

---

# Implicitly regularized interaction between SGD and the loss landscape geometry

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

1 We study unstable dynamics of stochastic gradient descent (SGD) and its impact  
2 on generalization in neural networks. We find that SGD induces an implicit  
3 regularization on the interaction between the gradient distribution and the loss  
4 landscape geometry. Moreover, based on the analysis of a concentration measure of  
5 the batch gradient, we propose a more accurate scaling rule, Linear and Saturation  
6 Scaling Rule (LSSR), between batch size and learning rate.

## 7 1 Introduction

8 SGD plays an important role in the success of deep learning. However, we still do not fully understand  
9 how SGD works from the perspectives of both optimization behavior and generalization performance.  
10 To be specific, SGD is a stochastic approximation of full-batch gradient descent (GD), but SGD  
11 generally yields better generalization with a small batch size [27, 23]. Moreover, GD is a discretization  
12 of gradient flow (GF) with a finite learning rate, i.e., GF is a GD in the limit of vanishing learning  
13 rate, but GD generally performs better with a large learning rate [2, 32, 28, 43]. There are some  
14 *scaling rules* [25, 10, 15, 45, 54] on how to tune the learning rate for varying batch sizes, but they  
15 fail when the batch size gets large [42, 38, 57, 43, 33]. Especially for a greater data-parallelism to  
16 accelerate the training process, we require a more accurate scaling rule for the large-batch regime.

17 There has been many studies to understand the SGD dynamics and its impacts on generalization in  
18 deep neural networks. While they provide some useful and intuitive explanations to help us understand  
19 these properties of SGD, unfortunately, some results often rely on impractical assumptions or only  
20 apply to a certain range of learning rates and batch sizes. For example, some approximate SGD as a  
21 stochastic differential equation (SDE) in the limit of vanishing learning rate [34, 35, 29, 16, 30, 31,  
22 18, 44, 4]. Therefore, in a practical finite learning rate regime, this may not properly describe the  
23 SGD dynamics. Moreover, Yaida [52] raises some theoretical issues about the SDE approximation  
24 and Li et al. [33] theoretically analyze a sufficient condition for the SDE approximation to fail.

25 In this paper, we aim to understand the dynamics and the implicit bias of SGD through the analysis of  
26 the *interaction* between SGD and the loss landscape of a neural network with minimal assumptions.  
27 To be specific, we investigate the unstable dynamics of SGD “at the edge of stability” [6] (Section  
28 4.1-4.2). This investigation leads to a more refined characterization of the edge of stability by the  
29 *interaction-aware sharpness* which extends the previous findings for full-batch GD to a general SGD.  
30 Then, we introduce a *concentration measure* of the the batch gradient distribution of SGD. By doing  
31 so, we find that SGD implicitly regularizes the interaction-aware sharpness and its regularization  
32 effect is controlled by the ratio of the concentration measure to learning rate (Section 5.1). Finally,  
33 we propose a more accurate scaling rule between batch size and learning rate, based on a novel

analysis of the implicit regularization and the concentration measure (Section 5.2). This can be applied to any batch size including the large-batch regime where the previous scaling rules fail [18, 38, 57, 42, 43, 46]. We name it *Linear and Saturation Scaling Rule* (LSSR).

## 2 Stochastic Gradient and Loss Landscape

In this section, we review some concepts required for further discussion. We also summarize the notations in Appendix A for a quick reference. We often omit the dependence on some variables and the subscript of the expectation operation when clear from the context.

For a learning task, we use a parameterized model (neural network) with model parameter  $\theta \in \Theta \subset \mathbb{R}^m$ . Then we train the model using training data  $\mathcal{D} = \{x_i\}_{i=1}^n$  and a loss function  $\ell(x; \theta)$ . We denote the (total) training loss by  $L(\theta) \equiv \frac{1}{n} \sum_{i=1}^n \ell(x_i; \theta)$  for training data  $\mathcal{D}$ . At time step  $t$ , we update the parameter  $\theta_t$  using GD:  $\theta_{t+1} = \theta_t - \eta \nabla_{\theta} L(\theta_t)$  with a learning rate  $\eta$ , or using SGD:  $\theta_{t+1} = \theta_t - \eta g_b(\theta_t)$  with a mini-batch gradient  $g_b(\theta_t) \equiv \frac{1}{b} \sum_{x \in \mathcal{B}_t} \nabla_{\theta} \ell(x; \theta_t) \in \mathbb{R}^m$  for a mini-batch  $\mathcal{B}_t \subset \mathcal{D}$  of size  $b$  ( $1 \leq b \leq n$ ).

Now, we are ready to introduce some important matrices,  $C_b$ ,  $S_b$ , and  $H$ . First, we define the covariance  $C_b(\theta) \equiv \text{Var}[g_b(\theta)] = \mathbb{E}[(g_b(\theta) - \mathbb{E}[g_b(\theta)])(g_b(\theta) - \mathbb{E}[g_b(\theta)])^{\top}] \in \mathbb{R}^{m \times m}$  and the second moment  $S_b(\theta) \equiv \mathbb{E}[g_b(\theta)g_b(\theta)^{\top}] \in \mathbb{R}^{m \times m}$  of the mini-batch gradient  $g_b(\theta)$  over batch sampling for a batch size  $1 \leq b \leq n$ .<sup>1</sup> The covariance  $C_b$  and the second moment  $S_b$  satisfy not only  $C_b = S_b - S_n$  but also the following equation [15, 29, 49]:

$$C_b = \frac{\gamma_{n,b}}{b}(S_1 - S_n) = \frac{\gamma_{n,b}}{b}C_1, \quad (1)$$

where  $\gamma_{n,b} = \frac{n-b}{n-1}$  for sampling *without* replacement and  $\gamma_{n,b} = 1$  for sampling *with* replacement. We provide a self-contained proof of (1) in Appendix B.1. We note that, for sampling without replacement, many previous works approximate  $\gamma_{n,b} \approx 1$  assuming  $b \ll n$  [18, 15, 46], but we consider the whole range of  $1 \leq b \leq n$  ( $0 \leq \gamma_{n,b} \leq 1$  with  $\gamma_{n,1} = 1$  and  $\gamma_{n,n} = 0$ ). Second, we define the Hessian  $H(\theta) = \nabla_{\theta}^2 L(\theta) = \mathbb{E}_{x \sim \mathcal{D}}[\nabla_{\theta}^2 \ell(x; \theta)] \in \mathbb{R}^{m \times m}$  and the operator norm (the top eigenvalue)  $\|H\| \equiv \sup_{\|u\|=1} \|Hu\|$  of  $H$ . We also denote the  $i$ -th largest eigenvalue and its corresponding normalized eigenvector by  $\lambda_i \in \mathbb{R}$  and  $q_i \in \mathbb{R}^m$ , respectively, for  $i = 1, \dots, m$ .

Therefore, with these matrices, we can write one of our goals as follows:

*We aim to understand how the gradient distribution ( $C_b$  and  $S_b$ ) and the loss landscape geometry ( $H$ ) interact with each other during SGD training.*

We investigate this ‘‘interaction’’ in terms of matrix multiplication  $HS_b$ . To be specific, we consider the trace  $\text{tr}(HS_b)$  or its normalized one  $\frac{\text{tr}(HS_b)}{\text{tr}(S_b)}$  (will be denoted by  $\|H\|_{S_b}$  in Definition 2 later).

## 3 Related Work

Some studies investigate the interaction between the gradient distribution and the loss landscape geometry represented by  $\text{tr}(HS_b)$  in the context of escaping efficiency [58, Section 3.1], stationarity [52, Section 2.2], and convergence [48, Section 3.1.1]. However, they require some additional assumptions like SDE approximation of SGD [58], the existence of a stationary-state distribution of the model parameter [52, Section 2.3.4], and strong convexity of the training loss function [48], respectively. In this paper, we provide a new insight into the interaction  $\text{tr}(HS_b)$  without these assumptions.

Convergence of full-batch GD ( $b = n$ ) has been instead analyzed with an upper bound on the interaction  $\text{tr}(HS_n)$  with further assumptions for the stable optimization, such as  $\beta$ -smoothness of

<sup>1</sup>These two matrices  $C_b$  and  $S_b$  are also called the second *central* and *non-central* moments, respectively. But to avoid confusion, we use the term ‘‘second moment’’ only for the non-central  $S_b$ .

the objective and  $0 < \eta < \frac{2}{\beta}$  (e.g.,  $\eta = \frac{1}{\beta}$ ) [39, 41, 37, 3].<sup>2</sup> However, it may lose useful information of the interaction between  $H$  and  $S_n$ . Moreover, when we train a standard neural network with GD in practice,  $\|H\| (\leq \beta)$  increases in the early phase of training and the iterate enters the regime called the edge of stability [6] where  $\|H\| \gtrsim \frac{2}{\eta}$ , i.e.,  $\eta \gtrsim \frac{2}{\|H\|} \geq \frac{2}{\beta}$ . This contradicts with the assumption for stable optimization and the iterate exhibits unstable behavior with a non-monotonically decreasing loss [51, 50, 6]. We further extend this discussion of unstable dynamics for GD to the case of SGD.

From the generalization perspective, many studies focus on the implicit bias of SGD toward a better generalization [40, 56, 47, 20, 21, 1, 46]. There are mainly two factors known to correlate with the generalization performance: the batch gradient distribution during training [15, 18, 44, 58] and the sharpness of the loss landscape at the minimum [14, 23, 8, 22, 9, 26]. We provide a link between the batch gradient distribution and the sharpness that the model is implicitly regularized to have a low sharpness when the second moment of the batch gradient is large (see Section 5.1).

## 4 Optimization through Loss Landscape

We start by investigating the optimization behavior of SGD through the interaction between SGD and the loss landscape *without* the stochastic differential equation (SDE) approximation.

### 4.1 Unstable Optimization

Using the second-order Taylor expansion, the change in total training loss  $L_t = L(\theta_t)$  as the SGD iterate moves from  $\theta_t$  to  $\theta_{t+1}$  at time step  $t$  can be expressed as follows:

$$L_{t+1} - L_t = -\eta \nabla L^\top g_b + \frac{\eta^2}{2} g_b^\top H g_b + O(\|\delta_t\|^3), \quad (2)$$

where  $\delta_t = \theta_{t+1} - \theta_t = -\eta g_b$ . Thus, we obtain the expected loss difference as follows:

$$\mathbb{E}[L_{t+1}] - L_t = -\eta \nabla L^\top \mathbb{E}[g_b] + \frac{\eta^2}{2} \mathbb{E}[g_b^\top H g_b] + \epsilon \quad (3)$$

$$= -\eta \|\nabla L\|^2 + \frac{\eta^2}{2} \text{tr}(\mathbb{E}[H g_b g_b^\top]) + \epsilon \quad (4)$$

$$= -\eta \text{tr}(S_n) + \frac{\eta^2}{2} \text{tr}(H S_b) + \epsilon \quad (5)$$

$$= \frac{\eta^2}{2} \text{tr}(S_n) \left[ \frac{\text{tr}(H S_b)}{\text{tr}(S_n)} - \frac{2}{\eta} \right] + \epsilon, \quad (6)$$

where  $\epsilon = O(\mathbb{E}[\|\delta_t\|^3])$  and  $\mathbb{E}[g_b] = \nabla L$  is used. For the moment, we make a minimal assumption that the training loss is locally quadratic, i.e.,  $\epsilon = 0$  near  $\theta_t$ , but we will revisit this assumption later (see Section 4.2). Then, the expected loss increases when the following *instability condition* is met:

**Definition 1** (Instability Condition).

$$\frac{\text{tr}(H S_b)}{\text{tr}(S_n)} > \frac{2}{\eta}. \quad (7)$$

We also define *unstable regime*  $\mathbb{U} = \{\theta \in \Theta : \frac{\text{tr}(H S_b)}{\text{tr}(S_n)} > \frac{2}{\eta}\}$  and *stable regime*  $\mathbb{S} \equiv \mathbb{U}^c$ . For a standard non-quadratic loss function, we will show in the following sections that the iterate tends not to stay within the unstable regime  $\mathbb{U}$  and operates near at the boundary  $\partial \mathbb{S}$  of the stable regime  $\mathbb{S}$ , called the edge of stability [6]. Cohen et al. [6] mark the edge of stability with  $\{\theta \in \Theta : \|H\| = \frac{2}{\eta}\}$  for GD, but we mark with  $\partial \mathbb{S} = \{\theta \in \Theta : \frac{\text{tr}(H S_b)}{\text{tr}(S_n)} = \frac{2}{\eta}\}$  for both SGD and GD which provides a more clear and generalized indication as shown in Figure 4 later. On the other hand, for a globally quadratic loss, when the GD iterate satisfies the instability condition, it diverges within the unstable regime [6]. We emphasize that many studies on the convergence of GD usually consider the optimization within

---

<sup>2</sup>  $L(\theta_{t+1}) - L(\theta_t) \leq \nabla L^\top (\theta_{t+1} - \theta_t) + \frac{\beta}{2} \|\theta_{t+1} - \theta_t\|^2 = -\eta \|\nabla L\|^2 + \frac{\beta \eta^2}{2} \|\nabla L\|^2 = -\eta(1 - \frac{\beta \eta}{2}) \|\nabla L\|^2$  and thus the loss monotonically decreases when  $0 < \eta < \frac{2}{\beta}$ .

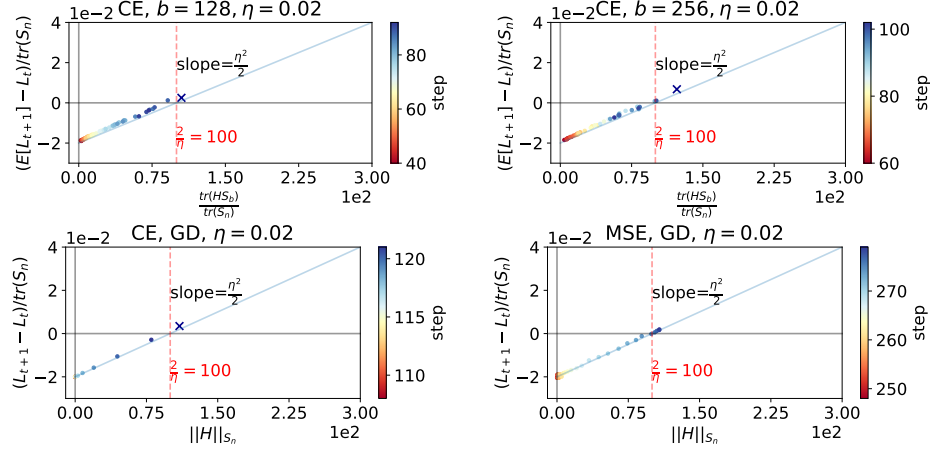


Figure 1: **[An empirical validation of (6) for SGD (top) and (9) for GD (bottom)]** In the early phase, until the iterate enters the edge of stability, it validates (6) and (9) with the blue line with the slope  $\frac{\eta^2}{2}$  and x-intercept  $\frac{2}{\eta}$ . For GD (bottom), they are plotted *after*  $\|H\|$  exceeds  $\frac{2}{\eta}$  after which  $\|H\|_{S_n}$  starts to increase from 0 to  $\frac{2}{\eta}$  in a few steps. For cross-entropy loss, we mark the end point with ‘x’ when the iterate enters the unstable regime. We train 6CNN with  $\eta = 0.02$ .

the stable regime [39, 41, 37, 3], but GD mostly occurs at the edge of stability after a few steps of training. We will argue that this behavior is crucial for generalization in neural networks.

For later use, we also define the *interaction-aware sharpness* as follows:

**Definition 2** (interaction-aware sharpness).

$$\|H\|_{S_b} \equiv \frac{\text{tr}(HS_b)}{\text{tr}(S_b)}. \quad (8)$$

Here,  $\text{tr}(HS_b) \leq \|H\| \text{tr}(S_b)$ , i.e.,  $\|H\|_{S_b} \leq \|H\|$ , and the equality holds only when every  $g_b$  is aligned in the direction of the top eigenvector of  $H$ .

Figure 1 (top row) empirically validates (6), showing the normalized loss difference  $\frac{\mathbb{E}[L_{t+1}] - L_t}{\text{tr}(S_n)}$  against  $\frac{\text{tr}(HS_b)}{\text{tr}(S_n)}$  in the early phase of training before entering the unstable regime. This result implies that the training loss  $L(\theta)$  is approximately locally quadratic, i.e.,  $\epsilon \approx 0$ , in the early phase. Especially, for full-batch GD ( $b = n$ ), the instability condition can be rewritten as  $\|H\|_{S_n} > \frac{2}{\eta}$  and we have the following relationship between the loss difference  $L_{t+1} - L_t$  and  $\|H\|_{S_n}$  from (6):

$$L_{t+1} - L_t = \frac{\eta^2}{2} \text{tr}(S_n) \left( \|H\|_{S_n} - \frac{2}{\eta} \right) + \epsilon. \quad (9)$$

Figure 1 (bottom row) shows  $\|H\|_{S_n}$  soars from 0 in a few steps after  $\|H\|$  exceeds  $\frac{2}{\eta}$  [6], satisfying (9) approximately with  $\epsilon \approx 0$ , before the iterate enters the edge of stability. This result is consistent with the following Proposition for a quadratic training loss  $L$ . The proof is deferred to Appendix B.2.

**Proposition 4.1.** *For GD with a quadratic  $L$ , if  $\|H\| > \frac{2}{\eta}$  and  $0 < \lambda_i < \frac{2}{\eta}$  for all  $i \neq 1$ , then  $|\cos(q_1, \nabla L(\theta_t))|$ ,  $|q_1^\top \nabla L(\theta_t)|$  and  $\|H\|_{S_n}$  increase to 1,  $\infty$  and  $\|H\|$ , respectively, as  $t \rightarrow \infty$ .*

## 4.2 Non-quadraticity, Asymmetric Valleys and the Edge of Stability

In the previous section, we have shown that the training loss is approximately locally quadratic *before* the iterate enters the edge of stability. However, *after* the iterate enters the edge of stability, i.e.,  $\frac{\text{tr}(HS_b)}{\text{tr}(S_n)}$  reaches and exceeds  $\frac{2}{\eta}$ , the step size is relatively large for the sharp loss landscape so that the iterate jumps across the valley [19], and the higher-order terms  $\epsilon$  in (6) and (9) become non-negligible and cause a different behavior of the iterate than in the stable regime.

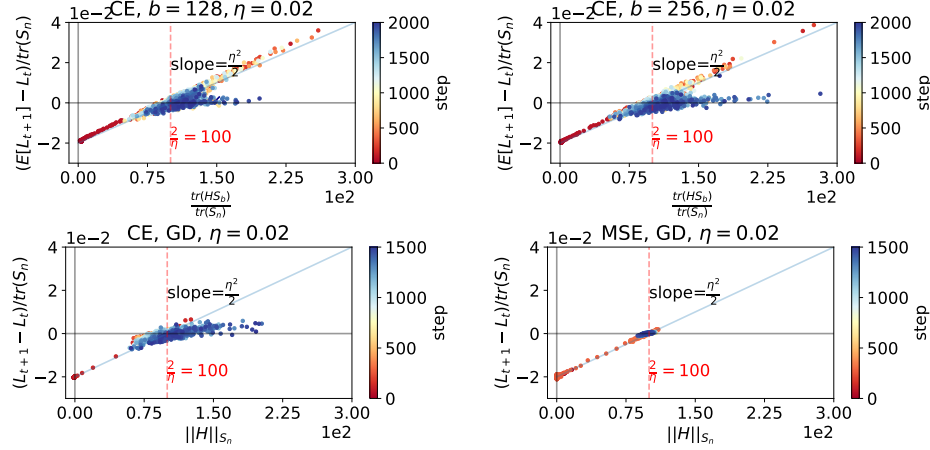


Figure 2: **[Non-quadraticity and overestimation]** The normalized loss difference  $\frac{\mathbb{E}[L_{t+1}] - L_t}{\text{tr}(S_n)}$  against  $\frac{\text{tr}(HS_b)}{\text{tr}(S_n)}$  during training. After the iterate enters the edge of stability, it often shows a more gentle slope than  $\frac{\eta^2}{2}$ , especially in the unstable regime.

Figure 2 shows empirical evidences for the *non-quadraticity*. After the SGD/GD iterate enters the edge of stability, when the instability condition  $\frac{\text{tr}(HS_b)}{\text{tr}(S_n)} > \frac{2}{\eta}$  is met, the normalized increase in the loss  $\left| \frac{\mathbb{E}[L_{t+1}] - L_t}{\text{tr}(S_n)} \right|$  is often smaller than  $\frac{\eta^2}{2} \left| \frac{\text{tr}(HS_b)}{\text{tr}(S_n)} - \frac{2}{\eta} \right|$  from (6) and (9) (blue line) when assuming a locally quadratic function. This results in a gentle slope less than  $\frac{\eta^2}{2}$ .

We hypothesise that due to this non-quadraticity of the training loss, the iterate is discouraged from staying within the unstable regime. Figure 3 demonstrates the asymmetric valley [12] that one side is sharp and the other is flat. In Figure 3 (left), we evaluate the directional sharpness  $\|H_\alpha\|_{S_n}$  along the gradient descent direction  $-\eta \nabla L(\theta)$  where  $H_\alpha \equiv H(\theta - \alpha \eta \nabla L(\theta))$  for  $\alpha \in \frac{1}{4} \times [1, 2, 3, 4, 5]$ , and compare  $\|H_\alpha\|_{S_n}(\theta)$  with  $\|H\|_{S_n}(\theta)$ . At the sharp side, it has a high  $\|H\|_{S_n} > \frac{2}{\eta}$  (blue) with the gradient  $\nabla L$  and the top eigenvector  $q_1(H)$  of the Hessian being highly aligned (cf. Prop. 4.1). However, when the loss landscape gets far from being quadratic, the Hessian and its top eigenvector can change abruptly,  $q_1(H_\alpha)$  would not always be aligned with  $q_1(H)$  and  $\nabla L(\theta)$ , and  $\|H_\alpha\|_{S_n}$  tends to decrease. This would be a possible explanation for the tendency of decreasing and then oscillating  $\|H\|_{S_n}$ . See Appendix C.3 for detailed empirical evidences of the above arguments. Figure 3 (right) similarly shows that when the iterate is at a sharp side of the valley, it tends to jump to the other side of a flatter area, and vice versa.

To summarize, we make the following observations for GD in order: (i)  $\|H\|$  increases in the beginning (the *progressive sharpening* [6]), (ii)  $\|H\|$  exceeds  $\frac{2}{\eta}$ , (iii) the gradient  $\nabla L$  becomes more aligned with the top eigenvector  $q_1(H)$  in a few steps, (iv)  $\|H\|_{S_n}$  reaches the threshold  $\frac{2}{\eta}$  and the iterate jumps across the valley, (v)  $\|H\|_{S_n}$  tends to decrease due to the non-quadraticity, and it repeats this process, while  $\|H\|_{S_n}$  oscillating around  $\frac{2}{\eta}$ . We observe a similar behavior with oscillating  $\frac{\text{tr}(HS_b)}{\text{tr}(S_n)}$  around  $\frac{2}{\eta}$  for SGD. It requires further investigation into the exact underlying mechanisms and we leave it as a future work.

**Remark** (Experiments in Section 4). *We report the experimental results using vanilla SGD/GD without momentum and weight decay, constant learning rate, and no data augmentation. We train a simple 6-layer CNN (6CNN,  $m = 0.51M$ ) on CIFAR-10-8k where DATASET- $n$  denotes a subset of DATASET with  $|\mathcal{D}| = n$  and  $k=2^{10} = 1024$ . See Appendix C.1-C.3 for the results from other datasets, learning rates and networks (ResNet-9 with  $m = 2.3M$  [13] and WRN-28-2 with  $m = 36M$  [55]).*

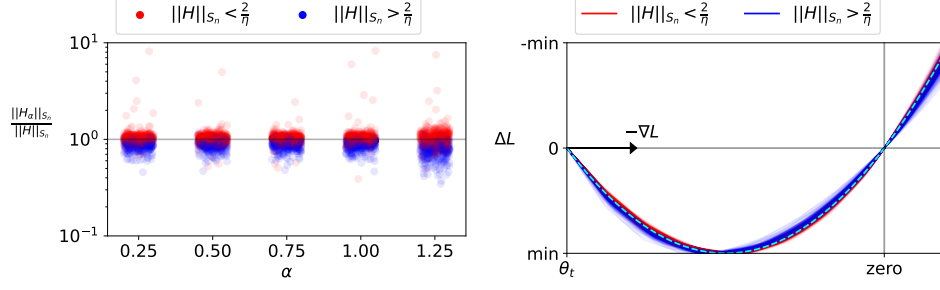


Figure 3: **[Asymmetric valleys]** Left: The ratio  $\frac{\|H_\alpha\|_{S_n}}{\|H\|_{S_n}}$  where  $H_\alpha = H(\theta - \alpha\eta\nabla L(\theta))$  for  $\alpha = \frac{1}{4} \times [1, 2, 3, 4, 5]$  for each  $t$  during training. When  $\|H\|_{S_n} < \frac{2}{\eta}$  (red),  $\|H_\alpha\|_{S_n}$  is usually larger than  $\|H\|_{S_n}$ . On the other hand, when  $\|H\|_{S_n} > \frac{2}{\eta}$  (blue),  $\|H_\alpha\|_{S_n}$  is usually smaller than  $\|H\|_{S_n}$ . Right: The training loss difference along the gradient descent direction, for each  $\theta_t$ . Each plot is normalized and translated to have the same minimum value and the same zero where  $\Delta L = 0$ . We also plot the quadratic baseline (cyan dashed curve). When  $\|H\|_{S_n} < \frac{2}{\eta}$  (red), it usually becomes sharper across the valley (right-shifted). On the other hand, when  $\|H\|_{S_n} > \frac{2}{\eta}$  (blue), it usually becomes flatter across the valley (left-shifted). We train 6CNN using GD with  $\eta = 0.04$ .

## 153 5 Generalization through Implicit Regularization

154 In the previous section, we have empirically demonstrated that the SGD iterate is implicitly discour-  
 155 aged from staying within the unstable regime. Now, we are ready to further analyze this property  
 156 from the regularization perspective.

### 157 5.1 Implicit Interaction Regularization (IIR)

158 First, to understand the effect of batch size  $b$  on the gradient distribution, we define the following  $\rho_b$ :

159 **Definition 3** (a concentration measure of the batch gradient). We define  $\rho_b$  as the ratio of the squared  
 160 norm of the total gradient  $\|\nabla L\|^2$  to the expected squared norm of the batch gradients  $\mathbb{E}[\|g_b\|^2]$ , i.e.,

$$\rho_b \equiv \frac{\|\nabla L\|^2}{\mathbb{E}[\|g_b\|^2]} = \frac{\text{tr}(S_n)}{\text{tr}(S_b)}. \quad (10)$$

161 Here, we can write  $\|\nabla L\|^2 = \|\mathbb{E}[g_b]\|^2$  and thus the ratio  $\rho_b = \frac{\|\mathbb{E}[g_b]\|^2}{\mathbb{E}[\|g_b\|^2]} \leq 1$  is similar to the square  
 162 of the mean resultant length  $\bar{R}_b^2 \equiv \|\mathbb{E}[\frac{g_b}{\|g_b\|}]\|^2 \leq 1$  of the batch gradient  $g_b$  [36], especially when  
 163  $\text{std}[\|g_b\|]$  is small compared to  $\mathbb{E}[\|g_b\|]$  (see Appendix C.5 for empirical evidences). Both  $\rho_b$  and  $\bar{R}_b^2$   
 164 are concentration measures and have lower values when the batch gradients  $g_b$  are more scattered.  
 165 Therefore, it is natural to expect that the ratio  $\rho_b$  is small for a small batch size  $b$ , and we will revisit  
 166 this in more detail in the following section (cf. (12)). We also note that  $\rho_n = \bar{R}_n^2 = 1$ .

167 Now, we can rewrite the instability condition  $\frac{\text{tr}(HS_b)}{\text{tr}(S_n)} > \frac{2}{\eta}$  (multiplying both sides by  $\rho_b$ ) as  $\|H\|_{S_b} >$   
 168  $\frac{2\rho_b}{\eta}$ . In other words, the interaction-aware sharpness  $\|H\|_{S_b}$  is implicitly regularized to be less than  
 169  $\frac{2\rho_b}{\eta}$ . We name this *Implicit Interaction Regularization (IIR)*.

**Definition 4** (Implicit Interaction Regularization (IIR)).

$$\|H\|_{S_b} \leq \frac{2\rho_b}{\eta}. \quad (11)$$

170 We argue that the upper constraint  $\frac{2\rho_b}{\eta}$  in IIR is crucial in determining the generalization performance.  
 171 With a low constraint, SGD strongly regularizes the interaction-aware sharpness  $\|H\|_{S_b}$ . We also  
 172 note that IIR affects not only the magnitude  $\|H\|$  but also the *directional* interaction. In other words,  
 173 IIR discourages the batch gradients from aligning with the top eigensubspace of the Hessian that is  
 174 spanned by a few largest eigenvectors of the Hessian (cf. [11]).

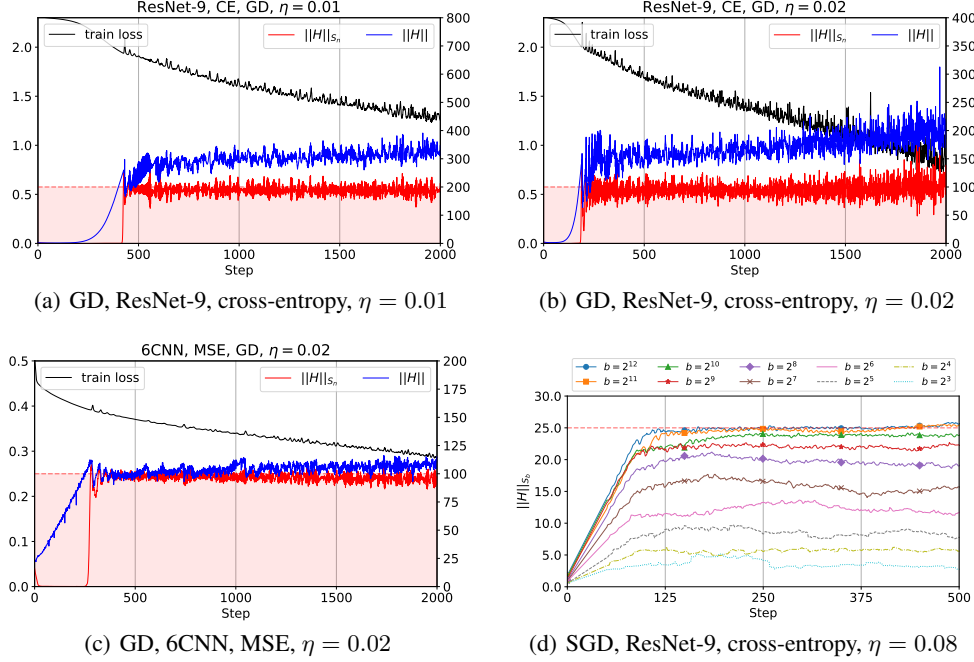


Figure 4: **[A clear indication of the edge of stability]** (a)-(c): After a few steps of full-batch training,  $\|H\|$  (blue) hovers **above**  $\frac{2}{\eta}$  [6], but  $\|H\|_{S_n}$  (red, defined in (8)) oscillates **around**  $\frac{2}{\eta}$  (red dashed horizontal line). The edge of stability is more evident in the latter (red). Curves are plotted for every step. We train a model on CIFAR-10-8k ( $n = 2^{13}$ ) using (a)/(b) cross-entropy loss with  $\eta = 0.01/0.02$ , respectively, and (c) MSE with  $\eta = 0.02$ . (d): We plot curves  $\|H\|_{S_b}$  when trained with various  $b$ 's. After a few steps (around 125), they reach the threshold which linearly increases as  $b$  becomes larger when  $b \ll n = 2^{13}$ , and saturates to  $\frac{2\rho_b}{\eta} \approx \frac{2}{\eta}$  when  $b$  is large. Curves are smoothed for visual clarity. We use SGD with  $b \in \{2^3, \dots, 2^{12}\}$  and  $\eta = 0.08$ .

Figures 4(a)-4(c) show that, for GD ( $\rho_n = 1$ ), the interaction-aware sharpness  $\|H\|_{S_n}$  (red) oscillates around  $\frac{2}{\eta}$  and exhibits IIR. This result is consistent with Cohen et al. [6] that  $\|H\|$  hovers above  $\frac{2}{\eta}$  for GD. This is because, as mentioned earlier,  $\frac{2}{\eta} \approx \|H\|_{S_n} \leq \|H\|$  and the equality holds only when the gradient  $\nabla L$  and the top eigenvector  $q_1$  of  $H$  are aligned, but generally they are not. For this reason, IIR provides a tighter relation and more clearly identifies the edge of stability than Cohen et al. [6]. These results are also consistent with Prop. 4.1 that  $\|H\|_{S_n}$  suddenly increases from 0 to  $\frac{2}{\eta}$  in a few steps after  $\|H\|$  exceeds  $\frac{2}{\eta}$  (see Appendix C.3-C.4 for more). Moreover, IIR also applies to a general SGD training with  $1 \leq b \leq n$ . Figure 4(d) shows IIR for SGD with different batch sizes  $b \in \{2^3, \dots, 2^{12}\}$ . The upper bound ( $2\rho_b/\eta$  according to (11)) of  $\|H\|_{S_b}$  is higher when using a larger batch size, but limited to less than  $2/\eta$  ( $\rho_b \leq 1$ ). We will further discuss this behavior with an investigation of  $\rho_b$  in the following section.

## 5.2 Linear and Saturation Scaling Rule (LSSR)

The ratio  $b/\eta$  of batch size  $b$  to learning rate  $\eta$  has long been believed as an important factor influencing the generalization performance, and the test accuracy has observed to be similar when trained with the same ratio  $b/\eta = b'/\eta'$ , i.e.,  $b' = kb$  and  $\eta' = k\eta$  for  $k > 0$ . This is called the linear scaling rule (LSR) [25, 10, 18, 44, 57]. They argue that LSR holds because  $\theta_{t+k} - \theta_t = -\frac{\eta}{b} \sum_{i=0}^{k-1} \sum_{x \in \mathcal{B}_{t+i}} \nabla \ell(x; \theta_{t+i}) \approx -\frac{\eta}{b} \sum_{i=0}^{k-1} \sum_{x \in \mathcal{B}_{t+i}} \nabla \ell(x; \theta_t) = -\frac{\eta'}{b'} \sum_{x \in \mathcal{B}_{t:t+k}} \nabla \ell(x; \theta_t)$  assuming  $\nabla \ell(\theta_{t+i}) \approx \nabla \ell(\theta_t)$  for  $0 \leq i < k$ , where  $\mathcal{B}_{t:t+k} \equiv \cup_{i=0}^{k-1} \mathcal{B}_{t+i}$  and  $|\mathcal{B}_{t:t+k}| = kb = b'$ . However, the assumption is false and the gradient oscillates mostly with a negative cosine value  $\cos(g_b(\theta_t), g_b(\theta_{t+1})) < 0$  between two consecutive gradients after entering the edge of stability

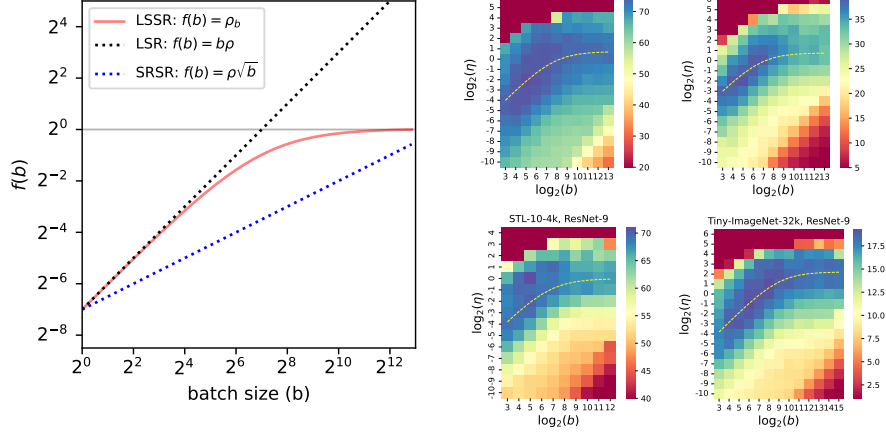


Figure 5: **[Linear and Saturation Scaling Rule (LSSR)]** Left: LSSR (red) in (12), LSR (black dotted line) [10] and SRSR (blue dotted line) [15]. For LSSR, we can observe both linear and saturation regions ( $n = 8k, \rho = 2^{-7}$ ). Right: Heatmaps of test accuracy for models trained with a large number of pairs of  $(b, \eta)$  on CIFAR-10-8k, CIFAR-100-8k, STL-10-4k, and Tiny-ImageNet-32k (from left to right, from top to bottom). It does not follow either LSR or SRSR, but LSSR. We also plot  $f(b) = \rho_b$  (yellow dashed curve) for some  $\rho$  on each heatmap. Note that they are all log-log plots and thus a slope of 1 means it is linear.

(see Appendix C.3). Moreover, LSR fails when the batch size is large [18, 38, 57, 43, 46]. On the other hand, Krizhevsky [25], Hoffer et al. [15] propose the square root scaling rule (SRSR) with another ratio  $\sqrt{b}/\eta$  to keep the covariance of the parameter update constant for  $b \ll n$  based on  $\text{Var}[\eta g_b] = \eta^2 C_b = \frac{\gamma_{n,b} \eta^2}{b} C_1 \approx \frac{\eta^2}{b} C_1$ . However, Shallue et al. [42] show that both LSR and SRSR do not hold in general.

Based on the analysis of IIR with a new ratio  $2\rho_b/\eta$  in the previous section, we explore why LSR fails in the large-batch regime and provide a more accurate rule to explain the generalization performance of the models trained with various choices of batch size and learning rate pairs  $(b, \eta)$ .

To this end, we investigate the concentration measure  $\rho_b = \text{tr}(S_n)/\text{tr}(S_b)$ . By combining two equations,  $C_b = S_b - S_n$  (by definition) and  $C_b = \frac{\gamma_{n,b}}{b}(S_1 - S_n)$  in (1), we can obtain  $S_b = C_b + S_n = \frac{\gamma_{n,b}}{b}S_1 + (1 - \frac{\gamma_{n,b}}{b})S_n$ . Therefore, we have  $\text{tr}(S_b) = \frac{\gamma_{n,b}}{b}\text{tr}(S_1) + (1 - \frac{\gamma_{n,b}}{b})\text{tr}(S_n)$ , which leads to the following equation:

$$\rho_b \equiv \frac{\text{tr}(S_n)}{\text{tr}(S_b)} = \frac{\text{tr}(S_n)}{\frac{\gamma_{n,b}}{b}\text{tr}(S_1) + (1 - \frac{\gamma_{n,b}}{b})\text{tr}(S_n)} = \underbrace{\frac{1}{\frac{\gamma_{n,b}}{b} \frac{1}{\rho} + (1 - \frac{\gamma_{n,b}}{b})}}_{(*)} \approx \begin{cases} \frac{b}{\gamma_{n,b}}\rho \approx b\rho & \text{if } b \text{ is small} \\ 1 & \text{if } b \text{ is large} \end{cases} \quad (12)$$

from (10) where  $\rho = \rho_1 = \text{tr}(S_n)/\text{tr}(S_1)$ . Note that  $\rho$  is (much) smaller than 1 because  $\nabla \ell(x_i)$  has different direction for each  $x_i$  and  $\text{tr}(S_n) = \|\nabla L\|^2 = \|\frac{1}{n} \sum_i \nabla \ell(x_i)\|^2 \leq \frac{1}{n} \sum_i \|\nabla \ell(x_i)\|^2 = \text{tr}(S_1)$ . In other words,  $1/\rho$  is (much) larger than 1 (see Appendix C.5).

Figure 5 (left) demonstrates a new scaling rule with the ratio  $\rho_b/\eta$ , called the *Linear and Saturation Scaling Rule* (LSSR), with the two regimes that (i)  $\rho_b$  is almost linear when  $b \ll n$  (linear regime) and (ii)  $\rho_b$  saturates when  $b$  is large (saturation regime), which are also shown in Figure 4(d). It depends on which part of the denominator  $(*)$  in (12) dominates the other. First, when  $b \ll n$ , then  $\gamma_{n,b}/b$  is not very small and the first term  $\frac{\gamma_{n,b}}{b} \frac{1}{\rho}$  dominates the second term  $1 - \frac{\gamma_{n,b}}{b}$  since  $\frac{1}{\rho} \gg 1$ . Second, as  $b$  becomes large,  $\gamma_{n,b}/b \approx 0$  and the second term ( $\approx 1$ ) dominates the first term. Thus,  $\rho_b$  saturates to 1 and is not linearly related to  $b$ , and LSR is no longer valid. The above arguments also hold for the batches sampled *with* replacement where the only modification is  $\gamma_{n,b} = 1, \forall b$  in (12). Figure 5 (right) empirically supports LSSR with the test accuracies when trained with various combinations of pairs  $(b, \eta)$ . To be specific, the optimal learning rate is almost linear when  $b$  is small, but it saturates



when  $b$  is large. We also plot  $f(b) = \rho_b$  (the yellow dashed curve) for some  $\rho$ . Note that Figure 8 of Shallue et al. [42, Section 4.7] shows similar “linear and saturation” behaviors supportive of LSSR on other datasets (see also Figure 7 of Zhang et al. [57, Section 4.3]).

**Remark** (Experiments in Section 5). *We train models using vanilla SGD/GD without momentum and weight decay, constant learning rate, and no data augmentation. For Figure 5, we use subsets of the datasets CIFAR-10 [24], CIFAR-100 [24], STL-10 [5], and Tiny-ImageNet (a subset of ImageNet [7] with  $3 \times 64 \times 64$  images and 200 object classes). We use a large number of epochs (800) and batch normalization [17] to achieve a zero training error even with a large  $b$  and a small  $\eta$ . However, in the lower right corner (red area) of each heatmap in Figure 5 (right), when  $b$  is too large or  $\eta$  is too small so that  $\|\theta_{t+1} - \theta_t\| = \eta\|g_b\|$  is too small, it requires an exponentially large number of steps for the iterate to enter the edge of stability. Thus, in this case, the assumption in Goyal et al. [10],  $\nabla\ell(\theta_t) \approx \nabla\ell(\theta_{t+i})$  for  $0 \leq i < k$ , approximately holds and the reasoning on LSR is valid. However, this only holds for a non-practical  $(b, \eta)$  which shows a suboptimal performance. See Appendix C.4-C.5 for the results from other networks and hyperparameters.*

## 6 Discussion

We provide a new insight on the link between the batch gradient distribution and the sharpness of the loss landscape. In this section, we reconcile our arguments with some previous studies.

Jastrzebski et al. [18] explain the optimization behavior of SGD with the SDE approximation  $d\theta_t = -\nabla L(\theta_t)dt + \sqrt{\frac{\eta}{b}}C_1^{1/2}dW(t)$  of the SGD where  $W$  is an  $m$ -dimensional Brownian motion. Therefore, the same ratio  $\frac{\eta}{b} = \frac{\eta'}{b'}$  leads to the same SDE, which implies LSR. Moreover, a large  $\frac{\eta}{b}$  implies a large diffusion in SDE, which has been linked with the escaping efficiency from a sharp local minimum in Zhu et al. [58]. We instead argue that a large second moment  $\text{tr}(S_b)$  (compared to  $\text{tr}(S_n)$ ) and a large  $\eta$  lead to a low constraint  $2\rho_b/\eta$  on the interaction-aware sharpness. We emphasize that we do not model SGD with SDE and thus our argument is applicable to a practical learning rate regime.

Wu et al. [49] empirically show that what is important for the generalization performance of a neural network is not the class to which the gradient distribution belongs, but the second moment of the distribution. This is consistent with our arguments with the interaction  $\text{tr}(HS_b)$  and the concentration measure  $\rho_b = \text{tr}(S_n)/\text{tr}(S_b)$ , because they depend on the second moment  $S_b$ , not on the class of the gradient distribution.

Recently, Li et al. [33] suggest a necessary condition that the “noise-to-signal ratio” needs to be large for LSR (and the SDE assumption) to hold. This is consistent with our result on the linear regime (where  $b$  and  $\rho_b$  are small) because the noise-to-signal ratio is approximately the inverse of the “signal-to-noise” ratio  $\rho_b = \text{tr}(S_n)/\text{tr}(S_b)$ , but defined for an equilibrium distribution. We provide not only the necessary condition but also the sufficient condition for LSR with a novel scaling rule LSSR applicable to every batch size including where LSR fails (the saturation regime).

## 7 Conclusion

From an analysis of unstable dynamics of SGD (Section 4.1) and the instability condition (Definition 1), we clearly mark the edge of stability (Figure 4) with the interaction-aware sharpness  $\|H\|_{S_b}$  (Definition 2) and show the presence of the implicit regularization effect on the interaction between the gradient distribution and the loss landscape geometry (IIR) (Section 5.1, Definition 4). Moreover, introducing the concentration measure  $\rho_b$  of the batch gradient (Definition 3, (12)), we link the second moment of the gradient distribution and the sharpness of the loss landscape, and propose a new scaling rule called Linear and Saturation Scaling Rule (LSSR) (Section 5.2, Figure 5). Due to the simplicity of the analysis, we hope that our insights will motivate the future work toward understanding various learning tasks.

## References

- [1] David Barrett and Benoit Dherin. Implicit gradient regularization. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=3q5IqUrkcF>.
- [2] Nils Bjorck, Carla P Gomes, Bart Selman, and Kilian Q Weinberger. Understanding batch normalization. *Advances in neural information processing systems*, 31, 2018.
- [3] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.
- [4] Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In *International Conference on Learning Representations*, 2018.
- [5] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.
- [6] Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=jh-rTtvkGeM>.
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [8] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028. PMLR, 2017.
- [9] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=6Tm1mposlrM>.
- [10] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.
- [11] Guy Gur-Ari, Daniel A Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.
- [12] Haowei He, Gao Huang, and Yang Yuan. Asymmetric valleys: Beyond sharp and flat local minima. *Advances in neural information processing systems*, 32, 2019.
- [13] 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*, pages 770–778, 2016.
- [14] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997.
- [15] Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. *Advances in neural information processing systems*, 30, 2017.
- [16] Wenqing Hu, Chris Junchi Li, Lei Li, and Jian-Guo Liu. On the diffusion approximation of nonconvex stochastic gradient descent. *Annals of Mathematical Sciences and Applications*, 4(1), 2019.

- [17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.
- [18] 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.
- [19] Stanisław Jastrzębski, Zachary Kenton, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. On the relation between the sharpest directions of DNN loss and the SGD step length. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=SkGEaj05t7>.
- [20] Stanisław Jastrzębski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho\*, and Krzysztof Geras\*. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=r1g87C4KwB>.
- [21] Stanisław Jastrzębski, 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 Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4772–4784. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/jastrzebski21a.html>.
- [22] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=SJgIPJBFvH>.
- [23] 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 *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL <https://openreview.net/forum?id=H1oyR1Ygg>.
- [24] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master’s thesis, Department of Computer Science, University of Toronto*, 2009.
- [25] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.
- [26] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks, 2021.
- [27] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.
- [28] 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.
- [29] Chris Junchi Li, Lei Li, Junyang Qian, and Jian-Guo Liu. Batch size matters: A diffusion approximation framework on nonconvex stochastic gradient descent. *stat*, 1050:22, 2017.
- [30] Qianxiao Li, Cheng Tai, and E Weinan. Stochastic modified equations and adaptive stochastic gradient algorithms. In *International Conference on Machine Learning*, pages 2101–2110. PMLR, 2017.

- [31] Qianxiao Li, Cheng Tai, and E Weinan. Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations. *The Journal of Machine Learning Research*, 20(1):1474–1520, 2019.
- [32] Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.
- [33] Zhiyuan Li, Sadhika Malladi, and Sanjeev Arora. On the validity of modeling sgd with stochastic differential equations (sdes). *Advances in Neural Information Processing Systems*, 34, 2021.
- [34] Stephan Mandt, Matthew Hoffman, and David Blei. A variational analysis of stochastic gradient algorithms. In *International conference on machine learning*, pages 354–363. PMLR, 2016.
- [35] 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.
- [36] Kanti V Mardia, Peter E Jupp, and KV Mardia. *Directional statistics*, volume 2. Wiley Online Library, 2000.
- [37] James Martens. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.
- [38] Dominic Masters and Carlo Luschi. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*, 2018.
- [39] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.
- [40] Behnam Neyshabur. Implicit regularization in deep learning. *arXiv preprint arXiv:1709.01953*, 2017.
- [41] Mark Schmidt. Convergence rate of stochastic gradient with constant step size. 2014.
- [42] Christopher J Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E Dahl. Measuring the effects of data parallelism on neural network training. *arXiv preprint arXiv:1811.03600*, 2018.
- [43] Samuel Smith, Erich Elsen, and Soham De. On the generalization benefit of noise in stochastic gradient descent. In *International Conference on Machine Learning*, pages 9058–9067. PMLR, 2020.
- [44] Samuel L Smith and Quoc V Le. A bayesian perspective on generalization and stochastic gradient descent. In *International Conference on Learning Representations*, 2018.
- [45] Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. Don’t decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.
- [46] Samuel L Smith, Benoit Dherin, David Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*, 2021. URL [https://openreview.net/forum?id=rq\\_Qr0c1Hyo](https://openreview.net/forum?id=rq_Qr0c1Hyo).
- [47] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.
- [48] Valentin Thomas, Fabian Pedregosa, Bart Merriënboer, Pierre-Antoine Manzagol, Yoshua Bengio, and Nicolas Le Roux. On the interplay between noise and curvature and its effect on optimization and generalization. In *International Conference on Artificial Intelligence and Statistics*, pages 3503–3513. PMLR, 2020.

- 396 [49] Jingfeng Wu, Wenqing Hu, Haoyi Xiong, Jun Huan, Vladimir Braverman, and Zhanxing Zhu.  
397 On the noisy gradient descent that generalizes as sgd. In *International Conference on Machine*  
398 *Learning*, pages 10367–10376. PMLR, 2020.
- 399 [50] Lei Wu, Chao Ma, and Weinan E. How sgd selects the global minima in over-parameterized  
400 learning: A dynamical stability perspective. In *Proceedings of the 32nd International Conference*  
401 *on Neural Information Processing Systems*, pages 8289–8298, 2018.
- 402 [51] Chen Xing, Devansh Arpit, Christos Tsirigotis, and Yoshua Bengio. A walk with sgd. *arXiv*  
403 *preprint arXiv:1802.08770*, 2018.
- 404 [52] Sho Yaida. Fluctuation-dissipation relations for stochastic gradient descent. In *International*  
405 *Conference on Learning Representations*, 2019. URL [https://openreview.net/forum?](https://openreview.net/forum?id=SkNks0RctQ)  
406 [id=SkNks0RctQ](https://openreview.net/forum?id=SkNks0RctQ).
- 407 [53] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural  
408 networks through the lens of the hessian. In *2020 IEEE International Conference on Big Data*  
409 *(Big Data)*, pages 581–590. IEEE, 2020.
- 410 [54] Yang You, Igor Gitman, and Boris Ginsburg. Scaling sgd batch size to 32k for imagenet training.  
411 *arXiv preprint arXiv:1708.03888*, 2017.
- 412 [55] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint*  
413 *arXiv:1605.07146*, 2016.
- 414 [56] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding  
415 deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):  
416 107–115, 2021.
- 417 [57] Guodong Zhang, Lala Li, Zachary Nado, James Martens, Sushant Sachdeva, George Dahl, Chris  
418 Shallue, and Roger B Grosse. Which algorithmic choices matter at which batch sizes? insights  
419 from a noisy quadratic model. *Advances in neural information processing systems*, 32, 2019.
- 420 [58] Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in  
421 stochastic gradient descent: Its behavior of escaping from sharp minima and regularization  
422 effects. In *International Conference on Machine Learning*, pages 7654–7663. PMLR, 2019.

## Checklist

- (a) For all authors...
  - (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
  - (b) Did you describe the limitations of your work? [Yes] We try to avoid theoretical analysis based on impractical assumptions. Therefore, some of our claims are supported by experiments and may require further theoretical investigation.
  - (c) Did you discuss any potential negative societal impacts of your work? [N/A]
  - (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
- (b) If you are including theoretical results...
  - (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Prop. 4.1.
  - (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix B.
- (c) If you ran experiments...
  - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
  - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
  - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
  - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A] We do not propose any new algorithm which requires to report the computational cost.
- (d) If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
  - (a) If your work uses existing assets, did you cite the creators? [Yes]
  - (b) Did you mention the license of the assets? [Yes]
  - (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
  - (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
  - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
- (e) If you used crowdsourcing or conducted research with human subjects...
  - (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
  - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
  - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

## 461 A Notations

462 We summarize the notations for a quick reference.

time step :	$t \in \mathbb{N}$
model parameter :	$\theta \in \Theta \subset \mathbb{R}^m$ (or indexed $\theta_t$ )
training sample :	$x \in \mathcal{X}$ (or indexed $x_i$ )
training data :	$\mathcal{D} = \{x_i\}_{i=1}^n;  \mathcal{D}  = n$
loss function :	$\ell(x; \theta)$
(total) training loss :	$L(\theta) \equiv \frac{1}{n} \sum_{i=1}^n \ell(x; \theta) = \frac{1}{ \mathcal{D} } \sum_{x \in \mathcal{D}} \ell(x; \theta); L_t = L(\theta_t)$
learning rate :	$\eta > 0$
batch :	$\mathcal{B} \subset \mathcal{D}$ (or indexed $\mathcal{B}_t$ )
batch size :	$b =  \mathcal{B} ; 1 \leq b \leq n$
batch gradient :	$g_b(\theta) \equiv \frac{1}{b} \sum_{x \in \mathcal{B}} \nabla \ell(x; \theta) = \frac{1}{ \mathcal{B} } \sum_{x \in \mathcal{B}} \nabla \ell(x; \theta)$
GD :	$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} L(\theta_t)$
SGD :	$\theta_{t+1} = \theta_t - \eta g_b(\theta_t)$
the covariance of the batch gradient :	$C_b(\theta) \equiv \text{Var}[g_b(\theta)] \in \mathbb{R}^{m \times m}$
the second moment of the batch gradient :	$S_b(\theta) \equiv \mathbb{E}[g_b(\theta) g_b(\theta)^{\top}] \in \mathbb{R}^{m \times m}$
sampling coefficient :	$\gamma_{n,b} = \begin{cases} \frac{n-b}{n-1} & \text{for sampling without replacement} \\ 1 & \text{for sampling with replacement} \end{cases}$
Hessian :	$H(\theta) \equiv \nabla_{\theta}^2 L(\theta) = \mathbb{E}_{x \sim \mathcal{D}}[\nabla_{\theta}^2 \ell(x; \theta)] \in \mathbb{R}^{m \times m}$
the Euclidean $\ell^2$ -norm :	$\ u\  = \sqrt{\sum_i u_i^2}$
the top eigenvalue of the Hessian :	$\lambda_1 = \ H\  \equiv \sup_{u \neq 0} \frac{\ Hu\ }{\ u\ }$
the $i$ -th largest eigenvalue of the Hessian :	$\lambda_i \in \mathbb{R}$
the corresponding $i$ -th eigenvector of the Hessian :	$q_i = q_i(H) \in \mathbb{R}^m$
trace :	$\text{tr}(A) = \sum_i A_{i,i}$
the interaction-aware sharpness :	$\ H\ _{S_b} \equiv \frac{\text{tr}(HS_b)}{\text{tr}(S_b)}$
the concentration measure of the batch gradient :	$\rho_b \equiv \frac{\text{tr}(S_n)}{\text{tr}(S_b)}$
the concentration measure of the per-example gradient :	$\rho \equiv \frac{\text{tr}(S_n)}{\text{tr}(S_1)}$
the unstable regime :	$\mathbb{U} = \{\theta \in \Theta : \ H\ _{S_b} > \frac{2\rho_b}{\eta}\}$
the stable regime :	$\mathbb{S} = \mathbb{U}^c = \{\theta \in \Theta : \ H\ _{S_b} \leq \frac{2\rho_b}{\eta}\}$
the edge of stability :	$\partial\mathbb{S} = \{\theta \in \Theta : \ H\ _{S_b} = \frac{2\rho_b}{\eta}\}$

## 463 B Proofs

### 464 B.1 Proof of (1)

465 We provide a proof of (1) to make the paper self-contained. Similar proofs are given in Li et al.  
466 [29], Hoffer et al. [15], Wu et al. [49].

467 *Proof.* We start with

$$C_b = \text{Var}[g_b] = \mathbb{E}[g_b g_b^\top] - \mathbb{E}[g_b] \mathbb{E}[g_b]^\top = \mathbb{E}[g_b g_b^\top] - \nabla L \nabla L^\top \quad (13)$$

$$= \nabla \mathcal{L} \mathbb{E}[w w^\top] \nabla \mathcal{L}^\top - \nabla \mathcal{L} \left( \frac{1}{n} \mathbf{1} \frac{1}{n} \mathbf{1}^\top \right) \nabla \mathcal{L}^\top \quad (14)$$

$$= \nabla \mathcal{L} \left( \mathbb{E}[w w^\top] - \frac{1}{n^2} \mathbf{1} \mathbf{1}^\top \right) \nabla \mathcal{L}^\top \quad (15)$$

$$= \nabla \mathcal{L} \text{Var}[w] \nabla \mathcal{L}^\top \quad (16)$$

$$= \frac{1}{b^2} \nabla \mathcal{L} \text{Var}[v] \nabla \mathcal{L}^\top \quad (17)$$

468 where  $\nabla \mathcal{L} = [\nabla \ell_1, \dots, \nabla \ell_n] \in \mathbb{R}^{m \times n}$ ,  $\ell_i = \ell(x_i)$ , the random vector  $w = [w_1, \dots, w_n]^\top$ , each  
469 element of which represents  $\frac{1}{b} \times$  "how many times the index  $i$  is sampled in  $\mathcal{B}$ ", and  $v = bw$ .

470 In case of sampling *with* replacement, we have  $v = v^{(1)} + \dots + v^{(b)}$  where  $v^{(i)}$  represents sampling  
471 of a single sample. Thus,  $\text{Var}[v] = b \text{Var}[v^{(1)}]$ . We have  $\mathbb{E}[v^{(1)}] = \frac{1}{n} \mathbf{1}$  and

$$\mathbb{E}[v^{(1)} v^{(1)\top}]_{i,j} = \begin{cases} P[i \in \mathcal{B}^{(1)}] = \frac{1}{n} & \text{if } i = j \\ P[i \in \mathcal{B}^{(1)}, j \in \mathcal{B}^{(1)}] = 0 & \text{else} \end{cases} \quad (18)$$

472 where  $|\mathcal{B}^{(1)}| = 1$ . Thus,

$$\text{Var}[v] = b \text{Var}[v^{(1)}] = b \left( \frac{1}{n} I - \frac{1}{n^2} \mathbf{1} \mathbf{1}^\top \right) \quad (19)$$

473 In case of sampling *without* replacement, we have  $\mathbb{E}[v] = \frac{b}{n} \mathbf{1}$  and

$$\mathbb{E}[v v^\top]_{i,j} = \begin{cases} P[i \in \mathcal{B}] = \frac{C(n-1, b-1)}{C(n, b)} = \frac{b}{n} & \text{if } i = j \\ P[i \in \mathcal{B}, j \in \mathcal{B}] = \frac{C(n-2, b-2)}{C(n, b)} = \frac{b(b-1)}{n(n-1)} & \text{else} \end{cases} \quad (20)$$

474 where  $C(n_1, r_1)$  is the number of  $r_1$ -combinations from a set of  $n_1$  elements. This leads to

$$\mathbb{E}[v v^\top] = \frac{b(b-1)}{n(n-1)} \mathbf{1} \mathbf{1}^\top + \left( \frac{b}{n} - \frac{b(b-1)}{n(n-1)} \right) I \quad (21)$$

475 and

$$\text{Var}[v] = \mathbb{E}[v v^\top] - \frac{b^2}{n^2} \mathbf{1} \mathbf{1}^\top = \left( \frac{b(b-1)}{n(n-1)} - \frac{b^2}{n^2} \right) \mathbf{1} \mathbf{1}^\top + \frac{b(n-b)}{n(n-1)} I \quad (22)$$

$$= \frac{b(b-n)}{n^2(n-1)} \mathbf{1} \mathbf{1}^\top + \frac{b(n-b)}{n(n-1)} I \quad (23)$$

$$= \frac{b(n-b)}{n-1} \left( \frac{1}{n} I - \frac{1}{n^2} \mathbf{1} \mathbf{1}^\top \right) \quad (24)$$

476 Putting the two cases together, from (17), (19) and (24), we have

$$C_b = \frac{1}{b^2} b \gamma_{n,b} \nabla \mathcal{L} \left( \frac{1}{n} I - \frac{1}{n^2} \mathbf{1} \mathbf{1}^\top \right) \nabla \mathcal{L}^\top = \frac{\gamma_{n,b}}{b} \left( \frac{1}{n} \nabla \mathcal{L} \nabla \mathcal{L}^\top - \frac{1}{n^2} (\nabla \mathcal{L} \mathbf{1})(\nabla \mathcal{L} \mathbf{1})^\top \right) \quad (25)$$

$$= \frac{\gamma_{n,b}}{b} \left( \frac{1}{n} \sum_i \nabla \ell_i \nabla \ell_i^\top - \nabla L \nabla L^\top \right) \quad (26)$$

$$= \frac{\gamma_{n,b}}{b} (S_1 - S_n) \quad (27)$$



477 where

$$\gamma_{n,b} = \begin{cases} 1 & \text{for the sampling with replacement} \\ \frac{n-b}{n-1} & \text{for the sampling without replacement} \end{cases} \quad (28)$$

478

□

## 479 B.2 Proof of Prop. 4.1

480 *Proof.* Put a quadratic training loss  $L(\theta) = \frac{1}{2}\theta^\top H\theta + b^\top \theta + c$ . We use the SVD of the symmetric  
481 positive-definite matrix  $H = Q\Lambda Q^\top$  and some translation of  $\theta$  to simplify the training loss  $L$  (up to  
482 constant) as follows:

$$L = \frac{1}{2}\theta^\top Q\Lambda Q^\top \theta = \frac{1}{2}\psi^\top \Lambda \psi, \quad (29)$$

483 where  $\psi = Q^\top \theta$ . Then  $\nabla_\psi L = \Lambda \psi$  and

$$\psi_{t+1} = \psi_t - \eta \nabla_\psi L(\psi_t) = \psi_t - \eta \Lambda \psi_t = (I - \eta \Lambda) \psi_t = (I - \eta \Lambda)^{t+1} \psi_0. \quad (30)$$

484 Therefore,

$$\nabla_\psi L(\psi_t) = \Lambda(I - \eta \Lambda)^t \psi_0 \quad (31)$$

485 which leads to

$$q_1^\top \nabla_\theta L(\theta_t) = [\nabla_\psi L(\psi_t)]_1 = \lambda_1(1 - \eta \lambda_1)^t [\psi_0]_1 \quad (32)$$

486 and

$$q_j^\top \nabla_\theta L(\theta_t) = [\nabla_\psi L(\psi_t)]_j = \lambda_j(1 - \eta \lambda_j)^t [\psi_0]_j \text{ for } j \neq 1, \quad (33)$$

487 where  $\lambda_i$  is the  $i$ -th largest eigenvalue of  $H$  with the corresponding eigenvector  $q_i$ . Because of the  
488 assumption  $\lambda_1 > \frac{2}{\eta}$  and  $0 < \lambda_j < \frac{2}{\eta}$  for  $j \neq 1$ , we have  $|1 - \eta \lambda_1| > 1$  while  $|1 - \eta \lambda_j| < 1$  for  
489  $j \neq 1$ . Therefore,

$$|q_1^\top \nabla_\theta L(\theta_t)| = \lambda_1 |1 - \eta \lambda_1|^t |[\psi_0]_1| = a_1(1 + r)^t \quad (34)$$

490 exhibits exponential growth with the growth rate  $r = \eta \lambda_1 - 2 > 0$  and the initial value  $a_0 = \lambda_1 |[\psi_0]_1|$ ,  
491 and  $|q_j^\top \nabla_\theta L(\theta_t)| \rightarrow 0$ . This makes the gradient  $\nabla L(\theta_t)$  to be aligned mostly with the top eigenvector  
492  $q_1$  of  $H$ , i.e.,  $\lim_{t \rightarrow \infty} |\cos(q_1, \nabla L(\theta_t))| = 1$ . Moreover,

$$\|H\|_{S_n} = \frac{\nabla L^\top H \nabla L}{\|\nabla L\|^2} = \sum \lambda_i (q_i^\top \frac{\nabla L(\theta_t)}{\|\nabla L\|})^2 = \sum \lambda_i \cos^2(q_i, \nabla L(\theta_t)) \rightarrow \lambda_1 = \|H\| \quad (35)$$

493 as  $t \rightarrow \infty$ . □

494 **Remark.** We implicitly ignore the set of measure zero that the initial point satisfies  $[\psi_0]_1 = 0$ .

495 **Remark.** Due to the exponential growth, it only takes a few steps (5-20) for  $\|H\|_{S_n}$  to exceed  $\frac{2}{\eta}$  (see  
496 Appendix C.3).

497 **Remark.**  $\|H\| = \lambda_1$  keeps increasing after it exceeds  $\frac{2}{\eta}$ . Therefore, we may relax the assumption as  
498  $L(\theta) = \frac{1}{2}\theta^\top Q\Lambda Q^\top \theta$  where the eigenvalues  $\lambda_1(\theta_t) > \frac{2}{\eta} + \epsilon_1$  and  $\epsilon_2 < \lambda_i(\theta_t) < \frac{2}{\eta} - \epsilon_3$  for  $i \neq 1$   
499 may change within the bounds over  $t$  for some  $\epsilon_j > 0$  ( $j = 1, 2, 3$ ), but  $Q$  is fixed. Then we obtain  
500 that, as  $t \rightarrow \infty$ ,

$$|q_1^\top \nabla_\theta L(\theta_t)| = |\lambda_1(\theta_t)[\psi_0]_1| \prod_{s=1}^t |1 - \eta \lambda_1(\theta_s)| \quad (36)$$

501 increases to  $\infty$  while

$$|q_i^\top \nabla_\theta L(\theta_t)| = |\lambda_i(\theta_t)[\psi_0]_i| \prod_{s=1}^t |1 - \eta \lambda_i(\theta_s)| \rightarrow 0, \text{ for } i \neq 1, \quad (37)$$

502 which draws the same conclusion except that the limit  $\|H\|_{S_n}$  may not exist because of varying  $\|H\|$   
503 according to  $t$ , but we can ensure that  $\|H\|_{S_n}$  eventually exceeds  $\frac{2}{\eta}$ .

## 504 C Experimental Settings and Additional Figures

505 We report the experimental results using vanilla SGD/GD without momentum and weight decay,  
 506 constant learning rate, and no data augmentation. We use a simple 6-layer CNN (6CNN,  $m = 0.51M$ ),  
 507 ResNet-9 [13]<sup>3</sup> ( $m = 2.3M$ ), WRN-28-2 [55] ( $m = 36M$ ). We use subsets of the datasets CIFAR-  
 508 10/100 [24]<sup>4</sup>, STL-10 [5]<sup>5</sup>, and Tiny-ImageNet<sup>6</sup> (a subset of ImageNet [7] with  $3 \times 64 \times 64$  images and  
 509 200 object classes) where DATASET- $n$  denotes a subset of DATASET with  $|\mathcal{D}| = n$  and  $k=2^{10} = 1024$ .  
 510 Moreover, we use PyHessian [53]<sup>7</sup> to compute the Hessian-vector product (e.g.,  $H \nabla L$ ), the top  
 511 eigenvalue  $\lambda_1$  and its corresponding eigenvector  $q_1$  of the Hessian. For these computations, we use  
 512 the power iterations with a batch size of 2k, the tolerance of 0.001, and the maximum iteration of 100.

### 513 C.1 Figure 1

514 In Figure 1 and Figure 2, we plot  $\frac{\mathbb{E}[L_{t+1}] - L_t}{\text{tr}(S_n)}$  against  $\frac{\text{tr}(HS_b)}{\text{tr}(S_n)}$ , which is equivalent to  $\frac{L_{t+1} - L_t}{\text{tr}(S_n)}$  against  
 515  $\|H\|_{S_n}$  for GD. Therefore, we expect the following linear relationship with the slope  $\frac{\eta^2}{2}$  and the  
 516 x-intercept  $\frac{2}{\eta}$  when the training loss  $L$  is locally quadratic, i.e.,  $\epsilon = 0$ :

$$\frac{\mathbb{E}[L_{t+1}] - L_t}{\text{tr}(S_n)} = \frac{\eta^2}{2} \left( \frac{\text{tr}(HS_b)}{\text{tr}(S_n)} - \frac{2}{\eta} \right) \quad (38)$$

$$\frac{L_{t+1} - L_t}{\text{tr}(S_n)} = \frac{\eta^2}{2} \left( \|H\|_{S_n} - \frac{2}{\eta} \right) \quad (39)$$

517 Figure 1 shows the behavior in the early phase, until the iterate enters the edge of stability. For GD,  
 518 they are plotted *after*  $\|H\|$  exceeds  $\frac{2}{\eta}$  after which  $\|H\|_{S_n}$  starts to increase from 0 to  $\frac{2}{\eta}$  in a few steps.  
 519 For cross-entropy loss, we mark the end point with ‘x’ when the iterate enters the unstable regime.

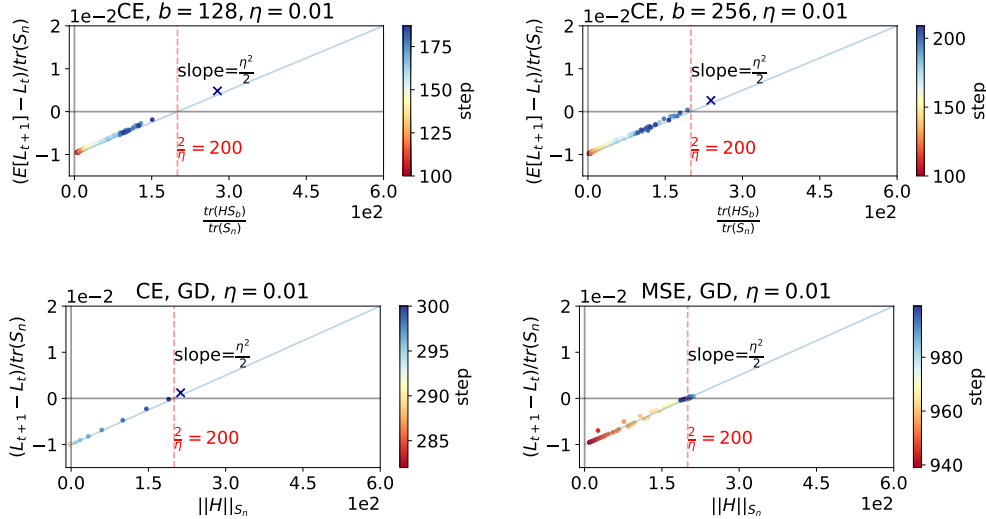


Figure 6: 6CNN with  $\eta = 0.01$ . See caption of Figure 1 for more details.

<sup>3</sup>from <https://github.com/wbaek/torchskeleton>, Apache-2.0 license

<sup>4</sup><https://www.cs.toronto.edu/~kriz/cifar.html>

<sup>5</sup><https://cs.stanford.edu/~acoates/stl10/>

<sup>6</sup><https://tiny-imagenet.herokuapp.com>

<sup>7</sup><https://github.com/amirgholami/PyHessian>, MIT license

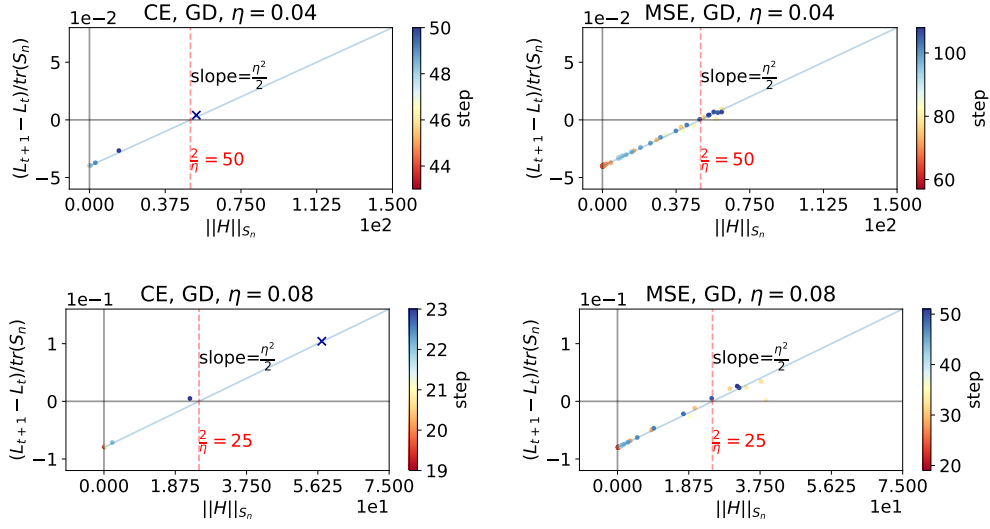


Figure 7: 6CNN with CE/MSE (left/right) and  $\eta = 0.04/0.08$  (top/bottom). See caption of Figure 1 for more details.

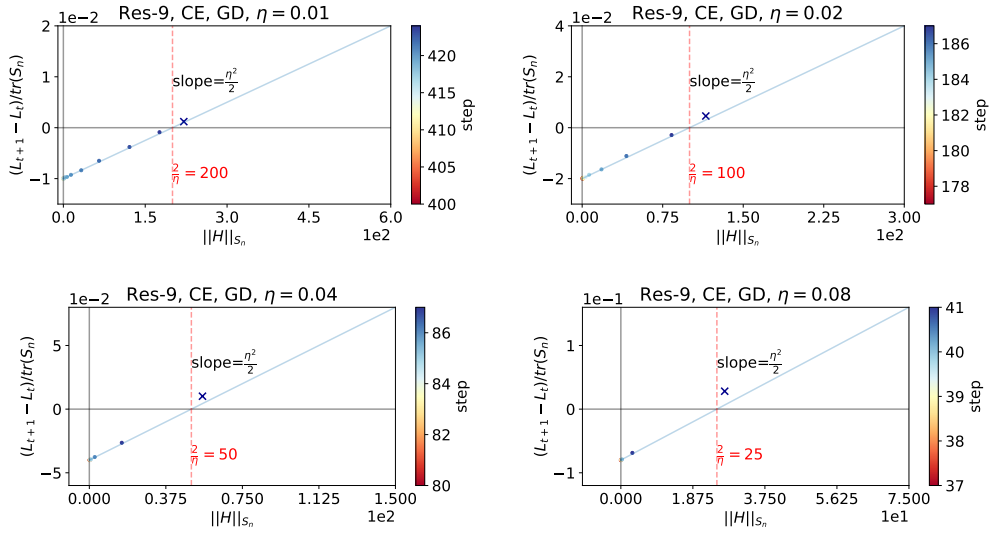


Figure 8: ResNet-9 with CE and  $\eta = 0.01/0.02/0.04/0.08$ . See caption of Figure 1 for more details.

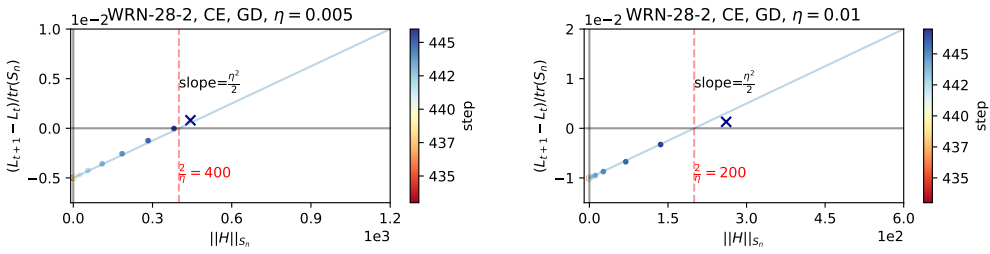


Figure 9: WRN-28-2 with  $\eta = 0.005/0.01$ . See caption of Figure 1 for more details.

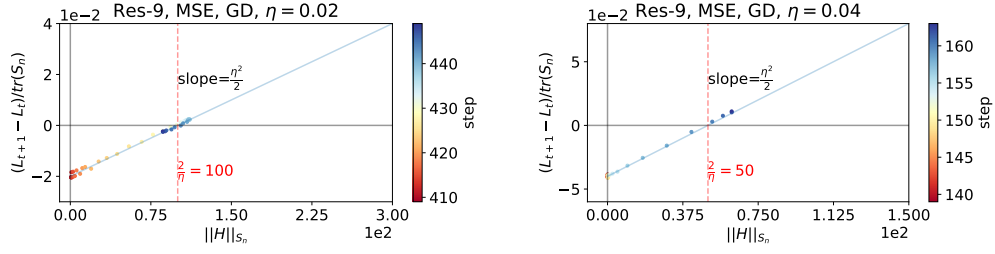


Figure 10: ResNet-9 with MSE and  $\eta = 0.02/0.04$ . See caption of Figure 1 for more details.

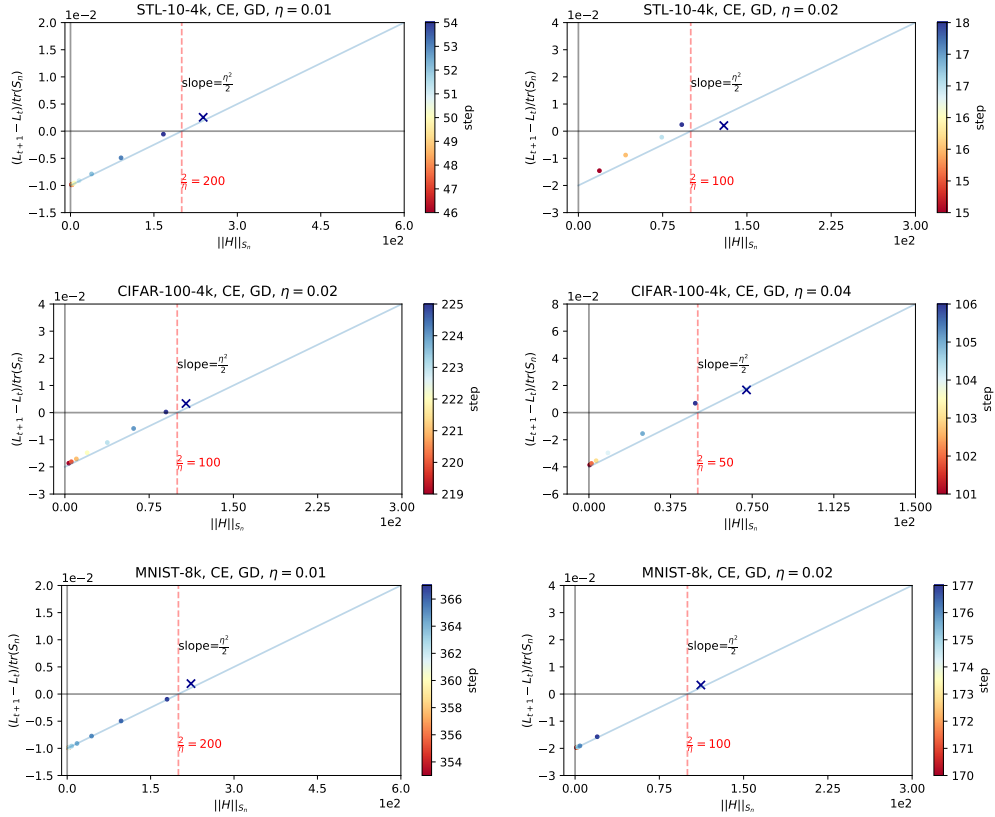


Figure 11: (DATASET,  $\eta$ ) = (STL-10-4k, 0.01/0.02), (CIFAR-100-4k, 0.02/0.04), (MNIST-8k, 0.02/0.04). See caption of Figure 1 for more details.

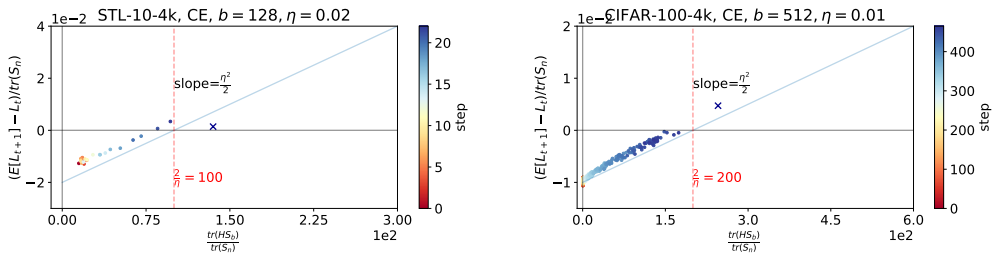


Figure 12: (DATASET,  $b$ ,  $\eta$ ) = (STL-10-4k, 128, 0.02), (CIFAR-100-4k, 512, 0.01), (MNIST-8k, 0.02/0.04). See caption of Figure 1 for more details.

520 **C.2 Figure 2**

521 Figure 2 shows the non-quadraticity (represented by deviation from the blue line in Figure 2 and the  
 522 following figures) of the training loss function  $L$  after the iterate enters the edge of stability.

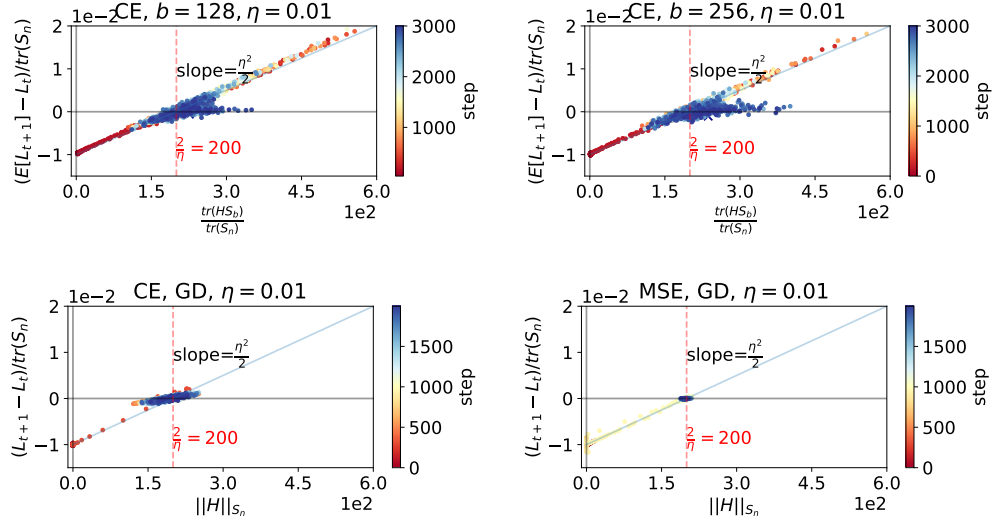


Figure 13: 6CNN with  $\eta = 0.01$ . See caption of Figure 2 for more details.

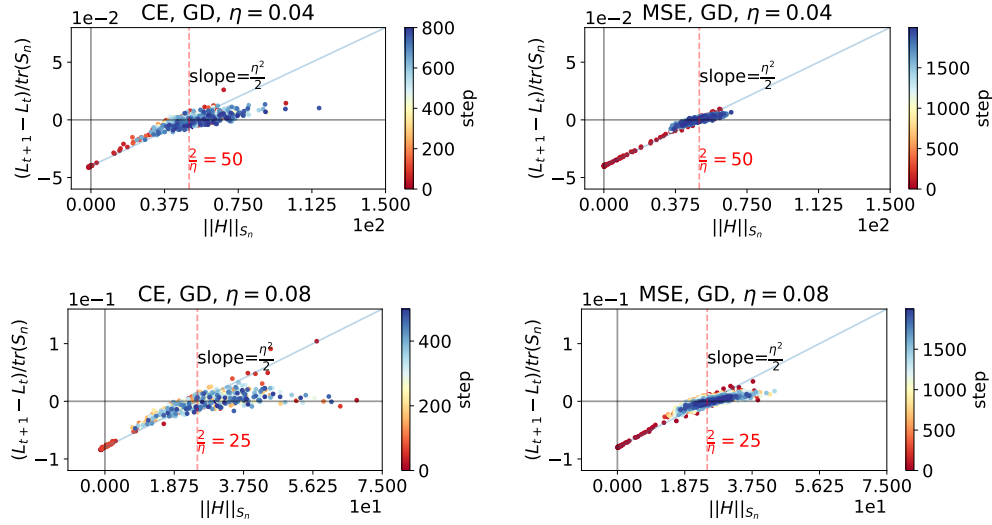


Figure 14: 6CNN with CE/MSE (left/right) and  $\eta = 0.04/0.08$  (top/bottom). See caption of Figure 2 for more details.

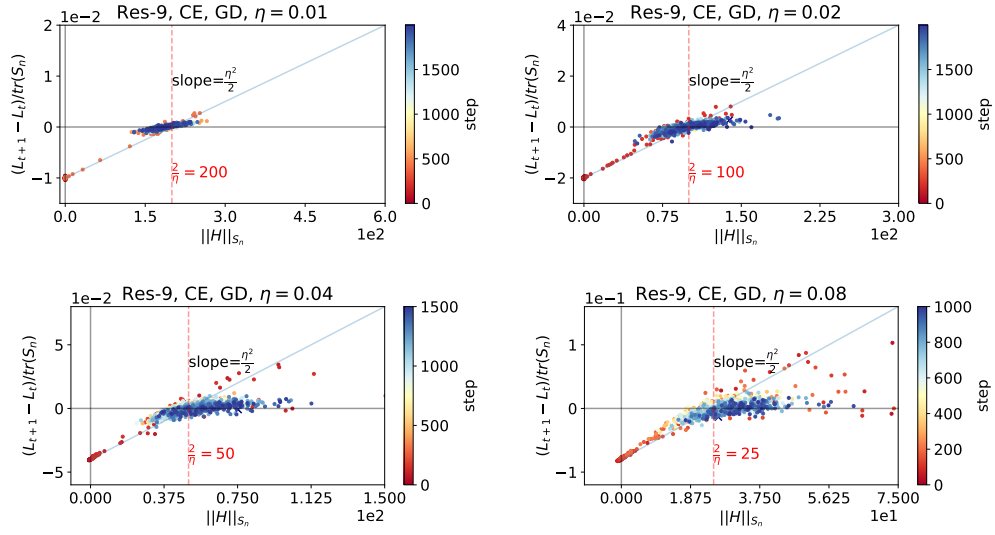


Figure 15: ResNet-9 with CE and  $\eta = 0.01/0.02/0.04/0.08$ . See caption of Figure 2 for more details.

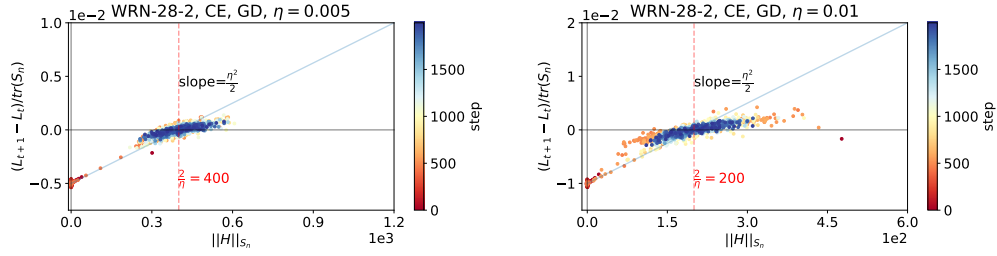


Figure 16: WRN-28-2 with  $\eta = 0.005/0.01$ . See caption of Figure 2 for more details.

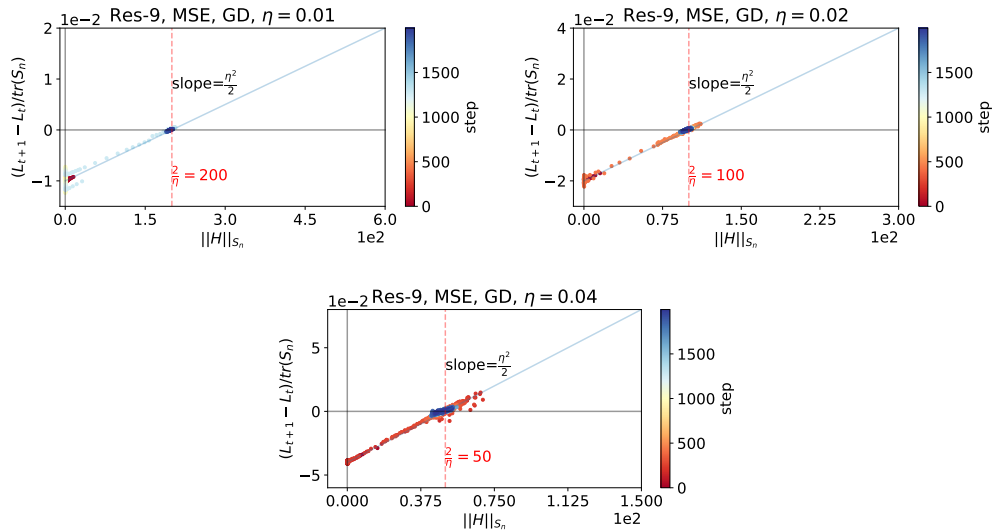


Figure 17: ResNet-9 with MSE and  $\eta = 0.01/0.02/0.04$ . See caption of Figure 2 for more details.

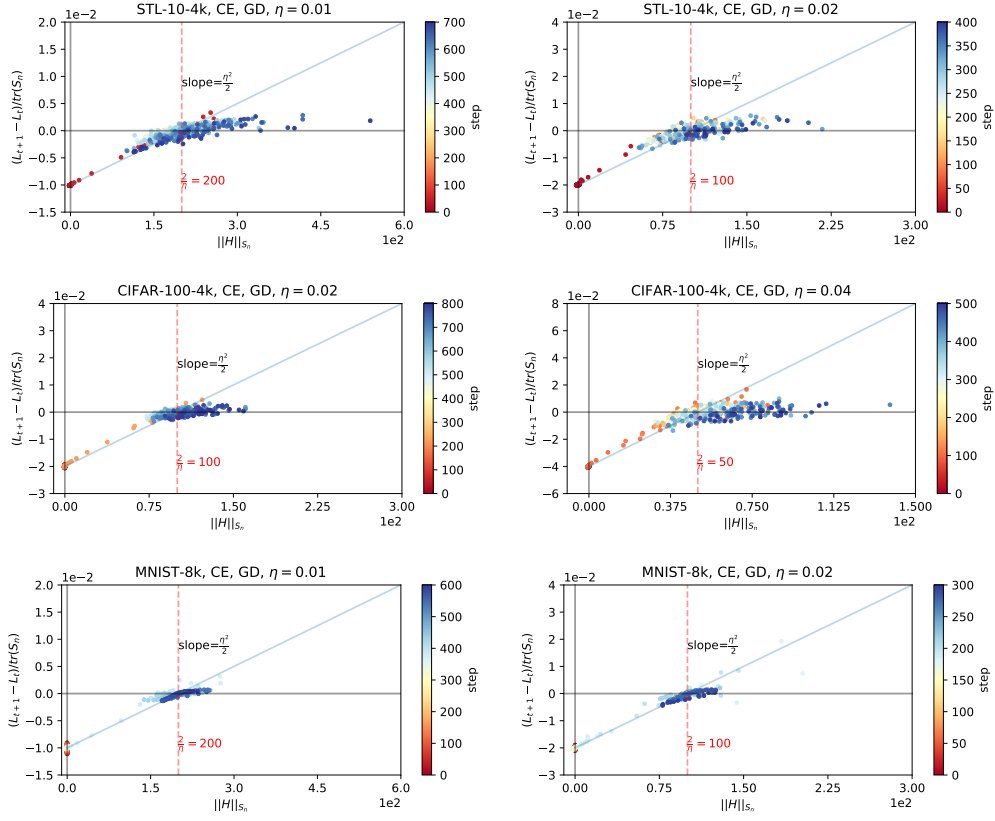


Figure 18: (DATASET,  $\eta$ ) = (STL-10-4k, 0.01/0.02), (CIFAR-100-4k, 0.02/0.04), (MNIST-8k, 0.01/0.02). See caption of Figure 2 for more details.

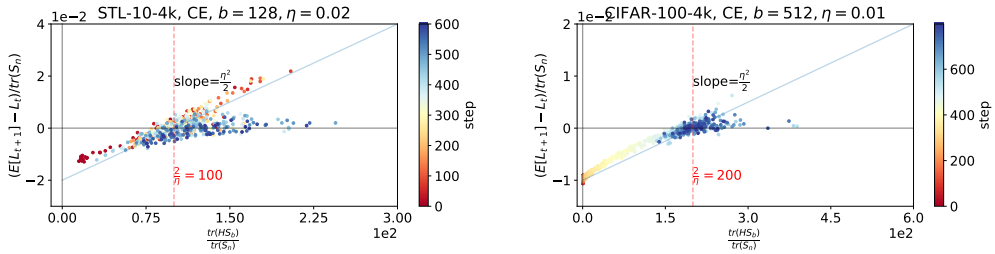


Figure 19: (DATASET,  $b$ ,  $\eta$ ) = (STL-10-4k, 128, 0.02), (CIFAR-100-4k, 512, 0.01). See caption of Figure 2 for more details.

523 **C.3 Figure 3 and Prop. 4.1**

524 Figure 20-21 provide additional information of Figure 3. After these figures, we focus on providing  
 525 some empirical evidences of Prop. 4.1 that  $\|H\|_{s_n}, |\cos(q_1, \nabla L)|$  and  $|q_1^\top \nabla L|$  increase in a few  
 526 steps after  $\|H\|$  exceeds  $\frac{2}{\eta}$ .

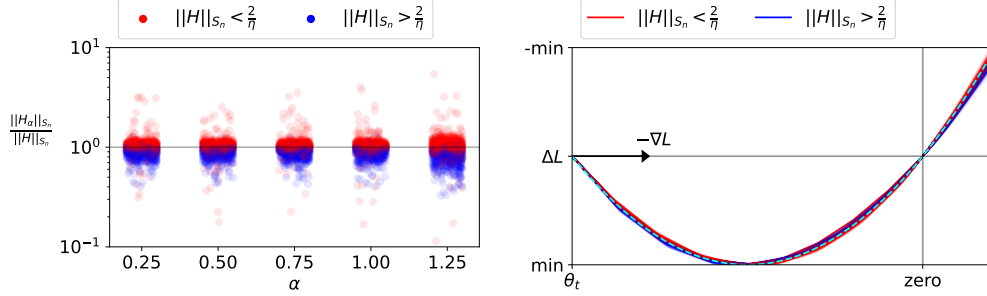


Figure 20: 6CNN with  $\eta = 0.02$ . See caption of Figure 3 for more details.

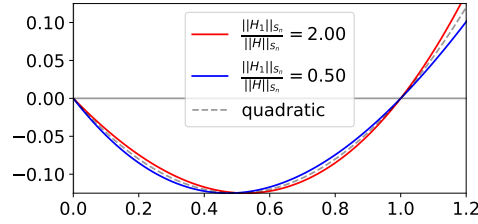


Figure 21: To gain some intuitions of Figure 3 and 20 (right), we plot three graphs—a quadratic function  $f(x) = \frac{1}{2}x(x-1)$  (black dashed line), and cubic functions  $f_1(x) = \alpha_1 x(x-1)(1 + \frac{1}{4}x)$  in red and  $f_2(x) = \alpha_2 x(x-1)(1 - \frac{1}{5}x)$  in blue. They are chosen to satisfy  $\frac{\partial^2 f_1}{\partial x^2}|_{x=1} / \frac{\partial^2 f_1}{\partial x^2}|_{x=0} = 2$  and  $\frac{\partial^2 f_2}{\partial x^2}|_{x=1} / \frac{\partial^2 f_2}{\partial x^2}|_{x=0} = 0.5$ . We also choose  $\alpha_1$  and  $\alpha_2$  for  $f_1$  and  $f_2$  to have the same minimum with  $f$  in  $x \in [0, 1]$ , i.e.,  $\min_{x \in [0, 1]} f(x) = \min_{x \in [0, 1]} f_i(x) = -\frac{1}{8}$  for  $i = 1, 2$ .

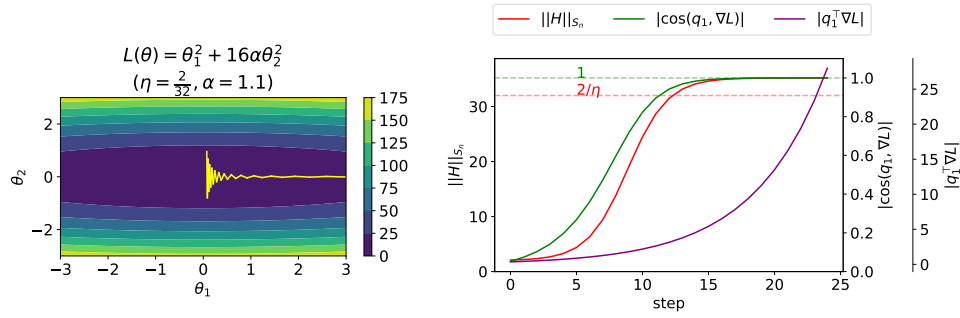


Figure 22: Left: A toy two-dimensional loss function  $L(\theta) = \theta_1^2 + 16\alpha\theta_2^2$  where  $\alpha = 1.1$ . We optimize the loss by GD with  $\eta = \frac{2}{32}$  so that  $\|H\| = 32\alpha > \frac{2}{\eta} = 32$ . We also plot the GD trajectory in yellow starting from  $(\theta_1, \theta_2) = (3, 0.1)$ . Right: We show the exponential increase in  $|q_1^\top \nabla L|$  (purple) and the S-shape increase in  $\|H\|_{s_n}$  (red) and  $|\cos(q_1, \nabla L)|$  (green) to  $\|H\|$  and 1, respectively, which empirically demonstrates Prop. 4.1. We also note that they start to increase in the order of  $|\cos(q_1, \nabla L)|$ ,  $\|H\|_{s_n}$  and  $|q_1^\top \nabla L|$ .



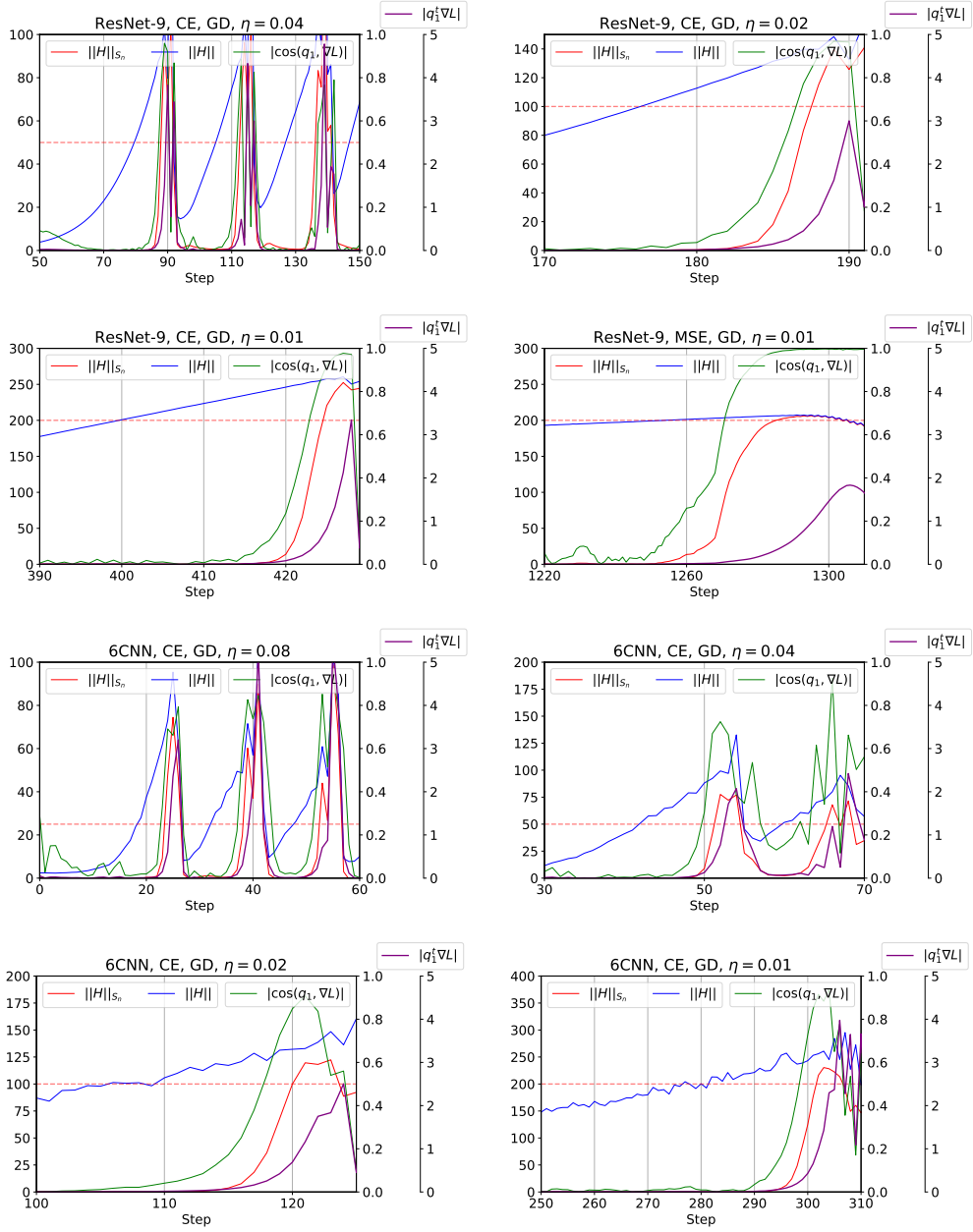


Figure 23: After  $\|H\|$  (blue) exceeds  $\frac{2}{\eta}$  (red dashed line) at  $t \approx 80/176/400/1250/19/43/110/280$ , in a few steps ( $\approx 5/6/18/5/5/6/5/15$ ),  $\|H\|_{S_n}$  (red) starts to increase. As expected in Prop. 4.1,  $\|H\|_{S_n}$  increases together with  $|\cos(q_1, \nabla L)|$  (green) and  $|q_1^T \nabla L|$  (purple). They are observed to start to increase in the order of  $|\cos(q_1, \nabla L)|$ ,  $\|H\|_{S_n}$  and  $|q_1^T \nabla L|$ , as shown in Figure 22.

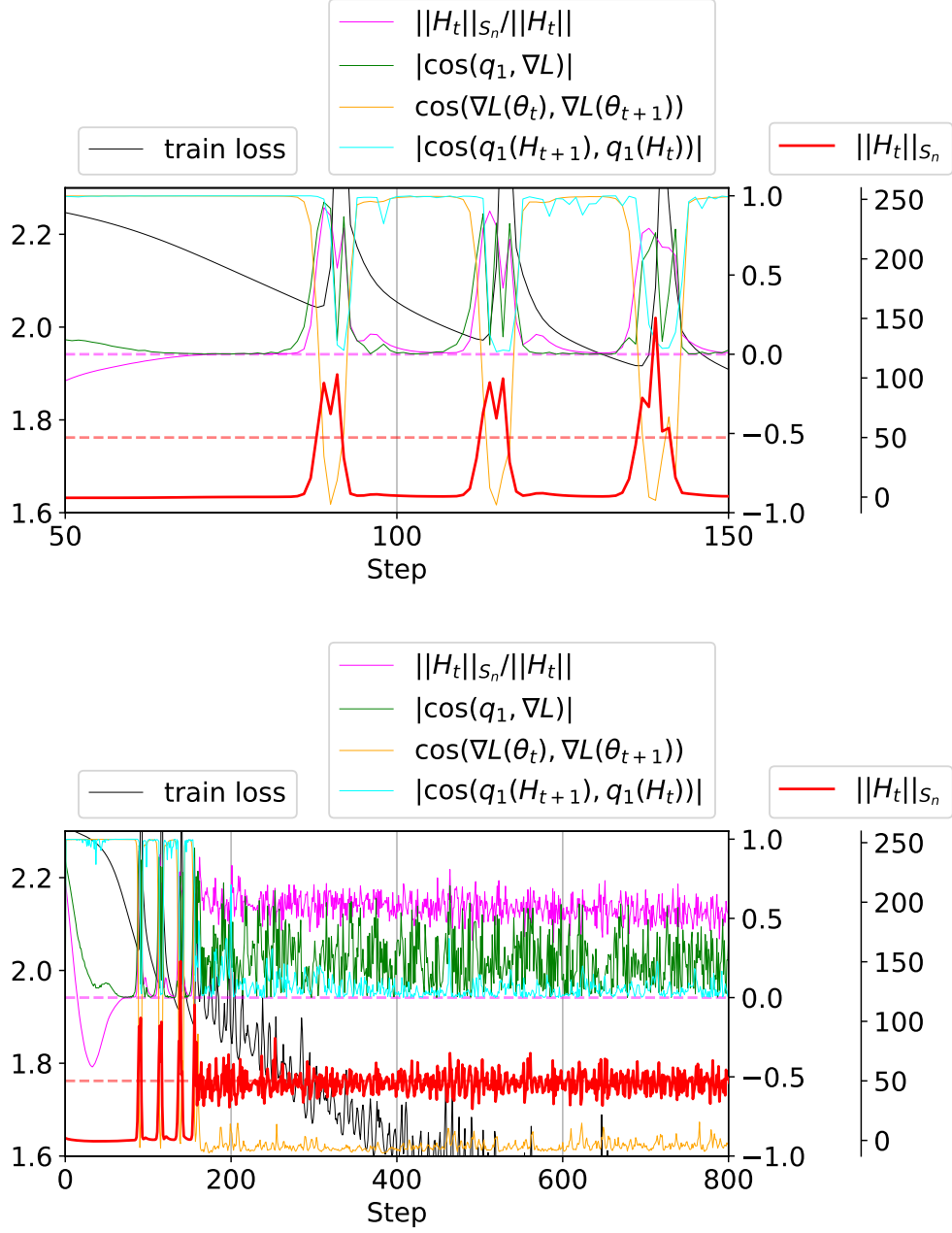


Figure 24: After  $\|H\|$  exceeds  $\frac{2}{\eta}$  (not shown in this Figure, see Figure 23), the cosine  $|\cos(q_1(H_t), \nabla L(\theta_t))|$  (green) between the sharpest direction and the gradient gets large, where  $H_t = H(\theta_t)$ . Simultaneously,  $\|H_t\|_{S_n}$  increases and exceeds  $\frac{2}{\eta}$  (red solid > red dashed), the iterate entering the unstable regime and oscillating with  $\cos(\nabla L(\theta_t), \nabla L(\theta_{t+1})) \approx -1$  (orange). However, due to the non-quadraticity, the sharpest direction changes with  $|\cos(q_1(H_{t+1}), q_1(H_t))|$  (cyan) close to 0. We train ResNet-9 on CIFAR-10-8k with  $\eta = 0.04$  (top: steps 50-150, bottom: steps 0-800).

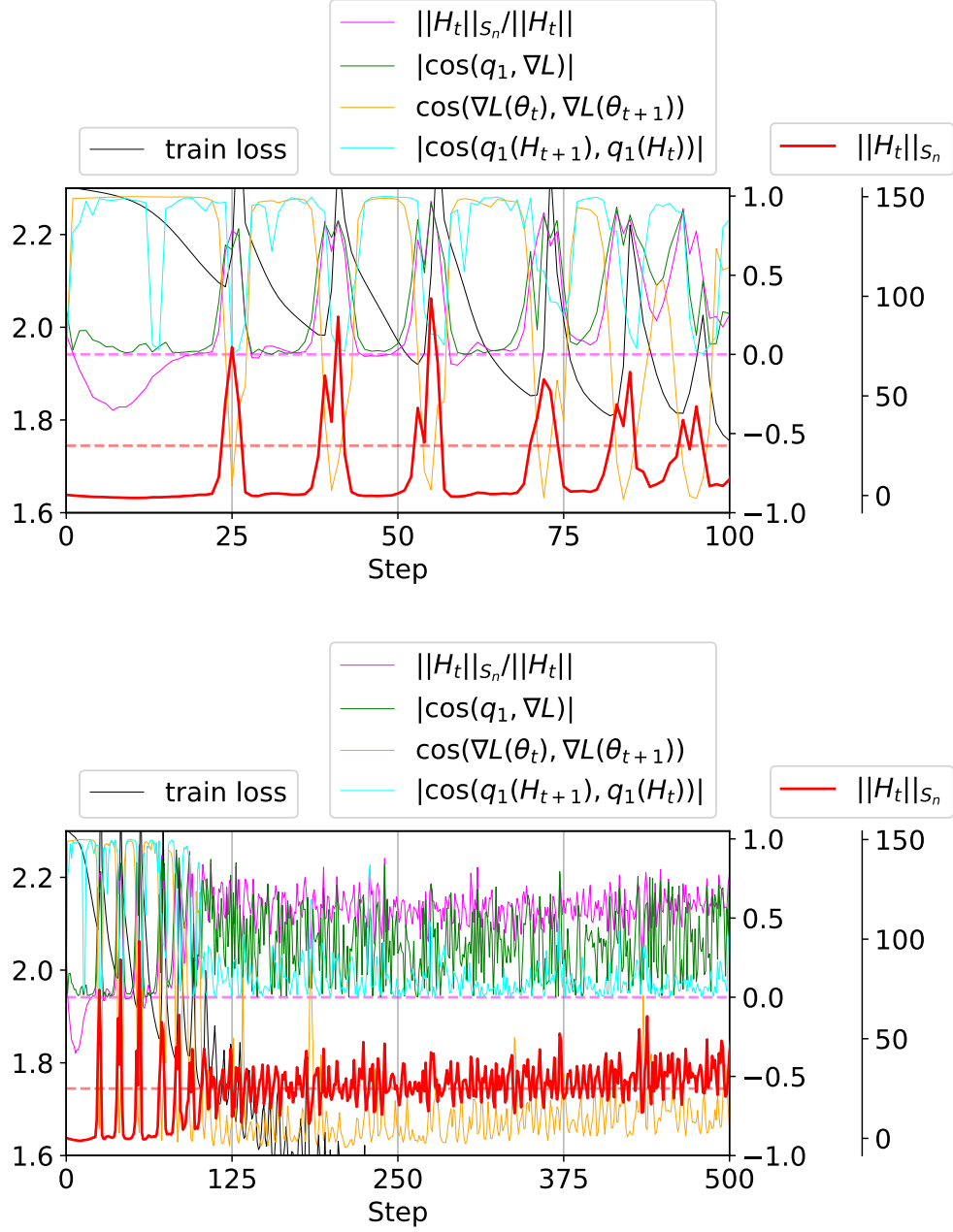


Figure 25: We train ResNet-9 on CIFAR-10-8k with  $\eta = 0.08$  (top: steps 0-100, bottom: steps 0-500). See caption of Figure 24 for more details.

527 **C.4 Figure 4**

528 **C.4.1 GD**

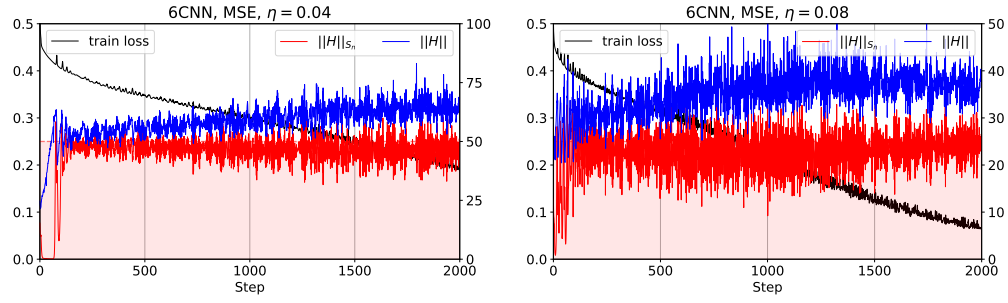


Figure 26: 6CNN, MSE, and  $\eta = 0.04/0.08$ . See caption of Figure 4 for more details.

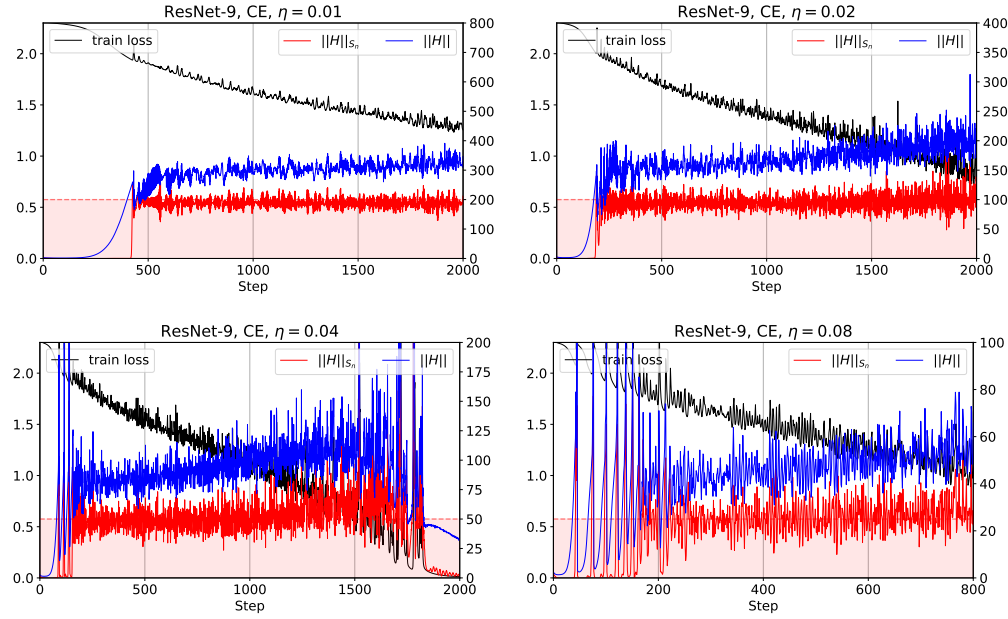


Figure 27: ResNet-9, CE, and  $\eta = 0.01/0.02/0.04/0.08$ . See caption of Figure 4 for more details.

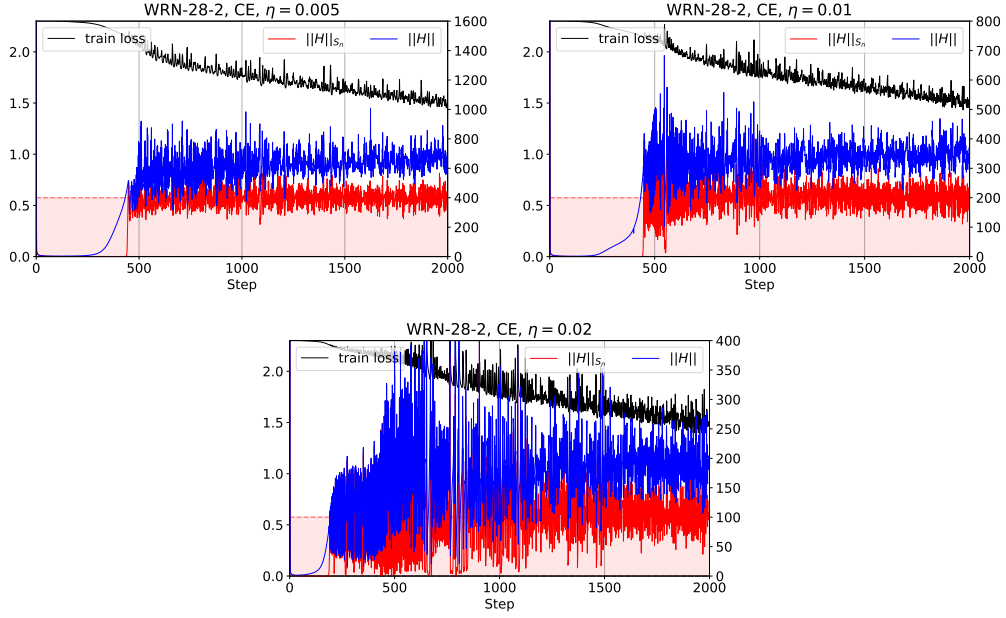


Figure 28: WRN-28-2, CE, and  $\eta = 0.005/0.01/0.02$ . See caption of Figure 4 for more details.

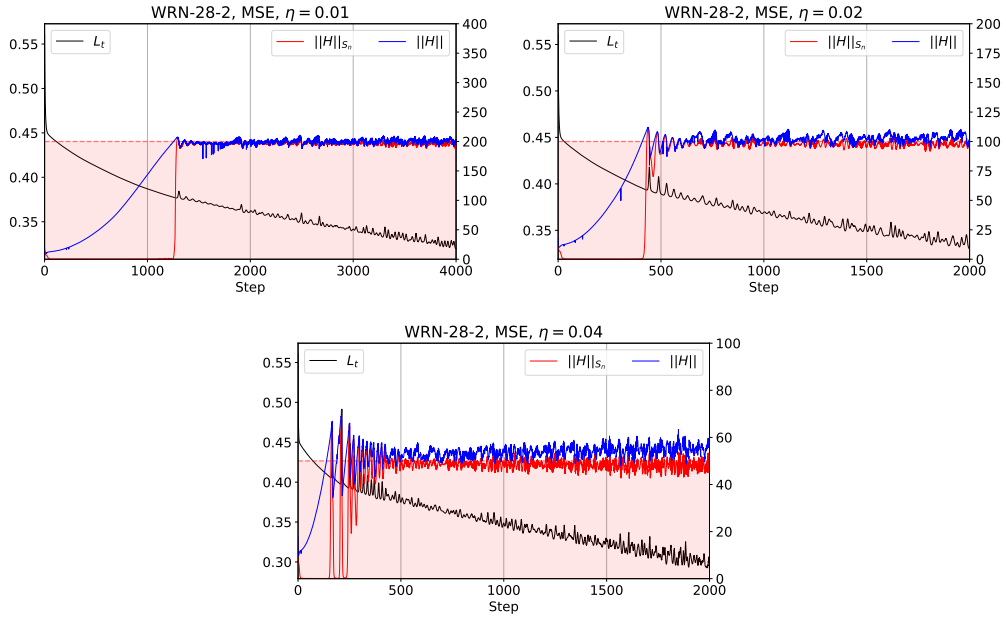


Figure 29: WRN-28-2, MSE, and  $\eta = 0.01/0.02/0.04$ . See caption of Figure 4 for more details.

## 529 C.4.2 SGD

530 We demonstrate two types of IIR, (i)  $\|H\|_{S_b} \leq \frac{2\rho_b}{\eta}$  and (ii)  $\frac{\text{tr}(HS_b)}{\text{tr}(S_n)} \leq \frac{2}{\eta}$  in Figure 30 and 31,  
 531 respectively. They are equivalent, but the latter shows the effects of IIR more clearly as it has the fixed threshold  $\frac{2}{\eta}$  regardless of  $b$ .

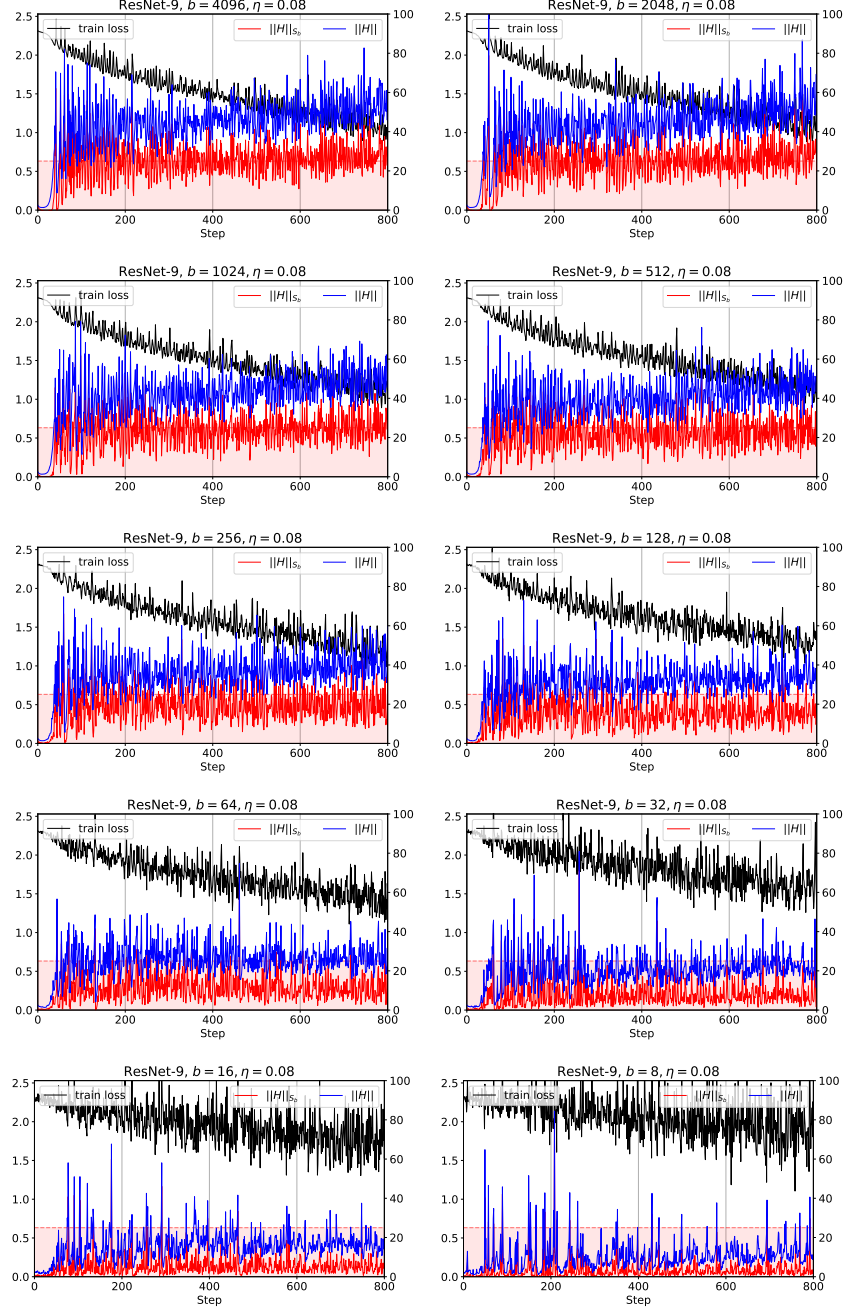


Figure 30: IIR of  $\|H\|_{S_b} \leq \frac{2\rho_b}{\eta}$  for SGD. We plot  $\frac{2}{\eta}$  (red dashed line) which is not the threshold for  $\|H\|_{S_b}$ . ResNet-9, CE,  $\eta = 0.08$ , and  $b \in \{2^{12}, 2^{11}, \dots, 2^3\}$ . See caption of Figure 4 for more details.

532

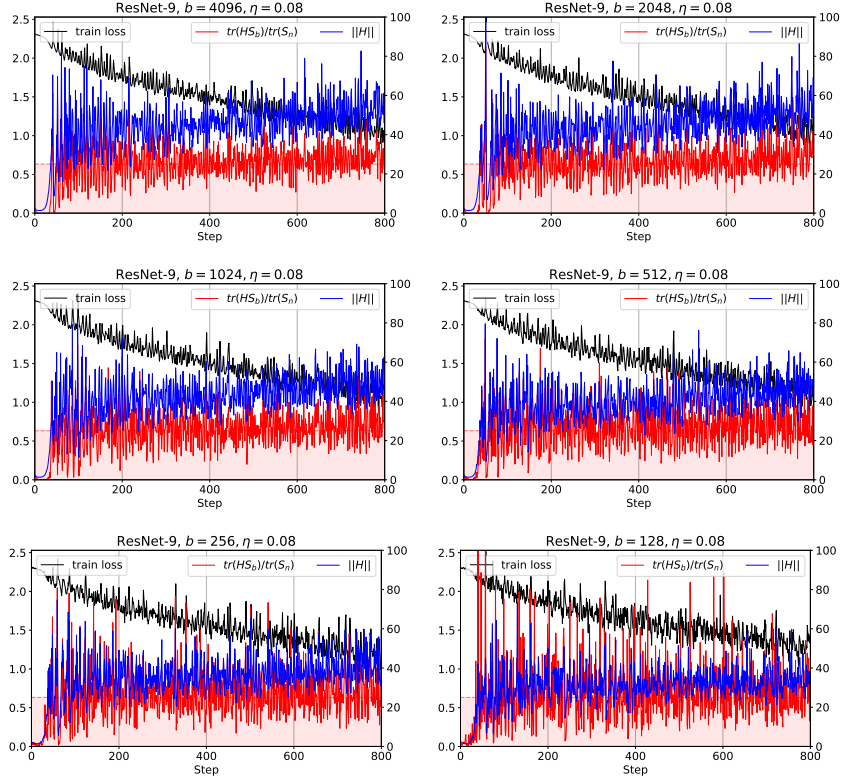


Figure 31: IIR of  $\frac{\text{tr}(HS_b)}{\text{tr}(S_n)} \leq \frac{2}{\eta}$  for SGD. With the upper bound  $\frac{2}{\eta}$  (red dashed line), this shows the effects of IIR more clearly than Figure 30. ResNet-9, CE,  $\eta = 0.08$ , and  $b \in \{2^{12}, 2^{11}, \dots, 2^7\}$ .

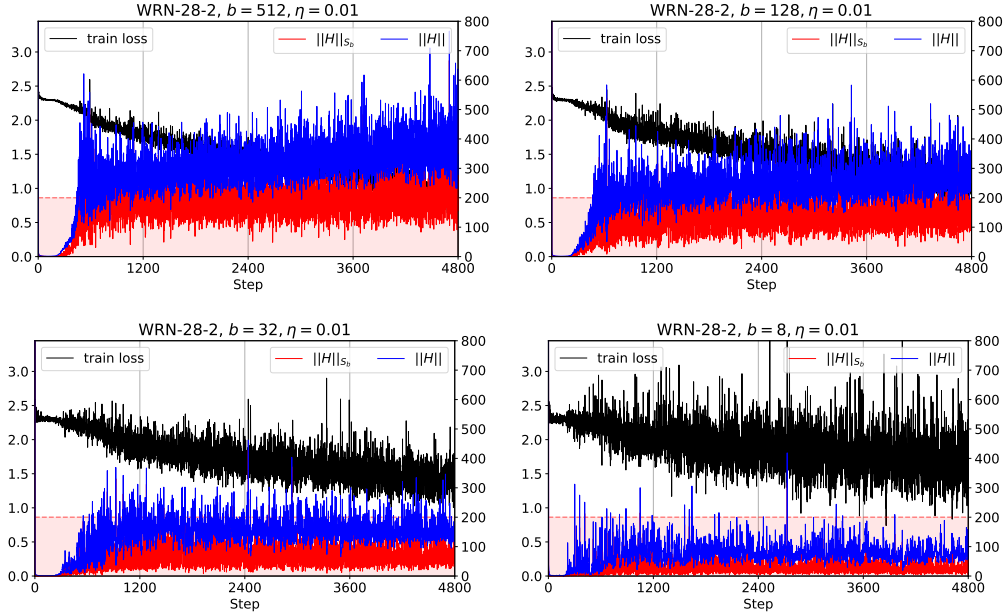


Figure 32: IIR of  $\|H\|_{S_b} \leq \frac{2\rho_b}{\eta}$  for SGD. WRN-28-2, CE,  $\eta = 0.01$ , and  $b \in \{2^9, 2^7, 2^5, 2^3\}$ . See caption of Figure 4 for more details.

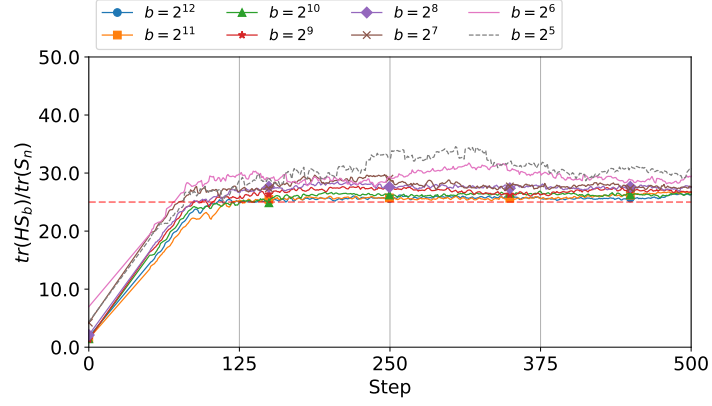


Figure 33: IIR of  $\frac{\text{tr}(HS_b)}{\text{tr}(S_n)} = \frac{\|H\|_{S_b}}{\rho_b} \leq \frac{2}{\eta}$  for SGD with different  $b$ 's. Curves are smoothed for visual clarity.

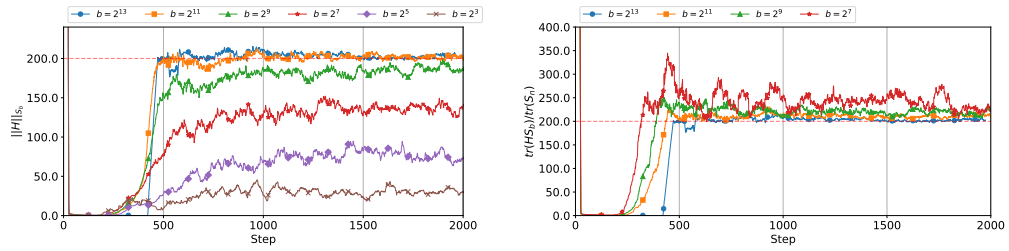


Figure 34: Left: See caption of Figure 4(d) for more details. Right: See caption of Figure 33 for more details. WRN-28-2, CIFAR-10-8k,  $\eta = 0.01$ .



533 **C.5 Figure 5**

534 Figure 35-36 provide some additional information of Figure 5. Figure 37 shows that  $1/\rho \approx 100$   
 535 is much larger than 1 and that  $\text{std}[\|g_b\|]$  is  $2\text{-}3\times$  smaller than  $\mathbb{E}[\|g_b\|]$  even in the case of  $b = 1$ .  
 536 Therefore, we use the approximation  $\|g_b\| \approx \mathbb{E}[\|g_b\|]$ , and thus the square of the mean resultant  
 537 length is similar to the concentration measure  $\rho_b$  as shown in the following approximation:

$$\bar{R}_b^2 \equiv \left\| \mathbb{E} \left[ \frac{g_b}{\|g_b\|} \right] \right\|^2 \approx \left\| \mathbb{E} \left[ \frac{g_b}{\mathbb{E}[\|g_b\|]} \right] \right\|^2 = \left\| \frac{\mathbb{E}[g_b]}{\mathbb{E}[\|g_b\|]} \right\|^2 = \rho_b. \quad (40)$$

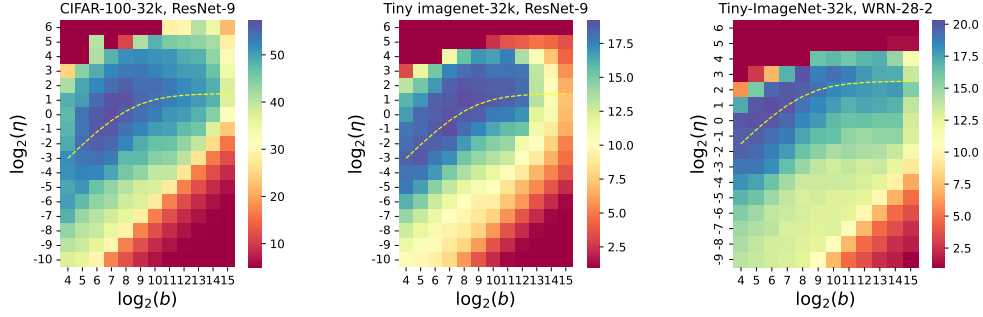


Figure 35: (DATASET, MODEL, EPOCHS) = (CIFAR-100-32k, ResNet-9, 800 epochs), (Tiny-ImageNet-32k, ResNet-9, 400 epochs), (Tiny-ImageNet-32k, WRN-28-2, 800 epochs). Middle: The model is trained for 400 epochs, which is short compared to Figure 5 (bottom right heatmap). See caption of Figure 5 (right) for more details.

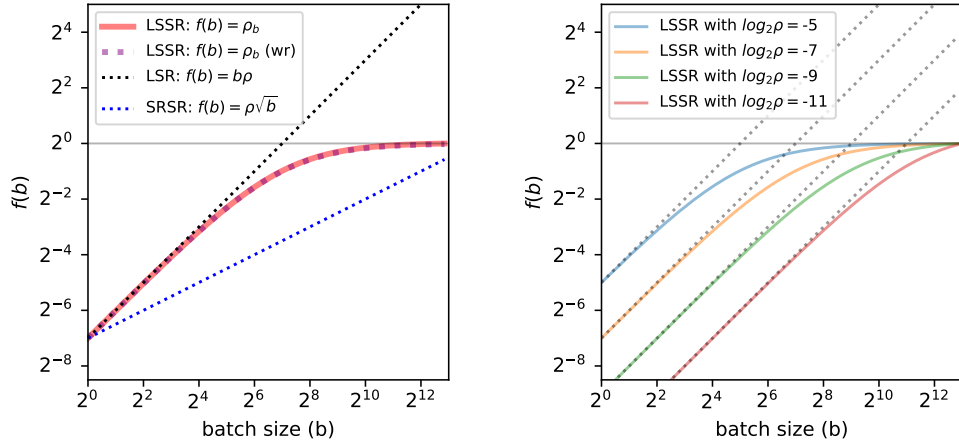


Figure 36: Left: We use  $\gamma_{n,b} = 1$  for sampling with replacement (‘wr’, dotted purple curve), which is almost equivalent to the “without replacement” counterpart (red). Right: LSSR for different  $\rho$  values with the corresponding LSR (dotted lines). See caption of Figure 5 (left) for more details.

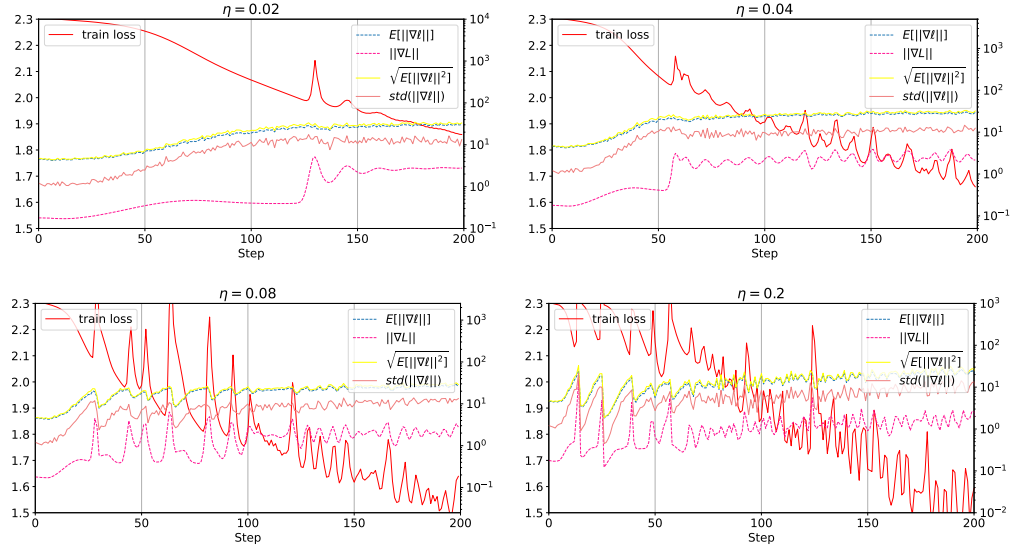


Figure 37: To further understand the concentration measure  $\rho$  of the per-example gradient, we plot  $\mathbb{E}[\|\nabla\ell\|]$ ,  $\|\nabla L\| = \|\mathbb{E}[\nabla\ell]\|$ ,  $\sqrt{\text{tr}(S_1)} = \mathbb{E}[\|\nabla\ell\|^2]$  and  $\text{std}[\|\nabla\ell\|]$ . We use 100 samples to compute the expectation values and the standard deviation. Here,  $\frac{1}{\rho} = \frac{\mathbb{E}[\|\nabla\ell\|^2]}{\|\nabla L\|^2}$  is about 100.  $\mathbb{E}[\|\nabla\ell\|]$  is 2-3  $\times$  larger than  $\text{std}[\|\nabla\ell\|]$ . We train a 6CNN with  $\eta = 0.02/0.04/0.08/0.2$  (GD).