13** (1.99↑)

B Proof of Lemma 1

Proof. We consider two cases.

(1) When $\mathcal{G} = \mathcal{G}'$ and $\ell(\cdot, f(\mathbf{w};\cdot))$ is L-smooth, then

$$\begin{aligned}\left\|\mathbf{w}_{t+1} - \mathbf{w}'_{t+1}\right\| &\le (1-\eta\lambda)\left\|\mathbf{w}_t - \mathbf{w}'_t\right\| + \eta\left\|\nabla_{\mathbf{u}}\ell\left(\mathbf{y}_t, f\left(\mathbf{u}_t;\mathbf{x}_t\right)\right) - \nabla_{\mathbf{u}}\ell\left(\mathbf{y}_t, f\left(\mathbf{u}'_t;\mathbf{x}_t\right)\right)\right\| \\ &\le (1-\eta\lambda)\left\|\mathbf{w}_t - \mathbf{w}'_t\right\| + \eta L\left\|\mathbf{u}_t - \mathbf{u}'_t\right\| \\ &\le (1+\eta L-\eta\lambda)\left\|\mathbf{w}_t - \mathbf{w}'_t\right\| + 2\eta\rho.\end{aligned}$$

(2) When $\mathcal{G} \neq \mathcal{G}'$ and $\ell(\cdot, f(\mathbf{w};\cdot))$ is B-Lipschitz, then

$$\begin{aligned}\left\|\mathbf{w}_{t+1} - \mathbf{w}'_{t+1}\right\| &\le (1-\eta\lambda)\left\|\mathbf{w}_t - \mathbf{w}'_t\right\| + \eta\left\|\nabla_{\mathbf{u}}\ell\left(\mathbf{y}_t, f\left(\mathbf{u}_t;\mathbf{x}_t\right)\right) - \nabla_{\mathbf{u}}\ell\left(\mathbf{u}'_t, f\left(\mathbf{w}'_t;\mathbf{x}'_t\right)\right)\right\| \\ &\le (1-\eta\lambda)\left\|\mathbf{w}_t - \mathbf{w}'_t\right\| + 2\eta B.\end{aligned}$$

□

C Proof of Theorem 1

Proof. By Theorem 3.2 of [29], we have

$$\begin{aligned}\Delta_{t+1} &:= \mathrm{E}\left[\left\|\mathbf{w}_{t+1} - \mathbf{w}'_{t+1}\right\|\right] \\ &\overset{(a)}{\le} \left(1-\frac{1}{n}\right)(1+\eta L-\eta\lambda)\Delta_t + \left(1-\frac{1}{n}\right)2\eta\rho + \frac{1}{n}\left((1-\eta\lambda)\Delta_t + 2\eta B\right) \\ &= \left[1+\left(1-\frac{1}{n}\right)\eta L - \eta\lambda\right]\Delta_t + \frac{2\eta B}{n} \\ &\overset{(b)}{\le} (1-\eta L)\Delta_t + \frac{2\eta B + 2(n-1)\eta\rho}{n} \\ &\overset{(c)}{\le} \frac{2\eta B + 2(n-1)\eta\rho}{n}\sum_{i=0}^{t_0}(1-\eta L)^i \overset{(d)}{\le} \frac{2B+2(n-1)\rho}{nL},\end{aligned} \tag{15}$$

where (a) uses Lemma 1; (b) uses $\lambda = 2L$; (c) uses $\Delta_{t_0} = 0$; (d) uses $\eta L < 1$. Then, by Lemma 3.11 of [29] and $\widehat{\ell}$ is $(L+\lambda)$-smooth, we have

$$\mathrm{E}\left[\widehat{\ell}\left(\mathbf{w}_t;\mathbf{z}\right) - \widehat{\ell}\left(\mathbf{w}_t';\mathbf{z}\right)\right] \le \frac{t_0}{n} + \frac{2(B+(n-1)\rho)(L+\lambda)}{nL} \le \frac{6(B+(n-1)\rho)+1}{n} \tag{16}$$

where the last inequality holds by selecting $t_0 = 1$ and $\lambda = 2L$. Then, by Theorem 2.2 of [29]

$$\mathrm{E}_{\mathcal{A},\mathcal{S}}\left[\widehat{F}\left(\mathbf{w}_t\right) - \widehat{F}_{\mathcal{S}}\left(\mathbf{w}_t\right)\right] \le \frac{6(B+(n-1)\rho)+1}{n} \tag{17}$$

which is the generalization error. Next, we will bound the optimization error. Since $\widehat{F}_{\mathcal{S}}$ is $(L+\lambda)$-smooth and satisfies μ-PL condition, then follow the standard analysis, we have

Since $\widehat{F}_{\mathcal{S}}$ is $(L+\lambda)$-smooth, we have

$$\begin{aligned}
&\widehat{F}_{\mathcal{S}}\left(\mathbf{w}_t\right) - \widehat{F}_{\mathcal{S}}\left(\mathbf{w}_{t-1}\right)\\
\le&\left\langle \nabla\widehat{F}_{\mathcal{S}}\left(\mathbf{w}_{t-1}\right), \mathbf{w}_t - \mathbf{w}_{t-1}\right\rangle + \frac{L+\lambda}{2}\left\|\mathbf{w}_t - \mathbf{w}_{t-1}\right\|^2\\
=&-\eta\left\langle \nabla\widehat{F}_{\mathcal{S}}\left(\mathbf{w}_{t-1}\right), \nabla_{\mathbf{u}}\widetilde{\ell}(\mathbf{y}, f\left(\mathbf{u}_t,\mathbf{x}\right))\right\rangle + \frac{\eta^2(L+\lambda)}{2}\left\|\nabla_{\mathbf{u}}\widetilde{\ell}(\mathbf{y}, f\left(\mathbf{u}_t,\mathbf{x}\right))\right\|^2\\
\le&-\eta\left\langle \nabla\widehat{F}_{\mathcal{S}}\left(\mathbf{w}_{t-1}\right), \nabla_{\mathbf{w}}\widehat{\ell}(\mathbf{y}, f\left(\mathbf{w}_{t-1},\mathbf{x}\right))\right\rangle - \eta\left\langle \nabla\widehat{F}_{\mathcal{S}}\left(\mathbf{w}_{t-1}\right), \nabla_{\mathbf{u}}\widetilde{\ell}(\mathbf{y}, f\left(\mathbf{u}_t,\mathbf{x}\right)) - \nabla_{\mathbf{w}}\widehat{\ell}(\mathbf{y}, f\left(\mathbf{w}_{t-1},\mathbf{x}\right))\right\rangle \qquad (18)\\
&+\frac{\eta^2(L+\lambda)B^2}{2}\\
\le&-\eta\left\langle \nabla\widehat{F}_{\mathcal{S}}\left(\mathbf{w}_{t-1}\right), \nabla_{\mathbf{w}}\widehat{\ell}(\mathbf{y}, f\left(\mathbf{w}_{t-1},\mathbf{x}\right))\right\rangle + \frac{\eta}{2}\left\|\nabla\widehat{F}_{\mathcal{S}}\left(\mathbf{w}_{t-1}\right)\right\|^2 + \frac{\eta\rho^2}{2} + \frac{\eta^2(L+\lambda)B^2}{2} \qquad (19)
\end{aligned}$$

where the last inequality is due to $\ell(\mathbf{y}, f(\mathbf{w},\mathbf{x}))$ is B-Lipschitz. Therefore, we get

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathrm{E}\left[\left\|\nabla\widehat{F}_{\mathcal{S}}\left(\mathbf{w}_t\right)\right\|^2\right] \le \frac{2\widehat{F}_{\mathcal{S}}\left(\mathbf{w}_0\right)}{\eta} + \rho + \eta(L+\lambda)B^2. \tag{20}$$

Since $\widehat{F}_{\mathcal{S}}$ satisfies μ-PL condition, then

$$\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1}\mathrm{E}_{\mathcal{S}}\left[\mathrm{E}_{\mathcal{A}}\left[\widehat{F}_{\mathcal{S}}\left(\mathbf{w}_t\right) - \widehat{F}_{\mathcal{S}}\left(\widehat{\mathbf{w}}_{\mathcal{S}}^*\right)\right]\right] &\le \frac{1}{2\mu T}\sum_{t=0}^{T-1}\mathrm{E}\left[\left\|\nabla\widehat{F}_{\mathcal{S}}\left(\mathbf{w}_t\right)\right\|^2\right]\\
&\le \frac{\widehat{F}_{\mathcal{S}}\left(\mathbf{w}_0\right)}{\mu\eta T} + \frac{\rho+\eta(L+\lambda)B^2}{2\mu} \qquad (21)
\end{aligned}$$

Then by (12), (17) and (21), we get

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathrm{E}_{\mathcal{A},\mathcal{S}}\left[\widehat{F}\left(\mathbf{w}_t\right)\right] - \mathrm{E}_{\mathcal{S}}\left[\widehat{F}_{\mathcal{S}}\left(\widehat{\mathbf{w}}_{\mathcal{S}}^*\right)\right] \le \frac{\widehat{F}_{\mathcal{S}}\left(\mathbf{w}_0\right)}{\mu\eta T} + \frac{\rho+\eta(L+\lambda)B^2}{2\mu} + \frac{6(B+(n-1)\rho)+1}{n} \tag{22}$$

By using the conditions that $\min_{\mathbf{w}\in\mathcal{W}}\widehat{F}_{\mathcal{S}}(\mathbf{w}) \le F_{\mathcal{S}}\left(\mathbf{w}_{\mathcal{S}}^*\right) + \frac{\lambda}{2}\left\|\mathbf{w}_t\right\|^2$, definitions (11), we get

$$F\left(\mathbf{w}_t\right) - F_{\mathcal{S}}\left(\mathbf{w}_{\mathcal{S}}^*\right) \le \widehat{F}\left(\mathbf{w}_t\right) - \widehat{F}_{\mathcal{S}}\left(\widehat{\mathbf{w}}_{\mathcal{S}}^*\right) \tag{23}$$

Therefore, we have the following inequality by (22) and (23):

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathrm{E}_{\mathcal{A},\mathcal{S}}\left[F\left(\mathbf{w}_t\right)\right] - \mathrm{E}_{\mathcal{S}}\left[F_{\mathcal{S}}\left(\mathbf{w}_{\mathcal{S}}^*\right)\right] \le \frac{\widehat{F}_{\mathcal{S}}\left(\mathbf{w}_0\right)}{\mu\eta T} + \frac{\rho+\eta(L+\lambda)B^2}{2\mu} + \frac{6(B+(n-1)\rho)+1}{n} \tag{24}$$

By Lemma 5.1 of [29], (24) implies the ERB is

$$\mathrm{E}_{R,\mathcal{A},\mathcal{S}}\left[F\left(\mathbf{w}_R\right)\right] - \mathrm{E}_{\mathcal{S}}\left[F\left(\mathbf{w}_*\right)\right] \le \frac{\widehat{F}_{\mathcal{S}}\left(\mathbf{w}_0\right)}{\mu\eta T} + \frac{\rho+\eta(L+\lambda)B^2}{2\mu} + \frac{6(B+(n-1)\rho)+1}{n} \tag{25}$$

By setting $\eta = O(1/\sqrt{n})$ and $T = O(n)$ then $\mathrm{E}_{R,\mathcal{A},\mathcal{S}}\left[F\left(\mathbf{w}_R\right)\right] - F\left(\mathbf{w}_*\right) \le O(1/\sqrt{n})$, where $\mathcal{A}$ is S^2-SAM. □

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We propose S^2-SAM which is tailored for sparse training. The proposed method is efficient with zero extra cost and achieves better accuracy on sparse training methods. We also demonstrate theoretical and experimental results in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The research is inherently scientific, and we anticipate no adverse societal impact stemming from its findings. We discuss the limitations in the experiments that our method is focusing on sparse training. But we also include dense training results to show potentials of our method. We will leave the dense model training of S^2-SAM in our future research.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

https://doi.org/10.52202/079017-1405

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide theoretical results in Section 2.3 and appendix.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All of our results can be reproduced and we will release our code after the paper is accepted.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

https://doi.org/10.52202/079017-1405

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All datasets used in the paper are publicly accessible. We will also release our code after the paper is accepted.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All important settings can be found in our paper and the papers we referenced.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We perform our experiments for 3 times as shown in our experimental settings. We report mean and standard deviation.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

https://doi.org/10.52202/079017-1405

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the training resources we use in the experimental settings.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impacts in the conclusion. The research is inherently scientific, and we anticipate no adverse societal impact stemming from its findings.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

https://doi.org/10.52202/079017-1405

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our method doesn't have high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We credit every referenced assets with citations.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

https://doi.org/10.52202/079017-1405

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: This paper introduces new code for sharpness-aware minimization, and we will release the code after the paper is accepted.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper doesn't perform crowdsourcing and research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper doesn't perform research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

https://doi.org/10.52202/079017-1405

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-1405