# GRADIENT NORM REGULARIZER SEEKS FLAT MINIMA AND IMPROVES GENERALIZATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

The heavy overparameterization of current deep neural networks requires model generalization guarantees. Recently, flat minima are proven to be effective for improving generalization and sharpness-aware minimization (SAM) achieves state-of-the-art performance. Yet we show that SAM fails to measure flatness/sharpness when there are multiple minima within the perturbation radius. We present a novel regularizer named Gradient Norm Regularizer (GNR) to seek minima with uniformly small curvature across all directions and measure sharpness even when multiple minima are within the perturbation radius. We show that GNR bounds both the maximum eigenvalue of Hessian at local minima and the regularization function of SAM. We present experimental results showing that GNR improves the generalization of models trained with current optimizers such as SGD and AdamW on various datasets and networks. Furthermore, we show that GNR can help SAM find flatter minima and achieve better generalization.

## 1 INTRODUCTION

Current neural networks have achieved promising results in a wide range of fields (Ren et al., 2015; Kipf & Welling, 2016; Simonyan & Zisserman, 2014; Young et al., 2018; Touvron et al., 2021; Zhou et al., 2021), yet they are typically heavily over-parameterized. Such heavy overparameterization leads to severe overfitting and poor generalization to unseen data when the model is learned simply with common loss functions (e.g., cross-entropy). Thus effective training algorithms are required to limit the negative effects of overfitting training data and find generalizable solutions.

Some studies have demonstrated that current optimization approaches, such as stochastic gradient descent, Adam (Kingma & Ba, 2015), AdamW (Loshchilov & Hutter, 2017), Adai (Xie et al., 2022b), and others (Duchi et al., 2011; Chen et al., 2018; Luo et al., 2019), can affect model generalization. However, it can be a trade-off between generalization ability and convergence speed (Keskar et al., 2017; Xie et al., 2022b). Different tasks and network architectures may agree with different optimizers (e.g., SGD is often chosen for ResNet (He et al., 2016) while AdamW (Loshchilov & Hutter, 2017) for ViTs (Dosovitskiy et al., 2020)). Thus selecting a proper optimizer is critical while the understanding of its relationship to model generalization remains nascent (Foret et al., 2021).

Many studies try to improve model generalization via modifying the training procedure, such as batch normalization (Ioffe & Szegedy, 2015), dropout (Hinton et al., 2012), and data augmentation (Zhang et al., 2018; Yun et al., 2019; Cubuk et al., 2020). Especially, some works discuss the connection between the geometry of the loss landscape and generalization (Izmailov et al., 2018; He et al., 2019; Foret et al., 2021). A branch of effective approaches, sharpness-Aware Minimization (SAM) (Foret et al., 2021) and its variants (Kwon et al., 2021; Zhuang et al., 2022) seek to minimize the sharpness of the loss landscape and achieve state-of-the-art performance on various image classification tasks. Foret et al. (2021) prove that optimizing the worst case of loss in the neighborhood of a weight point leads to flatter minima and lower generalization error.

Optimizing the worst case, however, relies on a reasonable choice of perturbation radius. We show that SAM may fail to indicate flatness/sharpness when there are multiple minima inside the perturbation radius. As a prefixed hyperparameter in SAM or a hyperparameter under parameter re-scaling in its variant, ASAM (Kwon et al., 2021), the perturbation radius $\rho$ can not always be a perfect choice (covering only a single minimal) in the whole training process.

To address this problem, we introduce a novel regularize, gradient norm regularizer (GNR), to control the maximum gradient norm in the neighborhood of minima. We show that when the perturbation radius is relatively small, both GNR and SAM control the maximum eigenvalue of the Hessian of the training loss, which is a proper sharpness/flatness measure indicating the worst-case loss increase under an adversarial perturbation to the weights (Keskar et al., 2017; Jiang et al., 2019). When the perturbation radius covers multiple minima, which we show is quite common, GNR discriminates more drastic jitters from real flat valleys. Moreover, SAM can be considered as a regularization term that constrains the gap between the worst-case loss and the current loss. We show that the SAM term is upper-bounded by the GNR term and thus, the constraint of GNR also constrains the SAM term.

We summarize our contributions as follows.

- We present a novel regularizer named gradient norm regularizer (GNR), which constrains the largest gradient norm in the neighborhood of minima. We show that the GNR term is a proper measure of the maximum eigenvalue of the Hessian and thus leads to flatter minima.
- We analyze the generalization error and the convergence of GNR.
- We empirically show that GNR considerably improves model generalization when combined with current optimizers such as SGD and AdamW across a wide range of datasets and networks. We show that GNR further improves the generalization of models trained with SAM.
- We empirically validate that GNR indeed finds flatter optima with lower Hessian spectra.

## 2 GRADIENT NORM REGULARIZER (GNR)

In this section, we introduce the overall framework of our Gradient Norm Regularizer (GNR) method. In Section 2.1, we formulate the regularizer and show its connection with the maximal eigenvalue of the Hessian, which is proven to be a proper flatness measure (Kaur et al., 2022) and closely related to the generalization ability (Foret et al., 2021). We further provide a generalization bound with respect to the empirical loss, the gradient norm regularizer, and high order terms, indicating that optimizing the regularizer could help improve generalization abilities. Motivated by these theoretical observations, in Section 2.2, we present the optimization framework based on GNR as shown in Algorithm 1. We then prove the convergence of this algorithm. Finally, we discuss the relationship between GNR and SAM in Section 2.3.

**Notations** Let $\mathcal{X}$ and $\mathcal{Y}$ be the sample space and label space, respectively. Let $\mathcal{D}$ denote the training distribution on $\mathcal{X} \times \mathcal{Y}$ and $S = \{(x_i, y_i)\}_{i=1}^n$ denote the training dataset with $n$ data-points drawn independently from $\mathcal{D}$. Let $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^d$ denote the parameters of the model. In addition, we use $B(\boldsymbol{\theta}, \rho)$ to denote the open ball of radius $\rho > 0$ centered at the point $\boldsymbol{\theta}$ in the Euclidean space, *i.e.*, $B(\boldsymbol{\theta}, \rho) = \{\boldsymbol{\theta}' : \|\boldsymbol{\theta} - \boldsymbol{\theta}'\| < \rho\}$[1].

Let $\ell : \Theta \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be the per-data-point loss function. Let $\hat{L}(\boldsymbol{\theta}) = \sum_{i=1}^n \ell(\boldsymbol{\theta}, x_i, y_i)$ and $L(\boldsymbol{\theta}) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(\boldsymbol{\theta}, x, y)]$ denote the empirical loss function and population-level loss function, respectively. $\nabla L(\boldsymbol{\theta})$ and $\nabla^2 L(\boldsymbol{\theta})$ ($\nabla \hat{L}(\boldsymbol{\theta})$ and $\nabla^2 \hat{L}(\boldsymbol{\theta})$) are the derivative and Hessian matrix of the function $L(\cdot)$ ($\hat{L}(\cdot)$) at point $\boldsymbol{\theta}$, respectively. Besides, for any $\boldsymbol{\theta} \in \Theta$, we use $\nabla\|\nabla \hat{L}(\boldsymbol{\theta})\|$ to represent the gradient of function $\|\nabla \hat{L}(\cdot)\|$ at point $\boldsymbol{\theta}$. In addition, we use $L^{\text{oracle}}(\boldsymbol{\theta})$ to denote an oracle loss function and it can be chosen as empirical loss function $\hat{L}(\boldsymbol{\theta})$, $\hat{L}(\boldsymbol{\theta})$ with the weight decay regularization, and other common loss functions.

### 2.1 GRADIENT NORM REGULARIZER

We first introduce the formulation of the regularizer, which measures the maximal gradient norm in the neighbourhood of a point $\boldsymbol{\theta} \in \Theta$.

[1] We use $\|\cdot\|$ to denote the L2 norm throughout the paper.

**Definition 2.1** (Gradient Norm Regularizer (GNR))**.** For any $\rho > 0$, the Gradient Norm Regularizer (GNR) is defined as

$$R_\rho^{\text{GNR}}(\boldsymbol{\theta}) \triangleq \rho \cdot \max_{\boldsymbol{\theta}' \in B(\boldsymbol{\theta}, \rho)} \left\| \nabla \hat{L}(\boldsymbol{\theta}') \right\|, \quad \forall \boldsymbol{\theta} \in \Theta. \tag{1}$$

Here $\rho$ is the perturbation radius that controls the magnitude of the neighbourhood.

Intuitively, GNR at a local minimum $\boldsymbol{\theta}^*$ can be explained as a flatness measure of the loss function $\hat{L}$ at $\boldsymbol{\theta}^*$. Specifically, when the function $\hat{L}$ is flat at $\boldsymbol{\theta}^*$, $\hat{L}$ should not change drastically in the neighbourhood of $\boldsymbol{\theta}^*$ (*i.e.*, $B(\boldsymbol{\theta}^*, \rho)$ for a constant $\rho$), indicating that the norm of $\nabla \hat{L}(\boldsymbol{\theta})$ should also be small in the neighbourhood. As a result, $R_\rho^{\text{GNR}}(\boldsymbol{\theta}^*)$ is small in this case. Conversely, when $\hat{L}$ is sharp at $\boldsymbol{\theta}^*$, there would probably exist a point $\boldsymbol{\theta}$ in the neighbourhood $B(\boldsymbol{\theta}^*, \rho)$ such that the gradient norm $\|\hat{L}(\boldsymbol{\theta})\|$ is large. Under this circumstance, $R_\rho^{\text{GNR}}(\boldsymbol{\theta}^*)$ would also be large.

More formally, as shown in Proposition 2.1, when the radius $\rho$ is small, GNR in Equation 1 is proportional to the maximal eigenvalue of the hessian matrix $\nabla^2 \hat{L}(\boldsymbol{\theta}^*)$, which is proven to be a proper flatness measure and closely related to generalization abilities (Jastrzębski et al., 2017; Keskar et al., 2017; Wen et al., 2019; Chaudhari et al., 2019; Kaur et al., 2022).

**Proposition 2.1.** *Let $\boldsymbol{\theta}^*$ be a local minimum of $\hat{L}$. Suppose $\hat{L}$ can be second order Taylor approximated in the neighbourhood $B(\boldsymbol{\theta}^*, \rho)$, i.e., $\forall \boldsymbol{\theta} \in B(\boldsymbol{\theta}^*, \rho)$, $\hat{L}(\boldsymbol{\theta}) = \hat{L}(\boldsymbol{\theta}^*) + (\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top \nabla^2 \hat{L}(\boldsymbol{\theta}^*)(\boldsymbol{\theta} - \boldsymbol{\theta}^*)/2$. Then*

$$\lambda_{\max}\left(\nabla^2 \hat{L}(\boldsymbol{\theta}^*)\right) = \frac{R_\rho^{GNR}(\boldsymbol{\theta}^*)}{\rho^2}. \tag{2}$$

**Remark.** The second order Taylor approximation assumption is commonly adopted in optimization-related literature (Mandt et al., 2017; Zhang et al., 2019; Xie et al., 2021; 2022b) to analyze the properties near critical points.

Since the maximal eigenvalue of Hessian matrices is usually difficult to approximate and optimize directly (Yao et al., 2018; 2020), GNR in Equation 1 becomes a proper surrogate. Moreover, we derive a generalization bound *w.r.t.* the regularizer as shown in Proposition 2.2.

**Proposition 2.2.** *Suppose the per-data-point loss function $\ell$ is differentiable and bounded by $M$. Fix $\rho > 0$ and $\boldsymbol{\theta} \in \Theta$. Then with probability at least $1 - \delta$ over training set $S$ generated from the distribution $\mathcal{D}$,*

$$\begin{aligned} &\mathbb{E}_{\epsilon_i \sim N(0, \rho^2/(\sqrt{d}+\sqrt{\log n})^2)}[L(\boldsymbol{\theta} + \boldsymbol{\epsilon})] \\ &\leq \hat{L}(\boldsymbol{\theta}) + R_\rho^{GNR}(\boldsymbol{\theta}) + \sqrt{\frac{\frac{1}{4} d \log\left(1 + \frac{\|\boldsymbol{\theta}\|^2 \left(\sqrt{d}+\sqrt{\log n}\right)^2}{d\rho^2}\right) + \frac{1}{4} + \log\frac{n}{\delta} + 2\log(6n + 3d)}{n - 1}} + \frac{M}{\sqrt{n}}. \end{aligned} \tag{3}$$

**Remark.** The left-hand side of Equation 3 is close to the population-level loss function $L(\boldsymbol{\theta})$ since the numbers of samples $n$ and parameters $d$ are often large. As a result, ignoring high-order terms, the population-level loss $L(\boldsymbol{\theta})$ is bounded by the empirical loss $\hat{L}(\boldsymbol{\theta})$ and GNR $R_\rho^{\text{GNR}}(\boldsymbol{\theta})$, which motivates us to use $R_\rho^{\text{GNR}}(\boldsymbol{\theta})$ as a regularizer to help improve the generalization abilities of models.

### 2.2 Optimization with Gradient Norm Regularizer

In this subsection, we propose a novel framework to incorporate the gradient norm regularizer $R_\rho^{\text{GNR}}(\boldsymbol{\theta})$ into optimization procedures.

Specifically, suppose we could obtain an oracle loss function $L^{\text{oracle}}(\boldsymbol{\theta})$ and calculate its gradient $\nabla L^{\text{oracle}}(\boldsymbol{\theta})$. $L^{\text{oracle}}(\boldsymbol{\theta})$ can be chosen as the empirical loss function $\hat{L}(\boldsymbol{\theta})$, empirical loss function with other regularizations (such as the weight decay), and many other loss functions (such as the SAM loss (Foret et al., 2021)). Inspired by Propositions 2.1 and 2.2, the overall loss function is given by

$$L^{\text{overall}}(\boldsymbol{\theta}) = L^{\text{oracle}}(\boldsymbol{\theta}) + \alpha R_\rho^{\text{GNR}}(\boldsymbol{\theta}). \tag{4}$$

The gradient of the loss function $L^{\text{overall}}(\boldsymbol{\theta})$ is given by $\nabla L^{\text{overall}}(\boldsymbol{\theta}) = \nabla L^{\text{oracle}}(\boldsymbol{\theta}) + \alpha \nabla R_\rho^{\text{GNR}}(\boldsymbol{\theta})$. Using similar techniques in (Foret et al., 2021), we could approximate $\nabla R_\rho^{\text{GNR}}(\boldsymbol{\theta})$ by

$$\nabla R_\rho^{\text{GNR}}(\boldsymbol{\theta}) \approx \rho \cdot \nabla \left\| \nabla \hat{L}(\boldsymbol{\theta}^{\text{adv}}) \right\|, \quad \boldsymbol{\theta}^{\text{adv}} = \boldsymbol{\theta} + \rho \cdot \frac{\boldsymbol{f}}{\|\boldsymbol{f}\|}, \quad \boldsymbol{f} = \nabla \left\| \nabla \hat{L}(\boldsymbol{\theta}) \right\|. \tag{5}$$

Details of the derivation of $\nabla R_\rho^{\text{GNR}}(\boldsymbol{\theta})$ can be found in Appendix A.1. Notice that

$$\forall \boldsymbol{\theta} \in \Theta, \quad \nabla \|\nabla \hat{L}(\boldsymbol{\theta})\| = \frac{\nabla^2 \hat{L}(\boldsymbol{\theta}) \cdot \nabla \hat{L}(\boldsymbol{\theta})}{\|\nabla \hat{L}(\boldsymbol{\theta})\|}. \tag{6}$$

As a result, Equation 5 can be calculated efficiently by the Hessian vector product. The pseudocode of the whole optimization procedure is shown in Algorithm 1.

**Convergence analysis** We further analyze the convergence properties of Algorithm 1. Firstly, we introduce the Lipschitz smoothness, which is common adopted in optimization-related literature (Allen-Zhu & Li, 2018; Xu et al., 2018; Zhuang et al., 2022).

**Definition 2.2.** A function $J : \Theta \to \mathbb{R}$ is $\gamma$-Lipschitz smooth if

$$\forall \boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \Theta, \quad \|\nabla J(\boldsymbol{\theta}_1) - \nabla J(\boldsymbol{\theta}_2)\| \leq \gamma \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|. \tag{7}$$

With Definition 2.2, we could prove the convergence property of Algorithm 1 as shown in Theorem 2.3.

**Theorem 2.3.** *Suppose $L^{oracle}(\boldsymbol{\theta})$ is $\gamma_1$-Lipschitz smooth and $\hat{L}(\boldsymbol{\theta})$ is $\gamma_2$-Lipschitz smooth. Suppose $|L^{oracle}(\boldsymbol{\theta})|$ is bounded by $M$. For any timestamp $t \in \{0, 1, \ldots, T\}$ and any $\boldsymbol{\theta} \in \Theta$, suppose we can obtain noisy and bounded observations $g_t^{loss}(\boldsymbol{\theta})$, $g_t^{norm}(\boldsymbol{\theta})$, and $\tilde{g}_t^{loss}(\boldsymbol{\theta})$ of $\nabla \hat{L}(\boldsymbol{\theta})$, $\nabla \|\nabla \hat{L}(\boldsymbol{\theta})\|$, and $\nabla L^{oracle}(\boldsymbol{\theta})$ such that*

$$\begin{aligned} &\mathbb{E}[g_t^{loss}(\boldsymbol{\theta})] = \nabla \hat{L}(\boldsymbol{\theta}), \|g_t^{loss}(\boldsymbol{\theta})\| \leq G^{loss}, \mathbb{E}[g_t^{norm}(\boldsymbol{\theta})] = \nabla \|\nabla \hat{L}(\boldsymbol{\theta})\|, \|g_t^{norm}(\boldsymbol{\theta})\| \leq G^{norm}, \\ &\mathbb{E}[\tilde{g}_t^{loss}(\boldsymbol{\theta})] = \nabla L^{oracle}(\boldsymbol{\theta}), \|\tilde{g}_t^{loss}(\boldsymbol{\theta})\| \leq \tilde{G}^{loss}. \end{aligned} \tag{8}$$

*Then with learning rate $\eta_t = \eta_0/\sqrt{t}$ and perturbation radius $\rho_t = \rho_0/\sqrt{t}$, Algorithm 1 could obtain*

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[ \left\| \nabla L^{overall}(\boldsymbol{\theta}_t) \right\|^2 \right] \leq \frac{C_1 + C_2 \log T}{\sqrt{T}}, \tag{9}$$

*for some constants $C_1$ and $C_2$ that only depend on $\gamma, G^{loss}, G^{norm}, \tilde{G}^{loss}, M, \eta_0, \rho_0$, and $\alpha$.*

**Remark.** The assumptions in Theorem 2.3 are common and standard when analyzing convergence of non-convex functions via SGD-based methods (Kingma & Ba, 2015; Reddi et al., 2018; Zhuang et al., 2022). In addition, the requirements on $L^{\text{oracle}}(\boldsymbol{\theta})$ (*i.e.*, $L^{\text{oracle}}(\boldsymbol{\theta})$ is Lipschitz smooth and we can obtain unbiased and bounded observations of $\nabla L^{\text{oracle}}(\boldsymbol{\theta})$) are mild and common. For example, when the empirical loss function $\hat{L}(\boldsymbol{\theta})$ satisfies the constraints, it is easy to check that $\hat{L}(\boldsymbol{\theta})$ with the weight decay regularization also meets the requirements. This indicates that GNR is a general regularization strategy and can be plugged into most commonly used loss functions.

### 2.3 Comparison with SAM

SAM (Foret et al., 2021) proposes to optimize the following loss function

$$L^{\text{sam}}(\boldsymbol{\theta}) = \hat{L}(\boldsymbol{\theta}) + \max_{\boldsymbol{\theta}' \in B(\boldsymbol{\theta}, \rho)} \left( \hat{L}(\boldsymbol{\theta}') - \hat{L}(\boldsymbol{\theta}) \right). \tag{10}$$

Let $R_\rho^{\text{SAM}}(\boldsymbol{\theta}) \triangleq \max_{\boldsymbol{\theta}' \in B(\boldsymbol{\theta}, \rho)} \left( \hat{L}(\boldsymbol{\theta}') - \hat{L}(\boldsymbol{\theta}) \right)$ and it can be considered as a regularization function. However, we highlight that GNR, as shown in Equation 1, can deal with common cases that can not be handled by SAM.

To be specific, fix a perturbation radius $\rho$ and consider a local minimum $\boldsymbol{\theta}^*$. When $\rho$ is small, both regularizers could approximate the maximal eigenvalue of the Hessian matrix at point $\boldsymbol{\theta}^*$ (Proposition 2.1 for GNR and Lemma 3.3 in (Zhuang et al., 2022) for SAM) and hence measure the flatness of the loss function at $\boldsymbol{\theta}^*$.

**Algorithm 1** Optimization with Gradient Norm Regularizer (GNR)

**Input:** Batch size $b$, Learning rate $\eta_t$, Perturbation radius $\rho_t$, Trade-off coefficient $\alpha$, Small constant $\xi$
$t \leftarrow 0$, $\boldsymbol{\theta}_0 \leftarrow$ initial parameters
**while** $\boldsymbol{\theta}_t$ not converged **do**
  Sample batch $W_t \leftarrow \{(x_1, y_1), (x_2, y_2), \ldots, (x_b, y_b)\}$
  $\boldsymbol{h}_t^{\text{loss}} \leftarrow \nabla L^{\text{oracle}}(\boldsymbol{\theta}_t)$ ▷ Calculate the oracle loss gradient $\nabla L^{\text{oracle}}(\boldsymbol{\theta}_t)$
  $\boldsymbol{f}_t \leftarrow \nabla^2 \hat{L}_{W_t}(\boldsymbol{\theta}_t) \cdot \frac{\nabla \hat{L}_{W_t}(\boldsymbol{\theta}_t)}{\|\nabla \hat{L}_{W_t}(\boldsymbol{\theta}_t)\| + \xi}$
  $\boldsymbol{\theta}_t^{\text{adv}} \leftarrow \boldsymbol{\theta}_t + \rho_t \cdot \frac{\boldsymbol{f}_t}{\|\boldsymbol{f}_t\| + \xi}$
  $\boldsymbol{h}_t^{\text{norm}} \leftarrow \rho_t \cdot \nabla^2 \hat{L}_{W_t}(\boldsymbol{\theta}_t^{\text{adv}}) \cdot \frac{\nabla \hat{L}_{W_t}(\boldsymbol{\theta}_t^{\text{adv}})}{\|\nabla \hat{L}_{W_t}(\boldsymbol{\theta}_t^{\text{adv}})\| + \xi}$ ▷ Calculate the norm gradient $\nabla R_\rho^{\text{GNR}}(\boldsymbol{\theta}_t)$
  $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta_t(\boldsymbol{h}_t^{\text{loss}} + \alpha \boldsymbol{h}_t^{\text{norm}})$
  $t \leftarrow t + 1$
**end while**
**return** $\boldsymbol{\theta}_t$

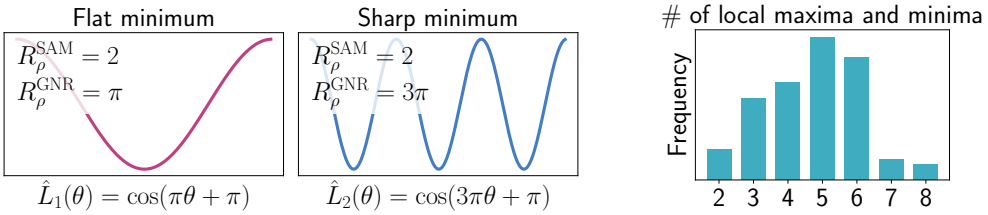



Figure 1: Subfigure (a) is a showcase of flat minimum (left) and sharp minimum (right). GNR ($R_\rho^{\text{GNR}}$) could distinguish the flatness of two functions while SAM ($R_\rho^{\text{SAM}}$) could not. Subfigure (b) reports the distribution of numbers of local minima and maxima within the perturbation radius $\rho$ after convergence.

When $\rho$ is large, there probably exist several other local minima in the neighborhood $B(\boldsymbol{\theta}^*, \rho)$. This case is common in practice as shown in Section 3.1. In addition, when the number of local minimum in $B(\boldsymbol{\theta}^*, \rho)$ becomes larger, $\boldsymbol{\theta}^*$ is expected to become sharper since the valley of $\boldsymbol{\theta}^*$ becomes narrower. However, SAM only measures the maximal gap of the loss function in $B(\boldsymbol{\theta}^*, \rho)$ and fails to distinguish the cases when the number of local minimums varies. By contrast, the maximal gradient norm in $B(\boldsymbol{\theta}^*, \rho)$ increases when the number of local minimum is larger, indicating that GNR can successfully characterize the sharpness in this case.

A detailed example can be found in Figure 1a. Fix $\rho = 1$. Consider two univariate functions $\hat{L}_1(\theta) = \cos(\pi\theta + \pi)$ and $\hat{L}_2(\theta) = \cos(3\pi\theta + \pi)$. $\theta$ takes value in $[-1, 1]$. By definition, we could obtain that $R_\rho^{\text{SAM}}(0) = 2$, $R_\rho^{\text{GNR}}(0) = \pi$ for $\hat{L}_1(\theta)$ while $R_\rho^{\text{SAM}}(0) = 2$, $R_\rho^{\text{GNR}}(0) = 3\pi$ for $\hat{L}_2(\theta)$. In this case, $\theta = 0$ in $\hat{L}_1(\theta)$ is flatter than that in $\hat{L}_2(\theta)$. However, $R_\rho^{\text{SAM}}$ could not distinguish the flatness of the minimum $\theta = 0$ for the two functions since $R_\rho^{\text{SAM}}(0)$ is equal for the two functions. By contrast, GNR could identify the flatter function $\hat{L}_1(\boldsymbol{\theta})$.

Moreover, in Proposition 2.4, we show that $R_\rho^{\text{SAM}}(\boldsymbol{\theta})$ is bounded by $R_\rho^{\text{GNR}}(\boldsymbol{\theta})$.

**Proposition 2.4.** *For any $\boldsymbol{\theta} \in \Theta$, $R_\rho^{SAM}(\boldsymbol{\theta})$ is bounded by $R_\rho^{GNR}(\boldsymbol{\theta})$, i.e., $R_\rho^{GNR}(\boldsymbol{\theta}) \geq R_\rho^{SAM}(\boldsymbol{\theta})$.*

Thus optimizing $R_\rho^{\text{GNR}}$ also leads to a smaller $R_\rho^{\text{SAM}}$. Proposition 2.4 gives an explanation of GNR covering wider scenarios compared with SAM.

Table 1: Results of GNR with state-of-the-art models on CIFAR-10 and CIFAR-100. The best results are highlighted in bold font.

| | CIFAR-10 | | | | CIFAR-100 | | | |
|---|---|---|---|---|---|---|---|---|
| Model | SGD | SGD + GNR | SAM | SAM + GNR | SGD | SGD + GNR | SAM | SAM + GNR |
| ResNet18 | $95.32_{\pm 0.13}$ | $\mathbf{96.17}_{\pm 0.21}$ | $96.10_{\pm 0.20}$ | $\mathbf{96.58}_{\pm 0.18}$ | $78.32_{\pm 0.32}$ | $\mathbf{79.53}_{\pm 0.30}$ | $79.27_{\pm 0.16}$ | $\mathbf{80.45}_{\pm 0.25}$ |
| ResNet101 | $96.35_{\pm 0.08}$ | $\mathbf{96.79}_{\pm 0.11}$ | $96.82_{\pm 0.16}$ | $\mathbf{97.20}_{\pm 0.15}$ | $80.47_{\pm 0.13}$ | $\mathbf{81.76}_{\pm 0.10}$ | $82.03_{\pm 0.12}$ | $\mathbf{83.13}_{\pm 0.07}$ |
| DenseNet121 | $91.16_{\pm 0.28}$ | $\mathbf{92.10}_{\pm 0.17}$ | $92.19_{\pm 0.20}$ | $\mathbf{92.56}_{\pm 0.29}$ | $69.25_{\pm 0.40}$ | $\mathbf{70.28}_{\pm 0.25}$ | $70.44_{\pm 0.19}$ | $\mathbf{70.82}_{\pm 0.25}$ |
| WRN28_2 | $94.82_{\pm 0.07}$ | $\mathbf{95.69}_{\pm 0.13}$ | $95.47_{\pm 0.08}$ | $\mathbf{95.85}_{\pm 0.08}$ | $75.45_{\pm 0.25}$ | $\mathbf{76.89}_{\pm 0.31}$ | $77.04_{\pm 0.18}$ | $\mathbf{77.55}_{\pm 0.20}$ |
| WRN28_10 | $95.73_{\pm 0.10}$ | $\mathbf{96.61}_{\pm 0.15}$ | $96.78_{\pm 0.80}$ | $\mathbf{97.29}_{\pm 0.11}$ | $81.40_{\pm 0.13}$ | $\mathbf{83.45}_{\pm 0.09}$ | $83.41_{\pm 0.04}$ | $\mathbf{84.31}_{\pm 0.06}$ |
| ResNeXt29-32x4d | $95.75_{\pm 0.31}$ | $\mathbf{96.40}_{\pm 0.25}$ | $96.32_{\pm 0.36}$ | $\mathbf{96.75}_{\pm 0.27}$ | $79.45_{\pm 0.29}$ | $\mathbf{81.18}_{\pm 0.33}$ | $81.35_{\pm 0.12}$ | $\mathbf{82.08}_{\pm 0.20}$ |
| PyramidNet110 | $96.19_{\pm 0.11}$ | $\mathbf{97.11}_{\pm 0.14}$ | $97.26_{\pm 0.05}$ | $\mathbf{97.51}_{\pm 0.09}$ | $82.74_{\pm 0.12}$ | $\mathbf{84.91}_{\pm 0.09}$ | $85.01_{\pm 0.09}$ | $\mathbf{85.25}_{\pm 0.06}$ |

# 3 EXPERIMENTS

We empirically show that the case discussed in Section 2.3 is common in practice. Then we evaluate GNR with random initialization on various state-of-the-art models and the transfer learning setting on various datasets. We show the Hessian spectra of GNR at convergence and discuss the computation overhead of GNR with the considerable improvement of model generalization.

## 3.1 THE DENSITY OF LOCAL MINIMA

To investigate the number of local minima within the perturbation radius, we train 3 ResNet-18 models with SAM on CIFAR-100 with proper hyperparameters for 200 epochs. The perturbation radius is set to 0.1 as suggested by Foret et al. (2021). We load the checkpoints at convergence and freeze the model weights for evaluation. We randomly generate 100 perturbation directions with the same size as the model weights for each model. For each direction, we repeatedly add a perturbation with the norm of 0.01 along the direction 10 times. We calculate the training loss after each addition. We report the distribution of the number of local maxima and minima along each perturbation direction within the perturbation radius $\rho$ of 0.1. As shown in Figure 1b, we find more than 1 local minima within $\rho$ for most of the directions, indicating that the case is common in practice. As discussed in Section 2.3, SAM fails to tell the sharpness caused by multiple minima while the GNR term increases as the sharpness grow. As shown in Section 3.4, GNR convergences to lower Hessian spectra.

## 3.2 TRAINING FROM SCRATCH

### 3.2.1 CIFAR-10 AND CIFAR-100

We conduct experiments on CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009) with ResNets (He et al., 2016), WideResNet (Zagoruyko & Komodakis, 2016), ResNeXt (Xie et al., 2017), PyramidNet (Han et al., 2017) and Vision Transformers (ViTs) (Dosovitskiy et al., 2020). All the models are trained with basic data augmentations (horizontal flip, padding by four pixels, and random crop) for 200 epochs from scratch. GNR has two hyperparameters, $\rho$ and $\alpha$. We conduct a grid search over $\{0.05, 0.1, 0.2, 0.5, 1.0, 2.0\}$ to tune $\rho$ and $\{0.1, 0.2, 0.5, 1.0, 2.0, 3.0, ..., 10.0\}$ for $\alpha$ using 10% of the training data as a validation set. For CIFAR-10 and CIFAR-100, we set both $\rho$ and $\alpha$ to 0.1.

As a gradient regularizer, GNR can be integrated with current optimizers such as SGD and Adam (Keskar et al., 2017). We also show that GNR can be combined with sharpness-aware training procedures such as SAM. As shown in Section 2.3, the GNR term bounds the regularization term in SAM. Yet the practical implementations of GNR and SAM rely on first-order Taylor expansion of different objective functions (GNR approximates the maximum gradient norm while SAM approximates the maximum loss). We empirically show that the combination of GNR and SAM outperforms both of them, indicating that they may strengthen each other with omitted items.

As shown in Table 1, GNR improves generalization for all models on CIFAR-10 and CIFAR-100. When combined with SGD, GNR achieves considerably higher test accuracy compared with SGD. Moreover, GNR further improves generalization combined with SAM. For example, GNR improves SAM performance by 1.18% and 1.10% on CIFAR-100 with ResNet-18 and ResNet-101, respectively, which are noticeable margins.

Table 2: Results of GNR with ResNet50 on ImageNet.

| Dataset | SGD | SGD + GNR (Ours) | SAM | SAM + GNR (Ours) |
|---|---|---|---|---|
| Top-1 | $75.53_{\pm 0.16}$ | $\mathbf{76.05}_{\pm 0.12}$ | $76.10_{\pm 0.10}$ | $\mathbf{76.56}_{\pm 0.11}$ |
| Top-5 | $92.59_{\pm 0.06}$ | $\mathbf{92.85}_{\pm 0.08}$ | $92.92_{\pm 0.08}$ | $\mathbf{93.19}_{\pm 0.07}$ |

Table 3: Results of GNR for finetuning EfficientNet-b0 and Swin Transformers on various datasets.

| | EfficientNet-b0 | | | | Swin-t | | | |
|---|---|---|---|---|---|---|---|---|
| Dataset | SGD | SGD + GNR | SAM | SAM + GNR | AdamW | AdamW + GNR | SAM | SAM + GNR |
| Stanford Cars | 82.14 | **83.54** | 83.21 | **83.45** | 83.50 | **84.70** | 83.55 | **84.11** |
| CIFAR-10 | 86.26 | **87.37** | 86.95 | **87.13** | 91.32 | **91.71** | 91.77 | **91.87** |
| CIFAR-100 | 63.75 | **64.59** | 63.52 | **63.64** | 72.88 | **73.30** | 73.05 | **73.10** |
| Oxford_IIIT_Pets | 91.03 | **91.66** | **91.36** | 91.09 | 93.49 | **93.62** | 93.45 | **93.68** |
| Food101 | 82.54 | **82.69** | 82.57 | **83.01** | 86.38 | **86.89** | 86.64 | **87.03** |

#### 3.2.2 IMAGENET

We use ResNet50 (He et al., 2016) for evaluations on ImageNet (Russakovsky et al., 2015) to evaluate GNR on large scale data. Following previous works (He et al., 2016), we resize and crop images to 224×224 resolution and normalize them to $N(0, 1)$. We set the batch size to 256, learning rate to 0.1, and weight decay to 0.0001. The learning rate is decayed by the factor of 0.1 every 30 epochs. As shown in Table 2, GNR consistently improves SGD performance on ImageNet. GNR also further improves the model generalization compared with SAM. The combination of GNR and SAM outperforms SGD and SAM.

### 3.3 TRANSFER LEARNING

Transfer learning shows the generalization of models when trained on sufficient labeled data and finetuned on a small dataset (Zhuang et al., 2020). We show that GNR improves generalization on all datasets in this setting.

We consider Stanford Cars (Krause et al., 2013), CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), Oxford_IIIT_Pets (Parkhi et al., 2012) and Food101 (Bossard et al., 2014) for this setting. We apply SGD, SAM, and GNR to finetuning EfficientNet-b0 (Tan & Le, 2019) and Swin-Transformer-t (Liu et al., 2021) on these datasets. Both EfficientNet-b0 and Swin-Transformer-t are pretrained on ImageNet.

We use ImageNet pretrained weights of EfficientNet-b0 and Swin-t except for the last linear layer for classification. Following previous works, we train for 40k steps since our batch size is 128. The initial learning rate is set to 2e-3 with cosine learning rate decay. Weight decay is set to 1e-5. We do not use any data augmentations for Stanford Cars, Oxford_IIIT_Pets and Food101. For CIFAR datasets, we employ the same data augmentations as previous experiments.

As seen in Table 3, GNR once again brings generalization improvement for SGD, AdamW, and SAM on both EfficientNet-b0 and Swin-t. For example, GNR improves AdamW by 1.2% on Stanford Cars with Swin-t and 1.11% on CIFAR-10 with EfficientNet-b0.

### 3.4 TOP EIGENVALUES OF HESSIAN AND HESSIAN TRACE

Proposition 2.1 shows that the GNR term can be an equivalent measure of the maximum eigenvalue of the Hessian, which is a well-known measure of flatness/sharpness. Thus optimizing the GNR term decreases the maximum eigenvalue of the Hessian and leads to flatter minima. To empirically validate that GNR finds optima with low curvature, we present the Hessian spectra of SGD, SAM, and GNR. We consider the maximum eigenvalue of Hessian, which measures the worst-case loss increase under an adversarial perturbation to the weights (Keskar et al., 2017) and the Hessian trace, which measures the expected loss increase under random perturbations to the weights (Kaur et al., 2022) as the measures of flatness. We empirically show that GNR significantly decreases both the

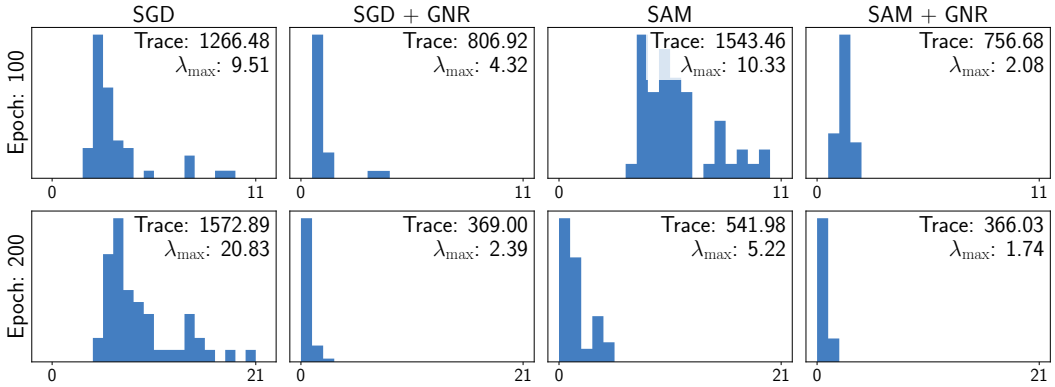



Figure 2: The distribution of top eigenvalues and the trace of Hessian at epoch 100 and 200 on CIFAR-100 with SGD, SGD + GNR, SAM, or SAM + GNR.

maximum eigenvalue and trace of Hessian during training compared with SGD and SAM, and thus finds flatter minima.

We compute the Hessian spectra of ResNet-18 trained on CIFAR-100 for 200 epochs with SGD, SAM, SGD + GNR, and SAM + GNR. We use power iteration (Yao et al., 2018) to compute the top eigenvalues of Hessian and Hutchinson's method (Avron & Toledo, 2011; Bai et al., 1996; Yao et al., 2020) to compute the Hessian trace. We report the histogram of the distribution of the top-50 Hessian eigenvalues for each method.

As shown in Figure 2, the model trained with SGD has a higher maximum Hessian eigenvalue and Hessian trace at convergence compared to the middle of training, indicating that optimizing directly with cross-entropy loss does not contribute to the lower Hessian spectra. In contrast, GNR leads to lower Hessian spectra and thus flatter minima. Moreover, GNR helps to reduce both top eigenvalues and the Hessian trace when combined with SAM, where Hessian spectra at convergence are lower than other methods. We show visualizations of landscapes of SGD, SAM, and GNR in Appendix C.

### 3.5 Computation Overhead

As discussed in Section 2.2, the GNR term can be easily calculated via the Hessian vector product, which is an efficient approach to calculating the dot product between the Hessian and a vector without the need to calculate the entire Hessian. However, it can still introduce extra computation when calculated in each iteration. To accelerate the training with GNR, we investigate applying GNR to only a few iterations in each epoch. Surprisingly, we show that only several iterations of learning with GNR (with higher $\alpha$ compared with applying GNR to all iterations) improve model generalization considerably. As shown in Table 4, with approximately 1/20 of iterations, GNR improves test accuracy for both SGD and SAM on CIFAR-10 and CIFAR-100. When applying GNR to 1/10 iterations of training, it shows similar effectiveness to applying GNR to all the iterations, while the extra computational cost for GNR is less than 25% of the original cost. Thus the computation overhead of GNR can be easily controlled.

## 4 Related Works

### 4.1 Optimizer

For accommodating numerous distinct learning tasks, researchers have proposed many optimizers with different properties, such as SGD, Adam (Kingma & Ba, 2015), Adai (Xie et al., 2022b), AdamW (Loshchilov & Hutter, 2017), AdaBound (Luo et al., 2019), Padam (Chen et al., 2018), RAdam (Liu et al., 2020), Yogi (Zaheer et al., 2018) and Adagrad (Duchi et al., 2011). SGD iteratively updates the parameters of deep neural networks by computing the gradient of the loss function with the randomly sampled batches. Adam optimizes the deep neural networks with adaptive learning rate and momentum based on SGD, therefore can make the training procedure faster. However, some previous literature find that Adam is more vulnerable to sharp minima than SGD (Wilson et al.,

Table 4: Accuracy and training speed of training with different ratios of iterations using GNR. Numbers in parentheses indicate the ratio of the training speed compared with the vanilla base optimizer SGD/SAM.

| | | | | | | |
|---|---|---|---|---|---|---|
| CIFAR-10 | | SGD | SGD + GNR$^{0.05}$ | SGD + GNR$^{0.1}$ | SGD + GNR$^{0.5}$ | SGD + GNR$^{1}$ |
| | Accuracy | 95.32 | 96.08 | 96.15 | 96.17 | 96.17 |
| | Images/s | 2,593 (100%) | 2,258 (87%) | 1,996 (77%) | 1,023 (39%) | 658 (25%) |
| | | SAM | SAM + GNR$^{0.05}$ | SAM + GNR$^{0.1}$ | SAM + GNR$^{0.5}$ | SAM + GNR$^{1}$ |
| | Accuracy | 96.10 | 96.54 | 96.62 | 96.65 | 96.58 |
| | Images/s | 1,314 (100%) | 1,247 (95%) | 1,184 (90%) | 858 (65%) | 629 (48%) |
| CIFAR-100 | | SGD | SGD + GNR$^{0.05}$ | SGD + GNR$^{0.1}$ | SGD + GNR$^{0.5}$ | SGD + GNR$^{1}$ |
| | Accuracy | 78.32 | 79.25 | 79.42 | 79.50 | 79.53 |
| | Images/s | 2,609 (100%) | 2,243 (86%) | 1,955 (75%) | 1,011 (39%) | 655 (25%) |
| | | SAM | SAM + GNR$^{0.05}$ | SAM + GNR$^{0.1}$ | SAM + GNR$^{0.5}$ | SAM + GNR$^{1}$ |
| | Accuracy | 79.27 | 80.08 | 80.44 | 80.40 | 80.45 |
| | Images/s | 1,318 (100%) | 1,251 (95%) | 1,172 (89%) | 848 (64%) | 628 (48%) |

2017), which results in worse generalization ability (Xie et al., 2022a; Hardt et al., 2016; Hochreiter & Schmidhuber, 1994). To overcome the generalization problem of Adam while maintaining the fast convergence speed, Xie et al. (2022b) propose Adai algorithm which focuses on adjusting the hyperparameters of momentum rather than learning rate.

### 4.2 Flat minima

The vanilla training procedure aims to search for a single parameter point that achieves a low loss value in the training dataset. However, when the model is developed for the testing dataset, the potential shift of the loss function landscape may lead to a drop of the generalization performance (Keskar et al., 2017). This phenomenon can be more severe when the neighborhood of minima is sharper. Therefore, some literature points out the necessity of seeking flat minima (Keskar et al., 2017; Zhuang et al., 2022).

Recently, Kaur et al. (2022) thoroughly reviews the literature related to generalization and sharpness of minima. It highlights the role of maximum Hessian eigenvalue in deciding the sharpness of minima (Keskar et al., 2017; Wen et al., 2019). And there also have been several simple strategies to achieve a smaller maximum Hessian eigenvalue, such as choosing a large learning rate (Lewkowycz et al., 2020; Cohen et al., 2021; Jastrzebski et al., 2020) and smaller batch size (Smith & Le, 2018; Lewkowycz et al., 2020; Jastrzębski et al., 2017).

Some previous works have been proposed to evaluate the sharpness of minima and minimize it. Sharpness-Aware Minimization (SAM) (Foret et al., 2021) and its variants (Zhuang et al., 2022; Kwon et al., 2021; Du et al., 2021; Liu et al., 2022; Du et al., 2022) are representative training algorithm to seek flat minima for better generalization. Particularly, SAM aims to minimize the perturbed loss, which is defined as the maximum loss in the neighborhood. However, the lower perturbed loss does not exactly imply flatter minima. To eliminate this inconsistency, Zhuang et al. (2022) proposes a two-step method GSAM to simultaneously minimize the perturbed loss and surrogate gap. Kwon et al. (2021) introduce the adaptive sharpness concept as the substitute and propose a learning algorithm based on it that eliminates the influence of scale dependency. In addition, Du et al. (2021) and Liu et al. (2022) respectively propose LookSAM and ESAM to improve the computation efficiency based on original SAM. Du et al. (2022) design SAF and MESA with a trajectory loss as a target which almost does not require extra computations.

## 5 Conclusion

We proposed a novel regularizer named Gradient Norm Regularizer (GNR) to seek minima with uniformly small curvature across all directions and measure sharpness when SAM fails. We showed that GNR bounded both the maximum eigenvalue of the Hessian and the regularization function of SAM. We empirically showed that GNR improved generalization for SGD, AdamW, and SAM.

## REFERENCES


Zeyuan Allen-Zhu and Yuanzhi Li. Neon2: Finding local minima via first-order oracles. *Advances in Neural Information Processing Systems*, 31, 2018.

Haim Avron and Sivan Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM (JACM)*, 58(2):1–34, 2011.

Zhaojun Bai, Gark Fahey, and Gene Golub. Some large-scale matrix computation problems. *Journal of Computational and Applied Mathematics*, 74(1-2):71–89, 1996.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12): 124018, 2019.

Jinghui Chen, Dongruo Zhou, Yiqi Tang, Ziyan Yang, Yuan Cao, and Quanquan Gu. Closing the generalization gap of adaptive gradient methods in training deep neural networks. *arXiv preprint arXiv:1806.06763*, 2018.

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. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=jh-rTtvkGeM`.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Jiawei Du, Hanshu Yan, Jiashi Feng, Joey Tianyi Zhou, Liangli Zhen, Rick Siow Mong Goh, and Vincent YF Tan. Efficient sharpness-aware minimization for improved training of neural networks. *arXiv preprint arXiv:2110.03141*, 2021.

Jiawei Du, Daquan Zhou, Jiashi Feng, Vincent YF Tan, and Joey Tianyi Zhou. Sharpness-aware training for free. *arXiv preprint arXiv:2205.14083*, 2022.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

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

Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5927–5935, 2017.

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

Haowei He, Gao Huang, and Yang Yuan. Asymmetric valleys: Beyond sharp and flat local minima. *Advances in neural information processing systems*, 32, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. *Advances in neural information processing systems*, 7, 1994.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho, and Krzysztof J. Geras. The break-even point on optimization trajectories of deep neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=r1g87C4KwB.

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.

Simran Kaur, Jeremy Cohen, and Zachary C Lipton. On the maximum hessian eigenvalue and generalization. *arXiv preprint arXiv:2206.10654*, 2022.

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. In *International Conference on Learning Representations*, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *International Conference on Learning Representations*, 2015.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009.

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, pp. 5905–5914. PMLR, 2021.

Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pp. 1302–1338, 2000.

Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv preprint arXiv:2003.02218*, 2020.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rkgz2aEKDr.

Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12360–12370, 2022.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. *arXiv preprint arXiv:1902.09843*, 2019.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18:1–35, 2017.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

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

Samuel L. Smith and Quoc V. Le. A bayesian perspective on generalization and stochastic gradient descent. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=BJij4yg0Z.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Yeming Wen, Kevin Luk, Maxime Gazeau, Guodong Zhang, Harris Chan, and Jimmy Ba. An empirical study of large-batch stochastic gradient descent with structured covariance noise. *arXiv preprint arXiv:1902.08234*, 2019.

Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. *Advances in neural information processing systems*, 30, 2017.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Zeke Xie, Issei Sato, and Masashi Sugiyama. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *International Conference on Learning Representations*, 2021.

Zeke Xie, Qian-Yuan Tang, Yunfeng Cai, Mingming Sun, and Ping Li. On the power-law spectrum in deep learning: A bridge to protein science. *arXiv preprint arXiv:2201.13011*, 2022a.

Zeke Xie, Xinrui Wang, Huishuai Zhang, Issei Sato, and Masashi Sugiyama. Adaptive inertia: Disentangling the effects of adaptive learning rate and momentum. In *International Conference on Machine Learning*, pp. 24430–24459. PMLR, 2022b.

Yi Xu, Rong Jin, and Tianbao Yang. First-order stochastic algorithms for escaping from saddle points in almost linear time. *Advances in neural information processing systems*, 31, 2018.

Zhewei Yao, Amir Gholami, Qi Lei, Kurt Keutzer, and Michael W Mahoney. Hessian-based analysis of large batch training and robustness to adversaries. *Advances in Neural Information Processing Systems*, 31, 2018.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, pp. 581–590. IEEE, 2020.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. *Advances in neural information processing systems*, 31, 2018.

Guodong Zhang, Lala Li, Zachary Nado, James Martens, Sushant Sachdeva, George Dahl, Chris Shallue, and Roger B Grosse. Which algorithmic choices matter at which batch sizes? insights from a noisy quadratic model. *Advances in neural information processing systems*, 32, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1): 43–76, 2020.

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

# A OMITTED DETAILS IN SECTION 2

## A.1 DERIVATION OF EQUATION 5

We follow the steps in (Foret et al., 2021) to approximate

$$\nabla R_\rho^{\text{GNR}}(\boldsymbol{\theta}) = \rho \cdot \nabla_{\boldsymbol{\theta}} \max_{\boldsymbol{\epsilon} \in B(0,\rho)} \left\| \nabla \hat{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon}) \right\|. \tag{11}$$

We first conduct the first-order Taylor expansion of $\|\nabla \hat{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon})\|$ and get that

$$\begin{aligned} \boldsymbol{\epsilon}^*(\boldsymbol{\theta}) &= \arg\max_{\boldsymbol{\epsilon} \in B(0,\rho)} R_\rho^{\text{GNR}}(\boldsymbol{\theta} + \boldsymbol{\epsilon}) \approx \arg\max_{\boldsymbol{\epsilon} \in B(0,\rho)} \left\| \nabla \hat{L}(\boldsymbol{\theta}) \right\| + \left( \nabla \left\| \nabla \hat{L}(\boldsymbol{\theta}) \right\| \right)^\top \boldsymbol{\epsilon} \\ &= \arg\max_{\boldsymbol{\epsilon} \in B(0,\rho)} \left( \nabla \left\| \nabla \hat{L}(\boldsymbol{\theta}) \right\| \right)^\top \boldsymbol{\epsilon} = \frac{\rho \cdot \boldsymbol{f}}{\|\boldsymbol{f}\|}, \end{aligned} \tag{12}$$

where $\boldsymbol{f} = \nabla \|\nabla \hat{L}(\boldsymbol{\theta})\|$.

As a result, by letting $\boldsymbol{\theta}^{\text{adv}} = \boldsymbol{\theta} + \epsilon^*(\boldsymbol{\theta})$,

$$\nabla R_\rho^{\text{GNR}}(\boldsymbol{\theta}) \approx \rho \cdot \nabla_{\boldsymbol{\theta}} \left\| \nabla \hat{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon}^*(\boldsymbol{\theta})) \right\| = \rho \cdot \nabla \left\| \nabla \hat{L}(\boldsymbol{\theta}^{\text{adv}}) \right\| + \rho \cdot \nabla \frac{\mathrm{d}\boldsymbol{\epsilon}^*(\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{\theta}} \cdot \left\| \nabla \hat{L}(\boldsymbol{\theta}^{\text{adv}}) \right\|. \tag{13}$$

In addition, similar to (Foret et al., 2021), we further drop the second-order term to accelerate the computation. Finally, the derivative $\nabla R_\rho^{\text{GNR}}(\boldsymbol{\theta})$ is given by

$$\nabla R_\rho^{\text{GNR}}(\boldsymbol{\theta}) \approx \rho \cdot \nabla \left\| \nabla \hat{L}(\boldsymbol{\theta}^{\text{adv}}) \right\|, \quad \boldsymbol{\theta}^{\text{adv}} = \boldsymbol{\theta} + \rho \cdot \frac{\boldsymbol{f}}{\|\boldsymbol{f}\|}, \quad \boldsymbol{f} = \nabla \left\| \nabla \hat{L}(\boldsymbol{\theta}) \right\|. \tag{14}$$

# B PROOFS

## B.1 PROOF OF PROPOSITION 2.1

*Proof.* By assumption, we have that for all $\boldsymbol{\theta} \in B(\boldsymbol{\theta}^*, \rho)$,

$$\hat{L}(\boldsymbol{\theta}) = \hat{L}(\boldsymbol{\theta}^*) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top \nabla^2 \hat{L}(\boldsymbol{\theta}^*)(\boldsymbol{\theta} - \boldsymbol{\theta}^*). \tag{15}$$

In addition,

$$\nabla \hat{L}(\boldsymbol{\theta}) = \nabla^2 \hat{L}(\boldsymbol{\theta}^*)(\boldsymbol{\theta} - \boldsymbol{\theta}^*). \tag{16}$$

As a result, according to the eigen-decomposition of $\nabla^2 \hat{L}(\boldsymbol{\theta}^*) = Q \Lambda Q^\top$, we have

$$\left\| \nabla \hat{L}(\boldsymbol{\theta}) \right\|^2 = (\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top \left( \nabla^2 \hat{L}(\boldsymbol{\theta}^*) \right)^2 (\boldsymbol{\theta} - \boldsymbol{\theta}^*) = (\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top Q \Lambda^2 Q^\top (\boldsymbol{\theta} - \boldsymbol{\theta}^*). \tag{17}$$

Let $\boldsymbol{\epsilon} = Q^\top(\boldsymbol{\theta} - \boldsymbol{\theta}^*) = (\epsilon_1, \epsilon_2, \ldots, \epsilon_d)$ and $\Lambda^2 = \operatorname{diag}(\lambda_1^2, \lambda_2^2, \ldots, \lambda_d^2)$. We have $\left\| \nabla \hat{L}(\boldsymbol{\theta}) \right\|^2 = \sum_{i=1}^d \epsilon_i^2 \lambda_i^2$. As a result,

$$\max_{\boldsymbol{\epsilon}: \sum_{i=1}^d \epsilon_i^2 \le \rho^2} \sum_{i=1}^d \epsilon_i^2 \lambda_i^2 = \rho^2 \lambda_{\max}^2. \tag{18}$$

Now the claim follows. □

## B.2 PROOF OF PROPOSITION 2.2

*Proof.* Define $h(\boldsymbol{\theta}) = \max_{\boldsymbol{\theta}' \in B(\boldsymbol{\theta}, \rho)} \left\| \nabla \hat{L}(\boldsymbol{\theta}) \right\|$. Fix $\sigma = \rho / (\sqrt{d} + \sqrt{\log n})$, we can obtain that with probability at least $1 - \delta$,

$$\mathbb{E}_{\epsilon_i \sim N(0,\sigma^2)} \left[ L(\boldsymbol{\theta} + \boldsymbol{\epsilon}) \right] \le \mathbb{E}_{\epsilon_i \sim N(0,\sigma^2)} \left[ \hat{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon}) \right] + \sqrt{\frac{\frac{1}{4} d \log\left(1 + \frac{\|\boldsymbol{\theta}\|_2^2}{d\sigma^2}\right) + \frac{1}{4} + \log \frac{n}{\delta} + 2 \log(6n + 3d)}{n - 1}}. \tag{19}$$

Since $\epsilon_i \sim N(0, \sigma^2)$, $\|\boldsymbol{\epsilon}\|^2/\sigma^2$ has a chi-square distribution. As a result, according to (Laurent & Massart, 2000, Lemma 1), we have that for any $t > 0$,

$$P\left(\|\boldsymbol{\epsilon}\|^2/\sigma^2 - d \geq 2\sqrt{dt} + 2t\right) \leq \exp(-t). \tag{20}$$

By letting $t = \frac{1}{2}\log n$, we can get that with probability at least $1 - 1/\sqrt{n}$,

$$\|\boldsymbol{\epsilon}\|^2 \leq \sigma^2\left(d + \sqrt{2d\log n} + \log n\right) \leq \sigma^2\left(\sqrt{d} + \sqrt{\log n}\right)^2 = \rho^2. \tag{21}$$

As a result,

$$\begin{aligned}
&\mathbb{E}_{\epsilon_i \sim N(0,\sigma^2)}\left[\hat{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon})\right] \\
\leq &\mathbb{E}_{\epsilon_i \sim N(0,\sigma^2)}\left[\hat{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon}) \mid \|\boldsymbol{\epsilon}\| \leq \rho\right] + \mathbb{E}_{\epsilon_i \sim N(0,\sigma^2)}\left[\hat{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon}) \mid \|\boldsymbol{\epsilon}\| > \rho\right] \\
\leq &\mathbb{E}_{\epsilon_i \sim N(0,\sigma^2)}\left[\hat{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon}) \mid \|\boldsymbol{\epsilon}\| \leq \rho\right] + \frac{M}{\sqrt{n}}.
\end{aligned} \tag{22}$$

According to the mean value theorem and Cauchy–Schwarz inequality, for any $\boldsymbol{\epsilon}$ such that $\|\boldsymbol{\epsilon}\| < \rho$, there exists a constant $0 \leq c \leq 1$, such that

$$\begin{aligned}
\hat{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon}) &= \hat{L}(\boldsymbol{\theta}) + \left(\nabla\hat{L}(\boldsymbol{\theta} + c\boldsymbol{\epsilon})\right)^\top \boldsymbol{\epsilon} \\
&\leq \hat{L}(\boldsymbol{\theta}) + \left\|\nabla\hat{L}(\boldsymbol{\theta} + c\boldsymbol{\epsilon})\right\| \cdot \|\boldsymbol{\epsilon}\| \\
&\leq \hat{L}(\boldsymbol{\theta}) + h(\boldsymbol{\theta})\rho.
\end{aligned} \tag{23}$$

Now the claim follows from Equations 19, 22, and 23. □

### B.3 Proof of Theorem 2.3

*Proof.* Observe that

$$\begin{aligned}
\left\|\nabla L^{\text{overall}}(\boldsymbol{\theta}_t)\right\|^2 &= \left\|\nabla L^{\text{oracle}}(\boldsymbol{\theta}_t) + \alpha\rho_t \cdot \nabla\left\|\nabla\hat{L}(\boldsymbol{\theta}_t^{\text{adv}})\right\|\right\|^2 \\
\leq 2\left(\left\|\nabla L^{\text{oracle}}(\boldsymbol{\theta}_t)\right\|^2 + \left\|\alpha\rho_t \cdot \nabla\left\|\nabla\hat{L}(\boldsymbol{\theta}_t^{\text{adv}})\right\|\right\|^2\right).
\end{aligned} \tag{24}$$

The claim follows from Propositions B.1 and B.2. □

**Proposition B.1.** *Assume the conditions in Theorem 2.3 hold (with parameters $\gamma_1, \gamma_2, G^{loss}, G^{norm}, \tilde{G}^{loss}, M, \eta_0, \rho_0, \alpha$). Then with learning rate $\eta_t = \eta_0/\sqrt{t}$ and perturbation radius $\rho_t = \rho_0/\sqrt{t}$, Algorithm 1 could obtain*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla L^{oracle}(\boldsymbol{\theta}_t)\right\|^2\right] \leq \frac{C_1' + C_2'\log T}{\sqrt{T}} \tag{25}$$

*for some constants $C_1'$ and $C_2'$ that only depend on $\gamma_1, \gamma_2, G^{loss}, G^{norm}, \tilde{G}^{loss}, M, \eta_0, \rho_0, \alpha$.*

*Proof.* By definition, we have $\boldsymbol{h}_t^{\text{loss}} = \tilde{g}_t^{\text{loss}}(\boldsymbol{\theta}_t)$ and $\boldsymbol{h}_t^{\text{norm}} = g_t^{\text{norm}}(\boldsymbol{\theta}_t^{\text{adv}})$. By assumption,

$$\begin{aligned}
L^{\text{oracle}}(\boldsymbol{\theta}_{t+1}) &\leq L^{\text{oracle}}(\boldsymbol{\theta}_t) + \left(\nabla L^{\text{oracle}}(\boldsymbol{\theta}_t)\right)^\top(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t) + \frac{\gamma_1}{2}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2 \\
&= L^{\text{oracle}}(\boldsymbol{\theta}_t) - \eta_t\left(\nabla L^{\text{oracle}}(\boldsymbol{\theta}_t)\right)^\top\left(\boldsymbol{h}_t^{\text{loss}} + \alpha\rho_t\boldsymbol{h}_t^{\text{norm}}\right) + \frac{\gamma_1\eta_t^2}{2}\left\|\boldsymbol{h}_t^{\text{loss}} + \alpha\rho_t\boldsymbol{h}_t^{\text{norm}}\right\|^2.
\end{aligned} \tag{26}$$

Take the expectation conditioned on the observations till timestamp $t$. By the assumption $\mathbb{E}[\boldsymbol{h}_t^{\text{loss}}] = \mathbb{E}[\tilde{g}_t^{\text{loss}}(\boldsymbol{\theta}_t)] = \nabla L^{\text{oracle}}(\boldsymbol{\theta}_t)$ and $\mathbb{E}[\boldsymbol{h}_t^{\text{norm}}] = \mathbb{E}[g_t^{\text{norm}}(\boldsymbol{\theta}_t^{\text{adv}})] = \nabla^2\hat{L}(\boldsymbol{\theta}^{\text{adv}}) \cdot \frac{\nabla\hat{L}(\boldsymbol{\theta}^{\text{adv}})}{\left\|\nabla\hat{L}(\boldsymbol{\theta}^{\text{adv}})\right\| + \xi}$, we can

obtain that

$$\begin{aligned}
&\mathbb{E}\left[L^{\text{oracle}}(\boldsymbol{\theta}_{t+1})\right] - L^{\text{oracle}}(\boldsymbol{\theta}_t) \\
\leq & -\eta_t \left\|\nabla L^{\text{oracle}}(\boldsymbol{\theta}_t)\right\|^2 - \eta_t\rho_t\alpha\mathbb{E}\left[\left(\nabla L^{\text{oracle}}(\boldsymbol{\theta}_t)\right)^\top \left(\nabla^2\hat{L}(\boldsymbol{\theta}^{\text{adv}}) \cdot \frac{\nabla\hat{L}(\boldsymbol{\theta}^{\text{adv}})}{\left\|\nabla\hat{L}(\boldsymbol{\theta}^{\text{adv}})\right\| + \xi}\right)\right] \\
&+ \frac{\gamma_1\eta_t^2}{2}\left\|\boldsymbol{h}_t^{\text{loss}} + \alpha\rho_t\boldsymbol{h}_t^{\text{norm}}\right\|^2
\end{aligned} \tag{27}$$

Because $\hat{L}$ is $\gamma_2$-Lipschitz smooth, the maximal absolute eigenvalue of $\nabla^2\hat{L}(\boldsymbol{\theta})$ is smaller than $\gamma_2$ and $\|\nabla^2\hat{L}(\boldsymbol{\theta})\| \leq \gamma_2$ for any $\boldsymbol{\theta} \in \Theta$. As a result,

$$\begin{aligned}
&= -\eta_t\rho_t\alpha\mathbb{E}\left[\left(\nabla L^{\text{oracle}}(\boldsymbol{\theta}_t)\right)^\top \left(\nabla^2\hat{L}(\boldsymbol{\theta}_t^{\text{adv}})\frac{\nabla\hat{L}(\boldsymbol{\theta}_t^{\text{adv}})}{\left\|\nabla\hat{L}(\boldsymbol{\theta}_t^{\text{adv}})\right\| + \xi}\right)\right] \\
&\leq \eta_t\rho_t\alpha\mathbb{E}\left[\left\|\nabla L^{\text{oracle}}(\boldsymbol{\theta}_t)\right\|\left\|\nabla^2\hat{L}(\boldsymbol{\theta}_t^{\text{adv}})\right\|\left\|\frac{\nabla\hat{L}(\boldsymbol{\theta}_t^{\text{adv}})}{\left\|\nabla\hat{L}(\boldsymbol{\theta}_t^{\text{adv}})\right\| + \xi}\right\|\right] \\
&\leq \eta_t\rho_t\alpha\tilde{G}^{\text{loss}}\gamma.
\end{aligned} \tag{28}$$

In addition,

$$\mathbb{E}\left[\left\|\boldsymbol{h}_t^{\text{loss}} + \alpha\boldsymbol{h}_t^{\text{norm}}\right\|^2\right] \leq \mathbb{E}\left[\left\|\boldsymbol{h}_t^{\text{loss}}\right\|^2\right] + \alpha^2\mathbb{E}\left[\left\|\boldsymbol{h}_t^{\text{norm}}\right\|^2\right] \leq \left(\tilde{G}^{\text{loss}}\right)^2 + \alpha^2\left(G^{\text{norm}}\right)^2. \tag{29}$$

Combining Equations 27, 28, and 29, we can get that

$$\eta_t\left\|\nabla L^{\text{oracle}}(\boldsymbol{\theta}_t)\right\|^2 \leq -\mathbb{E}\left[L^{\text{oracle}}(\boldsymbol{\theta}_{t+1})\right] + L^{\text{oracle}}(\boldsymbol{\theta}_t) + \eta_t\rho_t Z_1 + \eta_t^2 Z_2 \tag{30}$$

for some constants $Z_1$ and $Z_2$ that only depend on $\gamma, G^{\text{loss}}, G^{\text{norm}}, \tilde{G}^{\text{loss}}, \alpha$. Now perform telescope sum and take the expectations at each step, we can obtain that

$$\sum_{t=1}^{T}\eta_t\left\|\nabla L^{\text{oracle}}(\boldsymbol{\theta}_t)\right\|^2 \leq -\mathbb{E}\left[L^{\text{oracle}}(\boldsymbol{\theta}_{T+1})\right] + L^{\text{oracle}}(\boldsymbol{\theta}_1) + Z_1\sum_{t=1}^{T}\eta_t\rho_t + Z_2\sum_{t=1}^{T}\eta_t^2. \tag{31}$$

By letting $\eta_t = \eta_0/\sqrt{t}$ and $\alpha = \alpha_0/\sqrt{t}$, we can get that

$$\begin{aligned}
\frac{\eta_0}{\sqrt{T}}\sum_{t=1}^{T}\left\|\nabla L^{\text{oracle}}(\boldsymbol{\theta}_t)\right\|^2 &\leq \sum_{t=1}^{T}\eta_t\left\|\nabla L^{\text{oracle}}(\boldsymbol{\theta}_t)\right\|^2 \\
&\leq -\mathbb{E}\left[L^{\text{oracle}}(\boldsymbol{\theta}_{T+1})\right] + L^{\text{oracle}}(\boldsymbol{\theta}_1) + Z_1\sum_{t=1}^{T}\eta_t\rho_t + Z_2\sum_{t=1}^{T}\eta_t^2 \\
&\leq 2M + Z_1\eta_0\rho_0\sum_{t=1}^{T}\frac{1}{t} + Z_2\eta_0^2\sum_{t=1}^{T}\frac{1}{t} \\
&\leq Z_4 + Z_5\log T
\end{aligned} \tag{32}$$

for some constants $Z_4$ and $Z_5$ that only depend on $\gamma, G^{\text{loss}}, G^{\text{norm}}, \tilde{G}^{\text{loss}}, M, \eta_0, \rho_0, \alpha$. Divide the two sides of the equation by $\eta_0\sqrt{T}$ and the claim follows. □

**Proposition B.2.** *Assume the conditions in Theorem 2.3 hold (with parameters $\gamma_1, \gamma_2, G^{loss}, G^{norm}, \tilde{G}^{loss}, M, \eta_0, \rho_0, \alpha$). Then with perturbation radius $\rho_t = \rho_0/\sqrt{t}$, Algorithm 1 could obtain*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\alpha\rho_t \cdot \nabla\left\|\nabla\hat{L}(\boldsymbol{\theta}^{adv})\right\|\right\|^2\right] \leq \frac{C_1'' + C_2''\log T}{\sqrt{T}} \tag{33}$$

*for some constants $C_1''$ and $C_2''$ that only depend on $\gamma_1, \rho_0, \alpha$.*

*Proof.* For any $t \in \{1, 2, \ldots, T\}$,

$$\begin{aligned}
&\mathbb{E}\left[\left\|\alpha\rho_t \cdot \nabla \left\|\nabla \hat{L}(\boldsymbol{\theta}^{\text{adv}})\right\|\right\|^2\right] = \alpha^2\rho_t^2 \mathbb{E}\left[\left\|\nabla \left\|\nabla \hat{L}(\boldsymbol{\theta}^{\text{adv}})\right\|\right\|^2\right] \\
&= \alpha^2\rho_t^2 \mathbb{E}\left[\left\|\nabla^2 \hat{L}(\boldsymbol{\theta}_t^{\text{adv}}) \frac{\nabla \hat{L}(\boldsymbol{\theta}_t^{\text{adv}})}{\left\|\nabla \hat{L}(\boldsymbol{\theta}_t^{\text{adv}})\right\|}\right\|\right] \leq \alpha^2\rho_t^2 \mathbb{E}\left[\left\|\nabla^2 \hat{L}(\boldsymbol{\theta}_t^{\text{adv}})\right\| \left\|\frac{\nabla \hat{L}(\boldsymbol{\theta}_t^{\text{adv}})}{\left\|\nabla \hat{L}(\boldsymbol{\theta}_t^{\text{adv}})\right\|}\right\|\right] \\
&\leq \alpha^2\rho_t^2 \mathbb{E}[\gamma_1] = \alpha^2\rho_t^2\gamma_1.
\end{aligned} \tag{34}$$

By letting $\rho_t = \rho_0/\sqrt{t}$,

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[\left\|\alpha\rho_t \cdot \nabla \left\|\nabla \hat{L}(\boldsymbol{\theta}^{\text{adv}})\right\|\right\|^2\right] \leq \frac{1}{T}\alpha^2\gamma\rho_0^2 \sum_{t=1}^{T} \frac{1}{t} \leq \frac{C_1'' + C_2'' \log T}{\sqrt{T}} \tag{35}$$

for some constants $C_1''$ and $C_2''$ that only depend on $\gamma_1, \rho_0, \alpha$. □

### B.4 Proof of Proposition 2.4

*Proof.* Suppose $\boldsymbol{\epsilon}^* = \arg\max_{\boldsymbol{\epsilon} \in B(0,\rho)} \hat{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon})$. Then $R_\rho^{\text{SAM}}(\boldsymbol{\theta}) = \hat{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon}^*) - \hat{L}(\boldsymbol{\theta})$. According to the mean value theorem, there exists a constant $0 \leq c \leq 1$ such that

$$\hat{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon}^*) - \hat{L}(\boldsymbol{\theta}) = \left(\nabla \hat{L}(\boldsymbol{\theta} + c \cdot \boldsymbol{\epsilon}^*)\right)^\top \boldsymbol{\epsilon}^*. \tag{36}$$

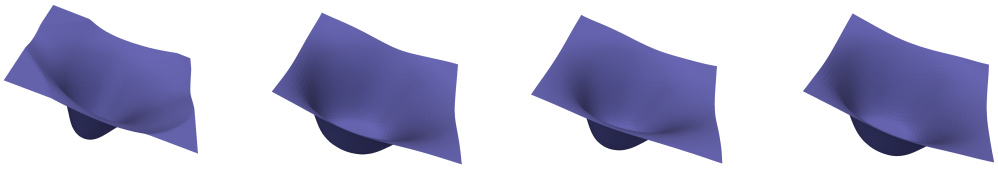
As a result,

$$\begin{aligned}
R_\rho^{\text{SAM}}(\boldsymbol{\theta}) = \hat{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon}^*) - \hat{L}(\boldsymbol{\theta}) &= \left(\nabla \hat{L}(\boldsymbol{\theta} + c \cdot \boldsymbol{\epsilon}^*)\right)^\top \boldsymbol{\epsilon}^* \leq \left\|\nabla \hat{L}(\boldsymbol{\theta} + c \cdot \boldsymbol{\epsilon}^*)\right\| \|\boldsymbol{\epsilon}^*\| \\
&\leq \max_{\boldsymbol{\epsilon} \in B(0,\rho)} \left\|\nabla \hat{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon})\right\| \cdot \rho = R_\rho^{\text{GNR}}(\boldsymbol{\theta}).
\end{aligned} \tag{37}$$

□

## C Visulization of Landscapes

We visualize the loss landscapes of models trained with SGD, SGD+GNR, SAM, SAM+GNR of the ResNet-18 model on CIFAR-100. All the models are trained with the same hyperparameters for 200 epochs as described in Section 3.2.1. As shown in Figure 3, GNR consistently helps SGD and SAM find flatter minima.



(a) SGD (b) SGD + GNR (c) SAM (d) SAM + GNR

Figure 3: Visualization of loss landscape for SGD, SGD+GNR, SAM, SAM+GNR