Understanding SAM's Robustness to Noisy Labels Through Gradient Down-weighting

Hoang-Chau Luong[1,2†] **Quang-Thuc Nguyen**[2†] **Dat Ba Tran**[3] **Minh-Triet Tran**[2]
[1] Rochester Institute of Technology, Rochester, NY, USA
[2] University of Science - VNUHCM, Vietnam
[3] Rowan University, Glassboro, NJ, USA
[†] These authors contributed equally

Abstract

Sharpness-Aware Minimization (SAM) was introduced to improve generalization by seeking flat minima, yet it also exhibits robustness to label noise, a phenomenon that remains only partially understood. Prior work has mainly attributed this effect to SAM's tendency to prolong the learning of clean samples. In this work, we provide a complementary explanation by analyzing SAM at the element-wise level. We show that when noisy gradients dominate a parameter direction, their influence is reduced by the stronger amplification of clean gradients. This slows the memorization of noisy labels while sustaining clean learning, offering a more complete account of SAM's robustness. Building on this insight, we propose SANER (Sharpness-Aware Noise-Explicit Reweighting), a simple variant of SAM that explicitly magnifies this down-weighting effect. Experiments on benchmark image classification tasks with noisy labels demonstrate that SANER significantly mitigates noisy-label memorization and improves generalization over both SAM and SGD. Moreover, since SANER is designed from the mechanism of SAM, it can also be seamlessly integrated into SAM-like variants, further boosting their robustness.

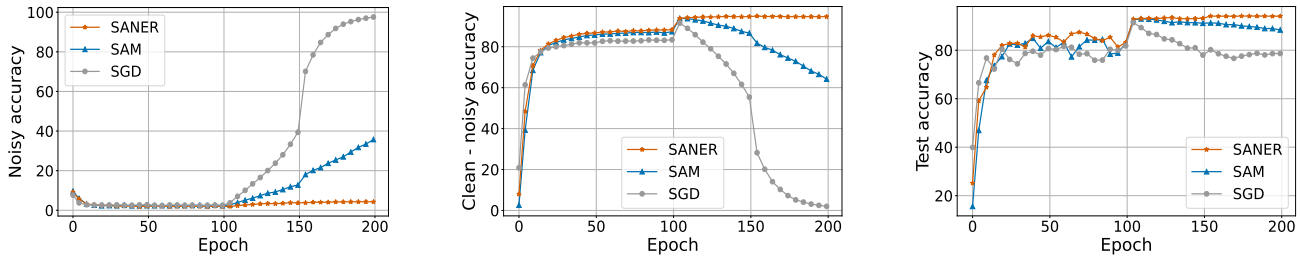
Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

1 INTRODUCTION

Robust learning under label noise is a fundamental challenge in deep learning, as real-world datasets often contain annotation errors that severely hinder generalization. Well-known datasets such as CIFAR-10N (Wei et al., 2022), CIFAR-100N (Wei et al., 2022), and WebVision (Li et al., 2017) highlight this issue, where large-scale human labeling introduces substantial noise. Over-parameterized deep neural networks (DNNs) are particularly prone to memorizing these noisy labels (Zhang et al., 2021a), which makes it harder to learn useful patterns and leads to poor performance on clean test data (Jiang et al., 2018; Nguyen et al., 2020). Additionally, because manually verifying labels at scale is costly, it is essential to develop training algorithms that robust to noisy labels while still leveraging clean supervision to ensure strong generalization.

Among recent optimization methods, SAM (Foret et al., 2021) has attracted attention not only for its ability to seek flat minima and improve generalization but also for its robustness to noisy labels. Moreover, several variants of SAM, though not explicitly developed to address label noise, also exhibit notable robustness in noisy-label scenarios (Kwon et al., 2021; Zhuang et al., 2022; Li et al., 2024; Li and Giannakis, 2024). Baek et al. (2024) observed that when trained with SAM under label noise, test accuracy peaks midway through training and does not improve with further epochs, indicating that its robustness cannot be attributed solely to flat-minima effects. Understanding this robustness is crucial, as it both deepens our theoretical understanding of noise-robust optimization and guides the design of more effective algorithms for learning with noisy labels.

Baek et al. (2024) suggested that SAM's robustness to label noise arises primarily from its ability to prolong the learning of clean samples by up-weighting



(a) The noisy training accuracy (b) The gap between clean and noisy (c) The test accuracy

Figure 1: Performance comparison of SGD, SAM, and SANER (ours) on ResNet18 trained on CIFAR-10 with 25% label noise. Noise accuracy measures how much the model memorizes noisy labels, where lower values are better. SAM reduces noisy label fitting (a) and increases the clean–noise accuracy gap (b), computed as clean accuracy minus noisy accuracy. SANER further strengthens this effect, resulting in improved test accuracy (c).

their gradients. While this explanation provides an important insight, it remains incomplete. In particular, SAM not only strengthens clean-sample gradients but also amplifies noisy-sample gradients, which raises the question of why it still achieves greater robustness than standard SGD. To address this gap, we analyze SAM at the element-wise gradient level and uncover characteristics that more fully explain its ability to slow down fitting to noisy labels.

Our theoretical analysis shows that during the transitional phase—when the model shifts from learning clean patterns to memorizing noisy labels—SAM down-weights element-wise gradients aligned with noisy supervision. This occurs because, when noisy gradients dominate a parameter direction, SAM rebalances the update. Particularly, it amplifies the contribution of clean gradients more strongly than that of noisy ones. This finding is significant because it complements SAM's tendency to prolong clean learning, offering a more complete explanation of its robustness.

We further validate this mechanism in DNNs through a series of controlled experiments, which consistently support our theoretical findings. Motivated by these insights, we propose **SANER** (**S**harpness-**A**ware **N**oise-**E**xplicit **R**eweighting), a simple modification of SAM that magnifies this down-weighting phenomenon to more aggressively resist noisy-label memorization. As shown in Figure 1, SANER significantly reduces overfitting to noisy labels and achieves better generalization than both SAM and SGD.

In summary, our contributions are threefold:

- We provide a theoretical analysis showing that SAM down-weights specific gradient elements, thereby slowing the learning of noisy labels.
- We empirically validate this behavior in DNNs, showing that the down-weighted elements align closely with gradients induced by label noise, and that their removal degrades SAM's robustness.
- We introduce SANER, a simple modification of SAM that more strongly suppresses noisy gradient elements. Extensive experiments demonstrate that SANER consistently outperforms SAM across different datasets, noise types, and architectures. Moreover, SANER can be seamlessly integrated with other SAM-like optimizers, further improving their robustness to label noise.

2 BACKGROUND & RELATED WORKS

Sharpness-Aware Minimization. SAM (Foret et al., 2021) was originally proposed to improve generalization by encouraging convergence to flat minima through a perturbation-based update. Since then, numerous variants have been developed and shown to improve performance in noisy-label settings (Kwon et al., 2021; Kim et al., 2022; Jiang et al., 2023; Li and Giannakis, 2024; Li et al., 2024). SAM has also been studied from diverse perspectives, including implicit bias (Andriushchenko and Flammarion, 2022), training dynamics (Bartlett et al., 2023), Hessian regularization (Wen et al., 2023), bias–variance trade-offs (Behdin and Mazumder, 2023), and benign overfitting (Chen et al., 2024). Its convergence has been analyzed in the framework of Inexact Gradient Descent (Khanh et al., 2023, 2024b), where SAM's perturbed gradient is treated as an approximation of the unperturbed gradient (Khanh et al., 2024a). Shin et al. (2023) showed that SAM benefits from over-parameterization under label noise, while Baek et al. (2024) linked its robustness to sample-wise gradients. We extend these analyses to the element-wise level, showing that SAM implicitly down-weights specific gradient components, thereby slowing noisy-label memorization.

Label Noise. Deep neural networks are prone to memorizing noisy labels, which can severely degrade generalization (Zhang et al., 2021a). To address this, a wide range of approaches have been developed. One line of work designs robust loss functions that are less sensitive to corrupted labels (Zhang and Sabuncu, 2018; Menon et al., 2020; Ma et al., 2020; Wei et al., 2023). Another introduces sample reweighting or selection strategies to prioritize clean data and reduce the effect of noisy examples (Liu and Tao, 2015; Ren et al., 2018; Jiang et al., 2018; Wei et al., 2020). Additional methods employ regularization (Tanaka et al., 2018; Lukasik et al., 2020; Bai et al., 2021; Zhang et al., 2021b; Liu et al., 2022) or leverage semi-supervised, meta-learning, and self-supervised training paradigms (Nguyen et al., 2020; Li et al., 2020; Shu et al., 2019; Li et al., 2022). While effective, these techniques often depend on noise modeling or auxiliary procedures, which limit their scalability in practice. In contrast, our approach leverages the inherent behavior of the SAM optimizer to mitigate label-noise memorization.

SAM algorithm. Let $f(\boldsymbol{x}_i;\boldsymbol{w})$ represent a mapping function from inputs $\boldsymbol{x_i} \in \mathbb{R}^d$ to outputs $\hat{y}_i \in \mathbb{R}$ which is parameterized by $\boldsymbol{w} \in \mathbb{R}^d$, and let $\ell(f(\boldsymbol{x}_i;\boldsymbol{w}), y_i)$ (shortened as $\ell_i(\boldsymbol{w})$) denote the loss function between the prediction $\hat{y}_i = f(\boldsymbol{x}_i;\boldsymbol{w})$ and the ground-truth label y_i. To enhance generalization performance, SAM (Foret et al., 2021) is proposed to seek a flat minimum of the empirical training loss by minimizing the following robust objective:

$$\min_{\boldsymbol{w}} \max_{||\boldsymbol{\epsilon}||_2 \le \rho} L(\boldsymbol{w}+\boldsymbol{\epsilon}), \text{ where } L(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(\boldsymbol{w}), \quad (1)$$

where perturbation radius $\rho \in \mathbb{R}$ represents the strength of the adversarial weight perturbation $\boldsymbol{\epsilon} \in \mathbb{R}^d$. Intuitively, the objective seeks a robust solution such that within a neighbor region, the loss can remain stable under any $\boldsymbol{\epsilon}$-perturbation. SAM employs a first-order Taylor approximation of the loss to efficiently optimize this objective, which approximates the worst-case $\hat{\boldsymbol{\epsilon}}$ as follows

$$\hat{\boldsymbol{\epsilon}} = \rho \frac{\boldsymbol{g}^{\text{SGD}}}{||\boldsymbol{g}^{\text{SGD}}||} = \rho \frac{\nabla_{\boldsymbol{w}} L(\boldsymbol{w})}{||\nabla_{\boldsymbol{w}} L(\boldsymbol{w})||}. \quad (2)$$

SAM gradient is computed at the perturbed point $\boldsymbol{w}+\hat{\boldsymbol{\epsilon}}$, and the base optimizer (e.g., SGD) with a learning rate η is used to update the model parameters:

$$\boldsymbol{w} = \boldsymbol{w} - \eta \boldsymbol{g}^{\text{SAM}} = \boldsymbol{w} - \eta \nabla_{\boldsymbol{w}} L(\boldsymbol{w})\Big|_{\boldsymbol{w}+\hat{\boldsymbol{\epsilon}}}. \quad (3)$$

This update steers the model parameters toward a perturbation-robust solution, requiring only one additional gradient computation per iteration.

3 ANALYSIS ELEMENT-WISE GRADIENT BEHAVIOR OF SAM

3.1 Problem Setup

We consider a binary classification setting. Given a training set of n samples $[(\boldsymbol{x}_i, y_i)]_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \{0,1\}$ is the corresponding label, our objective is to learn model parameters $\boldsymbol{w} \in \mathbb{R}^d$ that can give the correct prediction $\hat{y}_i = \sigma(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle)$, where $\sigma(x) = 1/(1+\exp(-x))$ by minimizing the empirical loss $L(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^n \ell(\boldsymbol{w}, \boldsymbol{x}_i, y_i)$. The binary cross-entropy loss is defined as

$$\ell(\boldsymbol{w}, \boldsymbol{x}_i, y_i) = H(y_i, \hat{y}_i) \quad (4)$$
$$= -y_i \log \hat{y}_i - (1-y)\log(1-\hat{y}_i). \quad (5)$$

To introduce label noise, we randomly flip the labels of a subset of training samples (e.g., from 0 to 1 or 1 to 0).

SGD gradient. The gradient of the loss with respect to $\boldsymbol{w}$ for a single sample $(\boldsymbol{x}_i, y_i)$, denoted as $\nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}; \boldsymbol{x}_i, y_i)$, is commonly referred to as a sample-wise gradient. It is the update direction used in standard SGD and is computed as:

$$\nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}, \boldsymbol{x}_i, y_i) = \Big(\sigma\big(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle\big) - y_i\Big)\boldsymbol{x}_i. \quad (6)$$

1-SAM gradient. We focus on 1-SAM variant of SAM, which computes an adversarial perturbation $\boldsymbol{\epsilon}_i$ for each individual training sample rather than over the mini-batch. This variant has been shown to yield strong performance (Foret et al., 2021) and is used for theoretical analysis in Baek et al. (2024). The 1-SAM sample-wise gradient for the i-th sample is defined as

$$\nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}+\boldsymbol{\epsilon}_i, \boldsymbol{x}_i, y_i) = \Big(\sigma\big(\langle \boldsymbol{w}+\boldsymbol{\epsilon}_i, \boldsymbol{x}_i \rangle\big) - y_i\Big)\boldsymbol{x}_i, \quad (7)$$

where the adversarial perturbation is

$$\boldsymbol{\epsilon}_i = \rho \frac{\big(\sigma(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle) - y_i\big)\boldsymbol{x}_i}{\big\|\big(\sigma(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle) - y_i\big)\boldsymbol{x}_i\big\|} = \rho(-1)^{y_i}\frac{\boldsymbol{x}_i}{\|\boldsymbol{x}_i\|}, \quad (8)$$

where the last equality follows from the fact that $\sigma\big(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle\big) - y_i > 0$ if $y_i = 0$, and $\sigma\big(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle\big) - y_i < 0$ if $y_i = 1$.

3.2 Theoretical Analysis in Linear Models

We theoretically show that SAM selectively down-weights specific element-wise gradients compared to standard SGD to mitigate noisy-label memorization.

Down-weighted gradient elements. We consider a mini-batch consisting of two samples from the same true class, where one is correctly labeled and the other

is corrupted due to label noise. Without loss of generality, let the clean sample be $(\boldsymbol{x}_c, y_c)$ with $y_c = 0$, and the noisy sample be $(\boldsymbol{x}_\eta, y_\eta)$ with $y_\eta = 1$. Because both samples are drawn from the same underlying class, we assume there exists at least one shared feature dimension j such that $x_c^j = x_\eta^j$. We define the ratio between the 1-SAM and SGD gradients at dimension j as $g^{\text{SAM}}/g^{\text{SGD}}$, or equivalently

$$r^j = \frac{\nabla_{\boldsymbol{w}}\ell(\boldsymbol{w}+\boldsymbol{\epsilon}_c, \boldsymbol{x}_c, y_c) + \nabla_{\boldsymbol{w}}\ell(\boldsymbol{w}+\boldsymbol{\epsilon}_\eta, \boldsymbol{x}_\eta, y_\eta)}{\nabla_{\boldsymbol{w}}\ell(\boldsymbol{w}, \boldsymbol{x}_c, y_c) + \nabla_{\boldsymbol{w}}\ell(\boldsymbol{w}, \boldsymbol{x}_\eta, y_\eta)}$$
$$= \frac{1-\sigma\big(\langle \boldsymbol{w}, \boldsymbol{x}_c\rangle + \rho\|\boldsymbol{x}_c\|\big) - \sigma\big(\langle \boldsymbol{w}, \boldsymbol{x}_\eta\rangle - \rho\|\boldsymbol{x}_\eta\|\big)}{1-\sigma\big(\langle \boldsymbol{w}, \boldsymbol{x}_c\rangle\big) - \sigma\big(\langle \boldsymbol{w}, \boldsymbol{x}_\eta\rangle\big)}.$$

We say that the gradient at dimension j is down-weighted by SAM if $0 < r^j < 1$, indicating that SAM reduces the gradient magnitude relative to SGD.

The transitional phase. Our analysis focuses on the transitional phase of training, during which the model shifts from fitting clean samples to memorizing noisy labels. This phase is particularly important because the difference between SAM and SGD becomes most evident here, as illustrated in Figure 1: after the 100th epoch, once the clean samples are well learned, both methods begin to fit noisy labels, but the rate at which SAM memorizes noisy labels is significantly lower than that of SGD. Prior work (Liu et al., 2020) has shown that in linear models, the early training dynamics are dominated by the majority of clean samples, and thus clean data are typically learned first. At this stage, the model can predict the correct true class for both clean and noisy samples. Since both $\boldsymbol{x}_c$ and $\boldsymbol{x}_\eta$ belong to the true class $y = 0$, the linear model outputs probabilities less than 0.5, or equivalently $\langle \boldsymbol{w}, \boldsymbol{x}_c\rangle < 0$ and $\langle \boldsymbol{w}, \boldsymbol{x}_\eta\rangle < 0$.

Remark 1. In the transitional phase, noisy gradient dominate the mini-batch gradient. Specifically, the gradients of the clean sample $\boldsymbol{x}_c$ and the noisy sample $\boldsymbol{x}_\eta$ take opposite signs, since $\sigma(\langle \boldsymbol{w}, \boldsymbol{x}_c\rangle)(\sigma(\langle \boldsymbol{w}, \boldsymbol{x}_\eta\rangle) - 1) < 0$. Furthermore, the noisy sample induces a larger gradient magnitude than the clean sample: $|\sigma(\langle \boldsymbol{w}, \boldsymbol{x}_c\rangle)| < |\sigma(\langle \boldsymbol{w}, \boldsymbol{x}_\eta\rangle) - 1|$. As a result, the mini-batch gradient is biased toward learning the noisy sample, which drives the model to fit incorrect labels.

Lemma 3.1. *Let $C > 0$ be a constant and $\sigma(x) = \frac{1}{1+e^{-x}}$ denote the sigmoid function. For any negative real numbers z_1, z_2 such that $z_1 > z_2 - C$, we have*

$$0 < \frac{1-\sigma(z_1+C)-\sigma(z_2-C)}{1-\sigma(z_1)-\sigma(z_2)} < 1. \tag{9}$$

The proof of Lemma 3.1 is provided in the Appendix A.1.

Remark 2. SAM slows down the learning of noisy labels. Let $z_1 = \langle \boldsymbol{w}, \boldsymbol{x}_c\rangle$, $z_2 = \langle \boldsymbol{w}, \boldsymbol{x}_\eta\rangle$, and $C = \rho\|\boldsymbol{x}_\eta\|$. Assume both samples yield similar confidence predictions and have equal norms $\|\boldsymbol{x}_c\| = \|\boldsymbol{x}_\eta\|$. Then the condition $z_1 > z_2 - C$ holds, and Lemma 3.1 guarantees that $0 < r^j < 1$. This confirms that SAM down-weights the noisy gradient relative to SGD. Combining this with Remark 1, which establishes that SGD gradient is biased toward overfitting to noisy labels, we conclude that SAM mitigates this bias by reducing the influence of such gradients.

Discussion. Eq. (9) can be reformulated to provide an intuitive interpretation. The condition $r^j < 1$ holds if:

$$(\sigma(z_2-C)-1)-(\sigma(z_2)-1) < \sigma(z_1+C)-\sigma(z_1). \tag{10}$$

Here, the left-hand side represents the change in the noisy sample's gradient induced by SAM, while the right-hand side reflects the corresponding change for the clean sample. This shows that SAM amplifies the clean gradient more than the noisy one, thereby reducing the influence of noise-driven gradients in the aggregated mini-batch gradient and giving rise to its down-weighting behavior.

3.3 Empirical Validation in DNNs

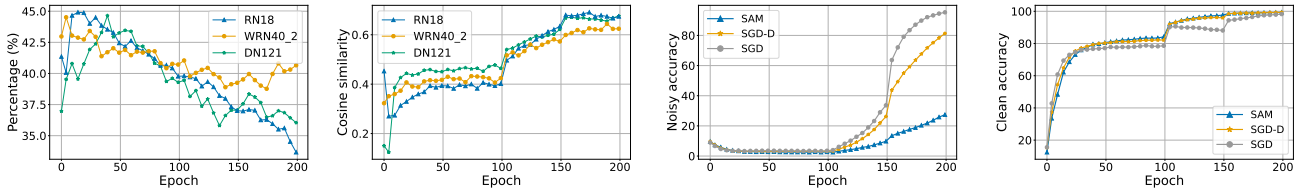
We empirically validate the theoretical analysis from Section 3.2 regarding SAM's gradient down-weighting behavior in DNNs. All experiments are conducted on CIFAR-10 with 50% symmetric label noise, using ResNet18 (RN18), WideResNet40-2 (WRN40_2), and DenseNet121 (DN121).

SAM consistently down-weights a significant portion of gradient elements. To assess the occurrence of SAM's down-weighting behavior, we compute the element-wise ratio between SAM and SGD gradients, $r_i = g_i^{\text{SAM}}/g_i^{\text{SGD}}$, for each update. Let d be the number of network parameters, we then define the set of down-weighted elements as

$$\mathcal{S}^{dw} = \{i \in \{1, 2, \ldots, d\} \mid 0 < r_i < 1\}, \tag{11}$$

As shown in Figure 2a, SAM consistently down-weights a substantial fraction of gradient elements (35–45%) throughout training across three different architectures, confirming that this is a systematic and nontrivial behavior.

Down-weighted gradient elements align with gradients from noisy labels. To validate that down-weighted elements correspond to updates induced by noisy labels, we measure the cosine similarity between (i) the SGD gradient computed on the entire mini-batch but projected onto the elements in $\mathcal{S}^{dw}$, denoted as $\boldsymbol{g}^{\mathcal{S}^{dw}}$, and (ii) the SGD gradient computed



(a) (b) (c) (d)

Figure 2: (a) Percentage (%) of down-weighted elements $\mathcal{S}^{dw}$. (b) Cosine similarity between $g^{\mathcal{S}^{dw}}$ and g^{noise}. (c) Noisy accuracy of SGD, SAM, and a SAM variant without gradient down-weighting (SGD-D). (d) Clean accuracy of SGD, SAM, and a SAM variant without gradient down-weighting (SGD-D).

using only the noisy examples within the same mini-batch, denoted as $\boldsymbol{g}^{\text{noise}}$. Specifically, we define the down-weighted SGD gradient at each element i as

$$g_i^{\mathcal{S}^{dw}} = \begin{cases} g_i^{\text{SGD}} & \text{if } i \in \mathcal{S}^{dw}, \\ 0 & \text{otherwise.} \end{cases}$$

Figure 2b shows that this similarity increases over time, peaking around ∼0.7 between epochs 150 and 175. This indicates that the elements down-weighted by SAM are closely aligned with gradients from noisy examples, supporting our claim that SAM mitigates noise-driven gradients at the element-wise level.

Isolating the effect of down-weighting on SAM's robustness. To evaluate the importance of SAM's down-weighting mechanism, we conduct an ablation study by creating a variant that nullifies this behavior. Specifically, we construct a variant, *SAM without down-weighting* (SGD-D), in which gradient elements in $\mathcal{S}^{dw}$ are replaced with their SGD counterparts:

$$g_i^{\text{SGD-D}} = \begin{cases} g_i^{\text{SGD}}, & \text{if } i \in \mathcal{S}^{dw}, \\ g_i^{\text{SAM}}, & \text{otherwise.} \end{cases}$$

As shown in Figure 2c, this variant substantially accelerates noisy-label memorization, reaching nearly 80% by the final epoch, whereas SAM remains around 30% at the same stage. Although SGD-D also exhibits a pronounced increase in memorizing noisy labels, its performance on clean samples remains comparable to SAM. This contrast highlights that SAM's resistance to noisy labels depends critically on its selective down-weighting of noise-aligned gradient elements.

4 ENHANCING NOISE ROBUSTNESS IN SAM VIA GRADIENT REWEIGHTING

Building on our analysis in the previous section, we introduce **SANER**, an optimizer designed to enhance SAM's robustness to label noise. The core insight is that the gradient elements implicitly down-weighted by SAM are those most responsible for noisy-label memorization. SANER leverages this finding by applying an explicit reweighting scheme that further suppresses these gradient components. We provide extensive experimental validation, showing that SANER consistently outperforms both SAM and SGD across various setups. In addition, SANER improves the performance of other SAM-like variants, while preserving their advantage in challenging overfitting scenarios.

4.1 The SANER Algorithm

SANER improves robustness by explicitly identifying and re-weighting the gradient elements that SAM naturally down-weights. Specifically, we introduce a binary mask vector $\boldsymbol{m} \in \{0,1\}^d$ to identify these elements and apply a scaling hyperparameter $\alpha \in \mathbb{R}$ to adjust their contribution. The binary mask for the i-th index is computed as

$$m_i = \begin{cases} 1, & \text{if } 0 < r_i < 1, \\ 0, & \text{otherwise,} \end{cases} \quad i = 1, \ldots, d. \tag{12}$$

Then, the SANER update is thus given by:

$$\boldsymbol{g}^{\text{SANER}} = \big(\mathbf{1} - \alpha \boldsymbol{m}\big) \odot \boldsymbol{g}^{\text{SAM}}. \tag{13}$$

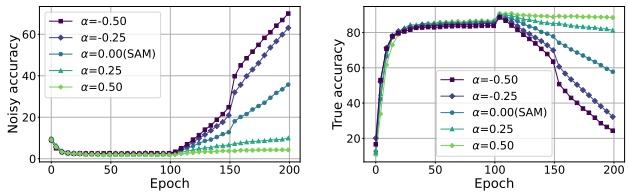
It is important to note that SANER *incurs the same computational cost as SAM*, since it does not require any additional gradient evaluations. The complete procedure is summarized in Algorithm 1.

Setting $\alpha > 0$ reduces noisy fitting. Our theoretical and empirical analyses in Section 3 show that SAM's down-weighted gradient elements slow the learning of noisy labels. Building on this, we hypothesize that explicitly amplifying this effect through a scaling parameter α can further improve robustness. To test this, we evaluate SANER with different values of $\alpha \in \{-0.50, -0.25, 0.00\,(\text{SAM}), 0.25, 0.50\}$ on ResNet18 with CIFAR-10. We report both the noisy training accuracy and the true accuracy on noisy samples, i.e., accuracy with respect to their clean labels.

Figure 3 shows that values $\alpha < 0$ increase noisy fitting compared to SAM and degrade performance on the

Algorithm 1 SANER

Input: Learning rate η, initial parameters $\boldsymbol{w}_0$, iterations T, perturbation size ρ, noise control α
Initialize model parameters: $\boldsymbol{w} = \boldsymbol{w}_0$
for $t = 0$ to T **do**
  Sample a mini-batch of m training examples to calculate gradient: $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$
  Compute the SGD gradient: $\boldsymbol{g}^{\text{SGD}} = \nabla_{\boldsymbol{w}} L(\boldsymbol{w})$
  Compute the SAM gradient: $\boldsymbol{g}^{\text{SAM}}$ via Eq. (3)
  Calculate the gradient ratio: $\boldsymbol{r} = \boldsymbol{g}^{\text{SAM}} / \boldsymbol{g}^{\text{SGD}}$ (element-wise division)
  Compute $\boldsymbol{m}$ and $\boldsymbol{g}^{\text{SANER}}$ via Eq. 16 and 17
  Update parameters: $\boldsymbol{w} = \boldsymbol{w} - \eta \boldsymbol{g}^{\text{SANER}}$
end for
Output: Learned parameters $\boldsymbol{w}$



(a) Noisy accuracy. (b) True accuracy on noisy.

Figure 3: (a) Effect of hyperparameter α on noisy accuracy and lower values of α enhance noise resistance. (b) shows the true accuracy on noisy samples (measured by the true labels of noisy samples).

true labels of noisy samples. In contrast, $\alpha > 0$ suppresses noisy fitting and better preserves performance on these samples. The results further demonstrate that α in our proposed method is strongly correlated with the degree of memorization of noisy labels.

4.2 Main Results

We conduct experiments on several datasets, including CIFAR-10/100 (Krizhevsky et al., 2009), Tiny-ImageNet (Le and Yang, 2015), and Mini-Webvision (Li et al., 2017). To evaluate robustness under different conditions, we consider four types of label noise: symmetric noise, asymmetric noise (Zhang and Sabuncu, 2018), instance-dependent noise (Xia et al., 2020), and real-world noise. More details on these noise types are provided in the Appendix B.

Training details. All models were trained for 200 epochs from scratch using SGD with a momentum of 0.9, a weight decay of 5×10^{-4}, and a batch size of 128. We applied standard data augmentation (random crops and horizontal flips). The learning rate was initialized to 0.1 and reduced by a factor of 10 at epochs 100 and 150, following prior works (Andriushchenko and Flammarion, 2022; Shin et al., 2023).

For computing the SAM gradient, we set the perturbation radius $\rho = 0.1$ in all experiments, following Foret et al. (2021). Our noise control hyperparameter α was set to 0.5, which provided stable performance across various settings from the set $\{0.10, 0.25, 0.50, 0.75, 0.9\}$. We linearly increase α from 0.0 to its target value over the first 50 epochs to avoid abrupt changes during early training. All results are reported as the mean and standard deviation of the best test accuracy, averaged over three runs with different random seeds.

CIFAR results. We evaluate SANER on ResNet18 (He et al., 2016) under various noise types and rates, with results summarized in Table 1. SANER consistently outperforms both SGD and SAM across all settings. On CIFAR-10, it achieves an average improvement of about 1% over SAM, with a maximum gain of 2.7%. On CIFAR-100, the improvement is more substantial—averaging 3% and reaching up to 8%. These results demonstrate that SANER effectively mitigates overfitting to noisy labels. The improvements on CIFAR-100 are particularly notable, as its larger number of classes makes it more susceptible to noise (Han et al., 2018), highlighting SANER's robustness in more challenging settings.

Tiny-ImageNet and Mini-WebVision results. Tiny-ImageNet (Le and Yang, 2015), a subset of ImageNet (Deng et al., 2009), contains 100,000 color images of size 64×64 across 200 classes. We introduce 25% symmetric label noise for this dataset. For Mini-WebVision, we follow the "Mini" setting from (Jiang et al., 2018), selecting the first 50 classes from the Google-resized subset and evaluating on the corresponding 50 classes from the clean ImageNet 2012 validation set (Russakovsky et al., 2015). As shown in Table 2, SANER outperforms SAM by approximately 4% on Tiny-ImageNet and improves test accuracy by 3% on Mini-WebVision.

Different architectures. We evaluate SGD, SAM, and SANER on CIFAR-100 across different architectures including ResNet34 (He et al., 2016), DenseNet121 (Huang et al., 2017), WideResNet40-2 and WideResNet28-10 (Zagoruyko and Komodakis, 2017) to assess SANER's adaptability across models. Table 3 shows that SANER consistently surpasses SAM across all settings, highlighting the role of SAM's element-wise down-weighting in enhancing robustness across diverse architectures.

Integration with SAM-like optimizers. Beyond designing SANER specifically for SAM, we extend our evaluation to other optimizers built upon SAM's theoretical foundation to examine the general effectiveness of SANER. To this end, we evaluate SANER's effec-

Table 1: Test accuracy comparison of SAM and SANER across different noise types and rates, trained on CIFAR-10 and CIFAR-100 with ResNet18. Bold values indicate the highest test accuracy. The teal values with an uparrow (↑) indicate the improvement of SANER over SAM.

Type	Noise rate	CIFAR-10		CIFAR-100	
		SAM	SANER	SAM	SANER
Symmetric noise	25%	$93.05_{\pm 0.17}$	$\mathbf{94.08}_{\pm 0.11}$(↑1.03)	$69.68_{\pm 0.07}$	$\mathbf{72.90}_{\pm 0.21}$(↑3.22)
	50%	$88.82_{\pm 0.08}$	$\mathbf{90.60}_{\pm 0.36}$(↑1.78)	$61.17_{\pm 0.14}$	$\mathbf{66.34}_{\pm 0.11}$(↑5.17)
Asymmetric noise	25%	$94.75_{\pm 0.28}$	$\mathbf{94.83}_{\pm 0.14}$(↑0.08)	$71.57_{\pm 0.30}$	$\mathbf{74.64}_{\pm 0.13}$(↑3.07)
	50%	$81.94_{\pm 0.71}$	$\mathbf{82.25}_{\pm 1.43}$(↑0.31)	$39.11_{\pm 0.50}$	$\mathbf{40.05}_{\pm 0.51}$(↑0.94)
Dependent noise	25%	$92.84_{\pm 0.18}$	$\mathbf{93.67}_{\pm 0.30}$(↑0.83)	$69.46_{\pm 0.24}$	$\mathbf{72.93}_{\pm 0.29}$(↑3.47)
	50%	$87.32_{\pm 1.17}$	$\mathbf{90.01}_{\pm 0.62}$(↑2.69)	$58.71_{\pm 0.69}$	$\mathbf{66.72}_{\pm 0.75}$(↑8.01)
Real label noise	-	$86.33_{\pm 0.07}$	$\mathbf{87.89}_{\pm 0.12}$(↑1.56)	$62.74_{\pm 0.59}$	$\mathbf{64.75}_{\pm 0.30}$(↑2.01)

Table 2: The test accuracy on the Tiny-ImageNet (Tiny-IN) for different models and the Top-1 validation accuracy on the clean ImageNet 2012 validation set for ResNet18 trained on Mini-WebVision (Mini-WV). Bold values indicate the highest test accuracy.

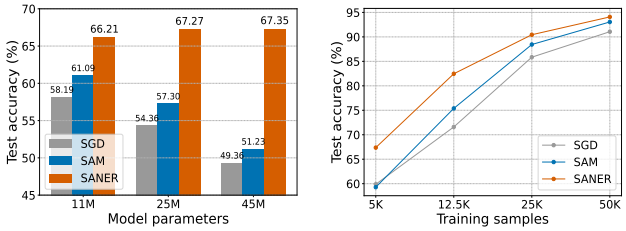
Dataset	Architecture	SGD	SAM	SANER
Tiny-IN	ResNet18	56.50	57.60	**61.60**
	ResNet34	56.82	59.30	**63.22**
	WRN28-10	57.94	59.84	**64.08**
Mini-WV	ResNet18	64.96	67.48	**70.84**

tiveness when integrated into SAM-like optimizers on CIFAR-10 and CIFAR-100 using ResNet18. Specifically, we compare standard versions of ASAM (Kwon et al., 2021), GSAM (Zhuang et al., 2022), FSAM (Li et al., 2024), and VaSSO (Li and Giannakis, 2024) with the SANER-enhanced versions. Detailed implementation settings and integration procedures are provided in Appendix C.1.

As shown in Table 4, SANER consistently improves test accuracy across all noise levels. The gains are particularly notable on CIFAR-100 under 50% symmetric noise, where SANER boosts performance by 4–5%. These results validate the shared behavior among SAM-like optimizers regarding down-weighted gradient elements.

Table 3: Test accuracy comparison of different architectures using SGD, SAM, and SANER on CIFAR-100 (Symmetric noise). Bold values indicate the highest test accuracy and γ denotes the noise rate.

Model	γ	SGD	SAM	SANER
ResNet34	25%	$69.07_{\pm 0.53}$	$71.10_{\pm 0.83}$	$\mathbf{74.02}_{\pm 0.22}$
	50%	$59.73_{\pm 1.26}$	$62.49_{\pm 1.18}$	$\mathbf{67.26}_{\pm 0.28}$
DN121	25%	$69.13_{\pm 0.48}$	$71.61_{\pm 0.49}$	$\mathbf{73.89}_{\pm 0.64}$
	50%	$58.19_{\pm 1.20}$	$60.74_{\pm 0.72}$	$\mathbf{64.26}_{\pm 0.62}$
WRN40-2	25%	$67.81_{\pm 0.27}$	$69.75_{\pm 0.26}$	$\mathbf{70.35}_{\pm 0.10}$
	50%	$60.51_{\pm 0.18}$	$62.58_{\pm 0.35}$	$\mathbf{64.71}_{\pm 0.55}$
WRN28-10	25%	$70.78_{\pm 0.20}$	$72.56_{\pm 0.18}$	$\mathbf{76.20}_{\pm 0.41}$
	50%	$61.94_{\pm 0.49}$	$64.12_{\pm 0.30}$	$\mathbf{70.80}_{\pm 0.28}$



(a) (b)

Figure 4: Test accuracy comparison of ResNet18 under different conditions and noise levels: (a) increasing layer width with 50% label noise and (b) varying CIFAR-10 training set size. SANER consistently outperforms other methods across all settings.

4.3 Ablation Studies

Various overfitting scenarios. We further evaluate SANER under two challenging settings known to induce overfitting: training wider models and using smaller datasets, following the protocol from (Nakkiran et al., 2020).

Increased model width. We observe that as model capacity increases, the performance of both SGD and SAM degrades, consistent with prior findings on overfitting noisy labels (Belkin et al., 2019; Nakkiran et al., 2020; Zhang et al., 2021a). In contrast, SANER benefits from the increased capacity, substantially widening its performance gap over SAM in highly overparameterized regimes, as shown in Figure 4a.

Limited dataset size. Overfitting also becomes more pronounced when the dataset size is small relative to model capacity. We evaluate SGD, SAM, and SANER on CIFAR-10 with training set sizes reduced to 10%, 25%, and 50%, while retaining the full test set. We omit CIFAR-100 due to its limited samples per class, which causes instability under further reduction. As

Table 4: Test accuracy comparison of different SAM-like optimizers with and without SANER integration on ResNet18 and CIFAR-10/CIFAR-100 (Symmetric noise). Bold values indicate the highest test accuracy. The teal values in parentheses (↑) show the improvement from integrating SANER.

Variants	Noise rate	CIFAR-10		CIFAR-100	
		Original	+SANER	Original	+SANER
ASAM	25%	$92.88_{\pm 0.13}$	$\mathbf{92.96}_{\pm 0.06}$ (↑0.08)	$70.67_{\pm 0.40}$	$\mathbf{72.44}_{\pm 0.10}$ (↑1.77)
	50%	$88.70_{\pm 0.18}$	$\mathbf{88.80}_{\pm 0.10}$ (↑0.10)	$63.04_{\pm 0.25}$	$\mathbf{66.62}_{\pm 0.13}$ (↑3.58)
GSAM	25%	$93.10_{\pm 0.12}$	$\mathbf{94.09}_{\pm 0.16}$ (↑0.99)	$69.65_{\pm 0.39}$	$\mathbf{72.97}_{\pm 0.27}$ (↑3.32)
	50%	$88.71_{\pm 0.15}$	$\mathbf{90.69}_{\pm 0.17}$ (↑1.98)	$61.25_{\pm 0.33}$	$\mathbf{66.19}_{\pm 0.15}$ (↑4.94)
FSAM	25%	$92.93_{\pm 0.08}$	$\mathbf{94.00}_{\pm 0.15}$ (↑1.07)	$69.49_{\pm 0.35}$	$\mathbf{72.94}_{\pm 0.58}$ (↑3.45)
	50%	$88.71_{\pm 0.13}$	$\mathbf{90.47}_{\pm 0.01}$ (↑1.76)	$61.24_{\pm 0.32}$	$\mathbf{66.25}_{\pm 0.15}$ (↑5.01)
VaSSO	25%	$92.35_{\pm 0.12}$	$\mathbf{93.31}_{\pm 0.32}$ (↑0.96)	$68.86_{\pm 0.18}$	$\mathbf{72.43}_{\pm 0.46}$ (↑3.57)
	50%	$87.93_{\pm 0.06}$	$\mathbf{89.66}_{\pm 0.57}$ (↑1.73)	$60.46_{\pm 0.05}$	$\mathbf{65.55}_{\pm 0.51}$ (↑5.09)

Table 5: Test accuracy of ResNet18 under 25% symmetric label noise across various SAM perturbation radii (ρ). Bold values highlight the highest accuracy.

Dataset	Opt	ρ			
		0.05	0.10	0.15	0.20
CIFAR-10	SAM	$92.15_{\pm 0.27}$	$93.05_{\pm 0.17}$	$93.78_{\pm 0.22}$	$93.90_{\pm 0.20}$
	SANER	$\mathbf{93.02}_{\pm 0.25}$	$\mathbf{94.08}_{\pm 0.11}$	$\mathbf{94.27}_{\pm \mathbf{0.18}}$	$\mathbf{94.30}_{\pm \mathbf{0.05}}$
CIFAR-100	SAM	$68.59_{\pm 0.27}$	$69.68_{\pm 0.07}$	$70.34_{\pm 0.26}$	$71.19_{\pm 0.29}$
	SANER	$\mathbf{72.10}_{\pm 0.40}$	$\mathbf{72.90}_{\pm 0.21}$	$\mathbf{73.21}_{\pm \mathbf{0.31}}$	$\mathbf{73.42}_{\pm 0.26}$

Table 6: Test accuracy comparison of SAM and SANER trained on clean datasets using ResNet18. Bold values highlight the highest test accuracy.

Dataset	SGD	SAM	SANER
CIFAR-10	$95.18_{\pm 0.09}$	$96.04_{\pm 0.04}$	$\mathbf{96.06}_{\pm 0.12}$
CIFAR-100	$78.06_{\pm 0.09}$	$79.19_{\pm 0.22}$	$\mathbf{79.63}_{\pm 0.36}$

shown in Figure 4b, SANER consistently outperforms both baselines, with a significant gain (7%) over SAM when trained on only 12,500 examples.

These experiments reveal a clear trend: the performance gap between SANER and SAM widens as conditions become more prone to overfitting. Whether through increased model capacity or reduced data, SANER's advantage grows, showcasing its effectiveness in preventing the memorization of noisy labels.

Integrating noise-robust training strategies with SAM and SANER. As an optimization technique, SANER can, in principle, be integrated into a wide range of label-noise learning algorithms. We evaluate the compatibility of SANER with existing noise-robust learning strategies, including hard bootstrapping (Reed et al., 2014) (BS) and CoTeaching (Han et al., 2018).

As shown in Table 7, integrating bootstrapping noticeably improves the performance of standard SAM, while its benefit for SANER is comparatively smaller. This suggests that SANER already provides strong robustness to label noise, leaving limited room for further gains from bootstrapping. For CoTeaching, the improvements are less consistent, particularly for SAM under higher noise levels. Nevertheless, SANER consistently achieves stronger performance than SAM across nearly all settings, both with and without additional noise-robust strategies, demonstrating that SANER serves as a reliable and complementary optimization framework for noisy-label training.

Impact of SAM's ρ on SANER performance. The perturbation radius ρ in SAM is a key hyperparameter that directly affects generalization performance (Foret et al., 2021). To understand how SANER interacts with this factor, we evaluate its performance across a range of ρ values (from 0.05 to 0.20), as recommended by Foret et al. (2021).

As shown in Table 5, SANER consistently outperforms SAM for all tested radii. While increasing ρ generally improves both methods, the gains introduced by SANER remain complementary and substantial. This effect is particularly evident on CIFAR-100, where the worst performance of SANER (72.57%) still surpasses the best result achieved by SAM (70.86%). Overall, these findings demonstrate that SANER provides a fundamental robustness improvement beyond what can be achieved by tuning ρ alone.

Noise-free scenarios. In addition to noisy settings, we evaluate SANER under noise-free conditions to examine its behavior when label noise is absent. Specifically, we test whether the proposed mechanism remains well-aligned with standard learning without introducing performance degradation.

As shown in Table 6, SANER performs on par with SAM in clean settings, without noticeable improvement. These results indicate that SANER preserves the baseline performance of SAM while avoiding any negative impact in the absence of noise. Overall, this demonstrates that SANER is a safe and broadly ap-

Table 7: Test accuracy comparison of SAM and SANER integrated with Bootstrap (BS) and CoTeaching (CT) on CIFAR-10/100 with symmetric label noise using ResNet18. Bold values indicate the best performance under each setting and γ denotes the noise rate.

Dataset	γ	Baseline		Bootstrap (BS)		CoTeaching (CT)	
		SAM	SANER	SAM+BS	SANER+BS	SAM+CT	SANER+CT
CIFAR-10	25%	$93.05_{\pm 0.17}$	$94.08_{\pm 0.11}$	$93.35_{\pm 0.07}$	$\mathbf{94.49}_{\pm 0.09}$	$92.86_{\pm 0.17}$	$93.41_{\pm 0.27}$
	50%	$88.82_{\pm 0.08}$	$90.60_{\pm 0.36}$	$89.57_{\pm 0.49}$	$\mathbf{90.98}_{\pm 0.49}$	$86.89_{\pm 0.16}$	$88.18_{\pm 0.23}$
CIFAR-100	25%	$69.68_{\pm 0.07}$	$72.90_{\pm 0.21}$	$70.81_{\pm 0.18}$	$73.27_{\pm 0.29}$	$72.46_{\pm 0.31}$	$\mathbf{73.89}_{\pm 0.38}$
	50%	$61.17_{\pm 0.14}$	$66.34_{\pm 0.11}$	$64.08_{\pm 0.42}$	$\mathbf{66.81}_{\pm 0.11}$	$61.35_{\pm 0.45}$	$64.26_{\pm 0.27}$

plicable optimization strategy across both noisy and noise-free scenarios.

5 CONCLUSION

In this work, we investigated the mechanisms behind SAM's robustness to label noise, identifying that element-wise gradient down-weighting is a key factor in mitigating the memorization of noisy samples during transitional phase. This down-weighting is supported by an up-weighting of the clean samples' gradients. Building on this insight, we proposed SANER, a simple modification that enhances this mechanism at no additional computational cost. Our experiments demonstrated that SANER consistently outperforms standard SAM across various datasets, noise types, and challenging overfitting scenarios. Moreover, our modification can be integrated into SAM-variants and shows consistent improvement. While our current theory is based on linear models, these strong empirical results suggest promising directions for future work, such as extending the analysis to deep models and exploring adaptive re-weighting mechanisms.

References

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

Christina Baek, J Zico Kolter, and Aditi Raghunathan. Why is SAM robust to label noise? In *International Conference on Learning Representations*, 2024.

Yingbin Bai, Erkun Yang, Bo Han, Yanhua Yang, Jiatong Li, Yinian Mao, Gang Niu, and Tongliang Liu. Understanding and improving early stopping for learning with noisy labels. *Advances in Neural Information Processing Systems*, 2021.

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

Kayhan Behdin and Rahul Mazumder. On statistical properties of sharpness-aware minimization: Provable guarantees. *arXiv preprint arXiv:2302.11836*, 2023.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 2019. doi: 10.1073/pnas.1903070116.

Zixiang Chen, Junkai Zhang, Yiwen Kou, Xiangning Chen, Cho-Jui Hsieh, and Quanquan Gu. Why does sharpness-aware minimization generalize better than sgd? *Advances in neural information processing systems*, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 2009.

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

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

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

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

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven

curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*, 2018.

Weisen Jiang, Hansi Yang, Yu Zhang, and James Kwok. An adaptive policy to employ sharpness-aware minimization. In *International Conference on Learning Representations*, 2023.

Pham Duy Khanh, Boris S Mordukhovich, and Dat Ba Tran. Inexact reduced gradient methods in smooth nonconvex optimization. *Journal of Optimization Theory and Applications, doi.org/10.1007/s10957-023-02319-9*, 2023.

Pham Duy Khanh, Hoang-Chau Luong, Boris S Mordukhovich, and Dat Ba Tran. Fundamental convergence analysis of sharpness-aware minimization. In *Advances in Neural Information Processing Systems*, 2024a. URL `https://arxiv.org/pdf/2401.08060`.

Pham Duy Khanh, Boris S Mordukhovich, and Dat Ba Tran. A new inexact gradient descent method with applications to nonsmooth convex optimization. *Optimization Methods and Software https://arxiv.org/abs/2303.08785*, 2024b.

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

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009.

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

Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. In *CS 231N: Convolutional Neural Networks for Visual Recognition, Stanford*, 2015.

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

Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020.

Shikun Li, Xiaobo Xia, Shiming Ge, and Tongliang Liu. Selective-supervised contrastive learning with noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

Tao Li, Pan Zhou, Zhengbao He, Xinwen Cheng, and Xiaolin Huang. Friendly sharpness-aware minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020.

Sheng Liu, Zhihui Zhu, Qing Qu, and Chong You. Robust training under label noise by overparameterization. In *International Conference on Machine Learning*, 2022.

Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 2015.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, 2020.

Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International conference on machine learning*, 2020.

Aditya Krishna Menon, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Can gradient clipping mitigate label noise? In *International Conference on Learning Representations*, 2020.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020.

Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. Self: Learning to filter noisy labels with self-ensembling. In *International Conference on Learning Representations*, 2020.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, 2018.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015.

Sungbin Shin, Dongyeop Lee, Maksym Andriushchenko, and Namhoon Lee. The effects of overparameterization on sharpness-aware minimization: An empirical and theoretical analysis. *ICML Workshop on High-dimensional Learning Dynamics*, 2023.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 2019.

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

Hongxin Wei, Lei Feng, Rundong Wang, and Bo An. Metainfonet: Learning task-guided information for sample reweighting. *arXiv preprint arXiv:2012.05273*, 2020.

Hongxin Wei, Huiping Zhuang, Renchunzi Xie, Lei Feng, Gang Niu, Bo An, and Yixuan Li. Mitigating memorization of noisy labels by clipping the model prediction. In *International Conference on Machine Learning*, 2023.

Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations*, 2022.

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. How sharpness-aware minimization minimizes sharpness? In *The Eleventh International Conference on Learning Representations*, 2023.

Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems*, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks, 2017. URL `https://arxiv.org/abs/1605.07146`.

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

Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. Learning with feature-dependent label noise: A progressive approach. In *ICLR*, 2021b.

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 2018.

Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha C Dvornek, sekhar tatikonda, James s Duncan, and Ting Liu. Surrogate gap minimization improves sharpness-aware training. In *International Conference on Learning Representations*, 2022.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]. See Section 3.2.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]. See Section 4.1.
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]. See Section 3.2.
   (b) Complete proofs of all theoretical results. [Yes]. See Appendix A.1.
   (c) Clear explanations of any assumptions. [Yes]. See Section 3.2.
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]. See Section 4.2.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]. See Section 4.2.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]. See Section B.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials

A MISSING PROOFS

A.1 Proof of Lemma 3.1

Proof. Let $h : \mathbb{R} \to \mathbb{R}$ be defined as $h(z) = \sigma(z+C) - \sigma(z)$ for all $z \in \mathbb{R}$. It follows from the derivative of sigmoid function that

$$\begin{aligned} h'(z) &= \sigma(z+C)\big(1-\sigma(z+C)\big) - \sigma(z)\big(1-\sigma(z)\big) \\ &= \big[\sigma(z+C) - \sigma(z)\big]\Big[1 - \sigma(z+C) - \sigma(z)\Big]. \end{aligned}$$

As $\sigma(z+C) - \sigma(z) > 0$ for all $z \in \mathbb{R}$ due to the increasing property of σ and $C > 0$, the inequality $h'(z) > 0$ is equivalent to $\sigma(z+C) + \sigma(z) < 1$, which means

$$\frac{1}{1+e^{-(z+C)}} + \frac{1}{1+e^{-z}} < 1.$$

It follows that $e^{-2z-C} > 1$, or equivalently $z < -\frac{C}{2}$, which means the function h is strictly increasing on $(-\infty, -\frac{C}{2})$. Using the property $\sigma(z) = 1 - \sigma(-z)$ for all $z \in \mathbb{R}$ of the sigmoid function, we deduce that

$$\begin{aligned} h\big(z\big) &= \sigma\big(z+C\big) - \sigma\big(z\big) \\ &= \sigma\big(-z\big) - \sigma\big(-z-C\big) = h\big(-z-C\big), \end{aligned} \tag{14}$$

for all $z \in \mathbb{R}$. Since $z_1, z_2 < 0$, and $z_1 > z_2 - C$, if $z_1 < \frac{-C}{2}$, then $h(z_1) > h(-z_2 - C) = h(z_2)$ due to the increasing property of h on $(-\infty, \frac{-C}{2})$. Otherwise, $-\frac{C}{2} \le z_1 < 0$ implies that $-\frac{C}{2} \ge -z_1 - C > -C$. Combining this with (14), the increasing property of h on $(-\infty, \frac{-C}{2})$ and $z_2 < 0$, we have

$$h(z_1) = h(-z_1 - C) > h(-C) > h(z_2 - C).$$

Using this with $h(z) = \sigma(z+C) - \sigma(z)$, we have $\sigma(z_1 + C) - \sigma(z_1) > \sigma(z_2) - \sigma(z_2 - C)$, or equivalently

$$1 - \sigma(z_1) - \sigma(z_2) > 1 - \sigma(z_1 + C) - \sigma(z_2 - C). \tag{15}$$

Using the property $\sigma(z) < 0.5$ for all $z < 0$ and $z_1, z_2 < 0$, we deduce that $1 - \sigma(z_1) - \sigma(z_2) > 0$. Combining this with (15), we have

$$\frac{1 - \sigma(z_1 + C) - \sigma(z_2 - C)}{1 - \sigma(z_1) - \sigma(z_2)} < 1.$$

We also aim to show that $1 - \sigma(z_1 + C) - \sigma(z_2 - C) = \sigma(-z_1 - C) - \sigma(z_2 - C) > 0$, which is equivalent to

$$\frac{1}{1+\exp(z_1 + C)} - \frac{1}{1+\exp(-z_2 + C)} > 0.$$

This inequality further simplifies to $\exp(-z_2) - \exp(z_1) > 0$, which holds because $-z_2 > 0 > z_1$. Combining this with $1 - \sigma(z_1) - \sigma(z_2) > 0$, we have

$$0 < \frac{1 - \sigma(z_1 + C) - \sigma(z_2 - C)}{1 - \sigma(z_1) - \sigma(z_2)},$$

which verifies the lemma. □

B IMPLEMENTATION DETAILS

In this paper, we use four types of noise as follows:

1. Symmetric noise: Each label is flipped to any other class with equal probability noise rate γ.
2. Asymmetric noise: Labels are flipped to similar, but not identical classes (Zhang and Sabuncu, 2018). For CIFAR-10, we generate asymmetric noisy labels by mapping specific classes to their most similar counterparts: TRUCK to AUTOMOBILE, BIRD to AIRPLANE, DEER to HORSE, CAT to DOG with noise rate γ, and leaving other labels unchanged. For CIFAR-100, each class is shifted circularly to the next class with noise rate γ.
3. Instance-dependent noise: The mislabeling probability of each instance depends on its input features. In our experiments, we use instance-dependent noise from PDN (Xia et al., 2020) with noisy rate γ, where the noise is synthesized based on DNN prediction errors.
4. Real-world noise: Labels are taken from the mislabeling of real-world human annotations. For CIFAR datasets, we use the "Worst" label set from CIFAR-10N and the "Fine" label set from CIFAR-100N (Wei et al., 2022).

All experiments are mainly conducted on an Ubuntu Linux machine equipped with an NVIDIA RTX 3090 GPU with 24GB of memory.

C ADDITIONAL EXPERIMENTS

C.1 Integration with SAM Variants

Experimental setup. To evaluate the effect of SANER on SAM-based optimizers, we conducted experiments on CIFAR-10 and CIFAR-100 using ResNet18. The SAM variants used include ASAM (Kwon et al., 2021), GSAM (Zhuang et al., 2022), FSAM (Li et al., 2024), and VaSSO (Li and Giannakis, 2024), tested both with and without SANER integration. The models were trained with label noise levels of 25% and 50%, and the SANER hyperparameter $\alpha = 0.5$ for all experiments as we setup when comparing with SAM. All other training configurations were kept consistent for fair comparison between methods.

Modification of SAM-based optimizers. SANER was integrated into the SAM-variants by modifying the update rules. Specifically, we compute a mask vector $\boldsymbol{m}$ at index i as

$$m_i = \begin{cases} 1, & \text{if } 0 < r_i < 1, \\ 0, & \text{otherwise,} \end{cases} \quad i = 1, \ldots, d, \tag{16}$$

then replace $\boldsymbol{r} = \boldsymbol{g}^{\text{SAM}^*}/\boldsymbol{g}^{\text{SGD}}$ and $\boldsymbol{g}^{\text{SAM}^*}$ as follows:

$$\boldsymbol{g}^{\text{SANER}^*} = \big(\mathbf{1} - \alpha \boldsymbol{m}\big) \odot \boldsymbol{g}^{\text{SAM}^*} \tag{17}$$

where $\boldsymbol{g}^{\text{SAM}^*}$ refers to the gradient of the specific SAM variant and $\boldsymbol{g}^{\text{SANER}^*}$ denotes the modified gradient under SANER integration.

Noisy train accuracy. As illustrated in Figure 5, the integration of SANER into SAM variants significantly reduces the number of noisy examples that are memorized during training compared to their original variants. This is particularly evident in high-noise scenarios such as 50%, where the noisy fitting curve rises more gradually in SANER-integrated models compared to their original counterparts. This indicates that SANER helps slow down the memorization of noisy labels, allowing the models to focus more on clean data, which leads to better generalization.

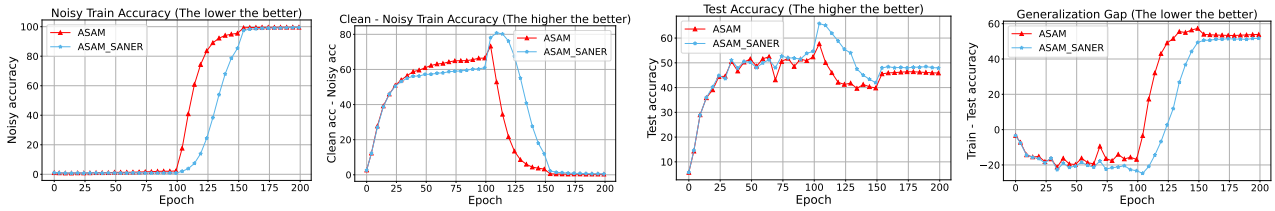
C.2 Effect of Increasing ResNet18 Width

To demonstrate the effectiveness of our method, we conduct experiments in overfitting-prone scenarios by increasing model parameters, as detailed in main paper. In this section, we visualize the training process under

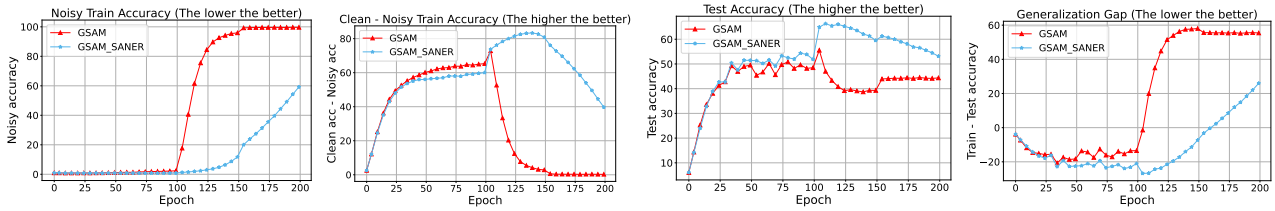
overparameterization by increasing the width of ResNet18 to provide further insights into the fitting rates of SGD, SAM, and SANER. As shown in Figure 6, increasing model width enhances overfitting, causing SAM to match the noisy fitting rate of SGD. In contrast, SANER maintains a slower noisy fitting rate while preserving the clean fitting rate, allowing the model to better leverage overparameterization and achieve higher test accuracy.

C.3 Various Architectures

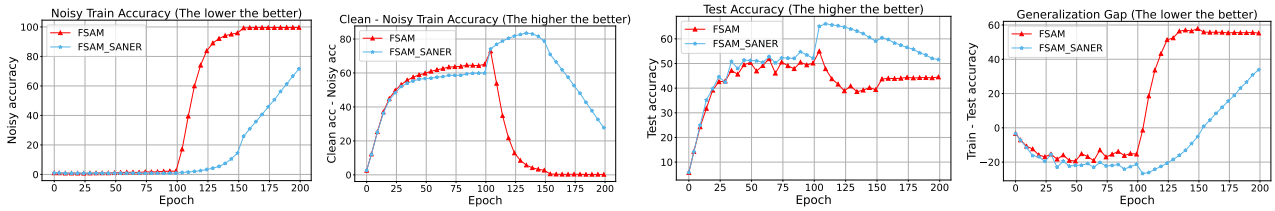
To evaluate SANER's robustness across different neural network architectures, we conducted experiments using ResNet34, DenseNet121, and WideResNet28-10 on CIFAR datasets with 25% and 50% label noise. We analyze the impact of SANER on the training process, specifically its ability to regulate noisy fitting rates, as shown in Figure 7. SANER consistently achieves better control over noisy fitting, thereby reducing overfitting and enhancing generalization performance. These results demonstrate SANER's effectiveness in handling noisy data across diverse architectures, yielding significant improvements over both SGD and SAM.



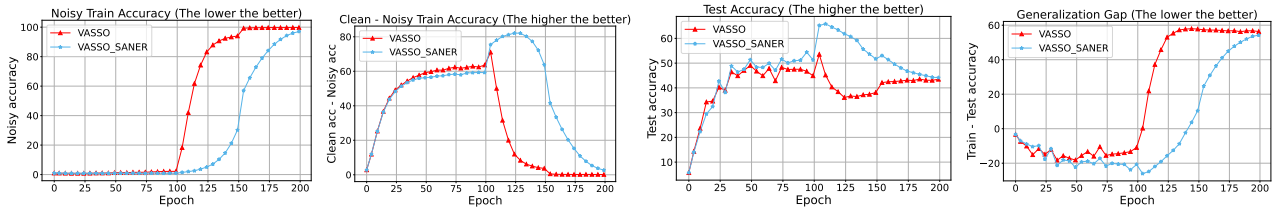
(a) ASAM



(b) GSAM

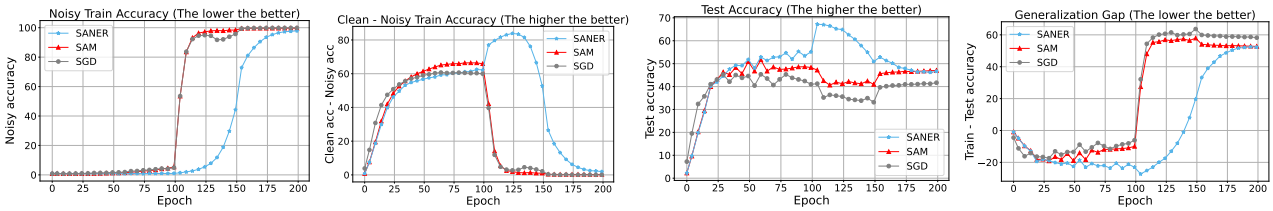


(c) FSAM

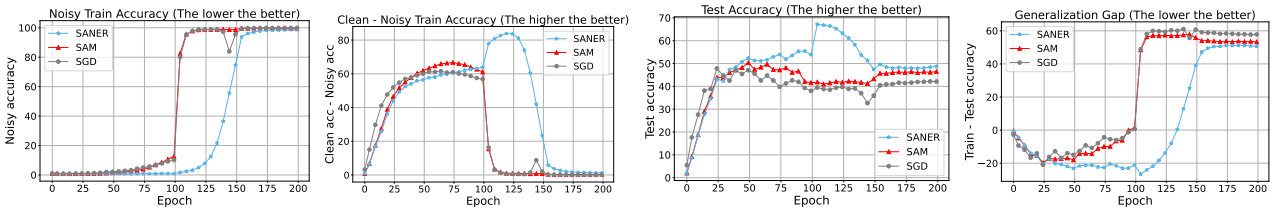


(d) VaSSO

Figure 5: Performance comparison of ASAM, GSAM, FSAM, and VaSSO (with and without SANER) trained on ResNet18 with CIFAR-100 under 25% label noise. The columns represent (from left to right): noisy training accuracy, gap between clean and noisy accuracy, test accuracy, and generalization gap. Overall, integrating SANER with these SAM variants slows the noisy fitting rate while preserving the clean fitting rate.

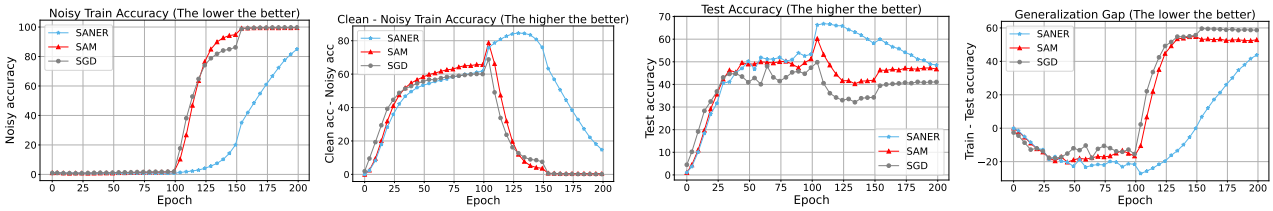


(a) ResNet18 (Width = 1.5)

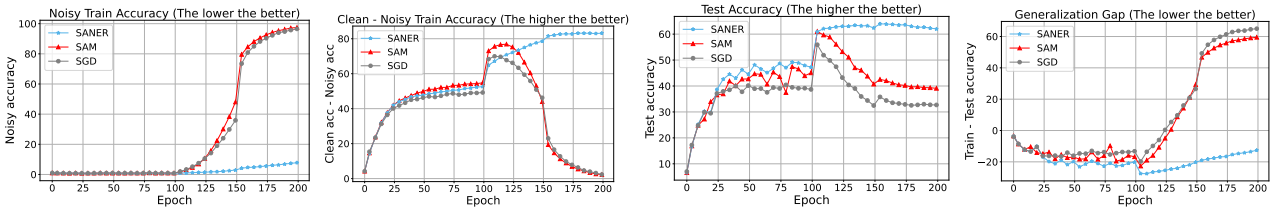


(b) ResNet18 (Width = 2)

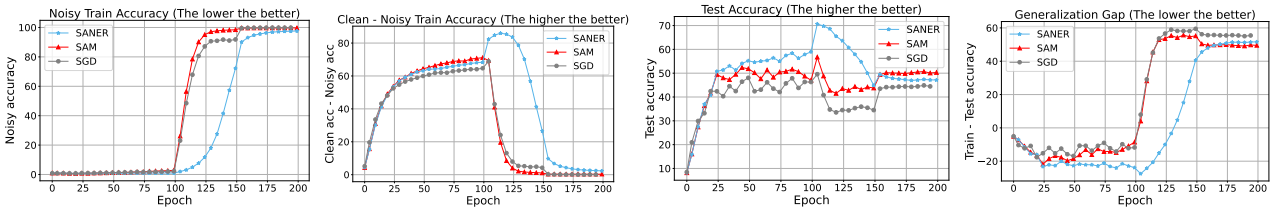
Figure 6: Performance comparison of SAM, SGD, and SANER (ours) when **increasing width** of ResNet18 trained on CIFAR-100 under 50% label noise. The columns represent (from left to right): noisy training accuracy, gap between clean and noisy accuracy, test accuracy, and generalization gap. The noisy fitting rate of SAM reaches that of SGD, whereas SANER keeps it low for a longer duration, resulting in better performance.



(a) ResNet34



(b) DenseNet121



(c) WideResNet28-10

Figure 7: Performance comparison of SAM, SGD, and SANER (ours) across **different models** trained on CIFAR-100 under 50% label noise. The columns represent (from left to right): noisy training accuracy, gap between clean and noisy accuracy, test accuracy, and generalization gap. SANER outperforms SAM in both noisy accuracy and the clean-noisy accuracy gap, demonstrating better generalization through higher test accuracy.