CASAM: Consistency-Anchored Sharpness-Aware Minimization for Improved Model Generalization

Chengli Tan[1,*] cltan023@nwpu.edu.cn
Tianyu Wang[2] wty111@stu.xjtu.edu.cn
Junmin Liu[2,3,*] junminliu@mail.xjtu.edu.cn

Yong Xu[1]
hsux3@nwpu.edu.cn
[1]School of Mathematics and Statistics, Northwestern Polytechnical University
[2]School of Mathematics and Statistics, Xi'an Jiaotong University
[3]SGIT AI Lab, State Grid Corporation of China

Abstract

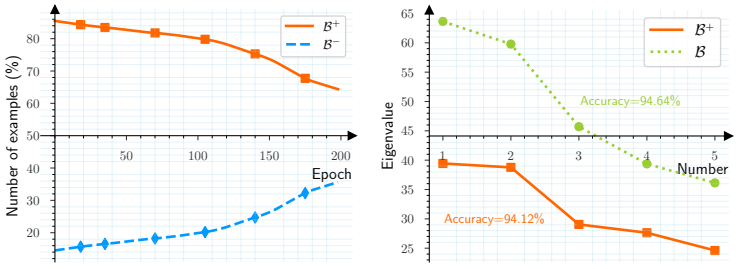
Many studies argue that the success of sharpness-aware minimization (SAM) is due to the implicit regularization of the gradient norm or the eigenvalues of the Hessian. However, they cannot fully explain why the flatness-based measures do not always correlate well with generalization. For instance, we can easily construct a counterexample by only optimizing with hard examples. In this paper, we propose to resolve such a contradiction from the perspective of anti-overfitting. First, we show that the sharpness at the mini-batch level approximately equals the sum of the gradient norm and the Kullback–Leibler (KL) divergence between the predictive distributions evaluated at the current point and its adversary. Second, we show that the *gradient consistency*, which approximately quantifies the inner product between the mini-batch gradient and the per-example gradient, decreases consistently during training. This result explains why SAM is less likely to suffer from overfitting. At last, building on these insights, we further introduce CASAM, an algorithm that reweights each example according to gradient consistency to enhance the generalization performance. Extensive experiments on popular benchmarks such as ImageNet-1K and Clothing1M corroborate its efficacy.

1 Introduction

Stochastic algorithms such as stochastic gradient descent (SGD) and Adam (Kingma & Ba, 2015) are the foundational tools to train modern neural networks, which often consist of millions (even billions) of learnable parameters. However, their performance on generalization is dramatically different. For example, on image classification tasks, Wilson et al. (2017) and Zhou et al. (2020) argue that SGD generalizes better than Adam. This finding is actually connected with the concept of flat minima (Hinton & van Camp, 1993; Hochreiter & Schmidhuber, 1997). While there exist some disputes (Dinh et al., 2017; Wen et al., 2023), it is widely appreciated that flat minima generalize well on unseen data (Dziugaite et al., 2020; Kaddour et al., 2022). Following this line, many popular optimizers (Chaudhari et al., 2019; Foret et al., 2021; Bisla et al., 2022) are proposed to bias the optimization towards the flat regions of the loss landscape. Among them, SAM (Foret et al., 2021) exhibits surprisingly effectiveness in improving generalization (Chen et al., 2022; Bahri et al., 2022; Lyu et al., 2022; Huang et al., 2023; Zou et al., 2024; Deng et al., 2025) and calibration (Zheng et al., 2021; Möllenhoff & Khan, 2023; Tan et al., 2026).

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Proportion (b) Eigenvalue

Figure 1: Easy examples from $\mathcal{B}^-$ cannot be simply eliminated when trained with SAM. The adopted neural network is ResNet-56 and trained on CIFAR-10 for 200 epochs with a perturbation radius $\rho = 0.05$. (a) Evolution of the size of $\mathcal{B}^+$ and $\mathcal{B}^-$ during training. Namely, how many examples of the mini-batch belongs to $\mathcal{B}^+$ and $\mathcal{B}^-$. (b) Top five eigenvalues of the Hessian for SAM using the mini-batch $\mathcal{B}$ and SAM only using the hard examples from $\mathcal{B}^+$. The average accuracy on the test set is also reported. It is worth noting that the performance deterioration is not due to the reduced mini-batch size ($|\mathcal{B}^+| < |\mathcal{B}|$). As many studies (Keskar et al., 2017; He et al., 2019) suggest, smaller mini-batches introduce more gradient noise and tend to yield better generalization.

Assume that $\boldsymbol{\omega} \in \mathbb{R}^d$ is the collection of learnable parameters and $\mathcal{B}$ is a random mini-batch from the training set $\mathcal{S}$. Define the empirical loss as

$$F_{\mathcal{B}}(\boldsymbol{\omega}) = \frac{1}{|\mathcal{B}|} \sum_{z \in \mathcal{B}} f_{\boldsymbol{\omega}}(z),$$

where $f_{\boldsymbol{\omega}}(z) : \mathbb{R}^d \times Z \mapsto \mathbb{R}_+$ is the loss incurred by a single example $z \in Z$. Unlike SGD, the objective function of SAM is formulated as

$$\min_{\boldsymbol{\omega} \in \mathbb{R}^d} \max_{\|\boldsymbol{\varepsilon}\|_2 \leq \rho} F_{\mathcal{B}}(\boldsymbol{\omega} + \boldsymbol{\varepsilon}), \tag{1}$$

where $\boldsymbol{\varepsilon} \in \mathbb{R}^d$ is a vector whose norm is bounded by the perturbation radius ρ. Essentially speaking, SAM is minimizing the worst-case loss within a neighborhood of $\boldsymbol{\omega}$. Solving the minimax problem (1) is not an easy task. Fortunately, with a first-order Taylor expansion, its solution can be recursively approximated by

$$\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k - \eta \nabla F_{\mathcal{B}_k}(\tilde{\boldsymbol{\omega}}_k), \quad \tilde{\boldsymbol{\omega}}_k = \boldsymbol{\omega}_k + \rho \frac{\nabla F_{\mathcal{B}_k}(\boldsymbol{\omega}_k)}{\|\nabla F_{\mathcal{B}_k}(\boldsymbol{\omega}_k)\|_2},$$

where η denotes the learning rate and $\tilde{\boldsymbol{\omega}}_k$ is referred to as the adversary of $\boldsymbol{\omega}_k$. While the implementation (first gradient ascent and then gradient descent) is simple, the solution generalizes substantially better than SGD (Foret et al., 2021). Some studies argue that this superiority comes from its implicit regularization. For example, Zhao et al. (2022a) infers SAM as a special case of gradient norm penalty, and Compagnoni et al. (2023) shows that SAM is implicitly minimizing a regularized loss with curvature-dependent noise. Meanwhile, Wen et al. (2022) and Luo et al. (2024) also demonstrate that the inductive bias of SAM is to decrease the largest eigenvalue of the Hessian. However, we argue that these insights cannot fully explain why SAM generalizes better than SGD. Below, we can easily establish a counterexample whose largest eigenvalue of the Hessian is smaller yet generalizes worse than SAM.

Due to gradient ascent, SAM splits the mini-batch $\mathcal{B}$ into two disjoint subsets at each step:

$$\mathcal{B}^+ = \{z \in \mathcal{B} | f_{\tilde{\boldsymbol{\omega}}}(z) - f_{\boldsymbol{\omega}}(z) \geq 0\},$$

and

$$\mathcal{B}^- = \{z \in \mathcal{B} | f_{\tilde{\boldsymbol{\omega}}}(z) - f_{\boldsymbol{\omega}}(z) < 0\},$$

which, for convenience, we refer to examples from $\mathcal{B}^+$ to be *hard* and from $\mathcal{B}^-$ to be *easy*. On one hand, the easy examples from $\mathcal{B}^-$ seem to be useless: minimizing $f_{\tilde{\boldsymbol{\omega}}}(z)$ neither decreases $f_{\boldsymbol{\omega}}(z)$ nor reduces the sharpness (because the component is negative). On the other hand, Figure 1a shows that

the proportion of easy examples keeps increasing during training. It is thus alluring to estimate the mini-batch gradient of the descent step only using examples from $\mathcal{B}^+$ to reduce the computational overhead (Du et al., 2022a; Xie et al., 2024a). Intuitively, this will drive the optimization towards flatter regions of the loss landscape. As shown in Figure 1b, we observe that models trained with $\mathcal{B}^+$ indeed decrease the largest eigenvalue of the Hessian. But, contrary to what is expected, they generalize much worse than the standard SAM. Therefore, the implicit regularization does not suffice to explain the superiority of SAM, and the goal of this paper is to fill this gap.

To this end, we propose to analyze SAM from the perspective of anti-overfitting (Chen et al., 2023). Different from previous studies that focus on the mini-batch gradient, we are more interested in isolating how each example affects the optimization direction. Generally, it is hard to know to what extent an individual example contributes to the mini-batch gradient. For example, when training neural networks with SGD or Adam, all that we can know is the mini-batch gradient because it is infeasible to backpropagate the per-example gradient. In contrast, thanks to the two consecutive steps of gradient backpropagation, SAM enables us to acquire an approximation of the inner product between the per-example gradient and the mini-batch gradient. It then allows us to adjust the importance of each example and bias the optimization towards the regions of interest. In brief, our contributions are as follows:

- We demonstrate that at the mini-batch level, the sharpness can be approximately decomposed into the sum of the gradient norm and the KL-divergence between the two predictive distributions evaluated at the current point and its adversary. Different from previous theoretical analysis (Zhao et al., 2022a; Compagnoni et al., 2023), our result quantitatively reveals that SAM not only penalizes the gradient norm but also penalizes the Hessian, which is implicitly determined by the KL-divergence.
- We show that the *gradient consistency*, which approximately quantifies the inner product between the per-example gradient and the mini-batch gradient, decreases during training. This finding justifies the superiority of SAM to counterattack overfitting.
- We propose a novel optimizer (namely, CASAM) to further improve generalization performance across various tasks. By reweighting each example according to its importance, CASAM provides us with more flexibility to handle the issue of overfitting. We also perform extensive experiments to validate its efficacy and find that CASAM performs surprisingly well on the noisy datasets.

The remainder of the paper is organized as follows. We first review the related works in Section 2. After presenting the theoretical analysis in Section 3, we further provide the experimental results in Section 4 and conclude in Section 5.

2 Related Works

Ever since SAM (Foret et al., 2021; Zheng et al., 2021) was proposed, it has become a powerful tool for discovering flat minima, and numerous variants have been introduced to boost its capability and efficiency.

Generalization improvement. While SAM attempts to seek flat minima by minimizing the perturbed loss, one caveat is that both sharp and flat minima can attain a low perturbed loss. To address this issue, GSAM (Zhuang et al., 2022) instead minimizes the so-called *surrogate gap* to seek a region with small loss and low sharpness. Building on the study of SAM's dynamics through higher-order stochastic differential equations, Luo et al. (2024) proposes Eigen-SAM to align the perturbation vector with the leading eigenvector of the Hessian. Some works, such as ASAM (Kwon et al., 2021) and Fisher SAM (Kim et al., 2022), also argue that the sharpness is not well defined and propose to reinforce its connection with generalization by adjusting the neighborhood ball properly. Similar to Bisla et al. (2022), Random SAM (Liu et al., 2022b) advocates for generating more accurate inner-maximization estimates by convoluting the loss landscape with Gaussian noise before the ascent step. In contrast, Yu et al. (2024) integrates SAM with Lookahead (Zhang et al., 2019) to smooth the optimization trajectory. In addition, VASSO (Li & Giannakis, 2023) highlights that the adversary perturbation of a particular mini-batch can become a friend on other examples and proposes a technique of variance suppression to alleviate this issue.

Training efficiency. One evident drawback of SAM is that it requires an extra gradient backpropagation per step, which, as a result, almost doubles the computational cost of SGD. As a remark, it is noteworthy that SAM with more ascent steps unfortunately does not promise to yield better generalization (Kim et al., 2023). Therefore, a lot of work has been devoted to boosting the efficiency of SAM. Early attempts include LookSAM (Liu et al., 2022a) and RST (Zhao et al., 2022b) that switch SAM and SGD periodically. A similar idea has also been explored by SALA (Tan et al., 2024) and AE-SAM (Jiang et al., 2023) that selectively minimize the sharpness only under certain conditions. Alternatively, some works, such as ESAM (Du et al., 2022a) and Sparse SAM (Mi et al., 2022), propose to perturb only a subset of the network parameters. Particularly, Mueller et al. (2023) demonstrates that perturbing only the affine parameters of the normalization layer can recover most of SAM's benefits. Furthermore, SAF (Du et al., 2022b) and S^2-SAM (Ji et al., 2024) both claim that their performance is on par with SAM without extra cost by leveraging the historical information. Finally, SAMPa (Xie et al., 2024b) parallelizes SAM's two gradient computations, which, under ideal communication, delivers up to a two-fold speedup. And RW-SAM (Luo et al., 2026) attempts to mimic the generalization benefits of m-sharpness while being parallelizable.

Theoretical explanations. The success of SAM has attracted a lot of attention to study the underlying mechanism. For example, Zhao et al. (2022a) argues that SAM actually penalizes the gradient norm, and Wen et al. (2022); Compagnoni et al. (2023) reveal that it implicitly regularizes the Hessian. Moreover, Dai et al. (2023) also points out that the normalization layer is crucial to stabilize training and permits drift along a continuum of minima. The implicit bias of SAM is also investigated by Andriushchenko & Flammarion (2022) and Andriushchenko et al. (2023), who show that SAM-trained networks tend to develop low-rank feature representation. From the perspective of learning dynamics, Bartlett et al. (2023) demonstrates that SAM naturally steers the training process towards the so-called *edge of stability*, a phenomenon associated with finding flat minima. In contrast, Zhou et al. (2025) focuses on the learning dynamics after applying SAM late in training. For a two-layer neural network, Chen et al. (2023) shows that SAM can prevent early memorization of label noise and thus generalizes better than SGD even in highly nonconvex regimes. Agiollo et al. (2024) shows how to use the solutions of SAM and SGD to measure to which degree each example is memorized, therefore enabling dataset pruning. Kim et al. (2026) demonstrates that SAM increases model vulnerability to membership inference attacks. At last, the convergence property of SAM is also extensively studied (Andriushchenko & Flammarion, 2022; Si & Yun, 2023; Khanh et al., 2024; Oikonomou & Loizou, 2025). Different from them, our work provides an explanation of why SAM is robust to overfitting and how it can be used to guide the design of new optimizers.

3 Methodology

In this section, we will elaborate on the details of why SAM is intrinsically immune to overfitting and how we can leverage this nature to further improve generalization performance. Before that, we now introduce some notations as follows.

Notations. Assume that the training set $\mathcal{S}$ is composed of n *i.i.d.* examples $\{z_i = (x_i, y_i)\}_{i=1}^n$ where $x_i \in X \subset \mathbb{R}^p$ and $y_i \in Y = \{0, 1, \cdots, K-1\}$. By default, the data distribution $Z = X \times Y$ is unknown. At each iteration, we randomly sample a mini-batch of size m from S and denote it by $\mathcal{B}$. It should be noted that the same $\mathcal{B}$ is used in both the ascent and descent steps of SAM. Given a neural network parameterized by $\boldsymbol{\omega} \in \mathbb{R}^d$, it actually defines a mapping $\Pi_{\boldsymbol{\omega}}$ from the input space X to the $(K-1)$-dimensional simplex $\Delta = \{\mathbf{p} \in [0,1]^K | \sum_{y=1}^K \mathbf{p}_y = 1\}$ due to the softmax operation. In classification task, $\hat{y} = \arg\max_{1 \le y \le K} \mathbf{p}_y$ corresponds to the prediction. For clarity, we focus on the widely used cross-entropy loss[2], which is given by $f_{\boldsymbol{\omega}}(z) = -\log\left[\Pi_{\boldsymbol{\omega}}(z)\right]_y = -\log \mathbf{p}_y$ for an arbitrary example $z = (x, y)$. Similarly, $f_{\tilde{\boldsymbol{\omega}}}(z) = -\log \tilde{\mathbf{p}}_y$ defines the loss evaluated at the adversary $\tilde{\boldsymbol{\omega}}$ and $\tilde{\mathbf{p}}_y$ denotes the probability of the correct class.

3.1 Decomposition of Sharpness

We begin with the concept of sharpness that lies at the core of SAM. Due to space limitations, all proofs are deferred to Appendix F.

[2]Besides cross-entropy loss, our analysis actually holds for the loss functions from the regular exponential family (see Definition F.1), including Poisson loss for count data and mean squared error (MSE) loss for regression.

Definition 3.1. *The* sharpness *of $\boldsymbol{\omega}$ within a neighborhood is defined as*

$$\zeta(\boldsymbol{\omega}, \rho) = \max_{\|\boldsymbol{\varepsilon}\|_2 \leq \rho} F_{\mathcal{B}}(\boldsymbol{\omega} + \boldsymbol{\varepsilon}) - F_{\mathcal{B}}(\boldsymbol{\omega}).$$

According to the PAC-Bayes analysis (for example, see Theorem 2 of Foret et al. (2021) and Theorem 3.2 of Kim et al. (2022)), the generalization bound is directly controlled by the sharpness. However, it is infeasible to acquire the precise value of sharpness. In practice, we often adopt its empirical counterpart (Foret et al., 2021; Bisla et al., 2022), which is given by

$$\widehat{\zeta}(\boldsymbol{\omega}, \rho) = F_{\mathcal{B}}(\tilde{\boldsymbol{\omega}}) - F_{\mathcal{B}}(\boldsymbol{\omega}), \text{and } \tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \rho \frac{\nabla F_{\mathcal{B}}(\boldsymbol{\omega})}{\|\nabla F_{\mathcal{B}}(\boldsymbol{\omega})\|_2}.$$

In the following, we show that this empirical sharpness can be further decomposed into two components, the mini-batch gradient norm and the variation along the statistical manifold.

Theorem 3.2. *Let $p(y|x, \tilde{\boldsymbol{\omega}})$ and $p(y|x, \boldsymbol{\omega})$ denote the predictive distributions at $\tilde{\boldsymbol{\omega}}$ and $\boldsymbol{\omega}$, respectively. For small $\rho > 0$, it follows that*

$$\widehat{\zeta}(\boldsymbol{\omega}, \rho) \cong \rho \|\nabla F_{\mathcal{B}}(\boldsymbol{\omega})\|_2 + \widehat{D}_{KL}(p(y|x, \tilde{\boldsymbol{\omega}}) || p(y|x, \boldsymbol{\omega})),$$

where $\widehat{D}_{KL}$ denotes the empirical KL-divergence over $\mathcal{B}$.

Remark 3.3. *With a Taylor expansion, Zhao et al. (2022a) shows that SAM is a special case of SGD with regularized gradient norm. In contrast, Compagnoni et al. (2023) models SAM as a second-order SDE, though the drift term also penalizes the gradient norm. Different from them, Theorem 3.2 reveals that SAM not only penalizes the gradient norm but also penalizes the KL-divergence between the two predictive distributions, which indicates that SAM also implicitly penalizes the Hessian. In practice, when the gradient norm approaches zero, the sharpness is generally still high. This is a situation that previous studies fail to explain, but Theorem 3.2 suggests that the sharpness is now dominated by the KL-divergence. Therefore, compared to SGD, SAM encourages the optimizer to seek flat solutions in different stages of training.*

To confirm the validity of Theorem 3.2, we trained a ResNet-56 on CIFAR-10/100. As shown in Figure E.1, we observe that the evolution of sharpness is precisely recovered by such an approximation. However, it remains unknown what we are actually optimizing on a single example, and we are going to address this issue in the next subsection.

3.2 Pitfalls of Not Optimizing Easy Examples

Recall that due to gradient ascent, the mini-batch $\mathcal{B}$ at each step is divided into the union of the hard examples $\mathcal{B}^+$ and the easy examples $\mathcal{B}^-$. When we optimize over an example $z \in \mathcal{B}^+$, notice that the KL-divergence $D_{KL}(p(y|x, \tilde{\boldsymbol{\omega}}) || p(y|x, \boldsymbol{\omega}))$ is decreasing. In contrast, when we are minimizing $f_{\tilde{\boldsymbol{\omega}}}(z)$ for an example $z \in \mathcal{B}^-$, we are nevertheless increasing the KL-divergence $D_{KL}(p(y|x, \tilde{\boldsymbol{\omega}}) || p(y|x, \boldsymbol{\omega}))$. Taking into account the equivalence between Hessian and Fisher information matrix, this observation happens to explain why eliminating $\mathcal{B}^-$ would lead to a flatter solution, see Figure 1b. But, surprisingly, this operation is detrimental to generalization. So, what is the optimizer secretly minimizing when optimizing over an example $z \in \mathcal{B}^-$? Do the examples from $\mathcal{B}^+$ and $\mathcal{B}^-$ have the same optimization goal? To answer these questions, we now introduce a concept as follows.

Definition 3.4. *The* gradient consistency *of an example $z \in \mathcal{B}$ after a gradient ascent step is defined as*

$$\mathcal{C}(z) = f_{\tilde{\boldsymbol{\omega}}}(z) - f_{\boldsymbol{\omega}}(z) - D_{KL}(\tilde{\mathbf{p}}_y || \mathbf{p}_y),$$

where $D_{KL}(\tilde{\mathbf{p}}_y || \mathbf{p}_y)$ denotes $D_{KL}(p(y|x, \tilde{\boldsymbol{\omega}}) || p(y|x, \boldsymbol{\omega}))$.

This concept describes the difference in the variations incurred by an example in the loss space and the probability space. Ideally, we would expect them to be as small as possible. Similar to Theorem 3.2, after a second-order Taylor expansion, we obtain

$$\begin{aligned} f_{\tilde{\boldsymbol{\omega}}}(z) &\cong f_{\boldsymbol{\omega}}(z) + \nabla f_{\boldsymbol{\omega}}(z)^T (\tilde{\boldsymbol{\omega}} - \boldsymbol{\omega}) + D_{KL}(\tilde{\mathbf{p}}_y || \mathbf{p}_y) \\ &= f_{\boldsymbol{\omega}}(z) + \frac{\nabla f_{\boldsymbol{\omega}}(z)^T \nabla F_{\mathcal{B}}(\boldsymbol{\omega})}{\|\nabla F_{\mathcal{B}}(\boldsymbol{\omega})\|_2 / \rho} + D_{KL}(\tilde{\mathbf{p}}_y || \mathbf{p}_y). \end{aligned}$$

Then, it follows that

$$\mathcal{C}(z) \approxeq \frac{\nabla f_{\boldsymbol{\omega}}(z)^T \nabla F_{\mathcal{B}}(\boldsymbol{\omega})}{\|\nabla F_{\mathcal{B}}(\boldsymbol{\omega})\|_2 / \rho}. \tag{2}$$

Therefore, up to a factor $\|\nabla F_{\mathcal{B}}(\boldsymbol{\omega})\|_2/\rho$, $\mathcal{C}(z)$ quantifies the inner product between the per-example gradient $\nabla f_{\boldsymbol{\omega}}(z)$ and the mini-batch gradient $\nabla F_{\mathcal{B}}(\boldsymbol{\omega})$. We also trained a ResNet-56 on CIFAR-100 to check whether Equation (2) holds in practice. As depicted in Figure 2, this linear correlation indeed holds and thus allows us to identify the importance of each example according to $\mathcal{C}(z)$.

For an easy example $z \in \mathcal{B}^-$, it is obvious that $\mathcal{C}(z) < 0$. This indicates that the angle between $\nabla f_{\boldsymbol{\omega}}(z)$ and the mini-batch gradient $\nabla F_{\mathcal{B}}(\boldsymbol{\omega})$ is obtuse. More importantly, the following theorem reveals that this trend is more pronounced as training continues.

Theorem 3.5. *Assume $0 < \mathbf{p}_y < \tilde{\mathbf{p}}_y < 1$, then the gradient consistency $\mathcal{C}(z)$ decreases with respect to $\tilde{\mathbf{p}}_y$.*

Remark 3.6. *Theorem 3.5 provides key insight into the operational mechanics of easy examples. Since $0 < \mathbf{p}_y < \tilde{\mathbf{p}}_y < 1$, the ascent step increases the probability of the correct class, yielding a negative gradient consistency $\mathcal{C}(z) < 0$. Furthermore, as $\tilde{\mathbf{p}}_y$ increases towards 1—signifying that the model becomes increasingly confident on these easy instances—$\mathcal{C}(z)$ drops monotonically. Geometrically, this means that for highly confident easy examples, the per-example gradient $\nabla f_{\boldsymbol{\omega}}(z)$ points in a direction that strongly opposes the aggregate mini-batch direction $\nabla F_{\mathcal{B}}(\boldsymbol{\omega})$. Consequently, during the descent step, updating parameters via these easy examples acts as a natural counterweight that prevents the model from overly committing to the dominant gradient directions, thereby serving as an intrinsic anti-overfitting mechanism.*

This result is critical as it suggests that optimizing over the easy examples biases the optimization towards the opposite direction of the mini-batch gradient $\nabla F_{\mathcal{B}}(\boldsymbol{\omega})$. Actually, this trend is not limited to the easy examples. For the hard examples, we can also observe the same changing trend.

Theorem 3.7. *Assume $0 < \tilde{\mathbf{p}}_y \leq \mathbf{p}_y \leq \frac{e^2}{e^2+1}$, it follows that $\mathcal{C}(z) \geq 0$ and decreases with respect to $\tilde{\mathbf{p}}_y$ as well.*

Remark 3.8. *Theorem 3.7 examines the dynamics of hard examples in the regime where prediction confidence remains below the threshold $\frac{e^2}{e^2+1} \approx 0.88$. For these examples, $\mathcal{C}(z) \geq 0$, confirming that their individual gradients align constructively with the mini-batch gradient $\nabla F_{\mathcal{B}}(\boldsymbol{\omega})$. Crucially, as the adversary decreases $\tilde{\mathbf{p}}_y$ (making the loss higher), $\mathcal{C}(z)$ increases. This implies that SAM dynamically amplifies the influence of those hard examples that suffer the greatest perturbation under adverse shift. As a result, hard examples drive the primary momentum of optimization, ensuring efficient representation learning before the model reaches high-confidence regimes.*

Notice that this theorem holds as long as $\tilde{\mathbf{p}}_y \leq \mathbf{p}_y$ and $\mathbf{p}_y \leq \frac{e^2}{e^2+1}$. When $\mathbf{p}_y > \frac{e^2}{e^2+1}$, things become more complicated. In order to arrive at the same conclusion, we additionally introduce an auxiliary lemma as follows.

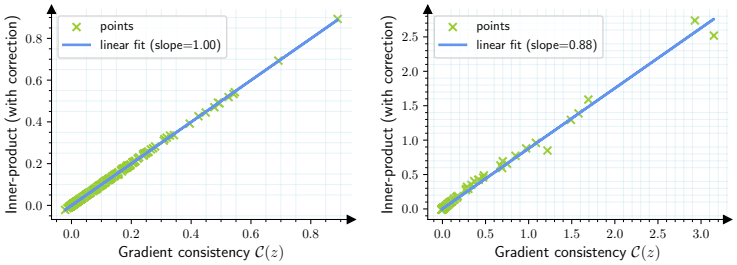
Lemma 3.9. *Assume that the loss function $f_{\boldsymbol{\omega}}(z)$ is G-Lipchitz continuous and L-smooth with respect to $\boldsymbol{\omega}$. Then, it follows that $\tilde{\mathbf{p}}_y \geq e^{-\rho(G+\rho L/2)}\mathbf{p}_y$.*

This lemma shows that the gradient ascent step of SAM will not significantly alter the prediction. Accompanied by Lemma 3.9, we can extend the result of Theorem 3.7 to the case of $\frac{e^2}{e^2+1} < \mathbf{p}_y < 1$.

Theorem 3.10. *Given the assumptions in Lemma 3.9. Let $0 < \tilde{\mathbf{p}}_y \leq \mathbf{p}_y < 1$ and $\frac{e^2}{e^2+1} < \mathbf{p}_y < 1$, then $\mathcal{C}(z) \geq 0$ and decreases with respect to $\tilde{\mathbf{p}}_y$ if $0 < \rho \leq \min\left(\frac{\kappa}{2G}, \sqrt{\frac{\kappa}{L}}\right)$, where $\kappa = \log \frac{\mathbf{p}_y}{2\mathbf{p}_y - 1}$.*

Remark 3.11. *Theorem 3.10 extends the monotonic property of gradient consistency to high-confidence hard examples under a mild constraint on ρ. Taken together, Theorems 3.5, 3.7, and 3.10 paint a unified picture of SAM's dynamic gradient allocation: across all examples, gradient consistency $\mathcal{C}(z)$ monotonically decreases with respect to $\tilde{\mathbf{p}}_y$. This monotonic relationship acts as an adaptive regularizer—automatically soft-damping high-confidence updates while prioritizing ambiguous or noisy boundary cases.*

Putting Theorems 3.5, 3.7, and 3.10 together, we can conclude that SAM is secretly maximizing the angular separation between the per-example gradient and the mini-batch gradient. As a result, compared to SGD, SAM significantly alleviates the issue of overfitting.



(a) Middle phase (b) Terminal phase

Figure 2: The gradient consistency $\mathcal{C}(z)$ linearly correlates with the inner product between the per-example gradient $\nabla f_{\boldsymbol{\omega}}(z)$ and the mini-batch gradient $\nabla F_{\mathcal{B}}(\boldsymbol{\omega})$. In the middle and the terminal phase of training, a random mini-batch is selected to visualize this correlation. The y-axis represents the inner product multiplied by the factor $\rho/\|\nabla F_{\mathcal{B}}(\boldsymbol{\omega})\|_2$.

3.3 Proposed Method

Since the gradient consistency $\mathcal{C}(z)$ quantifies how much the per-example gradient departs from the mini-batch gradient, it naturally offers us a flexible way to perturb the direction of $\nabla F_{\mathcal{B}}(\tilde{\boldsymbol{\omega}})$. For example, to reduce overfitting, we can reweight each example according to $\mathcal{C}(z)$. That is, we can reward examples with small $\mathcal{C}(z)$ and penalize examples with large $\mathcal{C}(z)$. Mathematically, we reformulate the loss function of the descent step of SAM as follows:

$$F_{\mathcal{B}}^{\gamma}(\tilde{\boldsymbol{\omega}}) = \frac{1}{|\mathcal{B}|} \sum_{z \in \mathcal{B}} -[1 + s(\mathcal{C}(z))]^{\gamma} \log \tilde{\mathbf{p}}_y, \tag{3}$$

where γ is a tuning hyperparameter, and $s(\cdot)$ is a squeezing function (e.g. sigmoid) that monotonically maps the input $\mathcal{C}(z)$ to the range of (0, 1). We introduce the squeezing function $s(\cdot)$ to stabilize the training process since $\mathcal{C}(z)$ sometimes can be very large (see Figure 2b). For convenience, we will refer to this variant as Consistency-Anchored Sharpness-Aware Minimization (CASAM) afterwards, and the pseudocode of the algorithm is summarized in Appendix E.

Note that when $\gamma = 0$, it degenerates to the standard SAM. Compared to other SAM variants, we do not alter the ascent direction. While the implementation is simple, it is worth noting that CASAM has one strength that has not been found in other SAM variants. As shown in Figure 3, the hyperparameter γ of CASAM quantitatively controls how much $\nabla F_{\mathcal{B}}(\tilde{\boldsymbol{\omega}})$ deviates from $\nabla F_{\mathcal{B}}(\boldsymbol{\omega})$. This modification incurs almost no computational overhead and is particularly useful for realistic training. For example, in the under-fitting regime, we can set $\gamma > 0$ to accelerate convergence. Also, we can set $\gamma < 0$ to avoid overfitting and memorizing noisy features.

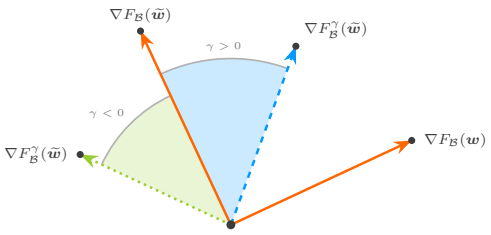
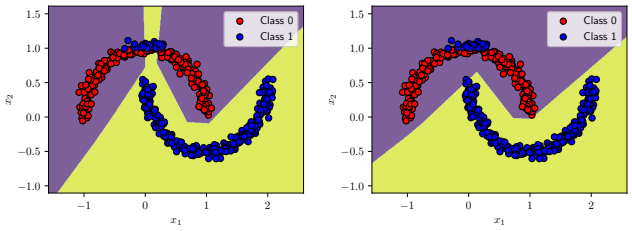


Figure 3: Illustration of how CASAM modulates the angle between the ascent direction $\nabla F_{\mathcal{B}}(\boldsymbol{\omega})$ and the descent direction $\nabla F_{\mathcal{B}}(\tilde{\boldsymbol{\omega}})$. For $\gamma > 0$, it compels them to align more closely. Conversely, when $\gamma < 0$, it drives them apart and widens the angle between them.

To illustrate that CASAM is more capable of anti-overfitting, we constructed a simple two-layer neural network containing 10 hidden nodes to classify the binary moon dataset. We first generated 500 training examples and then flipped the label of 50 examples from class 0 to class 1. We trained the network for 5000 epochs with a mini-batch size of 64. As depicted in Figure 4, SAM tends to overfit the mislabeled examples. In contrast, CASAM learns a decision boundary that correctly captures the underlying data structure.



(a) SAM (b) CASAM

Figure 4: Decision boundary on the noisy moon data where 10% of examples belonging to class 0 (red) are mislabeled as class 1 (blue).

Table 1: Test accuracy (%) of different neural networks trained on CIFAR-100.

	SGD	SAM	ASAM	GASAM	VASSO	EigenSAM	SSAMF	CASAM
ResNet-56	72.64 ± 0.13	75.13 ± 0.30	75.44 ± 0.17	75.21 ± 0.09	75.16 ± 0.07	75.42 ± 0.11	74.96 ± 0.21	$\mathbf{75.89 \pm 0.13}$
WRN-28-10	81.46 ± 0.11	83.58 ± 0.15	83.47 ± 0.15	83.69 ± 0.31	83.66 ± 0.12	83.72 ± 0.07	83.22 ± 0.18	$\mathbf{84.12 \pm 0.24}$
PyramidNet-110	82.26 ± 0.52	84.50 ± 0.29	84.06 ± 0.05	84.62 ± 0.17	84.68 ± 0.28	84.58 ± 0.13	84.17 ± 0.08	$\mathbf{85.06 \pm 0.22}$

4 Experiments

In this section, we present the empirical results on several popular tasks, including image classification and natural language processing (deferred to Section A due to space limitations).

4.1 Image Classification from Scratch

We first evaluate how CASAM performs on realistic datasets, including CIFAR-100 (Krizhevsky et al., 2009) and ImageNet-1K (Deng et al., 2009). The baselines include ASAM (Kwon et al., 2021), GASAM (Zhang et al., 2022), SSAMF (Mi et al., 2022), VASSO (Li & Giannakis, 2023) and EigenSAM (Luo et al., 2024). All results are reported based on three random runs, and the code to reproduce these results can be found at `https://github.com/aiu-group/casam2026/tree/main`. The detailed settings can be found in Appendix D.

Results on CIFAR-100. To verify the efficacy of CASAM, we trained it with three different kinds of neural networks, including ResNet-56 (He et al., 2016), WRN-28-10 (Zagoruyko & Komodakis, 2016), and PyramidNet-110 (Han et al., 2017). Both networks are trained up to 200 epochs to ensure that the datasets are fully learned. As shown in Table 1, both variants of SAM generalize better than the base optimizer SGD, and CASAM always achieves the highest test accuracy. This indicates that the issue of overfitting does not perish even when the networks are trained with SAM.

Results on ImageNet-1K. We continue to investigate the performance of CASAM on the large-scale ImageNet-1K dataset. We only adopt basic data augmentations, namely, resizing and cropping images to a 224-pixel resolution and then normalizing them. Apart from ResNet, we further provide the classification results of the vision transformer (ViT) (Dosovitskiy et al., 2021). The base optimizer for ResNet is SGD with momentum acceleration, and all networks are trained for 100 epochs. In contrast, the base optimizer for ViT is AdamW, and all networks are trained up to 300 epochs. Table 2 reveals that CASAM remains superior to SAM in terms of validation accuracy. Notably, CASAM improves the top-1 accuracy of ResNet-50 and ViT-S-32 from 76.19%, 69.65% to 76.73%, 70.11%, respectively. This improvement is substantial for ImageNet-1K. Interestingly, we also observe that SAM tends to be more effective when the base optimizer is AdamW. From the perspective of anti-overfitting, it implies that SAM encourages the optimization direction to deviate much more from the gradient descent direction. This explanation is reasonable because the optimization trajectory of AdamW on the loss landscape is more ragged.

4.2 Robustness to Label Noise

It is known that SAM is particularly effective in the presence of noise (Baek et al., 2024). In this part, we demonstrate that CASAM still performs better than SAM. To this end, we trained ResNet-56 on CIFAR-10 corrupted with asymmetric and instance-dependent noise (IDN) (Wang et al., 2025).

Table 2: Accuracy (%) on the validation set of ImageNet-1K.

	SGD/AdamW	SAM	GASAM	VASSO	CASAM
ResNet-18	69.95 ± 0.03	70.05 ± 0.07	70.07 ± 0.04	70.11 ± 0.06	$\mathbf{70.29 \pm 0.11}$
ResNet-50	75.78 ± 0.12	76.19 ± 0.08	76.28 ± 0.22	76.25 ± 0.15	$\mathbf{76.73 \pm 0.16}$
ViT-S-32	65.07 ± 0.08	69.65 ± 0.11	69.72 ± 0.15	69.86 ± 0.10	$\mathbf{70.31 \pm 0.21}$
ViT-S-16	70.84 ± 0.16	74.23 ± 0.09	74.29 ± 0.14	74.41 ± 0.22	$\mathbf{74.78 \pm 0.13}$

Table 3: Test accuracy (%) on corrupted CIFAR-10.

		$\tau = 0.1$	$\tau = 0.3$	$\tau = 0.5$	$\tau = 0.7$
Asymmetric	SAM	90.82 ± 0.23	87.76 ± 0.31	68.13 ± 0.18	57.83 ± 0.40
	VASSO	90.68 ± 0.14	88.18 ± 0.12	69.84 ± 0.23	57.48 ± 0.19
	GASAM	91.23 ± 0.08	87.16 ± 0.41	67.85 ± 0.26	56.45 ± 0.17
	SANER	91.57 ± 0.14	89.22 ± 0.08	74.98 ± 0.12	57.66 ± 0.21
	δ-SAM	87.88 ± 0.23	85.79 ± 0.14	71.71 ± 0.08	55.35 ± 0.16
	LASAM	90.76 ± 0.09	88.92 ± 0.11	65.01 ± 0.23	56.68 ± 0.17
	CASAM	$\mathbf{92.02 \pm 0.11}$	$\mathbf{89.66 \pm 0.31}$	$\mathbf{75.65 \pm 0.17}$	$\mathbf{58.99 \pm 0.30}$
IDN	SAM	88.9 ± 0.23	84.38 ± 0.16	78.48 ± 0.29	40.24 ± 0.65
	VASSO	88.38 ± 0.05	84.95 ± 0.14	76.45 ± 0.21	40.81 ± 0.34
	GASAM	89.24 ± 0.14	85.00 ± 0.08	78.62 ± 0.35	39.57 ± 0.24
	SANER	90.76 ± 0.09	88.92 ± 0.11	65.01 ± 0.23	56.68 ± 0.17
	δ-SAM	86.60 ± 0.12	82.81 ± 0.13	72.05 ± 0.09	36.27 ± 0.31
	LASAM	89.02 ± 0.11	84.86 ± 0.07	77.31 ± 0.14	44.08 ± 0.19
	CASAM	$\mathbf{89.75 \pm 0.18}$	$\mathbf{85.85 \pm 0.13}$	$\mathbf{78.72 \pm 0.18}$	$\mathbf{46.25 \pm 0.26}$

In particular, we use LDSAM to denote the variant of removing the KL term in $\mathcal{C}(z)$ and two other variants, δ-SAM (Zhou et al., 2022) and SANER (Luong et al., 2026), are included. To ensure successful training, we used a relatively small perturbation radius $\rho = 0.1$ for both runs. As shown in Table 3, both VASSO and GASAM do not consistently perform better than SAM. In some cases, they perform better when the corruption level τ is low, but sometimes they also perform worse when τ is large. In contrast, CASAM significantly and consistently performs better than SAM, particularly when $\tau = 0.5, 0.7$ ($\sim$1.5% improvement). This is because CASAM is designed to counteract overfitting when its hyperparameter $\gamma < 0$. Moreover, we can observe that LDSAM substantially performs worse than CASAM. This confirms our claim—it is not merely the loss differential that matters, but the gradient consistency relative to the mini-batch update direction that drives the anti-overfitting effect.

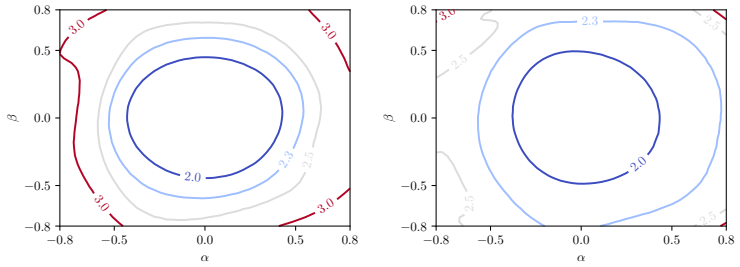
To investigate how CASAM behaves when confronted with realistic label noise, we further conduct a number of experiments on several popular noisy datasets, including Animal-10N (Song et al., 2019), Food-101N (Lee et al., 2018), and Clothing1M (Xiao et al., 2015). Table B.1 suggests that SAM ameliorates the generalization performance of SGD by 3$\sim$4% and CASAM can further improve by 0.5$\sim$1%. These results together indicate the crucial role of anti-overfitting during training. To probe how CASAM modifies the loss landscape, we also visualize the contour surface around the solution. As depicted in Figure 5, SAM and CASAM exhibit similar structure when the perturbation is very close to the solution. But when the perturbation becomes large, the loss of CASAM grows much more slowly than SAM, indicating its superior robustness against noise.

4.3 Hyperparameter Sensitivity

Note that CASAM introduces an additional hyperparameter γ, which controls the degree to which we either prefer to optimize over the hard examples ($\gamma > 0$) or the easy examples ($\gamma < 0$). The choice of γ is critical to the model generalization. To investigate how γ influences the test performance, we trained a number of ResNet-56 on CIFAR-100 with different γ, ranging from -2 to 2 with a step size of 0.5. As shown in Figure 6a, the test accuracy generally decreases with the value of γ. This indicates that prioritizing hard examples during training is not always as effective as expected. In practice, we recommend using $\gamma = -1$ or -2 when the training enters the overfitting regime.

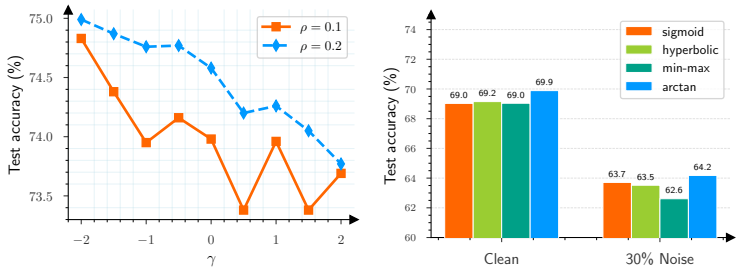
Apart from γ, CASAM also incorporates a function $s(x)$ to squeeze the consistency $\mathcal{C}(z)$ to the range of $(0, 1)$. It is interesting to investigate which types of functions are useful to improve the generalization performance. To this end, we tested four types of squeezing functions:

1. sigmoid: $s(x) = 1.0/[1.0 + \exp(-x)]$;



(a) SAM (b) CASAM

Figure 5: Loss landscape for ResNet-56 trained on CIFAR-10 with 70% symmetric noise. We perturb the converged parameters $\boldsymbol{\omega}$ along two orthogonal directions $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ that are first randomly sampled from $\mathcal{N}(0, \mathbb{I}_d)$ and then normalized to have a unit norm. The surface is computed by evaluating $F_S(\boldsymbol{\omega} + \alpha\boldsymbol{\omega}_1 + \beta\boldsymbol{\omega}_2)$ for $\alpha, \beta \in [-0.8, 0.8]$.



(a) γ (b) Squeezing function

Figure 6: (a) Investigation of how the hyperparameter γ affects the model generalization. Note that when $\gamma = 0$, CASAM degenerates to the vanilla SAM. (b) Influence of different types of squeezing function. Apart from the clean CIFAR-100 dataset, we also corrupt 30% of the true labels to increase the learning difficulty.

2. hyperbolic: $s(x) = \tanh(x)/2 + 1/2$;
3. min-max: $s(x) = [x - \min(x)]/[\max(x) - \min(x)]$;
4. arctan: $s(x) = \arctan(x)/\pi + 1/2$.

As shown in Figure 6b, we observe that the min-max function performs much worse than other squeezing functions, especially on the corrupted data. Meanwhile, the performance of sigmoid and hyperbolic functions is parallel with each other since both of them decay at an exponential rate. Surprisingly, the arctan function significantly outperforms other functions (an increase of 0.8% on clean data and 0.6% on corrupted data). The reason might be that the arctan function approaches its asymptotes much more slowly than the exponential functions. Throughout this study, we use arctan as the squeezing function by default.

5 Conclusion

In many tasks, such as image classification and object detection, SAM has proven to be a very promising optimizer. Many studies argue that its success is due to the implicit regularization of the gradient norm. Different from them, we showed that it can be further justified from the perspective of anti-overfitting. We demonstrated that the gradient consistency, which approximately quantifies the inner product between the per-example gradient and the mini-batch gradient, keeps decreasing in the training process. Notice that this finding does not invalidate prior research; rather, it serves as a supplement and refinement to the established studies. Building upon these insights, we propose CASAM to further enhance the ability of anti-overfitting. Note that our analysis is not limited to the cross-entropy loss, it is compatible with MSE loss as well.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 12501710 and 62276208, in part by the Natural Science Basic Research Program of Shaanxi Province under Grant 2024JC-JCQN-02, and in part by Fundamental Research Funds for the Central Universities.

References

Agiollo, A., Kim, Y. I., and Khanna, R. Approximating memorization using loss surface geometry for dataset pruning and summarization. In *Proceedings of the Conference on Knowledge Discovery and Data Mining*, pp. 17–28, 2024.

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

Andriushchenko, M., Bahri, D., Mobahi, H., and Flammarion, N. Sharpness-aware minimization leads to low-rank features. In *Advances in Neural Information Processing Systems*, pp. 47032–47051, 2023.

Baek, C., Kolter, J. Z., and Raghunathan, A. Why is SAM robust to label noise? In *International Conference on Learning Representations*, pp. 1–19, 2024.

Bahri, D., Mobahi, H., and Tay, Y. Sharpness-aware minimization improves language model generalization. In *Annual Meeting of the Association for Computational Linguistics*, pp. 7360–7371, 2022.

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

Bisla, D., Wang, J., and Choromanska, A. Low-pass filtering SGD for recovering flat optima in the deep learning optimization landscape. In *International Conference on Artificial Intelligence and Statistics*, pp. 8299–8339, 2022.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision*, pp. 446–461, 2014.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-SGD: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

Chen, X., Hsieh, C.-J., and Gong, B. When vision transformers outperform resnets without pre-training or strong data augmentations. In *International Conference on Learning Representation*, pp. 1–20, 2022.

Chen, Z., Zhang, J., Kou, Y., Chen, X., Hsieh, C.-J., and Gu, Q. Why does sharpness-aware minimization generalize better than SGD? In *Advances in Neural Information Processing Systems*, pp. 72325–72376, 2023.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 2924–2936, 2019.

Compagnoni, E. M., Biggio, L., Orvieto, A., Proske, F. N., Kersting, H., and Lucchi, A. An SDE for modeling SAM: Theory and insights. In *International Conference on Machine Learning*, pp. 25209–25253, 2023.

Dai, Y., Ahn, K., and Sra, S. The crucial role of normalization in sharpness-aware minimization. In *Advances in Neural Information Processing Systems*, pp. 67741–67770, 2023.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Deng, J., Zhu, Q., Pang, J., Yang, L., Fu, Z., and Zhang, B. Efficiently seeking flat minima for better generalization in fine-tuning large language models and beyond. In *Annual AAAI Conference on Artificial Intelligence*, pp. 1–14, 2025.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pp. 1019–1028, 2017.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, pp. 1–21, 2021.

Du, J., Yan, H., Feng, J., Zhou, J. T., Zhen, L., Goh, R. S. M., and Tan, V. Efficient sharpness-aware minimization for improved training of neural networks. In *International Conference on Learning Representations*, pp. 1–18, 2022a.

Du, J., Zhou, D., Feng, J., Tan, V., and Zhou, J. T. Sharpness-aware training for free. In *Advances in Neural Information Processing Systems*, pp. 23439–23451, 2022b.

Dziugaite, G. K., Drouin, A., Neal, B., Rajkumar, N., Caballero, E., Wang, L., Mitliagkas, I., and Roy, D. M. In search of robust measures of generalization. In *Advances in Neural Information Processing Systems*, pp. 11723–11733, 2020.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, pp. 1–19, 2021.

Han, D., Kim, J., and Kim, J. Deep pyramidal residual networks. In *Conference on Computer Vision and Pattern Recognition*, pp. 5927–5935, 2017.

He, F., Liu, T., and Tao, D. Control batch size and learning rate to generalize well: Theoretical and empirical evidence. In *Advances in Neural Information Processing Systems*, pp. 1–10, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hinton, G. E. and van Camp, D. Keeping neural networks simple. In *International Conference on Artificial Neural Networks*, pp. 11–18, 1993.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

Huang, Z., Zhu, M., Xia, X., Shen, L., Yu, J., Gong, C., Han, B., Du, B., and Liu, T. Robust generalization against photon-limited corruptions via worst-case sharpness minimization. In *Conference on Computer Vision and Pattern Recognition*, pp. 16175–16185, 2023.

Ji, J., Li, G., Fu, J., Afghah, F., Guo, L., Yuan, X., and Ma, X. A single-step, sharpness-aware minimization is all you need to achieve efficient and accurate sparse training. In *Advances in Neural Information Processing Systems*, pp. 44269–44290, 2024.

Jiang, W., Yang, H., Zhang, Y., and Kwok, J. An adaptive policy to employ sharpness-aware minimization. In *International Conference on Learning Representations*, pp. 1–19, 2023.

Kaddour, J., Liu, L., Silva, R., and Kusner, M. J. When do flat minima optimizers work? In *Advances in Neural Information Processing Systems*, pp. 16577–16595, 2022.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, pp. 1–16, 2017.

Khanh, P., Luong, H.-C., Mordukhovich, B., and Tran, D. Fundamental convergence analysis of sharpness-aware minimization. In *Advances in Neural Information Processing Systems*, pp. 13149–13182, 2024.

Kim, H., Park, J., Choi, Y., Lee, W., and Lee, J. Exploring the effect of multi-step ascent in sharpness-aware minimization. *arXiv preprint arXiv:2302.10181*, 2023.

Kim, M., Li, D., Hu, S. X., and Hospedales, T. Fisher SAM: Information geometry and sharpness aware minimisation. In *International Conference on Machine Learning*, pp. 11148–11161, 2022.

Kim, Y. I., Agiollo, A., Agrawal, P., Royset, J., and Khanna, R. Membership privacy risks of sharpness aware minimization. In *International Conference on Learning Representations*, pp. 80945–80973, 2026.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, pp. 1–15, 2015.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Kwon, J., Kim, J., Park, H., and Choi, I. K. ASAM: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, pp. 5905–5914, 2021.

Lee, K.-H., He, X., Zhang, L., and Yang, L. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Conference on Computer Vision and Pattern Recognition*, pp. 5447–5456, 2018.

Li, B. and Giannakis, G. Enhancing sharpness-aware optimization through variance suppression. In *Advances in Neural Information Processing Systems*, pp. 70861–70879, 2023.

Liu, Y., Mai, S., Chen, X., Hsieh, C.-J., and You, Y. Towards efficient and scalable sharpness-aware minimization. In *Conference on Computer Vision and Pattern Recognition*, pp. 12360–12370, 2022a.

Liu, Y., Mai, S., Cheng, M., Chen, X., Hsieh, C.-J., and You, Y. Random sharpness-aware minimization. In *Advances in Neural Information Processing Systems*, pp. 24543–24556, 2022b.

Loshchilov, I. and Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, pp. 1–16, 2017.

Luo, H., Truong, T., Pham, T., Harandi, M., Phung, D., and Le, T. Explicit eigenvalue regularization improves sharpness-aware minimization. In *Advances in Neural Information Processing Systems*, pp. 4424–4453, 2024.

Luo, H., Harandi, M., Phung, D., and Le, T. Unveiling m-sharpness through the structure of stochastic gradient noise. In *Advances in Neural Information Processing Systems*, pp. 68074–68112, 2026.

Luong, H.-C., Nguyen-Quang, T., and Tran, M.-T. Understanding SAM's robustness to noisy labels through gradient down-weighting. In *International Conference on Artificial Intelligence and Statistics*, pp. 1–16, 2026.

Lyu, K., Li, Z., and Arora, S. Understanding the generalization benefit of normalization layers: Sharpness reduction. In *Advances in Neural Information Processing Systems*, pp. 34689–34708, 2022.

Marcus, M., Santorini, B., and Marcinkiewicz, M. A. Building a large annotated corpus of English: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

Mi, P., Shen, L., Ren, T., Zhou, Y., Sun, X., Ji, R., and Tao, D. Make sharpness-aware minimization stronger: A sparsified perturbation approach. In *Advances in Neural Information Processing Systems*, pp. 30950–30962, 2022.

Möllenhoff, T. and Khan, M. E. SAM as an optimal relaxation of Bayes. In *International Conference on Learning Representations*, pp. 1–25, 2023.

Mueller, M., Vlaar, T., Rolnick, D., and Hein, M. Normalization layers are all that sharpness-aware minimization needs. In *Advances in Neural Information Processing Systems*, pp. 69228–69252, 2023.

Oikonomou, D. and Loizou, N. Sharpness-aware minimization: General analysis and improved rates. In *International Conference on Learning Representations*, pp. 30071–30097, 2025.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Si, D. and Yun, C. Practical sharpness-aware minimization cannot converge all the way to optima. In *Advances in Neural Information Processing Systems*, pp. 26190–26228, 2023.

Song, H., Kim, M., and Lee, J.-G. Selfie: Refurbishing unclean samples for robust deep learning. In *International Conference on Machine Learning*, pp. 5907–5915, 2019.

Tan, C., Zhang, J., Liu, J., and Gong, Y. Sharpness-aware lookahead for accelerating convergence and improving generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 10375 – 10388, 2024.

Tan, C., Zhou, Y., Ye, H., Dai, G., Liu, J., Song, Z., Zhang, J., Zhao, Z., Hao, Y., and Xu, Y. Towards understanding the calibration benefits of sharpness-aware minimization. In *International Conference on Learning Representations*, pp. 1–21, 2026.

Wang, J., Liu, X., Zhou, X., Hu, G., Zhai, D., Jiang, J., and Ji, X. Joint asymmetric loss for learning with noisy labels. In *International Conference on Computer Vision*, pp. 1947–1956, 2025.

Wen, K., Ma, T., and Li, Z. How does sharpness-aware minimization minimizes sharpness? In *Optimization for Machine Learning (NeurIPS Workshop)*, pp. 1–94, 2022.

Wen, K., Li, Z., and Ma, T. Sharpness minimization algorithms do not only minimize sharpness to achieve better generalization. In *Advances in Neural Information Processing Systems*, pp. 1024–1035, 2023.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pp. 1–11, 2017.

Xiao, T., Xia, T., Yang, Y., Huang, C., and Wang, X. Learning from massive noisy labeled data for image classification. In *Conference on Computer Vision and Pattern Recognition*, pp. 2691–2699, 2015.

Xie, W., Latorre, F., Antonakopoulos, K., Pethick, T., and Cevher, V. Improving SAM requires rethinking its optimization formulation. In *International Conference on Machine Learning*, pp. 54475–54487, 2024a.

Xie, W., Pethick, T., and Cevher, V. Sampa: Sharpness-aware minimization parallelized. In *Advances in Neural Information Processing Systems*, pp. 1–25, 2024b.

Yu, R., Zhang, Y., and Kwok, J. Improving sharpness-aware minimization by Lookahead. In *International Conference on Machine Learning*, pp. 57776–57802, 2024.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *British Machine Vision Conference*, pp. 1–15, 2016.

Zhang, M., Lucas, J., Ba, J., and Hinton, G. E. Lookahead optimizer: k steps forward, 1 step back. In *Advances in Neural Information Processing Systems*, pp. 1–12, 2019.

Zhang, Y., Zheng, S., Wu, P., Goswami, M., and Chen, C. Learning with feature-dependent label noise: A progressive approach. In *International Conference on Learning Representations*, pp. 1–13, 2021.

Zhang, Z., Luo, R., Su, Q., and Sun, X. GA-SAM: Gradient-strength based adaptive sharpness-aware minimization for improved generalization. In *Conference on Empirical Methods in Natural Language Processing*, pp. 3888–3903, 2022.

Zhao, Y., Zhang, H., and Hu, X. Penalizing gradient norm for efficiently improving generalization in deep learning. In *International Conference on Machine Learning*, pp. 26982–26992, 2022a.

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

Zheng, Y., Zhang, R., and Mao, Y. Regularizing neural networks via adversarial model perturbation. In *Conference on Computer Vision and Pattern Recognition*, pp. 8156–8165, 2021.

Zhou, P., Feng, J., Ma, C., Xiong, C., Hoi, S. C. H., et al. Towards theoretically understanding why SGD generalizes better than Adam in deep learning. In *Advances in Neural Information Processing Systems*, pp. 21285–21296, 2020.

Zhou, W., Liu, F., Zhang, H., and Chen, M. Sharpness-aware minimization with dynamic reweighting. In *Findings of the Association for Computational Linguistics*, pp. 5686–5699, 2022.

Zhou, Z., Wang, M., Mao, Y., Li, B., and Yan, J. Sharpness-aware minimization efficiently selects flatter minima late in training. In *International Conference on Learning Representations*, pp. 20949–20980, 2025.

Zhuang, J., Tang, T., Ding, Y., Tatikonda, S. C., Dvornek, N., Papademetris, X., and Duncan, J. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. In *Advances in Neural Information Processing Systems*, pp. 18795–18806, 2020.

Zhuang, J., Gong, B., Yuan, L., Cui, Y., Adam, H., Dvornek, N. C., s Duncan, J., Liu, T., et al. Surrogate gap minimization improves sharpness-aware training. In *International Conference on Learning Representations*, pp. 1–24, 2022.

Zou, Y., Kawaguchi, K., Liu, Y., Liu, J., Lee, M.-L., and Hsu, W. Towards robust out-of-distribution generalization bounds via sharpness. In *International Conference on Learning Representations*, pp. 31443–31482, 2024.

A Results on Natural Language Processing

We first trained several long short-term memory (LSTM) networks on the Penn Treebank dataset (Marcus et al., 1993). The mini-batch size is 20, and both models are trained up to 200 epochs. The base optimizers include Adam and Adabelief (Zhuang et al., 2020). As shown in Table A.1, SAM only performs on par with the base optimizers. In contrast, CASAM substantially reduces the perplexity. We also employed a Transformer-based model T5 (Raffel et al., 2020) to classify BoolQ (Clark et al., 2019), a question-answering dataset consisting of yes/no questions. Table A.2 further confirms the superiority of CASAM over SAM.

We also investigated the robustness of γ and the squeezing functions for LSTM. As shown in Table A.3, we can observe that $\gamma > 0$ generally performs worse than $\gamma < 0$, and $\gamma = \{-3, -4\}$ generally perform worse than $\gamma = \{-2, -1\}$. This suggests that γ should be smaller than 0 and should not be too small. Furthermore, the min-max squeezing function generally performs much worse than other functions. And, the arctan function is most recommended, just as in the vision task. Therefore, without extensive dataset-specific tuning, a principled and safe selection strategy is to select γ from {-2, -1}, and arctan can be used as the default squeezing function.

Table A.1: Test perplexity of LSTM on Penn Treebank (lower is better). The number of LSTM layers ranges from 1 to 3.

	# layer	Base	SAM	CASAM
Adam	1	87.40 ± 0.05	86.81 ± 0.07	$\mathbf{83.66 \pm 0.04}$
	2	70.12 ± 0.11	70.55 ± 0.08	$\mathbf{67.56 \pm 0.12}$
	3	67.79 ± 0.15	67.31 ± 0.06	$\mathbf{65.66 \pm 0.13}$
Adabelief	1	88.27 ± 0.09	88.61 ± 0.07	$\mathbf{85.26 \pm 0.12}$
	2	70.19 ± 0.12	69.97 ± 0.21	$\mathbf{67.63 \pm 0.16}$
	3	67.39 ± 0.15	67.35 ± 0.09	$\mathbf{64.91 \pm 0.03}$

B Results on Realistic Noisy Datasets

To investigate how CASAM behaves when confronted with realistic label noise, we further conduct a number of experiments on several popular noisy datasets, including Animal-10N (Song et al., 2019),

Table A.2: Test accuracy (%) of T5 on BoolQ dataset.

	Base	SAM	CASAM
Adam	77.12 ± 0.07	77.39 ± 0.13	$\mathbf{77.71 \pm 0.09}$
AdamW	77.20 ± 0.08	77.79 ± 0.06	$\mathbf{78.38 \pm 0.11}$

Table A.3: Hyperparameter sensitivity for LSTM on Penn Treebank (average perplexity over three runs).

	squeezing function	$\gamma = -4.0$	-3.0	-2.0	-1.0	1.0	2.0
ρ=0.05	sigmoid	66.95	65.24	64.84	65.02	69.77	69.92
	arctan	66.88	65.27	**64.76**	65.03	69.85	70.30
	hyperbolic	67.76	65.43	64.87	65.11	69.35	70.38
	min-max	67.41	66.21	65.77	66.17	69.75	70.43
ρ=0.2	sigmoid	66.93	66.38	65.68	65.85	70.64	70.70
	arctan	66.02	65.55	**65.49**	65.69	70.68	70.77
	hyperbolic	66.72	65.88	65.89	66.71	70.21	70.46
	min-max	68.11	67.36	66.09	66.85	70.39	70.48

Food-101N (Lee et al., 2018), and Clothing1M (Xiao et al., 2015). Note that Animal-10N is composed of human-labeled animal images for 10 classes with an approximately 8% label noise rate. Like CIFAR-10, it contains 50k training images and 5k test images, respectively. Moreover, Food-101N contains 310k training images for food classification and approximately has 20% label noise rate. The final accuracy is evaluated on the clean test set of Food-101 (Bossard et al., 2014), which contains 25k images with curated labels. In contrast, Clothing1M is a relatively large dataset with an estimated 38.5% noise and consists of 1 million clothing images crawled from online websites with 14 classes. This dataset is quite noisy and additionally provides 50k, 14k, and 10k manually verified clean data for training, validation, and testing. Following Zhang et al. (2021), however, we discard the 50k clean training data in our experiment and randomly construct a pseudo-balanced subset as the training set that approximately contains 260k images. We finally evaluate the classification accuracy on the 10k clean data. Table B.1 suggests that SAM ameliorates the generalization performance of SGD by 3$\sim$4% and CASAM can further improve by 0.5$\sim$1%. These results together indicate the crucial role of anti-overfitting during training.

Table B.1: Test accuracy (%) on several noisy datasets.

		SGD	SAM	CASAM
Animal-10N	ResNet-56	69.82 ± 0.12	74.82 ± 0.15	$\mathbf{75.78 \pm 0.21}$
Food-101N	ResNet-18	78.88 ± 0.05	82.25 ± 0.15	$\mathbf{82.74 \pm 0.06}$
	ResNet-50	81.81 ± 0.18	84.94 ± 0.13	$\mathbf{85.83 \pm 0.09}$
Clothing1M	ResNet-18	68.48 ± 0.12	71.45 ± 0.08	$\mathbf{71.91 \pm 0.14}$
	ResNet-50	68.86 ± 0.07	71.95 ± 0.11	$\mathbf{72.82 \pm 0.12}$

C Compatibility with MSE

The gradient consistency actually can be extended to encompass the exponential family of loss functions including MSE. In a probabilistic view, MSE is the Negative Log-Likelihood (NLL) of a model that assumes its noise follows a Gaussian with a constant variance. As such, one can again use the equivalence between the Fisher Information Matrix and Hessian, which plays a crucial role in our argument. To validate the effectiveness of the proposed CASAM, we trained ResNet-20/56 on CIFAR-10 using MSE loss. Table C.1 displays the test accuracy (%). It indicates that SAM is compatible with MSE loss and that CASAM can further enhance the generalization ($\sim$0.6% improvement).

Table C.1: Performance on MSE loss.

		ResNet-20	ResNet-56
Test Accuracy	SGD	90.86 ± 0.13	92.56 ± 0.23
	SAM	91.24 ± 0.18	92.90 ± 0.09
	CASAM	$\mathbf{91.89 \pm 0.07}$	$\mathbf{93.43 \pm 0.12}$
Test Loss	SGD	0.0138 ± 0.0004	0.0115 ± 0.0010
	SAM	0.0134 ± 0.0005	0.0108 ± 0.0007
	CASAM	$\mathbf{0.0122 \pm 0.0009}$	$\mathbf{0.0099 \pm 0.0008}$

D Hyperparameter Settings for CIFAR-100 and ImageNet-1K

For CIFAR-100, the default mini-batch size is 128, and the learning rate is decayed according to the popular cosine scheduler (Loshchilov & Hutter, 2017). To make a fair comparison, we first tune the hyperparameters, such as learning rate and weight decay coefficient, using the base optimizer SGD, and then apply them to the different variants of SAM. The learning rate and the weight decay coefficient are selected from $\{0.01, 0.05, 0.1\}$ and $\{1.0\times10^{-4}, 5.0\times10^{-4}, 1.0\times10^{-3}\}$, respectively. Following the suggestion of Foret et al. (2021), the perturbation radius ρ of SAM, GASAM, VASSO, and CASAM is swept over {0.05, 0.1, 0.2}. In contrast, the perturbation radius ρ of ASAM should be much larger, and we thus sweep it over {0.5, 1.0, 2.0}. For VASSO, we need to tune another hyperparameter $\theta \in (0, 1)$ and we choose it from {0.4, 0.9} according to Li & Giannakis (2023). At last, we vary the exponent γ of CASAM from {-0.5, -1.0, -2.0} because our primary purpose is to mitigate overfitting in this task.

For ImageNet-1K, since it is too expensive to tune the learning rate and weight decay, we generally follow the settings as shown in Table D.1, and the learning rate is also decayed according to the cosine scheduler. The additional hyperparameter tied to each optimizer follows the settings of the original paper. The γ of CASAM is selected from {-0.5, -1, -2}. While decreasing γ strongly suppresses memorization of sharp examples, pushing it to -2 on highly complex datasets like ImageNet over-penalizes informative but difficult examples, leading to optimization stagnation. Therefore, $\gamma = -1$ provides the optimal balance between regularization and maintaining sufficient learning capacity. Moreover, because CASAM only leverages the one-dimensional loss vector computed during the standard SAM forward/backward passes, the per-example weighting and squeezing function computations actually introduce lightweight scalar operations, as shown in Table D.2.

Table D.1: Hyperparameters for training ImageNet-1K with different neural networks. Note that all experiments are conducted on a server with 4 GPUs for parallelization.

	Peak learning rate	Mini-batch size	Weight decay	Perturbation radius	Epochs	Hyperparameter γ
ResNet-18	0.1	256×4	1.0e-4	0.05	100	-1
ResNet-50	0.1	256×4	1.0e-4	0.2	100	-1
ViT-S-16/32	4.0e-3	512×4	0.3	0.2	300	-1

Table D.2: Computational footprint of ResNet-18/50 architecture on ImageNet-1K. Compared to vanilla SAM, CASAM incurs almost negligible memory and computational cost.

		Wall-clock time(s/epoch)	Peak allocated memory (mb)	Throughput(images/s)
ResNet-18	SAM	199.68	6962.81	6416.08
	CASAM	201.34	6963.68	6363.12 (-0.82%)
ResNet-50	SAM	529.59	22639.02	2424.51
	CASAM	530.96	22640.11	2414.28 (-0.42%)

E Miscellaneous

The difference between CASAM and SAM is minor. We only modify the loss function of the gradient descent step so that the importance of each example can be considered. The pseudocode of CASAM is summarized in Algorithm 1 as follows.

Algorithm 1 Consistency-Anchored Sharpness-Aware Minimization (CASAM)

Input: Training set $S = \{z_i = (x_i, y_i)\}_{i=1}^n$, objective function $F_S(\boldsymbol{\omega})$, initial weight $\boldsymbol{\omega}_0 \in \mathbb{R}^d$, learning rate $\eta > 0$, perturbation radius $\rho > 0$, training iterations T, regularization coefficient γ, and base optimizer $\mathcal{A}$ (e.g. SGD)

Output: $\boldsymbol{w}_T$

1: **for** $k = 0, 1, \cdots, T-1$ **do**
2: Sample a mini-batch $\mathcal{B}_k = \{(x_1^k, y_1^k), \cdots, (x_m^k, y_m^k)\}$;
3: Compute cross-entropy loss $F_{\mathcal{B}_k}(\boldsymbol{\omega_k}) = -\frac{1}{m}\sum_{z_i \in \mathcal{B}_k} \log \mathbf{p}_{y_i}$, where $\mathbf{p}_{y_i}$ is the predictive probability associated with the true label;
4: Compute perturbed weight $\tilde{\boldsymbol{\omega}}_k = \boldsymbol{\omega}_k + \rho \cdot \frac{\nabla_{\boldsymbol{\omega}_k} F_{\mathcal{B}_k}(\boldsymbol{\omega}_k)}{\|\nabla_{\boldsymbol{\omega}_k} F_{\mathcal{B}_k}(\boldsymbol{\omega}_k)\|}$;
5: Compute perturbed loss $F^{\gamma}_{\mathcal{B}_k}(\tilde{\boldsymbol{\omega}}_k)$ per Equation (3);
6: Compute gradient $\tilde{\boldsymbol{g}}_k = \nabla_{\boldsymbol{\omega}} F^{\gamma}_{\mathcal{B}_k}(\tilde{\boldsymbol{\omega}}_k)|_{\boldsymbol{\omega}=\tilde{\boldsymbol{\omega}}_k}$;
7: Update weight with base optimizer $\mathcal{A}$, e.g. $\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k - \eta_k \tilde{\boldsymbol{g}}_k$;
8: **end for**

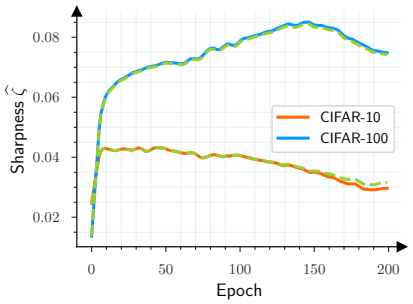


Figure E.1: Validity of Theorem 3.2. Sharpness $\widehat{\zeta}$ (solid line) can be approximately recovered from the sum of gradient norm and KL-divergence (dashed line) on CIFAR-10/100.

Dynamic scheduling of γ. Typically, the early epochs of training correspond to underfitting (feature learning), while late stages correspond to overfitting (memorization). Inspired by this, we implemented a dynamic schedule where γ changes in a cosine way. Table E.1 shows how the test accuracy changes with the schedule. We observe that a negative γ is preferred for improving generalization. And, a dynamic scheduling from large $\gamma = 0$ to small $\gamma = -1$ (73.83%) slightly exceeds the performance of a static $\gamma = -1$ (73.39%). This result indicates that we can remove the need for prior knowledge of the regime to design such a decaying schedule.

Table E.1: Dynamic schedule for γ (ResNet-56 on CIFAR-100, average over three runs).

γ_{init}	-1	0	1	1	1	-1
γ_{end}	-1	-1	1	0	-1	1
Test accuracy(%)	73.39	**73.83**	72.91	73.21	73.54	73.18

Full-parameter fine-tuning. We also conducted full-parameter fine-tuning experiments on CIFAR-10/100 using ViT-B/16 pretrained on ImageNet-1K. On CIFAR-10, CASAM obtains 98.45%, outperforming SGD and SAM by 0.37 and 0.22 percentage points, respectively. On the more challenging CIFAR-100 dataset, CASAM reaches 89.57%, improving upon SGD by 1.04 percentage points and standard SAM by 0.38 percentage points. These results indicate that CASAM can be effectively extended from training models from scratch to full-parameter fine-tuning of pretrained transformers.

Integration with other SAM variants. Since CASAM only modifies the objective function of the descent step, it can be easily integrated with other SAM variants. In the experiment on CIFAR-100 with ResNet-56, we can observe that CASAM is not a replacement but a complement to other SAM variants. Table E.3 shows that both SAM variants generalize better after integrating CASAM into them.

Table E.2: Fine-tuning results on CIFAR-10/100 using ViT-B/16 pretrained on ImageNet-1K.

	SGD	SAM	ASAM	VaSSO	GA-SAM	LE-SAM	CASAM
CIFAR-10	98.08	98.23	98.32	98.33	98.06	98.39	**98.45**
CIFAR-100	88.53	89.19	89.29	89.26	89.21	89.36	**89.57**

Table E.3: Perfomance of combining CASAM with other SAM variants.

	Test accuracy (%)		Test accuracy (%)
ASAM	73.36	ASAM+CASAM	**74.94**
VASSO	74.73	VASSO+CASAM	**75.22**
GASAM	74.18	GASAM+CASAM	**74.99**
EigenSAM	74.42	EigenSAM+CASAM	**75.16**

Limitations. We highlight that while CASAM excels at suppressing non-systematic noise and memorization, applying it to severely long-tailed/class-imbalanced datasets requires careful calibration of the parameter γ. For example, in long-tailed or minority-class distributions, rare or hard-but-clean examples may also produce gradients that diverge from the majority mini-batch trajectory. Because minority-class gradients contribute less to the overall mini-batch average, CASAM with $\gamma < 0$ could inadvertently downweight these minority-class or fine-grained feature updates during early training, potentially slowing down their representation learning relative to dominant classes. In this case, letting $\gamma > 0$ may be a better choice if our primary goal is to learn the minority.

F Missing Definitions and Proofs

Definition F.1 (Regular Exponential Family)**.** *A loss function is said to come from the regular exponential family if it satisfies the following conditions:*

1. *(NLL-Representability): The loss is the negative log-likelihood of a probability density $p(y|x, \boldsymbol{\omega})$: $f_{\boldsymbol{\omega}}(x, y) = -\ln p(y|x, \boldsymbol{\omega})$.*
2. *(Regularity of Support): The support Y of $p(y|x, \boldsymbol{\omega})$ is independent of $\boldsymbol{\omega}$, ensuring that $\nabla \int p(y)dy = \int \nabla p(y)dy$.*
3. *(Smoothness): $f_{\boldsymbol{\omega}}(x, y)$ is at least C^2 continuous with respect to $\boldsymbol{\omega}$, allowing the existence of Hessian.*

Common loss functions such as MSE and cross-entropy loss are members of this family. The reason why we limit our discussion to this family is that we can use the equivalence between the Hessian and the Fisher information matrix in our argument. To prove Theorem 3.2, we first need an auxiliary lemma as follows.

Lemma F.2. *Let $p(y|x, \boldsymbol{\omega})$ be a conditional probability density function parameterized by the vector $\boldsymbol{\omega} \in \mathbb{R}^d$. Then, for arbitrary small perturbation $\boldsymbol{\varepsilon}$, it follows that*

$$D_{KL}(p(y|x, \boldsymbol{\omega} + \boldsymbol{\varepsilon})||p(y|x, \boldsymbol{\omega})) \approxeq \frac{1}{2}\boldsymbol{\varepsilon}^T \mathcal{I}(\boldsymbol{\omega})\boldsymbol{\varepsilon}, \tag{F.1}$$

where $\mathcal{I}(\boldsymbol{\omega})$ is the Fisher information matrix,

$$\mathcal{I}(\boldsymbol{\omega}) = \mathbb{E}_{(x,y)\sim p(x,y|\boldsymbol{\omega})}\left[\{\nabla_{\boldsymbol{\omega}} \log p(y|x, \boldsymbol{\omega})\}\{\nabla_{\boldsymbol{\omega}} \log p(y|x, \boldsymbol{\omega})\}^T\right].$$

Proof. Consider the KL-divergence as a function $h(\boldsymbol{\varepsilon})$ with respect to the displacement $\boldsymbol{\varepsilon}$. Our purpose is to approximate $h(\boldsymbol{\varepsilon})$ near $\boldsymbol{\varepsilon} = \mathbf{0}$. According to the Taylor expansion,

$$h(\boldsymbol{\varepsilon}) \approxeq h(\mathbf{0}) + \nabla h(\mathbf{0})^T \boldsymbol{\varepsilon} + \frac{1}{2}\boldsymbol{\varepsilon}^T \mathcal{J}_h(\mathbf{0})\boldsymbol{\varepsilon}, \tag{F.2}$$

where $\nabla h(\mathbf{0})$ is the gradient vector and $\mathcal{J}_h(\mathbf{0})$ is the Hessian matrix evaluated at $\boldsymbol{\varepsilon} = \mathbf{0}$. First, note that $h(\mathbf{0}) = 0$. Then, we calculate the gradient of $h(\boldsymbol{\varepsilon})$ with respect to $\boldsymbol{\varepsilon}$:

$$\nabla h(\boldsymbol{\varepsilon}) = \int \nabla_{\boldsymbol{\varepsilon}} \left[p(y|x, \boldsymbol{\omega} + \boldsymbol{\varepsilon}) \ln p(y|x, \boldsymbol{\omega} + \boldsymbol{\varepsilon}) \right] \mathrm{d}y - \int \nabla_{\boldsymbol{\varepsilon}} p(y|x, \boldsymbol{\omega} + \boldsymbol{\varepsilon}) \ln p(y|x, \boldsymbol{\omega}) \mathrm{d}y \tag{F.3}$$

$$= \int \nabla_{\boldsymbol{\varepsilon}} p(y|x, \boldsymbol{\omega} + \boldsymbol{\varepsilon}) \left[\ln p(y|x, \boldsymbol{\omega} + \boldsymbol{\varepsilon}) - \ln p(y|x, \boldsymbol{\omega}) \right] \mathrm{d}y + \int \nabla_{\boldsymbol{\varepsilon}} p(y|x, \boldsymbol{\omega} + \boldsymbol{\varepsilon}) \mathrm{d}y \tag{F.4}$$

$$= \int \nabla_{\boldsymbol{\varepsilon}} p(y|x, \boldsymbol{\omega} + \boldsymbol{\varepsilon}) \left[\ln p(y|x, \boldsymbol{\omega} + \boldsymbol{\varepsilon}) - \ln p(y|x, \boldsymbol{\omega}) \right] \mathrm{d}y. \tag{F.5}$$

Thus, $\nabla h(\mathbf{0}) = \mathbf{0}$. Finally, we continue to differentiate $\nabla h(\boldsymbol{\varepsilon})$ to find the Hessian:

$$\mathcal{J}_h(\boldsymbol{\varepsilon}) = \int \nabla^2_{\boldsymbol{\varepsilon}} p(y|x, \boldsymbol{\omega} + \boldsymbol{\varepsilon}) \left[\ln p(y|x, \boldsymbol{\omega} + \boldsymbol{\varepsilon}) - \ln p(y|x, \boldsymbol{\omega}) \right] \mathrm{d}y + \tag{F.6}$$

$$\int \nabla_{\boldsymbol{\varepsilon}} p(y|x, \boldsymbol{\omega} + \boldsymbol{\varepsilon}) \nabla_{\boldsymbol{\varepsilon}} \left[\ln p(y|x, \boldsymbol{\omega} + \boldsymbol{\varepsilon}) - \ln p(y|x, \boldsymbol{\omega}) \right]^T \mathrm{d}y. \tag{F.7}$$

When $\boldsymbol{\varepsilon} = \mathbf{0}$, the first term is zero. And $\ln p(y|x, \boldsymbol{\omega})$ does not depend on $\boldsymbol{\varepsilon}$, yielding

$$\mathcal{J}_h(\mathbf{0}) = \int \nabla_{\boldsymbol{\varepsilon}} p(y|x, \boldsymbol{\omega}) \nabla_{\boldsymbol{\varepsilon}} \ln p(y|x, \boldsymbol{\omega})^T \mathrm{d}y \tag{F.8}$$

$$= \int p(y|x, \boldsymbol{\omega}) \nabla_{\boldsymbol{\varepsilon}} \ln p(y|x, \boldsymbol{\omega}) \nabla_{\boldsymbol{\varepsilon}} \ln p(y|x, \boldsymbol{\omega})^T \mathrm{d}y, \tag{F.9}$$

which is exactly the Fisher information matrix. Substituting the terms back into the Taylor expansion concludes the proof. □

Based on Lemma F.2, we are now ready to prove Theorem 3.2.

Proof of Theorem 3.2. Recall that $\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{\varepsilon}^*$, where

$$\boldsymbol{\varepsilon}^* = \rho \frac{\nabla F_{\mathcal{B}}(\boldsymbol{\omega})}{\|\nabla F_{\mathcal{B}}(\boldsymbol{\omega})\|_2}.$$

Obviously, $\|\boldsymbol{\varepsilon}^*\|_2 = \rho$. When ρ is small, for any $z_i \in \mathcal{B}$ it holds that

$$f_{\tilde{\boldsymbol{\omega}}}(z_i) = f_{\boldsymbol{\omega}}(z_i) + \nabla f^T_{\boldsymbol{\omega}}(z_i) \boldsymbol{\varepsilon}^* + \frac{1}{2} \boldsymbol{\varepsilon}^{*T} \nabla^2 f_{\boldsymbol{\omega}}(z_i) \boldsymbol{\varepsilon}^* + o(\rho^2). \tag{F.10}$$

Average over $\mathcal{B}$, yielding

$$F_{\mathcal{B}}(\tilde{\boldsymbol{\omega}}) = F_{\mathcal{B}}(\boldsymbol{\omega}) + \rho \|\nabla F_{\mathcal{B}}(\boldsymbol{\omega})\|_2 + \frac{1}{2} \boldsymbol{\varepsilon}^{*T} \left[\frac{1}{|\mathcal{B}|} \sum_{z_i \in \mathcal{B}} \nabla^2 f_{\boldsymbol{\omega}}(z_i) \right] \boldsymbol{\varepsilon}^* + o(\rho^2). \tag{F.11}$$

In the context of cross-entropy loss, denote the Hessian $\mathcal{J}(\boldsymbol{\omega})$ by

$$\mathcal{J}(\boldsymbol{\omega}) = \mathbb{E}_{(x,y) \sim p(x,y|\boldsymbol{\omega})} \left[-\nabla^2_{\boldsymbol{\omega}} \log p(y|x, \boldsymbol{\omega}) \right]$$

Therefore,

$$\hat{\mathcal{J}}(\boldsymbol{\omega}) = \frac{1}{|\mathcal{B}|} \sum_{z_i \in \mathcal{B}} \nabla^2 f_{\boldsymbol{\omega}}(z_i)$$

is an estimate of $\mathcal{J}(\boldsymbol{\omega})$. Actually, we can prove that $\mathcal{J}(\boldsymbol{\omega})$ equals the Fisher information matrix $\mathcal{I}(\boldsymbol{\omega})$. Following the chain rule, we obtain

$$-\nabla^2_{\boldsymbol{\omega}} \log p(y|x, \boldsymbol{\omega}) = -\frac{\nabla^2_{\boldsymbol{\omega}} p(y|x, \boldsymbol{\omega})}{p(y|x, \boldsymbol{\omega})} + \{\nabla_{\boldsymbol{\omega}} \log p(y|x, \boldsymbol{\omega})\} \{\nabla_{\boldsymbol{\omega}} \log p(y|x, \boldsymbol{\omega})\}^T. \tag{F.12}$$

Taking expectation with respect to $p(x, y|\boldsymbol{\omega})$, it follows that

$$\mathcal{J}(\boldsymbol{\omega}) = \mathcal{I}(\boldsymbol{\omega}) - \mathbb{E}_{(x,y)\sim p(x,y|\boldsymbol{\omega})}\left[\frac{\nabla^2_{\boldsymbol{\omega}} p(y|x,\boldsymbol{\omega})}{p(y|x,\boldsymbol{\omega})}\right] \tag{F.13}$$

$$= \mathcal{I}(\boldsymbol{\omega}) - \mathbb{E}_{x\sim p(x)}\mathbb{E}_{y\sim p(y|x,\boldsymbol{\omega})}\left[\frac{\nabla^2_{\boldsymbol{\omega}} p(y|x,\boldsymbol{\omega})}{p(y|x,\boldsymbol{\omega})}\right] \tag{F.14}$$

$$= \mathcal{I}(\boldsymbol{\omega}) - \mathbb{E}_{x\sim p(x)}\left[\sum_y \frac{\nabla^2_{\boldsymbol{\omega}} p(y|x,\boldsymbol{\omega})}{p(y|x,\boldsymbol{\omega})} \cdot p(y|x,\boldsymbol{\omega})\right] \tag{F.15}$$

$$= \mathcal{I}(\boldsymbol{\omega}), \tag{F.16}$$

where the last equality is due to the sum of $p(y|x, \boldsymbol{\omega})$ over y is 1. Thus, from Equation (F.11), we obtain

$$F_{\mathcal{B}}(\tilde{\boldsymbol{\omega}}) \approxeq F_{\mathcal{B}}(\boldsymbol{\omega}) + \rho\|\nabla F_{\mathcal{B}}(\boldsymbol{\omega})\|_2 + \frac{1}{2}{\boldsymbol{\varepsilon}^*}^T \mathcal{J}(\boldsymbol{\omega})\boldsymbol{\varepsilon}^* \tag{F.17}$$

$$= F_{\mathcal{B}}(\boldsymbol{\omega}) + \rho\|\nabla F_{\mathcal{B}}(\boldsymbol{\omega})\|_2 + \frac{1}{2}{\boldsymbol{\varepsilon}^*}^T \mathcal{I}(\boldsymbol{\omega})\boldsymbol{\varepsilon}^*. \tag{F.18}$$

Lemma F.2 further implies that

$$\hat{D}_{KL}(p(y|x,\tilde{\boldsymbol{\omega}})||p(y|x,\boldsymbol{\omega})) \approxeq \frac{1}{2}{\boldsymbol{\varepsilon}^*}^T \mathcal{I}(\boldsymbol{\omega})\boldsymbol{\varepsilon}^*, \tag{F.19}$$

thus concluding the proof. □

Proof of Theorem 3.5. Write $x = \mathbf{p}_y$ and $y = \tilde{\mathbf{p}}_y$ for notational simplicity. As a result, it yields that

$$D_{KL}(y||x) - \log x = -\log y + \log \cdot (\frac{y}{x})^{y+1} \cdot (\frac{1-y}{1-x})^{1-y} = -\log y + \xi(x, y). \tag{F.20}$$

Consider fixing x and calculating the partial derivative with respect to y, it leads to

$$\frac{\partial \xi(x,y)}{\partial y} = \log\frac{y}{1-y} - \log\frac{x}{1-x} + \frac{1}{y} > 0. \tag{F.21}$$

Therefore, the function is monotonically increasing with respect to y. Namely, $\mathcal{C}(z)$ decreases when y approaches 1 from below. □

Proof of Theorem 3.7. Following Theorem 3.5, we aim to verify whether $\xi(x, y) \leq 0$ holds when $0 < y \leq x < 1$.

First, note that $\xi(x, y) = 0$ when $x = y$. This is trivial, and we actually want to know how this function performs under this line. Fixing x, we now derive the partial derivative of $\xi(x, y)$ with respect to y,

$$\frac{\partial \xi(x,y)}{\partial y} = \log\frac{y}{1-y} - \log\frac{x}{1-x} + \frac{1}{y}. \tag{F.22}$$

In order to analyze the sign of $\frac{\partial \xi(x,y)}{\partial y}$, we continue to derive its second derivative with respect to y,

$$\frac{\partial^2 \xi(x,y)}{\partial y^2} = \frac{2y-1}{y^2(1-y)}. \tag{F.23}$$

This implies that $\frac{\partial \xi(x,y)}{\partial y}$ has a global minimum at $y = 1/2$. If the minimum is not smaller than zero, then it indicates that $\xi(x, y)$ is non-decreasing and, as a result, $\xi(x, y) \leq \xi(x, x) = 0$ for all $y \leq x$. Substituting $y = 1/2$, we have

$$\frac{\partial \xi(x,y)}{\partial y}|_{y=1/2} = 2 - \log\frac{x}{1-x} \geq 0, \tag{F.24}$$

if and only if $0 < x \leq \frac{e^2}{e^2+1}$. □

Proof of Lemma 3.9. According to the Taylor expansion, there always exists some $\hat{\boldsymbol{\omega}}$ such that

$$f_{\tilde{\boldsymbol{\omega}}}(z) = f_{\boldsymbol{\omega}}(z) + (\tilde{\boldsymbol{\omega}} - \boldsymbol{\omega})^T \nabla f_{\boldsymbol{\omega}}(z) + \frac{1}{2}(\tilde{\boldsymbol{\omega}} - \boldsymbol{\omega})^T \nabla^2 f_{\hat{\boldsymbol{\omega}}}(z)(\tilde{\boldsymbol{\omega}} - \boldsymbol{\omega}).$$

Since $\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \rho \nabla F_{\mathcal{B}}(\boldsymbol{\omega}) / \|\nabla F_{\mathcal{B}}(\boldsymbol{\omega})\|$, it follows that

$$\begin{aligned} -\log \tilde{\mathbf{p}}_y &= -\log \mathbf{p}_y + \rho \nabla f_{\boldsymbol{\omega}}(z)^T \frac{\nabla F_{\mathcal{B}}(\boldsymbol{\omega})}{\|\nabla F_{\mathcal{B}}(\boldsymbol{\omega})\|_2} + \frac{1}{2}(\tilde{\boldsymbol{\omega}} - \boldsymbol{\omega})^T \nabla^2 f_{\hat{\boldsymbol{\omega}}}(z)(\tilde{\boldsymbol{\omega}} - \boldsymbol{\omega}) \\ &\leq -\log \mathbf{p}_y + \rho G + \rho^2 L/2, \end{aligned}$$

thus concluding the proof. □

Proof of Theorem 3.10. Similar to the proof of Theorem 3.7, we write $x = \mathbf{p}_y$ and $y = \tilde{\mathbf{p}}_y$ for simplicity. According to Lemma 3.9, it follows that $y \geq e^{-\rho(G+\rho L/2)} x$. As such, our interest now is to check whether $\xi(x, y) \leq 0$ for $y \in [e^{-\rho(G+\rho L/2)} x, x)$.

Recall that $\frac{\partial \xi(x,y)}{\partial y}$ is increasing for $y > 1/2$. Hence, if we can assure that $\frac{\partial \xi(x,y)}{\partial y} \geq 0$ at $y = e^{-\rho(G+\rho L/2)} x$, then the claim consequently holds. Since

$$\begin{aligned} \frac{\partial \xi(x, y)}{\partial y} = \log \frac{y}{1-y} - \log \frac{x}{1-x} + \frac{1}{y} &\geq \log(1 + \frac{1}{y}) \cdot \frac{y}{1-y} \cdot \frac{x}{1-x} && \text{(F.25)} \\ &= \log \frac{1+y}{1-y} \cdot \frac{x}{1-x} && \text{(F.26)} \\ &\geq 0, && \text{(F.27)} \end{aligned}$$

if $y \geq 2x - 1$. Thus, by letting $e^{-\rho(G+\rho L/2)} x \geq 2x - 1$, namely, $\rho\,(G + \rho L/2) \leq \log \frac{x}{2x-1}$, we have that $\frac{\partial \xi(x,y)}{\partial y} \geq 0$ and thus concluding the proof. □

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: Our main claim is supported by the analysis in Section 3, and Section 4 offers detailed experiments.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: We include the limitation of our work in Section 5.

   Guidelines:

   - The answer [N/A] 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.

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 the assumptions in Section 3, and all the proofs are supplied in Appendix.

Guidelines:

- The answer [N/A] 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: We provide the hyperparameter settings of our method in Section 4.

   Guidelines:

   - The answer [N/A] 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.

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 the datasets are publicly available, and the code is released.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: Necessary details are provided in Appendix.

   Guidelines:

   - The answer [N/A] 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 show the standard error in most tables and plots, with an average of over three random seeds.

   Guidelines:

   - The answer [N/A] 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.
   - 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: The detail can be found in Appendix.

   Guidelines:

   - The answer [N/A] 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: We have read the code of ethics and have done our best to conform them.

   Guidelines:

   - The answer [N/A] 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: [N/A]

    Justification: Our work only involves small models and toy datasets. It does no impact the society at large, beyond deepening our understanding of certain aspects of deep learning.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We only train small models on toy-datasets, our work poses no risk of misuse.

    Guidelines:

    - The answer [N/A] 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: Both data and code are properly referenced.

    Guidelines:

    - The answer [N/A] 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.
    - 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: We will release our code base with included documentation files.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] 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.
    - 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.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We use LLM models only for writing and formatting.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.