UNDERSTANDING SAM UNDER LABEL NOISE

Anonymous authors
Paper under double-blind review

ABSTRACT

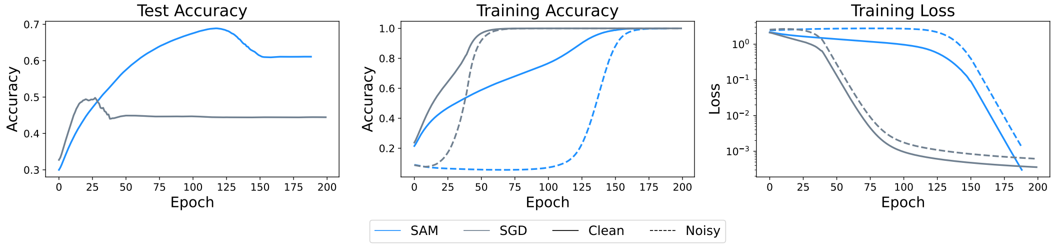
Sharpness-Aware Minimization (SAM) is most known for achieving state-of-the-art performances on natural image and language tasks. However, its most pronounced improvements (of tens of percent) is rather in the presence of label noise. Understanding SAM's label noise robustness requires a departure from characterizing the robustness of minimas lying in "flatter" regions of the loss landscape. In particular, the peak performance under label noise occurs with early stopping, far before the loss converges. We decompose SAM's robustness into two effects: one induced by changes to the logit term and the other induced by changes to the network Jacobian. The first can be observed in linear logistic regression where SAM provably upweights the gradient contribution from clean examples. Although this explicit upweighting is also observable in neural networks, when we intervene and modify SAM to remove this effect, surprisingly, we see no visible degradation in performance. We infer that SAM's effect in deeper networks is instead explained entirely by the effect SAM has on the network Jacobian. We theoretically derive the explicit regularization induced by this Jacobian effect in two layer linear networks. Motivated by our analysis, we see that cheaper alternatives to SAM that explicitly induce these regularization effects largely recover the benefits even in deep networks trained on real-world datasets.

1 INTRODUCTION

In recent years, there has been growing excitement about improving the generalization of deep networks by regularizing the sharpess of the loss landscape. Among optimizers that explicitly minimize sharpness, Sharpness Aware Minimization (SAM) (Foret et al., 2020) garnered popularity for achieving state-of-the-art performance on various natural image and language benchmarks. Compared to stochastic gradient descent (SGD), SAM provides consistent improvements of several percentage points. Interestingly, a less widely known finding from Foret et al. (2020) is that SAM's most prominent gains lie elsewhere, in the presence of synthetic label noise. In fact, SAM is more robust to label noise than SGD by tens of percentage points, rivaling the current best label noise robustness techniques (Jiang et al.; Zhang et al., 2017; Arazo et al., 2019; Liu et al., 2022).

In Figure 1, we demonstrate this finding in CIFAR10 with 30% label noise, where SAM's best test accuracy is 20% higher. Additionally, we find that the robustness gains are most obvious in a particular version of SAM called 1-SAM which applies the perturbation step to each sample in the minibatch separately. In this work, we investigate the behavior of 1-SAM before overfitting ("early learning regime") at a more mechanistic level. Decomposing the gradient of each example ("sample-wise" gradient) by chain rule into $\nabla_w \ell(f(w, x), y) = \partial \ell / \partial f \cdot \nabla_w f$, we analyze the effect of SAM's perturbation on the terms $\partial \ell / \partial f$ ("logit scale") and $\nabla_w f$ ("network Jacobian"), separately. We make the following key conclusions about how these components slow down overfitting and consequently, improve early-stopping test accuracy.

In the linear setting, we show the SAM's logit scale reduces to a re-weighting scheme that explicitly up-weights the gradient contribution of low loss points. Previous works show that when training with gradient descent, clean examples initially dominate the direction of the update and as a result, their corresponding loss initially decreases first before that of noisy examples (Liu et al., 2020; 2023) (See Figure 2). We show that similar to many existing label-noise techniques (Liu et al., 2020), SAM's explicit up-weighting keeps the gradient contribution of low-loss examples (which mostly consist of clean examples) large even after they are fit, thus slowing down overfitting.



Addresing Reviewer ATLV:

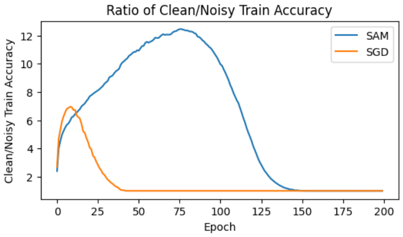


Figure 1: Trends in training loss of clean vs noisy data. We see that in both SGD and SAM, SAM fits much more clean data before overfitting to the noisy data. In particular, the ratio of clean training accuracy over noisy training accuracy peaks to a noticibly higher value with SAM.

Our experiments show that even in deep networks (e.g. ResNet18), SAM's logit scale term induces a similar effect of explicit up-weighting the gradients of low loss examples. However, we find that for range of SAM's perturbation utilized in practice, SAM's logit scale for deep networks has negligible impact on SAM's robustness. On the other hand, just keeping SAM's perturbation on the network Jacobian term *retains nearly identical performance* to SAM. This suggests that there is a fundamentally different mechanism that originates in SAM's Jacobian term that results in most of SAM's label noise robustness in the nonlinear case.

Motivated by this finding, we explicitly analyze the Jacobian term of SAM for a 2-layer linear network, and show that the resulting update decomposes into a combination of ℓ_2 regularization on the final layer weights and intermediate activations. Further, we show that including just these two terms for deep networks, while not achieving the full benefits of SAM, nonetheless substantially improves the performance of SGD under label noise. We emphasize that these methods are meant to be illustrative, and not intended to be a standalone method for label noise robustness. But in this vein, the findings suggest that one of the main mechanisms of SAM's label robustness may not come explicitly via its smoothness properties, but rather from the implicit regularization terms in its update.

2 Preliminaries

2.1 Problem Setup

Model We consider binary classification. The sign of the model's output $f : \mathbb{R}^d \to \mathbb{R}$ maps inputs $x \in \mathcal{X}$ to discrete labels $t \in \{-1, 1\}$. We will study two kinds of models – a linear model and a 2-layer deep linear network (DLN) in this work. We do not include the bias term.

$$\begin{aligned}&\textbf{Linear: } f(w, x) = \langle w, x \rangle \\ &\textbf{DLN: } f(v, W, x) = \langle v, Wx \rangle \text{ where } W \in \mathbb{R}^{d \times h}, v \in \mathbb{R}^h,\end{aligned} \tag{2.1}$$

where h denotes the dimension of the hidden layer. Let us denote the intermediate activation as $z = Wx$. We will abuse the notation to also refer to a generic parameterized model as $f(w, x)$ when clear from context.

Objective We consider the logistic loss $\ell(w, x, t) = -\log(\sigma(t \cdot f(w, x)))$ for sigmoid function $\sigma(z) = \frac{1}{1+\exp(-z)}$. Given n training points $[(x_i, t_i)]_{i=1}^n$ sampled from the data distribution $\mathcal{D}$, our training objective is

$$\min_{w} L(w) \text{ where } L(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(w, x_i, t_i) \tag{2.2}$$

By chain rule, we can write the sample-wise gradient with respect to the logistic loss $\ell(w, x, t)$ as

$$-\nabla_w \ell(x, t) = t \cdot \underbrace{\sigma(-tf(w, x))}_{\text{logit scale}} \underbrace{\nabla_w f(w, x)}_{\text{Jacobian}} \tag{2.3}$$

The logit term, which is the model's confidence that x belongs in the incorrect class $-t$, *scales* the Jacobian term. The logit term grows monotonically with the loss. We refer to the network Jacobian $\nabla_w f(w, x)$ as the "Jacobian term".

Although our mathematical analysis is for binary classification, the cross-entropy loss for multiclass classification observes a similar decomposition for its sample-wise gradient:

$$-\nabla_w \ell(x, y) = \langle \underbrace{e_t - \sigma(f(w + \epsilon_i, x))}_{\text{logit scale}}, \underbrace{\nabla_w f(w, x)}_{\text{Jacobian}} \rangle \tag{2.4}$$

where $\sigma(\cdot)$ is the softmax function and e_t is the one-hot encoding of the target label. Empirically, we will observe that the conclusions from our binary analysis empirically transfer to multi-class.

2.2 Sharpness Aware Minimization

Sharpness-aware Minimization (SAM) (Foret et al., 2020) attempts to find a flat minima of the training objective (Eq. 2.2) by minimizing the following objective

$$\min_{w} \max_{\|\epsilon\|_2 \leq \rho} L(w + \epsilon), \tag{2.5}$$

where ρ is the magnitude of the adversarial weight perturbation ϵ. The objective tries to find a solution that lies in a region where the loss does not fluctuate dramatically with any ϵ-perturbation. SAM approximates ϵ by first-order Taylor approximation of the loss to precisely be the normalized gradient $\rho \nabla_w L(w) / \|\nabla_w L(w)\|$.

1-SAM However, the naive SAM update that computes a single ϵ does not observe performance gains over SGD in practice unless paired with a small batch size (Foret et al., 2020; Andriushchenko & Flammarion, 2022a). Alternatively Foret et al. (2020) propose sharding the minibatch and calculating SAM's adversarial perturbation ϵ for each shard separately. At the end of this extreme is 1-SAM which computes ϵ for the loss of each example in the minibatch individually. Formally, this can be written as

$$w = w - \eta \left(\frac{1}{n} \sum_{i=1}^{n} \nabla_{w+\epsilon_i} \ell\left(w + \epsilon_i, x_i, t_i\right) \right) \text{ where } \epsilon_i = \rho \frac{\nabla_w \ell(x_i, t_i)}{\|\nabla_w \ell(x_i, t_i)\|_2} \tag{2.6}$$

In practice, 1-SAM is the version of SAM that achieves the most performance gain. In this paper, we focus on understanding 1-SAM and will use SAM to refer to 1-SAM unless explicitly stated otherwise.

2.3 Hybrid 1-SAM

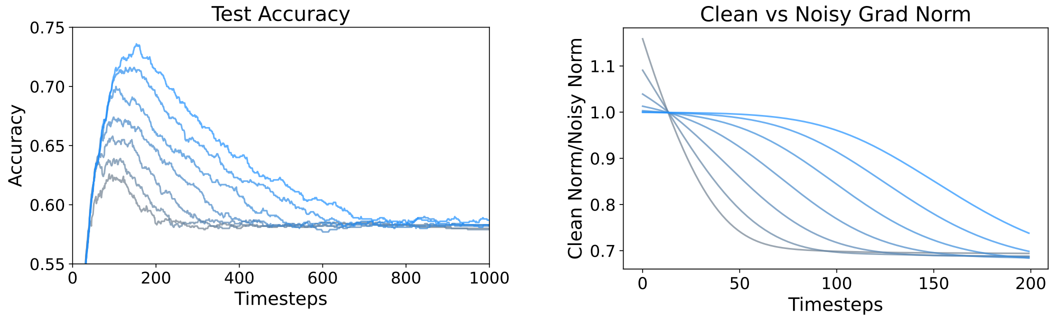
To understand the effects of 1-SAM, we try to isolate the robustness effects of 1-SAM coming from how the perturbation ϵ_i affects the logit scaling term and the Jacobian term of each sample-wise update. To do so, we will also pay attention to the following variants of 1-SAM:

$$\text{1-SAM}: \; \nabla_{w+\epsilon_i} \ell(w + \epsilon_i, x_i, t_i) = t\sigma(-tf(w + \epsilon_i , x))\nabla_{w+\epsilon_i} f(w + \epsilon_i , x)) \tag{2.7}$$

$$\text{Logit SAM}: \; \Delta^{\text{L-SAM}} \ell(w, x_i, t_i) = t\sigma(-tf(w + \epsilon_i , x))\nabla_w f(w , x_i) \tag{2.8}$$

$$\text{Jacobian SAM}: \; \Delta^{\text{J-SAM}} \ell(w, x_i, t_i) = t\sigma(-tf(w , x_i))\nabla_{w+\epsilon_i} f(w + \epsilon_i , x_i) \tag{2.9}$$

Logit SAM (L-SAM) only applies the SAM perturbation to the logit term for each sample-wise gradient while Jacobian SAM (J-SAM) only applies the SAM perturbation to the Jacobian term. We observe that in deep networks, J-SAM observes close to the same performance as 1-SAM while L-SAM does not.



(a) SAM test accuracy for $\rho \in [0, 0.18]$. Bluer curves denote larger ρ. Accuracy improves with larger ρ.

(b) Ratio of the average norm of the sample-wise gradient of clean versus noisy points

Figure 2: Linear models trained on the toy Gaussian data using SAM. SAM with higher ρ observe a slower decay in the gradient norm of clean points versus noisy points (Right) due to the preferential upweighting of low loss points, and this corresponds with higher early stopping test accuracy (Left).

3 Linear models: SAM up-weights the gradients of low loss examples

We first study the robustness affects of the logit scale term in linear models. Not only do we empirically observe that SAM's boost in early-stopping performance can be observed in linear models, the Jacobian term $\nabla_w f(w, x)$ for a linear model is equal to x and is independent of the weights. Thus, any robustness obtained by SAM in this setting is specifically due to SAM's logit scale.

3.1 Experimental investigation into SAM effect in linear models

We train our linear models over the following toy Gaussian data distribution $\mathcal{D} : \mathbb{R}^d \times \{-1, 1\}$.

$$\begin{gathered}\text{True Label: } y \sim \{-1, 1\} \text{ by flipping fair coin} \\ \text{Input: } x \sim y \cdot \left[B \in \mathbb{R}, z \sim \mathcal{N}\left(0, \frac{\gamma^2}{d-1} I_{d-1 \times d-1}\right)\right] \\ \text{Target: } t = y \cdot \varepsilon \text{ where } \varepsilon \sim \{-1 \text{ w.p } \Delta,\ 1 \text{ w.p } (1-\Delta)\}\end{gathered} \tag{3.1}$$

In x, the first dimension contains the true signal yB while the remaining $d - 1$ dimensions is uncorrelated Gaussian noise. We sample training datapoints (x, t) from this distribution where the random label noise ε corrupts Δ of the training data. We expect $\Delta < 0.5$ meaning the *majority* of the data is still clean. In the test data, we assume there are no mislabeled examples ($t = y$).

In Figure 2, we compare the performance of full-batch gradient descent and SAM on our toy Gaussian data with 40% label noise (See Appendix A for experimental details). Even in this simple setting SAM observes noticeably higher early stopping test accuracy over SGD. We run a grid search for ρ between 0 and 0.18 and observe that the early stopping test accuracy monotonically increases with ρ. In particular, for this toy data distribution, we will observe that performance saturates as ρ goes to infinity.

What causes SAM to outperform SGD? Correlated with this improved performance, when we plot the average norm of the sample-wise gradient of clean examples versus noisy examples along training, we observe that this ratio decays slower with larger ρ. This leads us to suspect that SAM's superior test performance is due to the faster rate at which clean training examples are learned versus noisy examples.

In particular, as shown by Liu et al. (2020) about gradient descent training in linear models, at the beginning of training, the clean majority and the noisy outliers have similar loss values, thus the gradient is more correlated with fitting the clean points. However, the dynamic quickly changes as training progresses. The loss of clean points begins to decrease quickly (at an exponential rate for cross-entropy) and thus the gradient contribution of noisy outliers begins to outweigh that of the clean examples. This manifests in a suboptimal test accuracy of the classifier.

In the next section, we will prove that SAM (by virtue of its adversarial weight perturbation) *preferentially* up-weights the gradient signal from low-loss points, therefore allowing the gradient signal from correct points to decay slower in the early training epochs. This allows SAM to learn a better classifier by initially learning more from the clean points rather than mislabeled data.

3.2 Analysis of SAM's logit scale

Recall that 1-SAM evaluates the gradient of each datapoint (x_i, t_i) at the weight perturbed by the corresponding normalized sample-wise gradient. In binary classification, the normalization factor causes the logit term to vanish leaving just the Jacobian term. In the linear model, the network Jacobian is the datapoint itself $\nabla_w f(w, x) = x$ irrespective of the weight value. Thus, the normalized gradient is simply the datapoint scaled by the label $\epsilon_i = -t_i \frac{x_i}{\|x_i\|_2}$ and SAM's update reduces to a *constant adjustment* of the logit scale by the norm of the datapoint

$$-\nabla_{w+\epsilon_i}\ell(w+\epsilon_i, x_i, y_i) = t_i\sigma(-t_i < w, x_i > + \underbrace{\rho\|x_i\|_2}_{\substack{\text{constant}\\ \text{adjustment}}})x_i. \tag{3.2}$$

Since the sigmoid function σ is monotonically increasing, the constant addition causes the gradient contribution of *all* training points to up-weighted. However, among points of the same norm $\|x_i\|$, this adjustment causes points where the confidence towards the incorrect class $\sigma(-t_i\langle w, x_i\rangle)$ is low (i.e. low loss points) to be up-weighted *by a larger coefficient* than those where the incorrect class confidence is high (i.e. high loss), as proven in the following lemma.

Lemma 3.1 (Preferential up-weighting of low loss points) *Consider the following function.*

$$f(z) = \frac{\sigma(-z+C)}{\sigma(-z)} = \frac{1+\exp(z)}{1+\exp(z-C)} \tag{3.3}$$

This function is strictly increasing if $C > 0$.

Proof Looking at the gradient

$$\frac{df}{dz} = \frac{\exp(z)}{1+\exp(z-C)} - \frac{(1+\exp(z))\exp(z-C)}{(1+\exp(z-C))^2} \tag{3.4}$$

$$= \frac{\exp(z)(1-\exp(-C)}{(1+\exp(z-C))^2} > 0 \tag{3.5}$$

■

Consider a particular datapoint x_i. We are interested in how much 1-SAM's sample-wise update up-weights the magnitude of its gradient in comparison to SGD. Note that

$$\frac{\|\nabla_{w+\epsilon_i}\ell(w+\epsilon_i, x_i, t_i)\|}{\|\nabla_w\ell(w, x_i, t_i)\|} = \frac{\sigma(-t_i\langle w, x_i\rangle + \rho\|x_i\|)}{\sigma(-t_i\langle w, x_i\rangle)}. \tag{3.6}$$

We can apply Lemma 3.1 directly by setting $z = t_i\langle w, x_i\rangle$ and $C = \rho\|x_i\|_2$ to show that across points of the same magnitude, low loss points are more affected by the up-weighting than high loss points. Since the loss is initially lower for correct points over incorrect points, SAM preferentially up-weights the gradients of correct points. The fact that the resulting early stopping test accuracy of SAM is higher than gradient descent follows by induction. In particular, for this toy data distribution, we observe that performance saturates as ρ goes to infinity. In particular, the logit scale for any datapoint converges to 1 and the solution converges to the class means which is the optimal classifier in this setting. Previous literature also observe that the naive SAM update (n-SAM) that computes a single perturbation ϵ utilizing the average gradient does not observe performance gains in practice. We find that this suboptimality is observable even in the linear setting and that n-SAM behaves strictly differently from 1-SAM. We provide further insight about the asymptotic behavior of 1-SAM and the behavior of n-SAM in Appendix B.

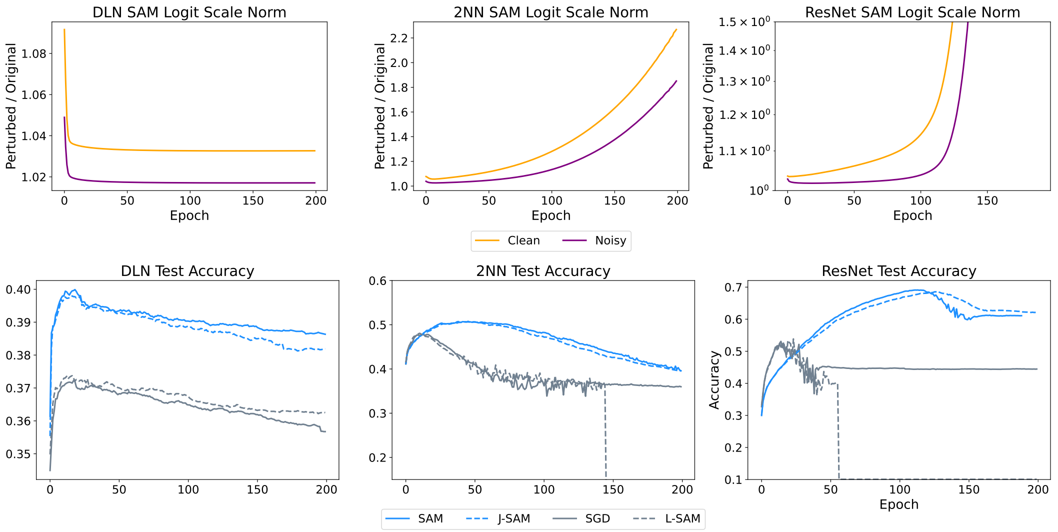


Figure 3: In 2-layer deep linear networks (DLN), 2-layer MLP with ReLU activation (2NN), and ResNet18 trained on noisy CIFAR10, we observe that SAM's perturbation to the logit scale preferentially upweights its norm for clean examples (top row). Yet, even when run J-SAM i.e. SAM is absent of the explicit reweighting effect, SAM's label noise robustness is preserved (bottom row).

4 Neural Networks: SAM robustness comes from the Jacobian

From our linear analysis, we learned that SAM's logit scale term preferentially up-weights the gradients of low-loss points. We verify that this effect also holds true for SAM neural networks. From this analysis, we would suspect that this explicit reweighting alone could potentially recover most of SAM's gains. However, we find that the opposite holds true in neural networks. Just conducting 1-SAM on the Jacobian term (J-SAM) recovers most of 1-SAM's gains while explicit logit reweighting alone (L-SAM) cannot. Exploring the effects of the Jacobian in particular, we find that a large proportion of the gains can be recovered through a simpler method.

4.1 Explicit re-weighting does not fully explain SAM's gains

We first note that the upweighting of low-loss points can be observed even in multi-class classification with neural networks. Recall the general form of the gradient for multi-class cross-entropy loss is $\nabla_w \ell(x,t) = \langle \sigma(f(w,x)) - e_t, \nabla_w f(w,x) \rangle$ where σ is the softmax function and e_t is the one-hot encoding of the label t. Similarly, a logit scale component $g(w,x,t) = \sigma(f(w,x)) - e_t$ emerges whose ℓ_2 norm scales with the loss. In Figure 3, we measure the change in the norm of this quantity for each example x_i in the minibatch after the 1-SAM perturbation, i.e., $\|g(w+\epsilon_i, x_i, t_i)\| / \|g(w, x_i, t_i)\|$. On the ResNet18 models trained on CIFAR10 using SAM, we indeed empirically observe that this quantity is higher in clean examples than noisy examples (Figure 3).

On the other hand, for the same magnitude of perturbation ρ, just the explicit upweighting through L-SAM observes marginal early-stopping performance gains above SGD (Figure 3). Alternatively, J-SAM recovers almost all of the gains of SAM. This suggests that SAM's rescaling of the logit term is not the main contributor to SAM's robustness *in neural networks*. We also find that this observation does not require an arbitrarily deep network but also holds true in simple 2-layer deep linear networks (DLN) and ReLU MLP's (2NN). Empirically, we observe the same trends in test accuracy when we train these models on flattened CIFAR10 images similarly with 30% label noise (Figure 3). A similar analysis of SAM under label noise in linear models was conducted by Andriushchenko & Flammarion (2022b), however they attribute the label noise robustness in neural networks to logit scaling. We claim the opposite: that the *direction* or the network Jacobian of SAM's update becomes much more important.

4.2 Analysis

Motivated by this, we look into the precise nature of "perturbing" the Jacobian for the nonlinear setting. In the linear case, the Jacobian term was constant and had no effect on the update, but in the nonlinear case the Jacobian is a function of the datapoint and weights. Furthermore, we find that this effect is observed even in a simple 2-layer deep linear network $f(x) = \langle v, Wx \rangle$ under binary classification. Here we find that J-SAM simply reduces to regularization on the norm of the intermediate representations and weight decay on the last layer weights, as proven by the following proposition.

Proposition 4.1 *For binary classification in a 2-layer deep linear network, a sample-wise J-SAM update reduces to an L2 regularization on the norm of the intermediate features and weight decay on the last layer weights.*

Proof We write the form of the J-SAM update for the first layer W of the deep linear network:

$$-\nabla_{W+\epsilon^{(1)}}\ell(w+\epsilon,x,t) = \sigma(-tf(w,x))\left(tv - \frac{\rho}{J}z\right)x^\top \tag{4.1}$$

$$= -\nabla_W \ell(w,x,t) - \frac{\rho\sigma(-tf(w,x))}{J}zx^\top \tag{4.2}$$

where $z = Wx$ is the intermediate feature and $J = \|\nabla f(x)\| = \sqrt{\|z\|^2 + \|x\|^2\|v\|^2}$ is a normalization factor. In the second layer, the gradient with respect to v is

$$-\nabla_{v+\epsilon^{(2)}}\ell(w+\epsilon,x,t) = \sigma(-tf(w,x))\left(tz - \frac{\rho\|x\|^2}{J}v\right) \tag{4.3}$$

$$= -\nabla_v \ell(w,x,t) - \frac{\rho\sigma(-tf(w,x))\|x\|^2}{J}v \tag{4.4}$$

From Equation 4.1, note that SAM adds a feature norm regularization to the first layer $zx^\top = \nabla_W \frac{1}{2}\|z\|_2^2$ scaled by some scalar dependent on ρ, $f(w,x)$, and J. Similarly, from Equation 4.3, note that SAM adds a weight norm penalty to the second layer weights $v = \nabla_v \frac{1}{2}\|v\|_2^2$ also multiplied by some scalar.

■

SAM's Jacobian term induces a norm penalty on the last layer weights and intermediate features. The normalization factor J scales the regularization be closer to the norm than the squared norm. We suspect that the explicit robustness effect from the logit scaling term and the Jacobian term are connected. In linear models, there exists a clear connection between weight decay and SAM's constant adjustment. A small weight norm and SAM's logit scale both balance the gradient contribution of low and high loss examples. Weight decay and small initialization is known to improve label noise robustness in high-dimensional settings (Advani & Saxe, 2017). In multi-layer models, the effect of regularizing the norm of the last layer weights and additionally, the intermediate activations, may have other implicit effects such as low-rank weight matrices (Galanti et al., 2023).

4.3 Experiments

Surprisingly, we show that if we add an ℓ_2 regularization on the last layer weights and last hidden layer intermediate activations to SGD, then this does improve robustness to label noise in the nonlinear setting (though not as much as SAM). This suggest that the regularization explains at least some of the benefits, but is insufficient to fully explain the entire benefit of SAM to label noise robustness. We further simplify down SAM's regularization to the following objective

$$\min_w L(w) + \gamma_z \frac{1}{N}\sum_{i=1}^{N} \|z\|_2 + \gamma_v \|v\|_2^2 \tag{4.5}$$

where v is the last layer weights and z is the last hidden layer intermediate representation.

We conduct our experiments on CIFAR10 with ResNet18. 1-SAM leads to unstable optimization with batch normalization as it requires passing through the datapoints individually through the network. Thus, we replace all batch normalization layers with layer normalization. Keeping all other hyperparameters such as learning rate, weight decay, and bawqdtch size the same, we compare the performance of SGD, 1-SAM, L-SAM, J-SAM, and the regularized SGD (Eq. 4.5). Although regularized SGD does not achieve exactly the same test accuracy as SAM, the gap is significantly closed from 17% to 9%.

Algorithm	Best Test Accuracy
1-SAM ($\rho = 0.01$)	69.47%
L-SAM ($\rho = 0.01$)	54.13%
J-SAM ($\rho = 0.01$)	69.17%
SGD	52.48%
SGD w/ Proposed Reg	60.8%

Table 1: Adding our proposed regularization on the last layer weights and logits boosts SGD performance by 10%. We employ no data augmentation

5 Related work

5.1 Connection between SAM and generalization

Although the reason SAM achieves better generalization remains poorly understood, several papers have independently tried to elucidate why the *per-example* regularization of 1-SAM may be important. Andriushchenko & Flammarion (2022a) show that in sparse regression on diagonal linear networks, 1-SAM is more biased towards sparser weights than n-SAM. Wen et al. (2022) differentiates 1-SAM and n-SAM by proving that the type of “flatness” that each algorithm regularizes is actually different. 1-SAM minimizes the trace of the Hessian, while n-SAM does not. Meng et al. (2023) analyzes the per-example gradient norm penalty, which also effectively minimizes sharpness and show that if the data has low signal-to-noise ratio, penalizing the per-example gradient norm dampens the noise allowing more signal learning. On the other hand, penalizing the average gradient norm does not sufficiently suppress noise. Our analysis differ from previous works as we focus on understanding 1-SAM’s behavior under label corruption, where models achieve best test accuracy with early-stopping. Although this is the regime where SAM’s achievements are the most notable, it has not been thoroughly studied in previous works which focus on SAM’s solution at convergence. Recently, Behdin & Mazumder (2023) studied the bias-variance tradeoff of SAM and its connection to SAM’s early label noise robustness, yet they restrict their analysis to n-SAM.

5.2 Flatness

Flatness of the loss landscape has been of interest as a proxy for predicting the generalization performance in neural networks. A family of optimization choices including minibatch noise in stochastic gradient descent (SGD), large initial learning rate, and dropout have shown to regularize the model towards solutions lying in *flat* basins or areas of low curvature (Keskar et al., 2016; Dziugaite & Roy, 2017; Cohen et al., 2021; Damian et al., 2022; Nar & Sastry, 2018; Wei et al., 2020). Several works also propose a notion of flatness as a generalization measure (Hochreiter & Schmidhuber, 1997; Dziugaite & Roy, 2017; Bartlett et al., 2017; Neyshabur et al., 2017). Some works design a regularization term or noise to add to GD to match the performance of minibatch SGD and also find a correlation with flatness (Damian et al., 2021; Orvieto et al., 2022; Jastrzebski et al., 2021). Yet, the positive correlation between generalization and flatness is very specific to those certain optimization choices (Jiang et al., 2019). For example, strong data augmentation can sharpen the landscape (Andriushchenko et al., 2023b). Flatness also appears in adversarial weight robustness where flatness is explicitly desired (Wu et al., 2020; Zheng et al., 2021). However, adversarial *weight* robustness is not obviously connected to generalization and moreover, has no direct connection to *label noise* robustness.

5.3 Learning with label noise

Mislabeled data is a persistent problem even in the most common benchmarks such as MNIST, CIFAR, and ImageNet (Müller & Markert, 2019) as their presence has an impact on model performance (Nakkiran et al., 2021; Rolnick et al., 2018). By filtering out noisy examples, smaller language models have shown to be able to match the performance of larger models (Gunasekar et al., 2023; West et al., 2021). The conclusions of our analysis in the linear setting are related to metrics inspired by the learning time to identify and avoid learning memorized examples(Zhang & Sabuncu, 2018; Lee et al., 2019; Chen et al., 2019; Huang et al., 2019; Jiang et al.; 2020a; Carlini et al., 2019; Jiang et al., 2020b; Arazo et al., 2019; Liu et al., 2022). Specifically, model tends to fit clean examples first, as proven in Liu et al. (2020) for linear models. More recently, Liu et al. (2023) showed that a similar effect is observable in neural networks. In particular, the gradient of clean and noisy examples often observe negative cosine similarity at the beginning of training, and thus we may reason about early learning of clean points by the magnitude of their contribution to the average gradient.

6 Discussion, Limitations, and Conclusion

Although the SAM optimizer has proven very successful in practice, there is a notable divide between the established motivation for SAM (increasing flatness of the solution), and the empirical behavior of the method. Fundamentally, the work we present here aims to justify the usage of SAM by appealing to a very different set of principles than those used to originally derive the algorithm. Specifically, we show that in the linear and nonlinear cases, there is an extent to which SAM "merely" acts by learning more clean examples before fitting noisy examples. This provides a natural perspective upon which to analyze the strong performance of SAM, especially in the setting of label noise.

In the linear setting, we identified that SAM up-weights the gradient signal from low loss points. This is quite similar in to well known label noise robustness methods (Liu et al., 2022; 2020) which also utilize learning time as a proxy for distinguishing between clean and noisy examples. In the nonlinear setting, we identify arguably a more interesting phenomena – *how* clean examples are fit can affect the learning time of noisy examples. Our observation of feature regularization is of close connection to Andriushchenko et al. (2023a) which show that SAM drives down the norm of the intermediate activation in a 2 layer ReLU network and this implicitly biases the activations to be low rank. Our paper similarly identifies that a regularization on the norm of the activations by SAM occurs *throughout training* and in particular improves label noise robustness.

Finally, we emphasize that despite their close connection, SAM has been surprisingly under explored in the label noise setting. The research community has developed a number of methods for understanding and adjusting to label noise, and it has so far been a mystery as to how SAM manages to unintentionally match the performance of such methods. Empirically however, we find that simulating even partial aspects of SAM's regularization of the network Jacobian such as simply regularizing the norm of the features and the norm of the last layer weights can largely preserve SAM's performance. As a secondary effect of this research, we hope our conclusions can inspire label-noise robustness methods that may ultimately have similar benefits to SAM (but ideally, without the additional runtime cost incurred by requiring the batchsize number of back-propagations per update needed by 1-SAM).

References

Madhu S. Advani and Andrew M. Saxe. High-dimensional dynamics of generalization error in neural networks, 2017.

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

Maksym Andriushchenko and Nicolas Flammarion. Understanding sharpness-aware minimization, 2022b. URL https://openreview.net/forum?id=qXa0nhTRZGV.

Maksym Andriushchenko, Dara Bahri, Hossein Mobahi, and Nicolas Flammarion. Sharpness-aware minimization leads to low-rank features. *arXiv preprint arXiv:2305.16292*, 2023a.

Maksym Andriushchenko, Francesco Croce, Maximilian Müller, Matthias Hein, and Nicolas Flammarion. A modern look at the relationship between sharpness and generalization. *arXiv preprint arXiv:2302.07011*, 2023b.

Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International conference on machine learning*, pp. 312–321. PMLR, 2019.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

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

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

Nicholas Carlini, Ulfar Erlingsson, and Nicolas Papernot. Distribution density, tails, and outliers in machine learning: Metrics and applications. *arXiv preprint arXiv:1910.13427*, 2019.

Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pp. 1062–1070. PMLR, 2019.

Jeremy M Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. *arXiv preprint arXiv:2103.00065*, 2021.

Alex Damian, Tengyu Ma, and Jason D Lee. Label noise sgd provably prefers flat global minimizers. *Advances in Neural Information Processing Systems*, 34:27449–27461, 2021.

Alex Damian, Eshaan Nichani, and Jason D Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. *arXiv preprint arXiv:2209.15594*, 2022.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Tomer Galanti, Zachary S. Siegel, Aparna Gupte, and Tomaso Poggio. Characterizing the implicit bias of regularized sgd in rank minimization, 2023.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997.

Jinchi Huang, Lie Qu, Rongfei Jia, and Binqiang Zhao. O2u-net: A simple noisy label detection approach for deep neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3326–3334, 2019.

Stanislaw Jastrzebski, Devansh Arpit, Oliver Astrand, Giancarlo B Kerg, Huan Wang, Caiming Xiong, Richard Socher, Kyunghyun Cho, and Krzysztof J Geras. Catastrophic fisher explosion: Early phase fisher matrix impacts generalization. In *International Conference on Machine Learning*, pp. 4772–4784. PMLR, 2021.

L Jiang, Z Zhou, T Leung, LJ Li, and L Fei-Fei. Mentornet: Regularizing very deep neural networks on corrupted labels (2017). *arXiv preprint arXiv:1712.05055*.

Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. Beyond synthetic noise: Deep learning on controlled noisy labels. In *International conference on machine learning*, pp. 4804–4815. PMLR, 2020a.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

Ziheng Jiang, Chiyuan Zhang, Kunal Talwar, and Michael C Mozer. Characterizing structural regularities of labeled data in overparameterized models. *arXiv preprint arXiv:2002.03206*, 2020b.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. In *International conference on machine learning*, pp. 3763–3772. PMLR, 2019.

Chaoyue Liu, Amirhesam Abedsoltan, and Mikhail Belkin. On emergence of clean-priority learning in early stopped neural networks. *arXiv preprint arXiv:2306.02533*, 2023.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342, 2020.

Sheng Liu, Zhihui Zhu, Qing Qu, and Chong You. Robust training under label noise by over-parameterization. In *International Conference on Machine Learning*, pp. 14153–14172. PMLR, 2022.

Xuran Meng, Yuan Cao, and Difan Zou. Per-example gradient regularization improves learning signals from noisy data. *arXiv preprint arXiv:2303.17940*, 2023.

Nicolas M Müller and Karla Markert. Identifying mislabeled instances in classification datasets. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2019.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

Kamil Nar and Shankar Sastry. Step size matters in deep learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

Antonio Orvieto, Hans Kersting, Frank Proske, Francis Bach, and Aurelien Lucchi. Anticorrelated noise injection for improved generalization. In *International Conference on Machine Learning*, pp. 17094–17116. PMLR, 2022.

David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise, 2018.

Colin Wei, Sham Kakade, and Tengyu Ma. The implicit and explicit regularization effects of dropout. In *International conference on machine learning*, pp. 10181–10192. PMLR, 2020.

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. How does sharpness-aware minimization minimize sharpness? *arXiv preprint arXiv:2211.05729*, 2022.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena D Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. Symbolic knowledge distillation: from general language models to commonsense models. *arXiv preprint arXiv:2110.07178*, 2021.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

Yaowei Zheng, Richong Zhang, and Yongyi Mao. Regularizing neural networks via adversarial model perturbation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8156–8165, 2021.

A Methods

All code was implemented in JAX (Bradbury et al., 2018), and we utilize the Flax neural network library. We utilize a normally distributed random initialization scheme. Our experiments were run on NVIDIA Quadro RTX A6000.

A.1 Synthetic Label Noise

We add synthetic label noise in the following manner. We randomly select Δ proportion of the training data to corrupt. For each datapoint (x, y), we select corrupted label t randomly from $t \in \{i \in [k]_{k=1}^{K} \mid i \neq y\}$.

A.2 Experiments on CIFAR10

For experiments conducted on CIFAR10, we train with 30% label corruption. We hyperparameter tuned the model for the best learning rate using SGD, and then hyperparameter tune 1-SAM's hyperparameter ρ utilizing the same learning rate. We do not utilize any learning rate schedule or data augmentation other than normalizing the image channels using mean (0.4914, 0.4822, 0.4465) and standard deviation (0.2023, 0.1994, 0.2010).

ResNet18 We have modified the ResNet18 architecture by replacing all Batch Norm layers with Layer Norm. This is necessary to safely run 1-SAM which requires a separate forward pass for each datapoint in the minibatch.

Parameter	Value
Batch size	128
Learning rate	0.01
Weight decay	0.0005
Epochs	200
ρ (for SAM)	0.01

2 Layer DLN/MLP with ReLU We do not include bias at any layer. The width of the intermediate layer is set to 500. We use the same hyperparameters as the ResNet18 experiments.

A.3 Experiments on Toy Data

Linear We set the parameters of our toy data distribution to be the following

Parameter	Value
Δ	0.4
B	2
γ	1
d	1000
Training samples	500
Test samples	1000

There is no weight decay. Learning rate is set to 0.01.

B LINEAR

Toy Data Setting We train our linear models over the following toy Gaussian data distribution $\mathcal{D} : \mathbb{R}^d \times \{-1, 1\}$.

$$\begin{gathered}\text{True Label: } y \sim \{-1, 1\} \text{ by flipping fair coin} \\ \text{Input: } x \sim y \cdot \left[B \in \mathbb{R}, n \sim \mathcal{N}\left(0, \frac{\gamma^2}{d-1} I_{d-1 \times d-1}\right)\right] \\ \text{Target: } t = y \cdot \varepsilon \text{ where } \varepsilon \sim \{-1 \;\text{ w.p } \Delta,\; 1 \;\text{ w.p } (1-\Delta)\}\end{gathered} \tag{B.1}$$

The test data is generated from a related distribution where the target is noiseless $t = y$.

Early Stopping Accuracy We are interested in assessing the early stopping test accuracy of SAM versus 1-SAM. The expected accuracy over the test distribution can be written as

$$\mathsf{Acc}(w) = \mathbb{E}_{x,y \sim \mathcal{D}_{test}} \left[\mathbb{1} \left[y(w^\top x) > 0 \right] \right] = P\left(y(w^\top x) > 0 \right) \tag{B.2}$$

$$= P\left(\frac{\gamma}{\sqrt{d-1}} w_{1+}^\top z > -w_1 B \right) = 1 - \Phi\left(-\frac{B\sqrt{d-1} w_1}{\gamma \|w_{1+}\|} \right) \tag{B.3}$$

where $w_{1+} \in \mathbb{R}^{d-2}$ denotes the vector consisting of the entries in w excluding the first. Therefore, the accuracy monotonically increases with $w_1 / \|w_{1+}\|$. Thus, the optimal linear classifier in this setting is precisely proportional to the first elementary vector:

$$w^* \propto e_1 \tag{B.4}$$

B.1 1-SAM Asymptotics

As we can observe in Figure 2, the test accuracy monotonically increases with ρ. We analyze 1-SAM in the limit as the perturbation magnitude converges to limit $\rho \to \infty$ and observe that it converges to the optimal classifier. We analyze the extreme of 1-SAM with $\rho = \infty$. In this regime, the 1-SAM update for each example simply becomes

$$\nabla_{w+\varepsilon_i} \ell(w+\varepsilon_i, x_i, y_i) = \lim_{\rho \to \infty} -t_i \sigma\left(-t_i \langle w_i, x_i \rangle + \rho \left\|x_i\right\|_2\right) x_i = -t_i x_i \tag{B.5}$$

and the ratio between the magnitude of any two points converges to 1, specifically for any two datapoints x_i and x_j

$$\lim_{\rho \to \infty} \frac{\|\nabla_{w+\varepsilon_i} \ell(w+\varepsilon_i, x_i, y_i)\|}{\left\|\nabla_{w+\varepsilon_j} \ell(w+\varepsilon_j, x_j, y_j)\right\|} = \lim_{\rho \to \infty} \frac{\sigma\left(-t_i \langle w, x_i \rangle + \rho \|x_i\| \,\right)\big)}{\sigma\left(-t_j \langle w, x_j \rangle + \rho \|x_j\| \,\right)} \tag{B.6}$$

$$= \lim_{\rho \to \infty} \frac{1 + \exp(t_i \langle w, x_i \rangle - \rho \|x_i\|)}{1 + \exp(t_j \langle w, x_j \rangle - \rho \|x_j\|)} = 1 \tag{B.7}$$

As a result, each gradient update of 1-SAM is precisely equal the empirical mean of the data scaled by the label $\hat{u}_{\mathcal{D}} = X^\top t$. Say that we are given fixed training examples $[x_i]_{i=1}^n$ independently sampled from $\mathcal{D}$ where Δ is corrupted. Note that as d grows (and $n = \Omega(\log(d))$), $\hat{u}_{\mathcal{D}}$ converges to the optimal classifier. Notably,

$$\hat{u}_{\mathcal{D}} = \left[(1-2\Delta)B, \sum_{i=1}^{n} \frac{\gamma}{n\sqrt{d-1}} z_i\right] \xrightarrow{d \to \infty} (1-2\Delta)Be_1 \tag{B.8}$$

where z_i are sampled from the standard normal. Also note that a same effect occurs with weight decay in regression, in the limit the solution for ridge regression also converges to the empirical mean scaled by the label

$$(XX^\top + \lambda I)^{-1} X^\top t \xrightarrow{\lambda \to \infty} X^\top t \tag{B.9}$$

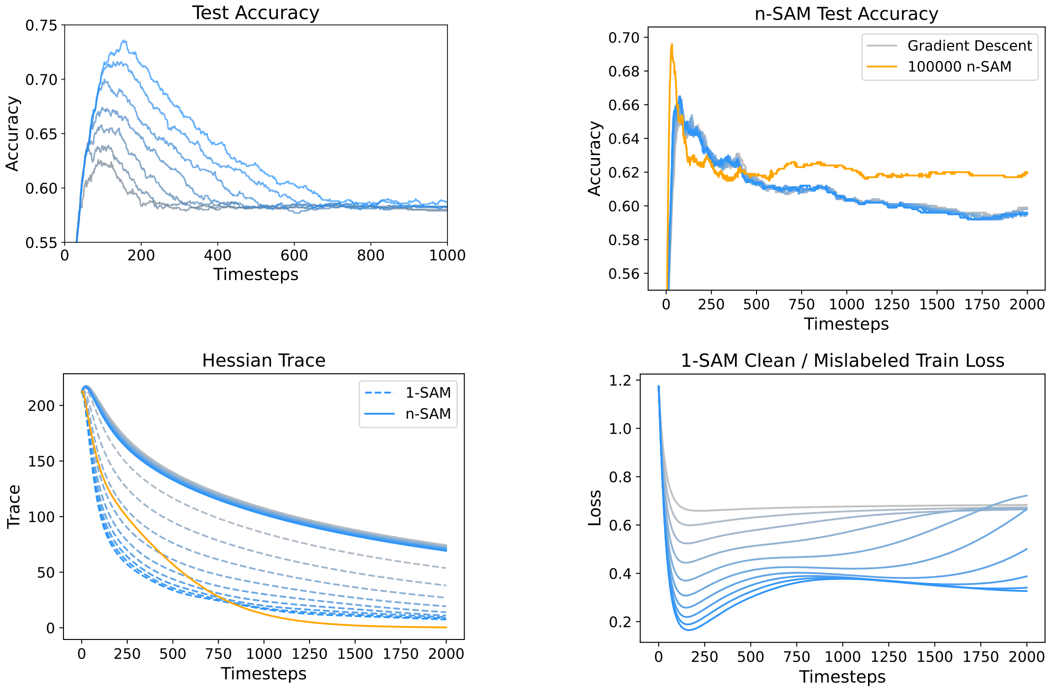


Figure 4: We train linear models on our toy model using 1-SAM. Bluer curves denote larger ρ. We initialize the model with random initialization, although we observe similar curves when initialized at the origin. To reduce interference, we set the learning rate to be small $\eta = 1e-2$ and we do not add weight decay.

B.2 Analysis of n-SAM

SAM is designed to minimize the following objective

$$\min_{w} \max_{\|\epsilon\|_2 \leq \rho} L(w+\epsilon)\| + \lambda \|w\|_2 \tag{B.10}$$

where w is the model parameters and L is the loss. The goal is to find a solution that lies in a region where the loss does not fluctuate dramatically with any ϵ-perturbation. SAM approximates ϵ by taking the first-order approximation of the loss, leading to the following update

$$w = w - \eta \left(\frac{1}{n} \sum_{i=1}^{n} \nabla_{w+\epsilon} \ell \left(w+\epsilon, x_i, y_i\right) + \lambda w \right) \tag{B.11}$$

where $\epsilon = \rho \frac{\nabla_w L(w)}{\|\nabla_w L(w)\|_2}$ and $L(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(w, x_i, y_i)$ and ρ is a hyperparameter. The original and follow up works Foret et al. (2020); Andriushchenko & Flammarion (2022a) show that the SAM update paired with full-batch gradient descent (n-SAM) does not have any advantage over gradient descent in practice. We observe that even in the linear setting, when a linear model is trained using n-SAM on our toy Gaussian data, this naive SAM update observes little early stopping improvements unless ρ is scaled up dramatically. In Figure 4 for example, we set $\rho = 100000$ for n-SAM.

Comparing 1-SAM and n-SAM in the linear setting, we find that they have fundamentally different effects. n-SAM's perturbation is not a constant that is only a function of the data norm. The magnitude of the perturbation is proportional to the loss, so SAM does not preferentially upweight low loss points except for the early training steps. Let us consider that the noise n_i of the datapoints are orthogonal for ease of analysis. Then

$$\nabla_w^{n-SAM} \ell(w, x_i, y_i) = -t_i \sigma \left(-t_i \langle w + \rho \frac{\nabla_w L(w)}{\|\nabla_w L(w)\|}, x_i \rangle \right) x_i \tag{B.12}$$

$$= -t_i \sigma \left(-t_i \langle w, x_i \rangle + \frac{\rho}{n \|\nabla_w L(w)\|} \left(O(B^2) + \frac{\gamma^2}{d-1} \|n_i\| \sigma(-t_i \langle w, x_i \rangle) \right) \right) x_i \tag{B.13}$$

Assuming gradient descent starting at $w = 0$, class balance, and the same number of mislabeled datapoints in each class, n-SAM at each iteration, perturbs each point proportional to $\sigma(-t\langle w, x_i \rangle)$ which is smaller for low loss points and higher for high loss points. We do observe in Figure 3 (see orange curve) that if ρ is sufficiently large, we are able to see some gains with n-SAM in the first couple timesteps.

C REBUTTAL

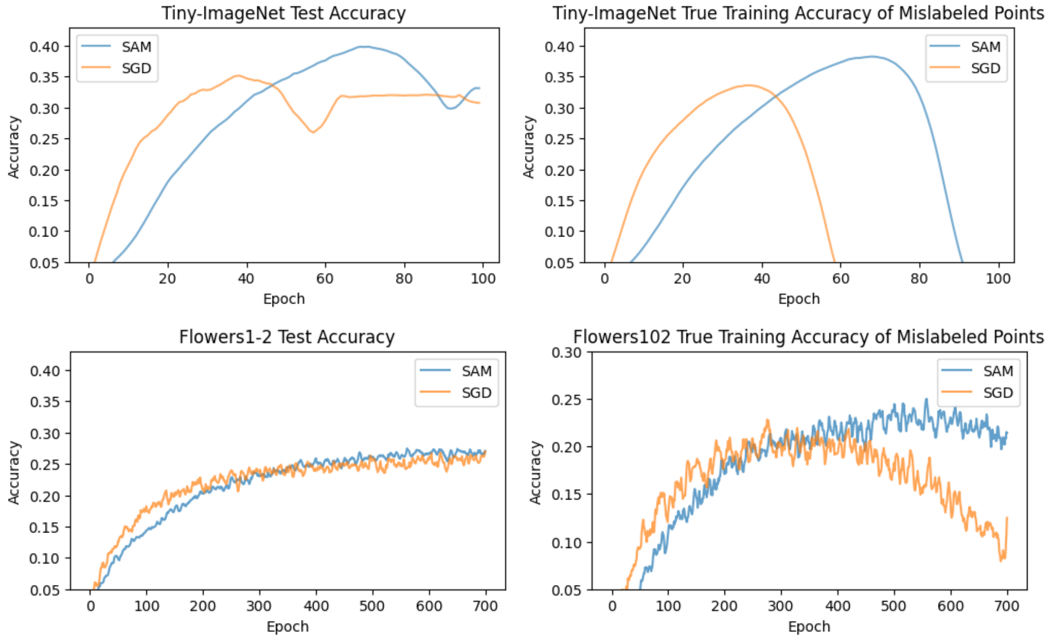


Figure 5: SAM observes higher true accuracy of noisy examples and this corresponds with better test accuracy on Tiny-ImageNet with 30% label noise and Flower102 with 20% label noise. Do note that contrary to trends in CIFAR10, in a low data settings such as Flowers102, the test accuracies of SAM and SGD do not drop even when the model starts to overfit. Models are ResNet18.

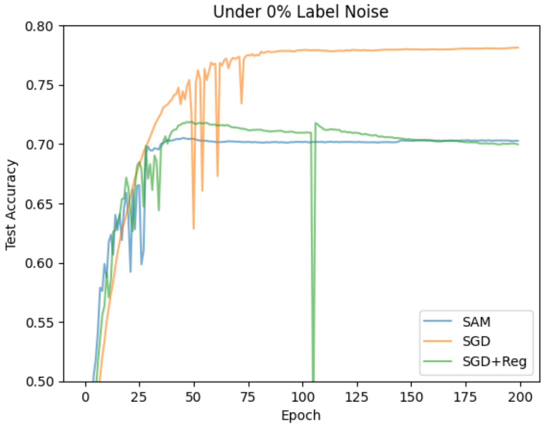


Figure 6: SAM ($\rho = 0.01$) and SGD have a smaller 8% difference in performance (less in comparison to the 20% difference with 30% label noise). Our weight and feature norm penalty observes improvements but only by a small factor of 1% and the performance degrades over time.

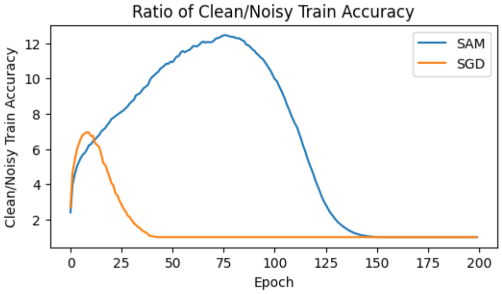


Figure 7: Under label noise, SAM learns clean examples faster than noisy examples. We conduct this study on CIFAR10 30% label noise using ResNet18. In particular, the ratio of clean and noisy training accuracy reaches a much higher value for SAM than SGD.

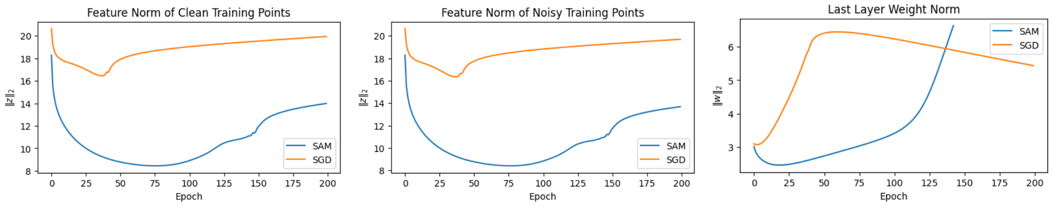


Figure 8: When training ResNet18 with SAM, the norm of the last hidden layer features and last layer weights is implicitly regularized to be much smaller. This is consistent with our analysis of 2-layer deep linear networks.

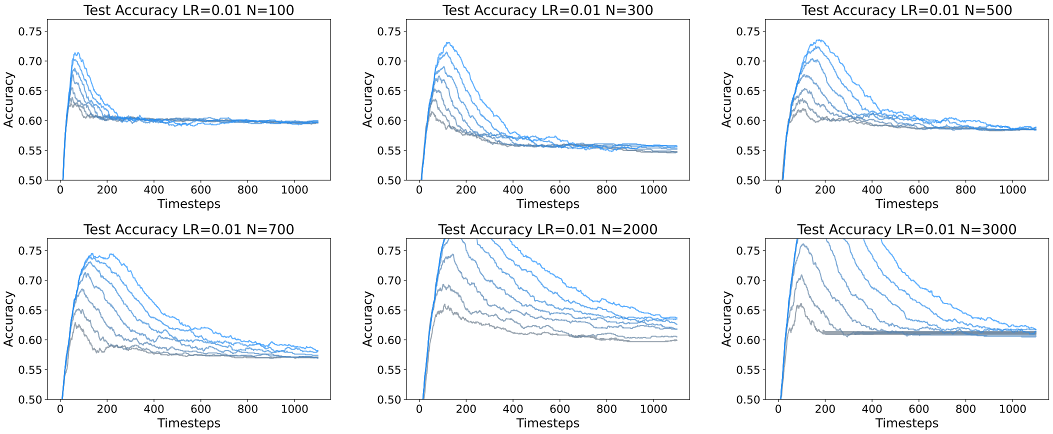


Figure 9: **Behavior with number of training examples for linear model/toy Gaussian data with 40% label noise.** Linear models trained with different subsets of the toy Gaussian data observe different levels of benefit with SAM. ρ is scaled between 0.03 and 0.18, bluer curves signifying higher ρ. The data is 1000 dimensions. Note that performance improves with SAM even in underparametrized regimes.

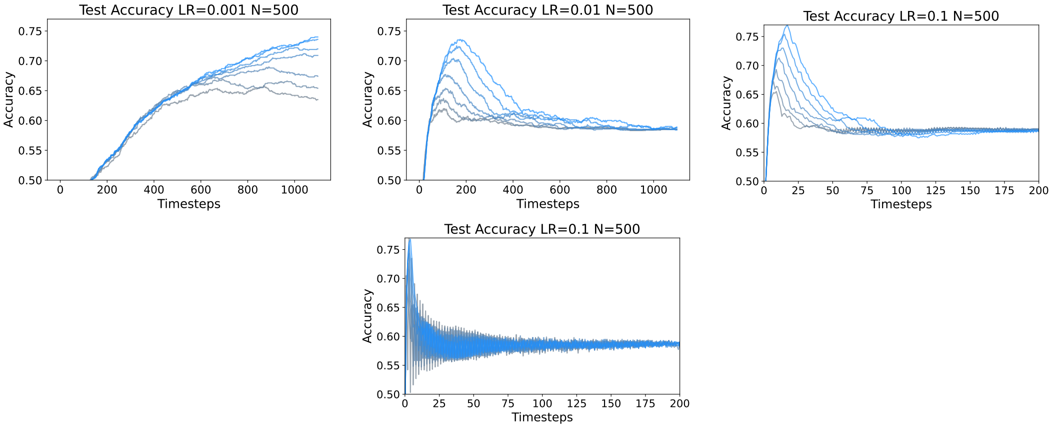


Figure 10: **Behavior with learning rate for linear model/toy Gaussian data with 40% label noise** As the learning rate increases, we generally observe a slight improvement in early stopping accuracy in both SGD and SAM.

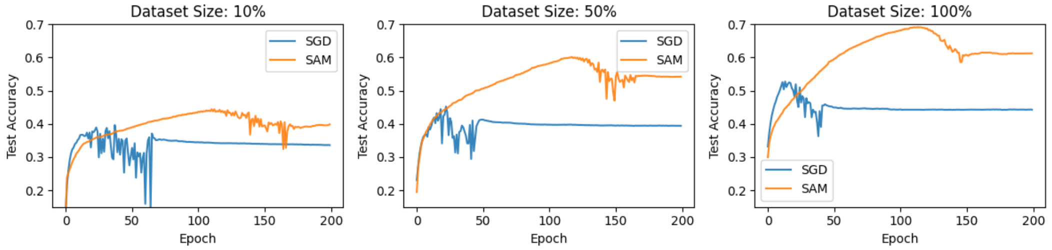


Figure 11: **Behavior with number of training examples for ResNet18/CIFAR10 with 30% label noise** We compare SGD and SAM trained on different number of training examples $(10, 50, 100\%$ of the training data). We see that the difference between SAM $(\rho = 0.01)$ and SGD increases as the dataset size increases.

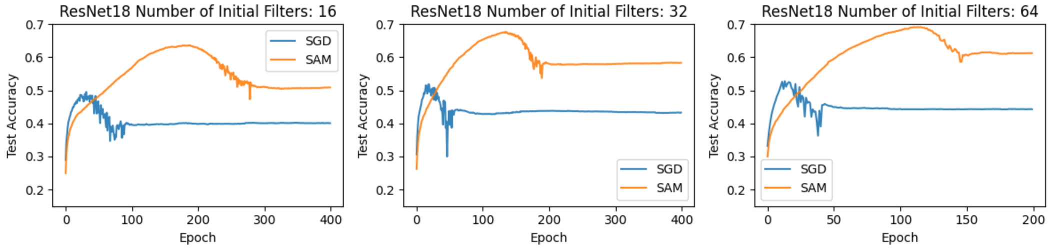


Figure 12: **Behavior with model width for ResNet18/CIFAR10 with 30% label noise** ResNet18 starts with 64 convolutional filters, and the filters double every two convolutional layers. We reduce the width of ResNet18 by 1/2 and 1/4 by starting with 32 and 16 initial number of filters, respectively. We see that SAM $(\rho = 0.01)$ and SGD both improve in terms of performance as model width increases.

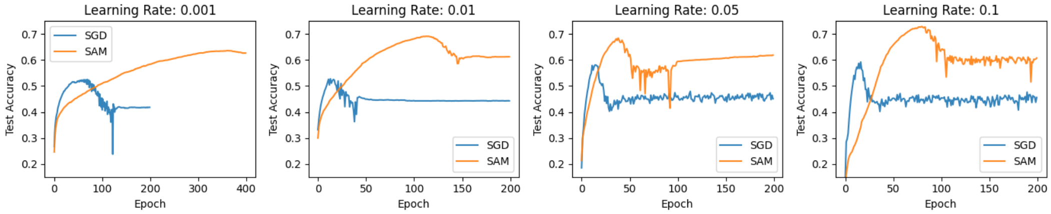


Figure 13: **Behavior with learning rate for ResNet18/CIFAR10 with 30% label noise**. For each learning rate, we choose the best ρ for SAM found by hyperparameter search. As learning rate increases, we observe that both SAM and SGD both improve in terms of performance as learning rate increases. For small learning rate 0.001, we found it difficult for SAM to observe significant improvements upon SGD unless trained for at least double the time.