Edge of Stochastic Stability: Revisiting the Edge of Stability for SGD

Anonymous authors
Paper under double-blind review

Abstract

Recent findings by Cohen et al. (2021) demonstrate that when training neural networks with full-batch gradient descent with step size η, the largest eigenvalue $\lambda_{\max}$ of the full-batch Hessian consistently stabilizes at $\lambda_{\max} = 2/\eta$. These results have significant implications for convergence and generalization. This, however, is not the case of mini-batch stochastic gradient descent (SGD), limiting the broader applicability of its consequences. We show that SGD trains in a different regime we term Edge of Stochastic Stability (EoSS). In this regime, what stabilizes at $2/\eta$ is *Batch Sharpness*: the expected directional curvature of mini-batch Hessians along their corresponding stochastic gradients. As a consequence, $\lambda_{\max}$—which is generally smaller than *Batch Sharpness*—is suppressed, aligning with the long-standing empirical observation that smaller batches and larger step sizes favor flatter minima. We further discuss implications for mathematical modeling of SGD trajectories.

1 Introduction

The choice of training algorithm is a key ingredient in the deep learning recipe. Extensive evidence, e.g. (Keskar et al., 2016), indeed shows that performance consistently depends on the optimizer and hyperparameters. What machinery induces this optimizer-dependence is a central question of theory of deep learning.

Cohen et al. (2021; 2024) answered this question for Gradient Descent (GD): it optimizes neural networks in a regime of instability, they termed Edge of Stability (EoS). With a constant step size η, the highest eigenvalue of the Hessian of the full-batch loss—denoted here as $\lambda_{\max}$—grows until $2/\eta$ and hovers right above, subject to small oscillations (Cohen et al., 2021; 2022; Jastrzębski et al., 2019; 2020; Xing et al., 2018). Although, classical convex optimization theory call this step size "too large", the loss continues to decrease. These works established a number of surprising facts: **(1)** we require an optimization theory which works in more general scenarios then the classical $\eta < 2/L$; **(2)** what a source of instability of (pre-)training is. **(3)** how location of convergence depends on the choice of hyperparameters.

While real-world training is almost always *mini-batch*—given the large amounts of data—existing EoS analyses **explicitly** do not apply to this case: no curvature-type quantities, as $\lambda_{\max}$, are known to similarly affect SGD while training neural networks. We bridge this gap by establishing that:

Mini-batch SGD trains in a similar regime of instability which we term Edge of Stochastic Stability (EoSS). Precisely, *Batch Sharpness*, our notion of curvature,

$$\textit{Batch Sharpness}(\theta) \;:=\; \mathbb{E}_{B\sim\mathcal{P}_b}\left[\frac{\nabla L_B(\theta)^\top \mathcal{H}(L_B)\,\nabla L_B(\theta)}{\|\nabla L_B(\theta)\|^2}\right], \qquad \text{with } L_B \text{ loss on the batch } B \text{ sampled from } \mathcal{P}_b.$$

hovers around $2/\eta$ and implicitly functions as sharpness for SGD. This implies that:
stability for SGD is stability on the mini-batch landscape

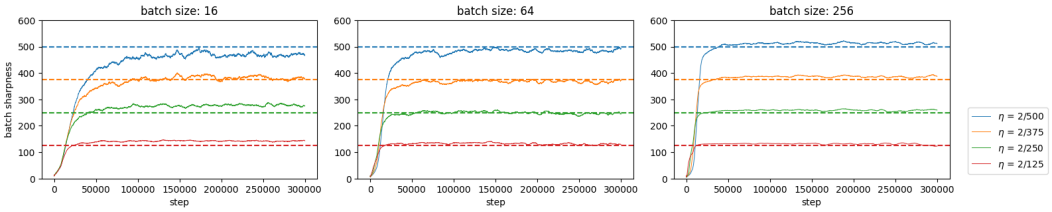


Figure 1: **SGD at EoSS under different step sizes and batch sizes.** MLP on an 8k subset of CIFAR-10 with step size $\eta > 0$. *Batch Sharpness* stabilizes at the $2/\eta$ threshold across varying batch sizes and step sizes.

Organization and Contributions. Section 2 reviews related work and outlines the key open questions we tackle. Oscillations are central to these phenomena, as a necessary step, in Section 3 we distinguish SGD oscillations between noise-driven (as in Robbins-Monro–type of stochastic optimization when the step size is kept fixed) and curvature-driven—which are the ones we are interested in. In Section 4, we introduce, properly characterize, and empirically validate the phenomenon of Edge of Stochastic Stability. In Section 5 we give a mathematical treatment of SGD stability. Finally, our results are yet another proof of the fact that the dynamics of noise-injected GD or SDEs and the dynamics of mini-batch SGD are qualitatively different and studying the firsts could be misleading for inducing properties of the second. We discuss this implication in Section 6.

Throughout the rest of this paper $B \subset \mathcal{D}$ denotes a random mini-batch of size b drawn from a fixed sampling distribution $\mathcal{P}_b$. For model parameters $\theta \in \mathbb{R}^d$ let $L_B(\theta) = \frac{1}{b}\sum_{(x_i,y_i)\in B}\tilde{L}\big(f_\theta(x_i),\, y_i\big)$, $L(\theta) = \mathbb{E}_{B\sim\mathcal{P}_b}\big[L_B(\theta)\big]$ be the mini-batch and full-batch losses, respectively. Write $\mathcal{H}(L_B) = \nabla^2_\theta L_B(\theta)$.

2 Related Work

Progressive sharpening. Early studies observed that the local shape of the loss landscape changes rapidly at the beginning of the training, by means of growth of different estimators of the curvature (Keskar et al., 2016; Jastrzębski et al., 2019; LeCun et al., 2012; Achille et al., 2017; Jastrzębski et al., 2018; Fort and Ganguli, 2019; Sagun et al., 2016; Fort and Scherlis, 2019). Subsequently, Jastrzębski et al. (2019; 2020) and Cohen et al. (2021) precisely characterized this behavior, demonstrating a steady rise in $\lambda_{\max}$ along GD and SGD trajectories, typically following a brief initial decline. This phenomenon was termed *progressive sharpening* by Cohen et al. (2021).

Full-batch edge of stability. Prior research (Goodfellow et al., 2016; Li et al., 2019; Jiang et al., 2019; Lewkowycz et al., 2020) found that large initial learning rates often enhance generalization despite delaying initial loss reduction. Jastrzębski et al. (2020) attributed this effect to a phase transition, termed the break-even point, marking the end of progressive sharpening. Unlike progressive sharpening, this phenomenon is considered to result from algorithmic instability rather than inherent landscape properties. Indeed, Jastrzębski et al. (2019; 2020); Cohen et al. (2021; 2022) demonstrated that this phase transition comes at different points for different algorithms on the same landscapes. Cohen et al. (2021; 2022) later showed that it comes at the instability thresholds, in the case of full-batch optimization algorithms. Precisely, GD and full-batch Adam train in the EoS oscillatory regime (Cohen et al., 2021; 2022), where the $\lambda_{\max}$ stabilizes and oscillates around a characteristic value. The name is due to the fact that, in the case of full-batch GD, the $\lambda_{\max}$ hovers at $2/\eta$ which is the stability threshold for optimizing quadratics. Observations from Cohen et al. (2021; 2022) indicate that, under mean square error (MSE), the bulk of training dynamics occur within this regime, effectively determining $\lambda_{\max}$ of the final solution. Lee and Jang (2023) explained why in this regime $\lambda_{\max}$ often slightly exceeds $2/\eta$: this deviation arises primarily from nonlinearity of the loss gradient, which shifts the required value depending on higher-order derivatives, and the EoS being governed by the Hessian along the gradient direction, rather than $\lambda_{\max}$ alone. A growing body of research analyzes the surprising

mechanism underlying EoS dynamics observed during training with GD. Classically, when gradients depend linearly on parameters, divergence occurs locally if $\eta > \frac{2}{\lambda_{\max}}$, as illustrated by one-dimensional quadratic models (Cohen et al., 2021). In contrast, neural networks often converge despite violating this classical stability condition, presumably due to the problem's non-standard geometry. Damian et al. (2023) propose an explanation under some, empirically tested, assumptions of alignment of third derivatives and gradients.

The work on EoS is about full-batch methods. While the empirical behavior of EoS for full-batch algorithms is relatively well-understood, neural networks are predominantly trained using mini-batch methods. As explicitly noted by Cohen et al. (2021, Section 6, Appendices G and H), their observations and analysis do not directly apply to mini-batch training. In particular, they emphasize:

> *[...] while the sharpness does not flatline at any value during SGD (as it does during gradient descent), the trajectory of the sharpness is heavily influenced by the step size and batch size Jastrzębski et al. (2019; 2020), which cannot be explained by existing optimization theory. Indeed, there are indications that the "Edge of Stability" intuition might generalize somehow to SGD, just in a way that does not center around the (full-batch) sharpness. [...] In extending these findings to SGD, the question arises of how to model "stability" of SGD.*

We show that the EoS phenomenon does indeed generalize to SGD, and we identify the key quantity governing this generalization (*Batch Sharpness* in Definition 3). We model stability of SGD on the neural networks landscapes: our answer is that *SGD is stable if on average the step is stable on the mini-batch landscape—not on the full-batch landscape.*

What was empirically known for SGD. In the context of mini-batch algorithms, **(i)** Jastrzębski et al. (2019; 2020) noticed that for SGD the phase transition happens earlier for smaller η or smaller batch size b, but they did not quantify when. **(ii)** Cohen et al. (2021); Gilmer et al. (2021) established that initialization and architecture choices affect stability of SGD, without providing a definitive condition. **(iii)** When $\lambda_{\max}$ stabilizes, that always happens at a level they could not quantify which is below the $2/\eta$ threshold (Cohen et al., 2021; Keskar et al., 2016), see Figure 2, often without a proper progressive sharpening phase. This leaves the most basic questions open: *In what way the location of convergence of SGD acclimates to the choice of hyperparameters? What are the key quantities involved?* To be more specific, can we characterize the training phenomena in **(i), (ii), (iii)** above? What determines them? Does SGD train in an unstable regime?

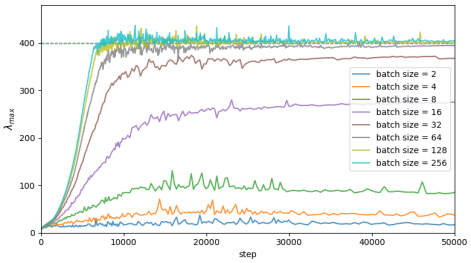


Figure 2: **SGD on CIFAR-10:** $\eta = 1/400$. The full-batch Hessian's $\lambda_{\max}$ plateaus *below* $2/\eta$. Smaller batch sizes lead to lower plateau values.

Previous Works on SGD Stability. A series of works, Wu et al. (2018); Ma and Ying (2021); Granziol et al. (2021); Wu et al. (2022); Mulayoff and Michaeli (2024), mathematically study stability for constant-step-size SGD on quadratic losses. However, **(1)** *they do not establish empirically whether—and in what sense—SGD* ***trains*** *in an EoS-like regime or not for neural networks.* Specifically, they compare to our Section 5 but not to the rest of the article. And **(2)** there exist multiple different notions of stability for stochastic algorithms, they discuss *some* of them, not the one we use here to describe EoSS[1]. It is also important to remark that empirically, multiple works—e.g. Xing et al. (2018), Cohen et al. (2021, Appendix H), Ahn et al. (2022), Lee and Jang (2023)—establish the presence of oscillations in the SGD trajectory for neural networks. However, did not distinguish between

[1]Moreover, some of them can not be tested in high-dimensional settings as neural networks. Further discussion in Appendix B.1.

noise and curvature-driven oscillations—the ones directly relevant to the question of how the curvature adapts to the hyper parameters and the possible generalization of previous EoS implications to SGD[2]. Agarwala and Pennington (2024) showed that for SGD, edge of stability may be sensitive to the trace of the NTK rather than the $\lambda_{\max}$.

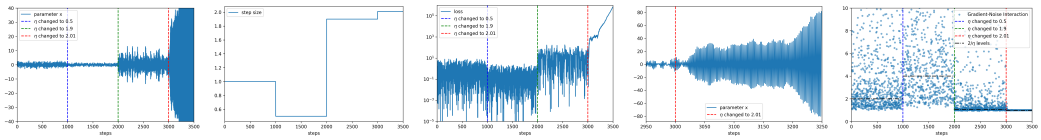
Flatness and Generalization. SGD-trained networks consistently generalize better than GD-trained ones, with smaller batch sizes further enhancing generalization performance (Keskar et al., 2016; LeCun et al., 2012; Jastrzębski et al., 2018; Goyal et al., 2017; Masters and Luschi, 2018; Smith et al., 2021; Beneventano et al., 2024). This advantage has been widely attributed to some notion of flatness of the minima (Jiang et al., 2019; Jastrzębski et al., 2021; Hochreiter and Schmidhuber, 1994; Neyshabur et al., 2017; Wu et al., 2017; Kleinberg et al., 2018; Xie et al., 2020). Training algorithms explicitly designed to find flat minima have indeed demonstrated strong performance across various tasks (Izmailov et al., 2019; Foret et al., 2021). Our result is inherently a result about mini-batch training improving flatness. Specifically, we explain why: *Training with smaller batches constraints the dynamics to areas with smaller eigenvalues of the full-batch Hessian.* This quantifies and characterizes prior observations that SGD tends to locate flat minima and that smaller batch sizes result in reduced Hessian sharpness (Keskar et al., 2016; Jastrzębski et al., 2021).

3 Preliminaries: Noise-Driven *vs* Curvature-Driven

The key defining aspect of EoS is about the solutions found by the algorithm adapting to the optimizer's hyperparameters. In the case of full-batch algorithms, this manifests through the emergence of an oscillatory regime. Mini-batch SGD, however, always oscillates because its gradient is noisy and the step size does not vanish. The central question, therefore, is *which* oscillations signal curvature-limited dynamics (EoS-like). We define stable and unstable oscillations based on the induction of catapults.

Definition 1 (Quadratic instability and Catapults)**.** Consider the quadratic approximation of all the data point loss landscapes $\frac{1}{2}(\theta - x_i)^\top \mathcal{H}_i(\theta - x_i)$. We say that a set of hyperparameters is unstable if the trajectory exits all the compacts in which the quadratic approximation holds up to $\mathcal{O}(\eta)$[3]. We say the algorithm experienced a *catapult* when this event happened.

We define *Type-1* (Noise-Driven Oscillation) those that are stable under the definition above, e.g., when we increase the step size and the trajectory re-stabilizes within the neighborhood. We call *Type-2* (Curvature-induced) the oscillations which saturate stability, i.e., the ones for which a small change in the hyperparameters induces a catapult as defined in Definition 1. Interestingly, both types of oscillation involve quantities stabilizing near the critical threshold of $2/\eta$, yet they differ.



(a) Trajectory (b) Step size sched. (c) Loss (d) Zoomed-in (e) GNI of Eq. (5)

Figure 3: **Quadratics:** Dynamics of SGD on a 1-D quadratic with N datapoints, $L(x) = \frac{1}{2N}\sum_i (x - a_i)^2$, where $a_i \sim \mathcal{N}(0, 1)$. Oscillations are present for any step size. Yet, only when the step size becomes larger than $2/\lambda_{\max} = 2$ (after the red line), the oscillations become unstable (d) and the loss diverges (c). Meanwhile, *GNI* consistently stays at $2/\eta$.

[2] As we discuss in Section 3, 5, and Appendix B and E.

[3] This means that either SGD seen as a linear dynamical system is diverging **or** that the re-stabilization would happen be at a level which exits the largest region in which the quadratic approximation holds and so the dyanmics *changes region*.

3.1 *Type-1*: Noise-Driven Oscillation

SGD can wobble around a stationary point simply because gradients vary across batches and the step size is not annealed. This occurs even if the Hessian is small as, with fixed step-size, mini-batch noise has lower-bounded variance. Such noise-driven behavior is well-studied in classical stochastic approximation (Robbins and Monro, 1951; Mandt et al., 2016; Bottou et al., 2018; Mishchenko et al., 2020). We call *Type-1* oscillations any stochastic or chaotic trajectory which does not leave the region defined in Definition 1. *Type-1* oscillations occur even for simple quadratics—see Figure 3, Proposition 1, Appendix C.

Definition 2. Denote $\mathcal{H}$ the Hessian of $L(\theta)$. We define Gradient-Noise Interaction (*GNI*):

$$\textit{Gradient-Noise Interaction}(\theta) \quad := \quad \frac{\mathbb{E}_{B\sim\mathcal{P}_b}\left[\nabla L_B(\theta)^\top \mathcal{H}\, \nabla L_B(\theta)\right]}{\|\nabla L(\theta)\|^2}\,^{4}. \tag{1}$$

GNI is defined by dividing the two terms in the Descent Lemmas. The trajectory oscillates, no matter the reason, if and only if $GNI \approx 2/\eta$, indeed:

Lemma 1. $\mathbb{E}_{B\sim\mathcal{P}_b}\left[L(\theta_{t+1})\right] \approx L(\theta_t)$ *if and only if*

$$-\eta\|\nabla L(\theta)\|^2 + \frac{\eta^2}{2}\mathbb{E}_{B\sim\mathcal{P}_b}\left[\nabla L_B(\theta)^\top \mathcal{H} \nabla L_B(\theta)\right] \;\approx\; 0 \quad \Longleftrightarrow \quad GNI \;\approx\; \frac{2}{\eta}. \tag{2}$$

This regime has been previously documented by measuring the expected total loss decrease by e.g., Cohen et al. (2021, Appendix H), Ahn et al. (2022), and Lee and Jang (2023) that tracked GNI explicitly[5]. Notably, GNI is a quantity that is *centered* around $2/\eta$ *whenever* the trajectory oscillates. No matter the reason of the oscillation, see Figure 5. In particular GNI centers at $2/\eta$ over the stationary distribution if any, see Proposition 1.

3.2 *Type-2*: Curvature-Driven Oscillation

Once the *local*, or *perceived*, curvature saturates with respect to the hyperparameters, the updates become unstable in a manner analogous to the classic EoS Cohen et al. (2021). We define *Type-2* oscillation the trajectory in the setting of Definition 1 of the optimizer with hyperparameters at the boundary between stable and unstable[6]. *Batch Sharpness* (Definition 3), instead of GNI (Definition 2), is the quantity that governs this saturation, as we will see in Section 4 and 5.

Definition 3 (Batch Sharpness)**.** We define *Batch Sharpness* as the ratio

$$\textit{Batch Sharpness}(\theta) \quad := \quad \mathbb{E}_{B\sim\mathcal{P}_b}\left[\frac{\nabla L_B(\theta)^\top\, \mathcal{H}(L_{\mathbf{B}})\, \nabla L_B(\theta)}{\|\nabla L_{\mathbf{B}}(\theta)\|^2}\right]^{7}. \tag{3}$$

Batch Sharpness is thus the expectation of the Rayleigh quotient between the mini-batch gradient ∇L_B and the mini-batch Hessian $\mathcal{H}(L_B)$. It therefore measures the *expected directional curvature of the mini-batch loss surface along the step induced by it.*

Relation to earlier notions: (i) Lee and Jang (2023) noted that the ratio $\nabla L^\top \mathcal{H} \nabla L / \|\nabla L\|^2$ settles at $2/\eta$ during EoS and coincides with $\lambda_{\max}(\mathcal{H})$ only when the gradient aligns with the top eigenvector—this eventually happens in the quadratic case, but not in the general case. Batch Sharpness extends that directional viewpoint to the stochastic setting by replacing $(\nabla L, \mathcal{H})$ with their mini-batch counterparts $\left(\nabla L_B, \mathcal{H}(L_B)\right)$ *before* taking expectations[8]. **(ii)** As we show in Section 5, *Batch Sharpness*, generalizes $\lambda_{\max}$ as "if greater than $2/\eta$, the norm squared of the mini-batch gradients explode", see Theorem 1.

[4] Note that both the Hessian $\mathcal{H}$ and the gradient at the denominator are on the full-batch loss.

[5] In their notations $\mathrm{tr}(HS_b)/\,\mathrm{tr}(S_n)$. See Appendix B for further comparison with previous work.

[6] These exist whenever we search hyperparameters in a compact set.

[7] We used bold for the "**B**" to highlight the difference with *GNI* in Definition 2.

[8] Importantly, see Appendix Q, the right notion of curvature/stability for mini-batch algorithms has to depend on different statistics or moments of the mini-batch Hessians, not simply on the average of the Hessians as in full-batch.

Importantly, a reason why EoSS was difficult to observe, is that the measure of curvature does not generalize $\lambda_{\max}$, **(iii)** Also, up to higher orders in the step size, single mini batch step is stable on the mini-batch landscape (seen as full-batch step on a restricted dataset B) if the ratio within the expectation of *Batch Sharpness* is smaller than $2/\eta$:

$$\frac{\nabla L_B^\top \mathcal{H}(L_B) \nabla L_B}{\|\nabla L_B\|^2} \leq \frac{2}{\eta} \iff -\eta \|\nabla L_B\|^2 + \frac{\eta^2}{2} \nabla L_B^\top \mathcal{H}(L_B) \nabla L_B \leq 0. \tag{4}$$

4 SGD Typically Occurs at the EoSS

We characterize here the phenomenon of the Edge of Stochastic Stability. We verify the emergence of EoSS across of a range of step sizes, batch sizes and architectures (Figure 6 and Appendix R); datasets (CIFAR-10 and SVHN, Appendix S); and dataset sizes (8k and 32k subsets, Figure 7).

1. Stabilization of *Batch Sharpness*. SGD typically traind in an EoS-like regime:

> *SGD tends to train in a regime we call Edge of Stochastic Stability. Precisely, after a phase of progressive sharpening,* Batch Sharpness *reaches a stability level of* $2/\eta$*, and hovers there.*

In particular, the level of plateau of *Batch Sharpness* is $2/\eta$ independent of the batch size (Figure 1). Importantly, *Type-1* oscillations happen throughout most of the training as highlighted by the quantity of Proposition 1, see Figure 5, but they do not impact progressive sharpening which leads to the second phase of EoSS stabilization and *Type-2* oscillations. Importantly, analogously to EoS, training continues and the loss continues to decrease while *Batch Sharpness* is constrained by the step size magnitude.

2. Stabilization of $\lambda_{\max}$ and *GNI*. Crucially, stabilization of *Batch Sharpness* around $2/\eta$ happens while *GNI* has stabilized at $2/\eta$ already, and induces a corresponding stabilization of $\lambda_{\max}$. However, $\lambda_{\max}$ consistently settles at a lower level, due to a batch-size–dependent gap between the two. This is also influenced by the specific optimization trajectory, Figures 6 and 7. See Section J for factors determining their gap.

3. Catapults. Unlike in EoS, in the EoSS regime what is stabilized is the *expectation* of a quantity which the algorithm sees one observation at time. Occasionally, a sequence of sampled batches exhibits anomalously high sharpness–that is too high for the stable regime–and steps overshoot, triggering a catapult effect, where *Batch Sharpness* spikes before rapidly decreasing (Figure 4). This is typically followed by renewed progressive sharpening, eventually returning to the EoSS regime. This results in a catapult phase for the training loss, aligning with, and maybe explaining, previous observations about catapult behaviors, e.g., (Lewkowycz et al., 2020; Zhu et al., 2024).

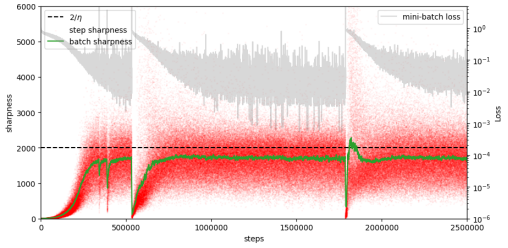
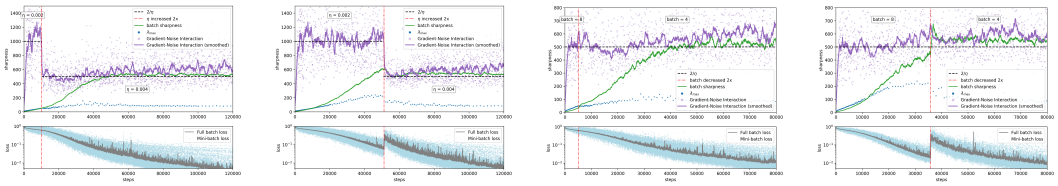


Figure 4: **Catapults at EoSS.** During EoSS, *Batch Sharpness* goes through cycles of progressive sharpening and stabilizations. Notations follow Figure 6.

4.1 *Batch Sharpness* Governs EoSS

Following Cohen et al. (2021) and the discussion in Section 3, we track how the training dynamics change when perturbing the hyperparameters mid-training. Overall, we find that *Batch Sharpness* governs EoSS behavior—mirroring how $\lambda_{\max}$ operates in the full-batch EoS—while the full-batch $\lambda_{\max}$ lags behind or settles inconsistently, underlining the mini-batch nature of SGD stability, see Appendix J. Increasing the step size η or decreasing the batch size b triggers a *catapult* spike in all the quantities in considerations and the training loss, before *Batch Sharpness* re-stabilizes near the updated threshold $2/\eta$, see Figures 7a



(a) LR increased early (b) LR increased late (c) Batch decr. early (d) Batch decr. late

Figure 5: **(1)** The whole training happens with *Type-1* oscillations (see Proposition 1, $GNI \approx 2/\eta$), however, **(2)** *GNI* being $2/\eta$ does not govern *Type-2* oscillations—in particular, highlighting the difference in the two types of oscillations. **(3)** *Batch Sharpness* is instead an indicator of *Type-2* oscillations, as illustrated by the fact that catapults happen only when the shift in hyperparametes occurs **after** *Batch Sharpness* reaches $2/\eta$.

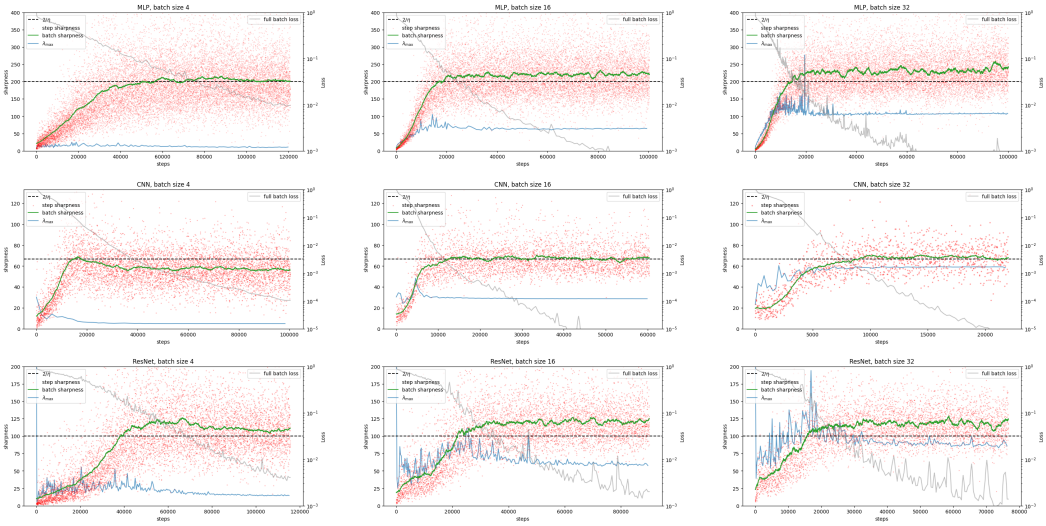
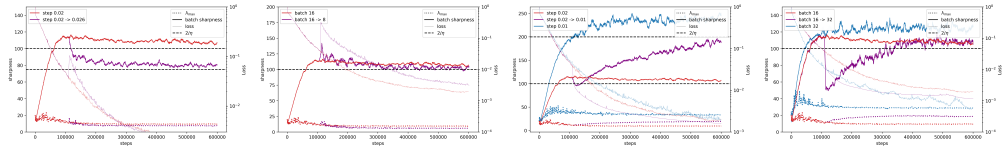


Figure 6: **Comparing different sharpness measures.** Red: observed sharpness on the current step's mini batch—essentially *Batch Sharpness* without the expectation; Green: *Batch Sharpness* (Definition 3); Blue: full-batch $\lambda_{\max}$. Top row: MLP (2 hidden layers of width 512); middle: 5-layer CNN; bottom: ResNet-14; all trained on an 8k subset of CIFAR-10.



(a) Increasing step (b) Decreasing batch (c) Decreasing step (d) Increasing batch

Figure 7: **Effects of changing step size or batch size in EoSS.** *Catapults:* (a) Increasing the step size η causes a catapult spike before Batch Sharpness re-settles at the new $2/\eta$. (b) Decreasing the batch size b increases *Batch Sharpness* and causes a catapult. *Restarting PS:* (c) Decreasing η prompts renewed progressive sharpening. (d) Increasing b lowers *Batch Sharpness* and re-starts progressive sharpening. The experiments are conducted on a 32k subset of CIFAR-10 to ensure sufficient complexity remains in the dataset, which is necessary for observing renewed progressive sharpening, consistent with observations by Cohen et al. (2021).

and 7b. This therefore pushes $\lambda_{\max}$ lower. Conversely, reducing η raises the $2/\eta$ threshold. Analogously, increasing the batch size leaves $\lambda_{\max}$ unchanged but reduces *Batch Sharpness*. These changes prompt a new phase of progressive sharpening, see Figures 7c and 7d. Notice that, *instantaneously*, the change in batch size does not change the full-batch loss landscape, but only changes the mini-batch landscapes—the fact that this causes a catapult/restarts PS is an indicator that it is indeed the mini-batch landscape (and therefore *Batch Sharpness*) that governs the stability/instability of SGD. Here, $\lambda_{\max}$ also rises, but ultimately stabilizes at a lower value than if the entire training had run with the smaller step size/larger. Again, if stability was governed by $\lambda_{\max}$, this step-size adjustment would have had the same effect as starting from scratch with the new step size.

5 On Stability

The previous section empirically demonstrated that mini-batch SGD generally settles into the EoSS regime, where *Batch Sharpness* hovers around $2/\eta$. In classical (full-batch) gradient descent, the condition $\eta < 2/\lambda_{\max}$ guarantees local stability by preventing divergence along the direction of the largest eigenvalue of a fixed Hessian. Here, is *Batch Sharpness* at $2/\eta$ *saturating the stability regime*? We answered positively empirically by showing that when you perturb hyperparameters you have explosions, see Figure 7. This proves empirically that we are at the Edge of Stability according to Definition 1. We show this mathematically in this section. Analogously, *is Batch Sharpness hovering at* $2/\eta$ *the cause of EoSS or a byproduct of something else happening?*. We already established empirically that the stabilization of $\lambda_{\max}$ is a byproduct of it, see Figure 7 and Section J. We establish this causality proving Theorem 1 and Proposition 1 below. Precisely, Theorem 1 shows that the trajectory is unstable with respect to Definition 1 when *Batch Sharpness* is bigger than $2/\eta$ and Proposition 1 shows that even in the case of stable trajectories *GNI* is eventually centered at $2/\eta$ but *Batch Sharpness is smaller than that.*

Importantly, there exist many stochastic notions of stability, depending on different moments of the *random variable* $\mathcal{H}(L_B)$. Some of which, depend on quantities that can not be computed in high-dimensional experiments. See Appendix B.1 for a discussion. One of our findings is essentially that the good one is the one of Definition 1.

5.1 *Type-2* is about *Batch Sharpness*

The next theorem implies that SGD is unstable on quadratics also when $\lambda_{\max} < 2/\eta$ if *Batch Sharpness* is bigger than $2/\eta$.

Theorem 1. *Let* $\eta \leq 2/\lambda_{\max}$. *There exists a absolute constant* $c > 0$ *such that if Batch Sharpness is strictly bigger than* $2/\eta + c\eta$ *then* $\mathbb{E}[\|\nabla L_B\|^2]$ *locally increases exponentially with the SGD step and the trajectory is unstable in the sense of Definition 1.*

The proof relies on Jensen and Cauchy-Schwarz inequalities, see Appendix G and Proposition 5. The use of these inequalities is its main limitation—we can not show the if and only if. However, Theorem 1 is the first (in)stability results that relies on a quantity we can efficiently estimate or compute in high-dimensional settings as neural networks. Note indeed that stability for quadratics is classically established by checking when $\mathbb{E}[\|\theta\|^2]$ diverges and when does not, see (Ma and Ying, 2021; Mulayoff and Michaeli, 2024) and Appendix B.1. *Batch Sharpness* is not directly related to these proofs and to the size of $\mathbb{E}[\|\theta\|]$.

5.2 *Type-1* is about *GNI*

In Proposition 1 (proof in Appendix C) we show that when SGD is performed with a stable fixed step size converges to a stationary distribution π—as known since Robbins' analysis (Robbins and Monro, 1951; Mandt et al., 2016; Bottou et al., 2018; Mishchenko et al., 2020). For $\theta \sim \pi$, the distribution of $GNI(\theta)$ is centered in $2/\eta$, however with big variance.

Proposition 1. *Around a local minimum* θ^*, *fix* $\eta > 0$ *such that* $\mathcal{H}(L_B)$ *satisfy that* $\|(I - \eta\mathcal{H})^2 + \frac{\eta^2}{b}\mathbb{E}_B[\mathcal{H}(L_B)^2 - \mathcal{H}^2]\|_2^2 < 1$. *Then the trajectory of SGD settles in a stationary distribution* $\theta \sim \pi$ *characterized by Type-1 oscillations but not Type-2 and satisfying*

$$\frac{\mathbb{E}_{\theta\sim\pi}\left[\mathbb{E}_{B\sim\mathcal{P}_b}\left[\nabla L_B(\theta)^\top \mathcal{H}\, \nabla L_B(\theta)\right]\right]}{\mathbb{E}_{\theta\sim\pi}\left[\|\nabla L(\theta)\|^2\right]} \quad = \quad \frac{2}{\eta}\Big[1 + \mathcal{O}(\eta)\Big], \tag{5}$$

Independently of the moments of the Hessians $\mathcal{H}$ *and* $\mathcal{H}(L_B)$.

Crucially, the appearance of some quantity—GNI—being $2/\eta$ implies the system is oscillating, not *why*. It does not mean, in principle, that the landscape or the curvature **adapted** to the hyper parameters. In the case of *Type-1*, $2/\eta$ is about the ratio between the covariance of the gradients and the size of the full-batch gradient. Importantly, in this setting by perturbing the hyper parameters the system does not show catapults (as defined in Definition 1). When the size of oscillations increases (bigger step or smaller batch) the dynamics just increases the size of the oscillations—quickly restabilizing.

6 Implications: How Noise-Injected GD Differs from SGD

SGD vs. Noisy Gradient Descent. A common belief is that SGD's regularization stems from its "noisy" gradients, which find flatter minima. Our analysis highlights how the noise in the Hessians as crucial. To test this, we compare mini-batch SGD (batch size 16) against three noisy GD variants[9]: **(i)** *Gaussian reweighting on the samples (Wu et al., 2020)* which maintains the noise structure in the Hessians; **(ii)** *Isotropic/Anisotropic diagonal noise* (Zhu et al., 2019); and SDE dynamics (Li et al., 2017). As shown in Figure 8 and Appendix H, only noise which maintains the higher moments of the Hessian(s) (and thus implicitly preserves the mini-batch landscape structure) leads to an EoSS-like regime with $\lambda_{\max}$ stabilizing well below $2/\eta$. Classical analyses of neural network optimization often assume noisy trajectories on a single, static, landscape. This is a further proof that the community has to be careful when modeling SGD as noise-injected GD or SDEs.

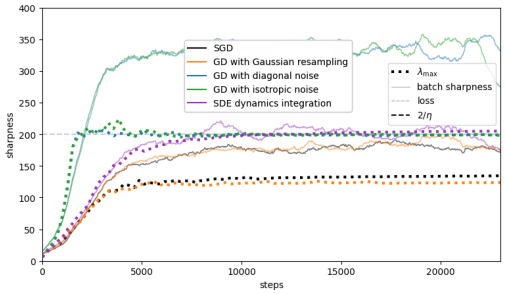


Figure 8: **SGD vs. Noisy GD vs SDE.** Only noise preserving the mini-batch structure of SGD leads to $\lambda_{\max}$ plateauing below $2/\eta$ (akin to EoSS and as observed by (Keskar et al., 2016)). Noise injection fails to reproduce this behavior even with the same covariance SGD's.

7 Conclusions, Limitations, and Future Work

Conclusions. We have addressed the longstanding question of *if* and *how* mini-batch SGD enters a regime reminiscent of the "Edge of Stability" previously observed in full-batch methods. Contrary to the usual focus on the global Hessian's top eigenvalue, we uncovered that *Batch Sharpness*—the expected directional curvature of the mini-batch landscape in the direction of its own gradient—consistently rises (progressive sharpening) and then hovers around $2/\eta$, independent of batch size. This behavior characterizes a new regime "Edge of Stochastic Stability", which explains how mini-batch training can exhibit catapult-like surges and settle into flatter minima even when the full-batch Hessian remains below $2/\eta$. Our analysis clarifies why smaller batch sizes and larger step sizes both constrain the final curvature to a lower level, thereby linking these hyperparameters to flatter solutions and often improved generalization. Furthermore, we show that this phenomenon depends on the noise injected into the Hessians by mini-batch optimizers, highlighting important limitations of SDE-based approximations. Overall, the EoSS framework unifies several empirically observed effects—catapult phases, dependence on batch size, and progressive sharpening—under a single perspective focused on the *mini-batch* landscape and its directional curvature.

Limitations. (*i*) We have tested only image-classification tasks, leaving open whether similar phenomena arise in NLP, RL, or other domains. (*ii*) Our experiments mainly use fixed step sizes and standard architectures, so very large-scale or large-batch settings remain less explored. (*iii*) We have not analyzed momentum-based or adaptive methods (e.g. Adam), even though full-batch EoS has been seen there (Cohen et al., 2022).

Future Work. Beyond addressing these limitations, several directions remain: Understanding (*i*) *where* $\lambda_{\max}$ stabilizes; (*ii*) how EoSS and EoS affect performances and the features learned by the neural network, e.g. (Lyu et al., 2023; Arora et al., 2022; Ahn et al., 2023; Zhu et al., 2023; Wang et al., 2022; Beneventano and Woodworth, 2025); (*iii*) consequently if it is benign effect or not; (*iv*) what the *other* sources of instability are there in the (pre-)training; (*v*) better describing the phenomenon of progressive sharpening; and (*vi*) understanding its causes.

[9] See details in Appendix M.

References

Jeremy M. Cohen, Simran Kaur, Yuanzhi Li, J. Zico Kolter, and Ameet Talwalkar. Gradient Descent on Neural Networks Typically Occurs at the Edge of Stability. *arXiv:2103.00065 [cs, stat]*, June 2021. URL http://arxiv.org/abs/2103.00065. arXiv: 2103.00065.

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

Jeremy M. Cohen, Alex Damian, Ameet Talwalkar, Zico Kolter, and Jason D. Lee. Understanding Optimization in Deep Learning with Central Flows, October 2024. URL http://arxiv.org/abs/2410.24206. arXiv:2410.24206.

Jeremy M. Cohen, Behrooz Ghorbani, Shankar Krishnan, Naman Agarwal, Sourabh Medapati, Michal Badura, Daniel Suo, David Cardoze, Zachary Nado, George E. Dahl, and Justin Gilmer. Adaptive Gradient Methods at the Edge of Stability, July 2022. URL http://arxiv.org/abs/2207.14484. arXiv:2207.14484 [cs].

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, December 2019. URL http://arxiv.org/abs/1807.05031. arXiv:1807.05031 [stat].

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. *arXiv:2002.09572 [cs, stat]*, February 2020. URL http://arxiv.org/abs/2002.09572. arXiv: 2002.09572.

Chen Xing, Devansh Arpit, Christos Tsirigotis, and Yoshua Bengio. A Walk with SGD, May 2018. URL http://arxiv.org/abs/1802.08770. arXiv:1802.08770 [cs, stat].

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.

Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical Learning Periods in Deep Neural Networks, 2017. URL https://arxiv.org/abs/1711.08856v3.

Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three Factors Influencing Minima in SGD. *arXiv:1711.04623 [cs, stat]*, September 2018. URL http://arxiv.org/abs/1711.04623. arXiv: 1711.04623.

Stanislav Fort and Surya Ganguli. Emergent properties of the local geometry of neural loss landscapes, 2019. URL https://arxiv.org/abs/1910.05929v1.

Levent Sagun, Leon Bottou, and Yann LeCun. Eigenvalues of the Hessian in Deep Learning: Singularity and Beyond, November 2016. URL https://openreview.net/forum?id=B186cP9gx.

Stanislav Fort and Adam Scherlis. The Goldilocks Zone: Towards Better Understanding of Neural Network Loss Landscapes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3574–3581, July 2019. ISSN 2374-3468. doi: 10.1609/aaai.v33i01.33013574. URL https://ojs.aaai.org/index.php/AAAI/article/view/4237. Number: 01.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. In *Advances in Neural Information Processing Systems*, pages 11669–11680, 2019.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic Generalization Measures and Where to Find Them. *arXiv:1912.02178 [cs, stat]*, December 2019. URL http://arxiv.org/abs/1912.02178. arXiv: 1912.02178.

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:2003.02218 [cs, stat]*, March 2020. URL http://arxiv.org/abs/2003.02218. arXiv: 2003.02218.

Sungyoon Lee and Cheongjae Jang. A new characterization of the edge of stability based on a sharpness measure aware of batch gradient distribution. In *International Conference on Learning Representations*, 2023. URL https://api.semanticscholar.org/CorpusID:259298833.

Alex Damian, Eshaan Nichani, and Jason D. Lee. Self-Stabilization: The Implicit Bias of Gradient Descent at the Edge of Stability, April 2023. URL http://arxiv.org/abs/2209.15594. arXiv:2209.15594 [cs, math, stat].

Justin Gilmer, Behrooz Ghorbani, Ankush Garg, Sneha Kudugunta, Behnam Neyshabur, David Cardoze, George Dahl, Zachary Nado, and Orhan Firat. A loss curvature perspective on training instability in deep learning, 2021. URL https://arxiv.org/abs/2110.04369.

Lei Wu, Chao Ma, and Weinan E. How SGD Selects the Global Minima in Over-parameterized Learning: A Dynamical Stability Perspective. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/6651526b6fb8f29a00507de6a49ce30f-Abstract.html.

Chao Ma and Lexing Ying. On linear stability of SGD and input-smoothness of neural networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=yAvCV6NwWQ.

Diego Granziol, Stefan Zohren, and Stephen Roberts. Learning rates as a function of batch size: A random matrix theory approach to neural network training, 2021. URL https://arxiv.org/abs/2006.09092.

Lei Wu, Mingze Wang, and Weijie Su. The alignment property of sgd noise and how it helps select flat minima: A stability analysis, 2022.

Rotem Mulayoff and Tomer Michaeli. Exact mean square linear stability analysis for sgd, 2024. URL https://arxiv.org/abs/2306.07850.

Kwangjun Ahn, Jingzhao Zhang, and Suvrit Sra. Understanding the unstable convergence of gradient descent. In *Proceedings of the 39th International Conference on Machine Learning*, June 2022. URL https://proceedings.mlr.press/v162/ahn22a.html.

Atish Agarwala and Jeffrey Pennington. High dimensional analysis reveals conservative sharpening and a stochastic edge of stability. *arXiv preprint arXiv:2404.19261*, 2024.

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.

Dominic Masters and Carlo Luschi. Revisiting Small Batch Training for Deep Neural Networks, April 2018. URL http://arxiv.org/abs/1804.07612. arXiv:1804.07612.

Samuel L. Smith, Benoit Dherin, David G. T. Barrett, and Soham De. On the Origin of Implicit Regularization in Stochastic Gradient Descent. *arXiv:2101.12176 [cs, stat]*, January 2021. URL http://arxiv.org/abs/2101.12176. arXiv: 2101.12176.

Pierfrancesco Beneventano, Andrea Pinto, and Tomaso Poggio. How Neural Networks Learn the Support is an Implicit Regularization Effect of SGD. *arXiv:2406.11110 [cs, math, stat]*, June 2024. doi: 10.48550/arXiv.2406.11110. URL http://arxiv.org/abs/2406.11110. arXiv:2406.11110 [cs, math, stat].

Stanisław Jastrzębski, Devansh Arpit, Oliver Astrand, Giancarlo Kerg, Huan Wang, Caiming Xiong, Richard Socher, Kyunghyun Cho, and Krzysztof Geras. Catastrophic Fisher Explosion: Early Phase Fisher Matrix Impacts Generalization. *arXiv:2012.14193 [cs, stat]*, June 2021. URL http://arxiv.org/abs/2012.14193. arXiv: 2012.14193.

Sepp Hochreiter and Jürgen Schmidhuber. SIMPLIFYING NEURAL NETS BY DISCOVERING FLAT MINIMA. In *Advances in Neural Information Processing Systems*, volume 7. MIT Press, 1994. URL https://proceedings.neurips.cc/paper/1994/hash/01882513d5fa7c329e940dda99b12147-Abstract.html.

Behnam Neyshabur, Srinadh Bhojanapalli, David Mcallester, and Nati Srebro. Exploring Generalization in Deep Learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5947–5956. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7176-exploring-generalization-in-deep-learning.pdf.

Lei Wu, Zhanxing Zhu, and Weinan E. Towards Understanding Generalization of Deep Learning: Perspective of Loss Landscapes. *arXiv:1706.10239 [cs, stat]*, November 2017. URL http://arxiv.org/abs/1706.10239. arXiv: 1706.10239.

Robert Kleinberg, Yuanzhi Li, and Yang Yuan. An Alternative View: When Does SGD Escape Local Minima?, August 2018. URL http://arxiv.org/abs/1802.06175. arXiv:1802.06175 [cs].

Zeke Xie, Issei Sato, and Masashi Sugiyama. A Diffusion Theory For Deep Learning Dynamics: Stochastic Gradient Descent Exponentially Favors Flat Minima. *arXiv:2002.03495 [cs, stat]*, November 2020. URL http://arxiv.org/abs/2002.03495. arXiv: 2002.03495.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging Weights Leads to Wider Optima and Better Generalization, February 2019. URL http://arxiv.org/abs/1803.05407. arXiv:1803.05407 [cs, stat].

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-Aware Minimization for Efficiently Improving Generalization, April 2021. URL http://arxiv.org/abs/2010.01412. arXiv:2010.01412 [cs, stat].

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Stephan Mandt, Matthew Hoffman, and David Blei. A variational analysis of stochastic gradient algorithms. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 354–363, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/mandt16.html.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization Methods for Large-Scale Machine Learning, February 2018. URL http://arxiv.org/abs/1606.04838. arXiv: 1606.04838.

Konstantin Mishchenko, Ahmed Khaled, and Peter Richtarik. Random Reshuffling: Simple Analysis with Vast Improvements. In *Advances in Neural Information Processing Systems*, volume 33, pages 17309–17320. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/c8cc6e90ccbff44c9cee23611711cdc4-Abstract.html.

Libin Zhu, Chaoyue Liu, Adityanarayanan Radhakrishnan, and Mikhail Belkin. Quadratic models for understanding catapult dynamics of neural networks, May 2024. URL http://arxiv.org/abs/2205.11787. arXiv:2205.11787 [cs].

Jingfeng Wu, Wenqing Hu, Haoyi Xiong, Jun Huan, Vladimir Braverman, and Zhanxing Zhu. On the noisy gradient descent that generalizes as sgd, 2020. URL https://arxiv.org/abs/1906.07405.

Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The Anisotropic Noise in Stochastic Gradient Descent: Its Behavior of Escaping from Sharp Minima and Regularization Effects, June 2019. URL http://arxiv.org/abs/1803.00195. arXiv:1803.00195 [cs, stat].

Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and adaptive stochastic gradient algorithms. *arXiv:1511.06251 [cs, stat]*, June 2017. URL http://arxiv.org/abs/1511.06251. arXiv: 1511.06251.

Kaifeng Lyu, Zhiyuan Li, and Sanjeev Arora. Understanding the Generalization Benefit of Normalization Layers: Sharpness Reduction, January 2023. URL http://arxiv.org/abs/2206.07085. arXiv:2206.07085 [cs].

Sanjeev Arora, Zhiyuan Li, and Abhishek Panigrahi. Understanding Gradient Descent on Edge of Stability in Deep Learning, October 2022. URL http://arxiv.org/abs/2205.09745. arXiv:2205.09745 [cs].

Kwangjun Ahn, Sébastien Bubeck, Sinho Chewi, Yin Tat Lee, Felipe Suarez, and Yi Zhang. Learning threshold neurons via the "edge of stability", October 2023. URL http://arxiv.org/abs/2212.07469. arXiv:2212.07469 [cs, math].

Xingyu Zhu, Zixuan Wang, Xiang Wang, Mo Zhou, and Rong Ge. UNDERSTANDING EDGE-OF-STABILITY TRAINING DYNAMICS WITH A MINIMALIST EXAMPLE, 2023.

Yuqing Wang, Minshuo Chen, Tuo Zhao, and Molei Tao. Large learning rate tames homogeneity: Convergence and balancing effect. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=3tbDrs77LJ5.

Pierfrancesco Beneventano and Blake Woodworth. Gradient Descent Converges Linearly to Flatter Minima than Gradient Flow in Shallow Linear Networks, January 2025. URL http://arxiv.org/abs/2501.09137. arXiv:2501.09137 [cs].

Pierfrancesco Beneventano. On the Trajectories of SGD Without Replacement, December 2023. URL http://arxiv.org/abs/2312.16143. arXiv:2312.16143.

Chang-Han Rhee and Peter W. Glynn. Lyapunov conditions for differentiability of markov chain expectations: The absolutely continuous case, 2017. URL https://arxiv.org/abs/1707.03870.

Sean P. Meyn and Richard L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press, Cambridge, second edition, 2009. ISBN 978-0-521-73182-9.

Léon Bottou. *Stochastic Gradient Learning in Neural Networks*. PhD thesis, Université Pierre et Marie Curie (Paris 6), 1991. URL http://leon.bottou.org/papers/bottou-91c.

Sho Yaida. Fluctuation-dissipation relations for stochastic gradient descent. *arXiv preprint arXiv:1810.00004*, 2018.

Mert Gürbüzbalaban, Asuman Ozdaglar, and Pablo Parrilo. Why Random Reshuffling Beats Stochastic Gradient Descent. *Mathematical Programming*, 186(1-2):49–84, March 2021. ISSN 0025-5610, 1436-4646. doi: 10.1007/s10107-019-01440-w. URL http://arxiv.org/abs/1510.08560. arXiv:1510.08560 [math].

Hanlin Zhang, Depen Morwani, Nikhil Vyas, Jingfeng Wu, Difan Zou, Udaya Ghai, Dean Foster, and Sham Kakade. How Does Critical Batch Size Scale in Pre-training?, November 2024. URL http://arxiv.org/abs/2410.21676. arXiv:2410.21676 [cs].

Zhiyuan Li, Sadhika Malladi, and Sanjeev Arora. On the Validity of Modeling SGD with Stochastic Differential Equations (SDEs). *arXiv:2102.12470 [cs, stat]*, June 2021. URL http://arxiv.org/abs/2102.12470. arXiv: 2102.12470.

Jeff Z. HaoChen, Colin Wei, Jason D. Lee, and Tengyu Ma. Shape Matters: Understanding the Implicit Bias of the Noise Covariance. *arXiv:2006.08680 [cs, stat]*, June 2020. URL http://arxiv.org/abs/2006.08680. arXiv: 2006.08680.

Alex Damian, Tengyu Ma, and Jason Lee. Label Noise SGD Provably Prefers Flat Global Minimizers. *arXiv:2106.06530 [cs, math, stat]*, June 2021. URL http://arxiv.org/abs/2106.06530. arXiv: 2106.06530.

Zhiyuan Li, Tianhao Wang, and Sanjeev Arora. What Happens after SGD Reaches Zero Loss? –A Mathematical Framework. *arXiv:2110.06914 [cs, stat]*, February 2022. URL http://arxiv.org/abs/2110.06914. arXiv: 2110.06914.

Daniel A. Roberts. SGD Implicitly Regularizes Generalization Error. *arXiv:2104.04874 [cs, stat]*, April 2021. URL http://arxiv.org/abs/2104.04874. arXiv: 2104.04874.

Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations, 2018. URL https://arxiv.org/abs/1811.01558.

Lei Wu and Weijie J. Su. The Implicit Regularization of Dynamical Stability in Stochastic Gradient Descent, June 2023. URL http://arxiv.org/abs/2305.17490. arXiv:2305.17490 [stat].

Contents

A Acknowledgement of LLMs usage

We acknowledge the use of DeepSeek, Claude Code, Codex, and ChatGPT for code assistance. We used ChatGPT and Claude for text editing suggestions, proof-reading, and LaTeX editing help.

B Comparison with Previous Stability Work

B.1 Stability of SGD

Stochastic stability notions are not unique. Cohen et al. (2021) introduces EoS making an argument about convergence vs divergence of the loss, studying $L(\theta^{t+1}) \lesseqgtr L(\theta^t)$. This is the correct thing to do and does not limit the applicability of their theory, when the loss is approximately quadratic with full-batch GD any quantity diverges or converges given the relationship between η and $\lambda_{\max}$.

In the case of SGD the story is different *(but still quadratic)*, different moments of the Hessian may appear. As an example, in the setting of Definition 1, with loss $\frac{1}{2}(\theta_t - x_i)^\top \mathcal{H}_i(\theta_t - x_i)$ at the step t, the evolution of the expectation of the norm squared of the gradients of the mini batch loss is

$$-2\eta \mathbb{E}_{i,j}\left[(\theta_t - x_i)^\top \mathcal{H}_i^2 \mathcal{H}_j(\theta_t - x_j)\right] + \eta^2 \mathbb{E}_{i,j}\left[(\theta_t - x_j)^\top \mathcal{H}_j \mathcal{H}_i^2 \mathcal{H}_j(\theta_t - x_j)\right], \tag{6}$$

from time t to $t+1$, and the one for the full batch loss is

$$-\eta \mathbb{E}_{i,j}\left[(\theta_t - x_i)^\top \mathcal{H}_i \mathcal{H}_j(\theta_t - x_j)\right] + \frac{\eta^2}{2} \mathbb{E}_{i,j}\left[(\theta_t - x_j)^\top \mathcal{H}_j \mathcal{H}_i \mathcal{H}_j(\theta_t - x_j)\right]. \tag{7}$$

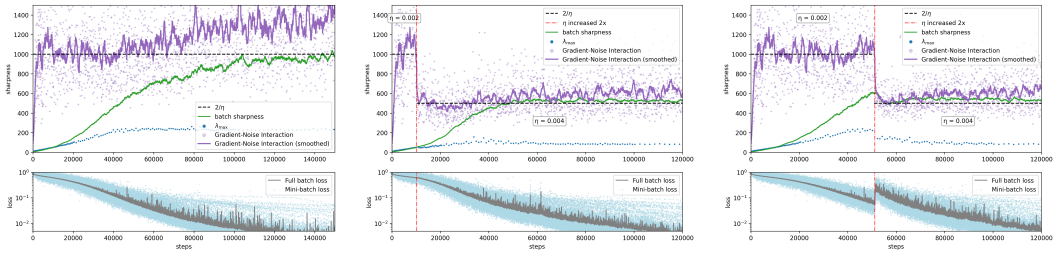
The stability of the expectation of Eq. (7) reduces to $GNI \lesseqgtr 2/\eta$, the one of Eq. (6) is about higher moments. One can in principle find examples in which they behave differently. The picture becomes even more nuanced for non-quadratics: the third order terms are different and vary in size.

Seeking the correct (or a computable) one. It is non-trivial which the *correct one is* and that it may be different for different tasks. Even more, the correct one may be not computable. For instance, previous work by Cohen et al. (2021, Appendix H), Ahn et al. (2022), and Lee and Jang (2023) study the behavior of the loss of SGD—Eq. (7), GNI—and show that in the final part of the training it oscillates without decreasing. We show that *GNI* gets to threshold $2/\eta$ extremely early on but that does not saturate the progressive sharpening, Figure 5. This implies that the loss stability is not the right quantity to look at under the type of progressive sharpening we have in deep learning. Further previous work, Wu et al. (2018); Ma and Ying (2021); Wu et al. (2022) analyzes the stability threshold for the quantity $\mathbb{E}[\|\theta\|^2]$. The threshold is η such that $\|(I - \eta\mathcal{H})^2 + \frac{\eta^2}{b}\mathbb{E}_B[\mathcal{H}(L_B)^2 - \mathcal{H}^2]\|_2^2 \lesseqgtr 1$ (Ma and Ying, 2021). This threshold is not computable in practice in high dimensional settings, indeed, it entails actually checking quantities which are d^2 dimensional. This lead Mulayoff and Michaeli (2024) to develop computable bounds on η, i.e., quantities X such that $\|(I - \eta\mathcal{H})^2 + \frac{\eta^2}{b}\mathbb{E}_B[\mathcal{H}(L_B)^2 - \mathcal{H}^2]\|_2^2 \lesseqgtr 1$ when approximately $\eta \lesseqgtr X$. *Batch Sharpness* is the first quantity proposed, to our knowledge, that is computable in high dimension (invertible in the step size in the sense above) and has been linked to the saturation of the stable region, as in to *Type-2* oscillations.

Path dependency and implications on progressive sharpening. understanding the inner working and the phenomenology of progressive sharpening is a key open ingredient to understand stability and location of convergence of optimization algorithms. In principle,

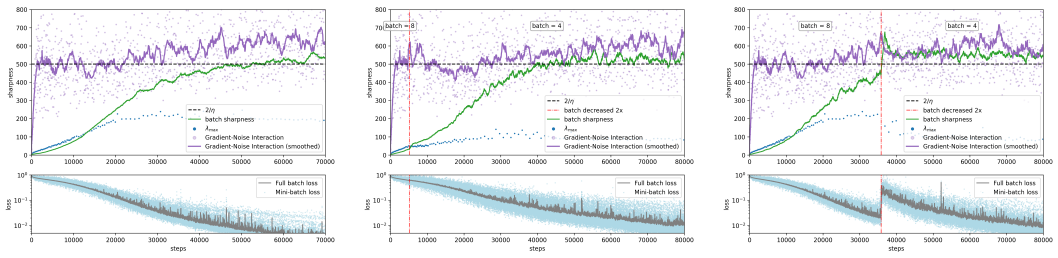
instability could emerge while moving toward infinity in any possible direction on the space. Previous work dealt with sufficient and necessary conditions for the stability of SGD *in general*, (Ma and Ying, 2021; Mulayoff and Michaeli, 2024), i.e., when does SGD become unstable given *any possible notion of progressive sharpening*. We show that the trajectory goes to infinity in *precise* directions even if we can not properly characterize it. The fact that *Batch Sharpness* is the quantity to look at implies that progressive sharpening is acting in a certain way on the datapoint Hessians. Precisely, that progressive sharpening takes the trajectory *in that place* of the boundary of the high dimensional open set of stability *such that* $GNI = BatchSharpness = 2/\eta$. Empirically, indeed, the iterates become unstable precisely because of *Batch Sharpness* reaching $2/\eta$, not because of, e.g., $\lambda_{\max}$ increasing but its variance remaining stable (which would keep *Batch Sharpness* $< GNI-constant$ until the instability threshold. This, we think, shows one more time why it is key for a deep learning optimization theory to advance to focus on path-dependent properties of the correct paths. Importantly, in Appendix F we also establish exactly when $GNI \geq$ *Batch Sharpness*. This fact depends *only* on the kurtosis of the gradients (which rarely changes much), $\mathbb{E}[\|\mathcal{H}(L_B) - \mathcal{H}\|_2^2]$, and the alignment of the steps with the full-batch Hessian. We conjecture that *Batch Sharpness* reaches GNI at $2/\eta$ by means of increasing $\mathbb{E}[\|\mathcal{H}(L_B) - \mathcal{H}\|_2^2]$ more than the alignment of gradients with the top eigenvectors the full-batch Hessian. This would imply that progressive sharpening increases the *variance* of the Hessian over the batches at least as fast as $\lambda_{\max}$. This conjecture would agree with the red cloud amplifying around the green line in Figure 6.

B.2 Previous Empirical Work



(a) Constant step size (b) Step size increased early (c) Step size increased late

Figure 9: We demonstrate that the saturation of *GNI* does not govern a sharpness-related regime of instability typical of Type-2 oscillations - and in particular, highlighting the difference in the two types of oscillations. When we double the step size after *batch sharpness* is at least half of $2/\eta$ threshold (so that it is beyond the new $2/\eta$ level), training exhibits a catapult surge in the loss (c). But if we make the same change *before* batch sharpness crosses that level—despite *GNI* already saturating—no catapult occurs. (b)



(a) Constant batch size (b) Batch size decreased early (c) Batch size decreased late

Figure 10: Similarly, reducing the batch size only triggers catapults if batch sharpness, not *GNI*, exceeds the threshold.

Lee and Jang (2023) introduce several quantities crucial for understanding neural network training dynamics. Below, we discuss the relationships among $\lambda_{\max}$, *Batch Sharpness*, and Interaction-Aware Sharpness (IAS, Lee and Jang (2023)), emphasizing that a comprehensive

theory of mini-batch dynamics should explain their distinct plateau timings and interconnected behaviors. We conjecture that a complete theory of stochastic gradient descent (SGD) dynamics would elucidate these metrics' precise interrelations and their different plateau timings.

Interaction-Aware Sharpness Lee and Jang (2023) introduce Interaction-Aware Sharpness (IAS), denoted $\|\mathcal{H}\|_{S_b}$:

$$\|\mathcal{H}\|_{S_b} \quad := \quad \frac{\mathbb{E}_{B\sim\mathcal{P}_b}\left[\nabla L_B(\mathbf{x})^\top \mathcal{H}\, \nabla L_B(\mathbf{x})\right]}{\mathbb{E}_{B\sim\mathcal{P}_b}\left[\|\nabla L_B\|^2\right]}.$$

This quantity shares structural similarities with both *Batch Sharpness* (Definition 3) and the *Gradient-Noise Interaction* (Proposition 1), differing from the latter only in the denominator. The key distinction from *Batch Sharpness* lies in which Hessian is evaluated: IAS measures the directional curvature of the **full-batch** loss landscape L along mini-batch gradient directions, while *Batch Sharpness* measures the directional curvature of **mini-batch** loss landscape L_B along their corresponding gradients. This distinction is crucial, as mini-batch Hessians vary with batch selection while the full-batch Hessian remains fixed.

Notably, with full-batch GD, IAS serves as a directional alternative to the maximal Hessian eigenvalue, $\lambda_{\max}$, introduced by Cohen et al. (2021). IAS aligns closely with the $2/\eta$ threshold, unlike $\lambda_{\max}$, which often remains slightly above this threshold during EoS, especially at the beginning of it. Since IAS measures *directional* curvature, we have $\|\mathcal{H}\|_{S_n} \leq \lambda_{\max}$. Consequently, in the mini-batch setting, IAS stabilizes below $2/\eta$, consistent with empirical observations from Jastrzębski et al. (2019; 2020); Cohen et al. (2021) and our Figure 2. Notably, when $B = n$, our *Batch Sharpness* coincides with IAS rather than λmax, reinforcing the interpretation of *Batch Sharpness* as the relevant metric stabilizing at $2/\eta$ even under full-batch conditions.

Relation to Gradient-Noise Interaction Another metric from Lee and Jang (2023) is defined as:

$$\frac{\mathrm{tr}(HS_b)}{\mathrm{tr}(S_n)} = \frac{\mathbb{E}_{B\sim\mathcal{P}_b}\left[\nabla L_B(\mathbf{x})^\top \mathcal{H}\, \nabla L_B(\mathbf{x})\right]}{\|\nabla L\|^2}$$

which coincides exactly with our definition of GNI (Proposition 1). As detailed in Section 3 and Appendix C, the stabilization of GNI around $2/\eta$ signals the presence of oscillations, at least Type-1 oscillations. Lee and Jang (2023) provide extensive empirical evidence demonstrating that neural networks spend much of their training within this oscillatory regime (see also Figures 9a and 10a). This contrasts traditional theoretical analyses (Bottou et al. (2018); Mandt et al. (2016)), which consider oscillations only near the manifold of minima.

Distinguishing oscillation types It is crucial to note that GNI around $2/\eta$ does not inherently indicate instability. As clarified in Sections 3, 4 and Appendix C, not all oscillations are inherently unstable. Figures 5, 9b, 10b illustrate that altering hyperparameters when GNI is around $2/\eta$ typically does not trigger instability (catapult-like divergence), contrary to expectations if the system was in an EoS-like regime of instability. Instead, as shown in Figures 5, 9c, 10c, *Batch Sharpness* more reliably predicts a regime of instability. Additionally, Figure 11 highlights GNI's independence from progressive sharpening, a necessary precursor to Type-2 (curvature-driven) oscillations and EoS-like instabilities, as detailed in Appendix E.

Missing Progressive Sharpening. Extensively, both in our experiments and in the ones of Lee and Jang (2023), GNI grows to $2/\eta$ in a few initial steps (and sometimes from the very beginning if the intialization size is large) without ever being in subject to a phase of progressive sharpening unlike *Batch Sharpness* and $\lambda_{\max}$. The phase of growth of GNI is generally short and independent of the size, the behavior, and the phase in which *Batch Sharpness* and $\lambda_{\max}$ are.

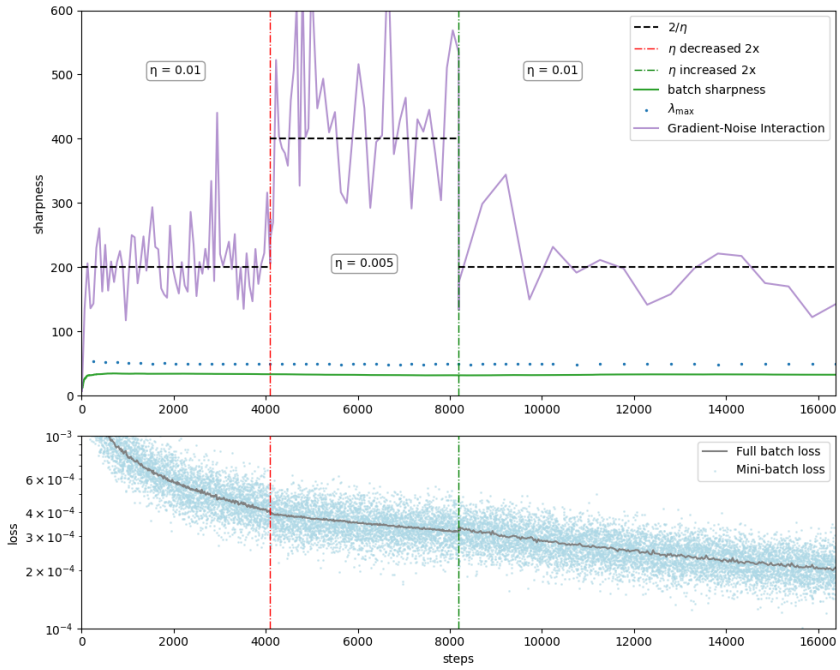


Figure 11: We construct a 32k-point "easy" CIFAR-10, where we "pull apart" all the 10 classes, so the classes become linearly separable. In this case, there is virtually no "learning" to be done, and therefore, there is barely any progressive sharpnening happening (as established Cohen et al. (2021), progressive sharpening does not happen if the dataset "is not complex enough"). Yet, *GNI* still stabilizes at the initial level of $2/\eta$. More importantly, when we decrease and then increase the step size, the *GNI* measure restabilizes to the corresponding new thresholds, while $\lambda_{\max}$ does not change. That means that *GNI* is *independent of the curvature of the loss landscape* and is unrelated to progressive sharpening, and thus Type-2 oscillations and EoS-like instability regimes.

C On the Two Types of Oscillations in SGD Dynamics

A fundamental challenge in analyzing SGD compared to GD stems from the inherent oscillations induced by mini-batch gradient noise. This appendix, together with Appendix E (also see proofs in D and G), extends the discussion in Section 3 by formally distinguishing between two distinct types of oscillations: noise-driven (Type-1) and curvature-driven (Type-2). This distinction is crucial because Type-1 oscillations occur independently of the loss landscape's curvature and thus do not exert a regularizing effect on the sharpness of the final solution. In contrast, Type-2 oscillations are directly caused by landscape curvature and induce an implicit regularization effect by discouraging convergence towards sharp minima.

We begin with a minimalistic example to illustrate the nature of Type-1

C.1 A Minimalistic Quadratic Example.

Consider a regression problem with two datapoints, 1 and -1, and a linear model $f(x) = x$ under the quadratic loss. The (scaled) full-batch loss is given by:

$$L(x) \;=\; \frac{1}{4}(x-1)^2 \;+\; \frac{1}{4}(x+1)^2.$$

Batch-1 SGD updates with step-size $0 < \eta < 2$ result in oscillatory behavior around the optimum $x = 0$ due entirely to gradient noise, with amplitude approximately $\sqrt{\frac{\eta}{2-\eta}}$. Cru-

cially, the Hessian in this example is small ($\frac{d^2L}{dx^2} = 1$), demonstrating that these persistent oscillations are entirely noise-driven (Type-1).

Formally, the SGD update is:

$$x_{t+1} = x_t - \eta \nabla \ell_{i_t} = (1-\eta)x_t + \eta \xi_t$$

where l_{i_t}s are the individual datapoint losses, and ξ_ts are i.i.d Rademacher random variables. Thus, we obtain the first two moments explicitly:

$$\mathbb{E}[x_t] = (1-\eta)\mathbb{E}[x_{t-1}] = (1-\eta)^t x_0$$

$$\mathbb{E}[x_t^2] = (1-\eta)^2 \mathbb{E}[x_{t-1}^2] + \eta^2 = (1-\eta)^{2t} x_0 + \frac{\eta^2}{1-(1-\eta)^2}\left(1-(1-\eta)^{2t}\right)$$

This implies convergence in expectation for $0 < \eta < 2$, with a limiting variance given by:

$$\lim_{t\to\infty} \mathbb{E}[x_t^2] = \frac{\eta}{2-\eta}$$

and divergence for $\eta > 2$.

A key observation is that increasing η to any value $\eta_1 < 2$ merely changes the amplitude of oscillations to $\sqrt{\frac{\eta_1}{2-\eta_1}}$ without triggering any catapult-like behavior. The only step size for which we observe Type-2 (curvature-driven) oscillations and an EoS-like[10] instability is precisely $\eta = 2$, where the dynamics effectively become a random walk, and any larger step size leads to divergence.

See Figure 3 for a plot of this phenomena for a one-dimensional example with many datapoints, with the same stability thresholds (also see Example 3.1 in Bottou et al. (2018)).

Crucially, when $\eta < 2$ oscillations occur persistently on the full-batch loss, despite the individual steps on the mini-batch loss remaining stable.

The oscillation is due to the fact that the mini-batch loss landscape shifts from step to step, not to the fact that the steps are unstable.

C.2 General Case: From One-Dimensional Toy to Multidimensional Lyapunov Analysis

The simple one-dimensional regression in §C.1 already demonstrates how *noise-driven (Type-1) oscillations* can persist indefinitely and yield a stable "two-cycle" around the optimum, independent of the actual Hessian magnitude. In higher dimensions, the story is similar: when the step size η is fixed, the *randomness* in mini-batch gradients still injects a continual "kick" at each iteration, causing the iterates to hover in a noisy neighborhood of the minimum. The main difference is that now there can be many directions—some with higher curvature than others, or even flat ($\lambda = 0$) directions. Nonetheless, the essential mechanism remains:

$$\Delta_{t+1} \;=\; \left(I - \eta\,\mathcal{H}\right)\,\Delta_t \;-\; \eta\,\xi_t,$$

where $\Delta_t = x_t - x^\star$ is the displacement from the optimum, $\mathcal{H}$ is the Hessian at $x^\star$, and ξ_t encodes the random fluctuation (gradient noise). Once Δ_t settles into a *stationary distribution*, the covariance Σ_x can be found by solving a discrete Lyapunov equation similar to the one-dimensional case.

[10] The key difference between these oscillations and genuine EoS behavior in neural networks is that, in the quadratic case, the full-batch loss does not decrease, making this scenario inherently less informative. In contrast, neural networks exhibit a surprising, albeit non-monotonic, decrease in loss within this instability regime, an effect arising from the multidimensional nature of their optimization landscape (Damian et al., 2023)

Key References and the Road Ahead. A number of works formalize this "SGD noise equilibrium" by viewing the updates as a linear Markov chain in a neighborhood of $x^\star$. Classical references include Mandt et al. (2016) for the stochastic differential analogy (Ornstein–Uhlenbeck process), and Bottou et al. (2018) for a thorough discussion of how the constant stepsize prevents exact convergence. Intuitively, the argument for Proposition 1 goes thus as follows:

1. The iterates oscillate with a stationary covariance Σ_x around x^*.
2. Full-batch (expected) gradient is zero at x^* and grows roughly linearly with distance for small deviations (by Taylor expansion $\nabla L(x) \approx \mathcal{H}(x - x^*)$). So on average *over the iterations* we have
$$\mathbb{E}_k[\|\nabla L(x_k)\|^2] = \mathrm{Tr}\big(\mathcal{H}\Sigma_x\mathcal{H}\big).$$
3. The stationary covariance of the gradients is Σ_g satisfies:
$$\Sigma_x \approx \frac{\eta}{2}(\mathcal{H}^{-1}\Sigma_g).$$

Putting all together, this implies that Σ_g of the gradients is about $\frac{2}{\eta}\mathcal{H}^{-1}$ bigger than the full-batch $\nabla L \nabla L^\top$. Precisely the following quantity (where L_i is the loss on the $i-th$ data point):
$$\frac{\mathbb{E}_i\big[\nabla L_i(\mathbf{x})^\top \mathcal{H} \nabla L_i(\mathbf{x})\big]}{\|\nabla L\|^2} = \frac{\mathrm{Tr}\big(\mathcal{H}\Sigma_g\big)}{\mathrm{Tr}\big(\mathcal{H}\Sigma_x\mathcal{H}\big)}, \tag{8}$$
and this can be rewritten as
$$\textit{Gradient-Noise Interaction (GNI)} = \frac{\mathbb{E}_i\big[\nabla L_i(\mathbf{x})^\top \mathcal{H} \nabla L_i(\mathbf{x})\big]}{\|\nabla L\|^2} = \frac{2}{\eta} \cdot \underbrace{\frac{\mathrm{Tr}\big(\mathcal{H}\Sigma_x\mathcal{H}\big)}{\mathrm{Tr}\big(\mathcal{H}\Sigma_x\mathcal{H}\big)}}_{=1} = \frac{2}{\eta}. \tag{9}$$
This $2/\eta$ thus comes out of the only fact of oscillating and it is *unrelated* to the Hessian value. Moreover, EoS happens only in the eigenspace of the highest eigenvalue, *Type-1* noise on the whole subspace spanned by the eigenspaces of the *non-negative* eigenvalues.

On Stability For this reason, linear stability analyses of stochastic gradient descent and noise-injected gradient descent on quadratic objectives–originally explored by Wu et al. (2018) and further developed by Ma and Ying (2021); Wu et al. (2022); Mulayoff and Michaeli (2024)—explicitly exclude *Type-1* oscillations by categorizing them as *stable*. Specifically, Ma and Ying (2021) restrict their analysis to interpolating minima, where all individual gradients vanish, thus effectively eliminating noise-driven oscillations and isolating the curvature-driven (*Type-2*) scenario. Mulayoff and Michaeli (2024) extends this to a more general class of minima by restricting the analysis to the orthogonal complement of the null space of the Hessian, and demonstrating that the noise-driven oscillations do not affect stability. Lee and Jang (2023) empirically established that Gradient-Noise Interaction (*GNI*) consistently remains around $2/\eta$ throughout training. From the above, this implies that most training occurs in an oscillatory regime (at least *Type-1*)—see Figure 10 and Appendix B.2. In contrast, our study specifically investigates the emergence and implications of *Type-2* oscillations, given their significant role in implicitly regularizing the loss landscape.

From Informal to Formal. In the next Appendix D, we present a general *discrete Lyapunov* proof of Lemma 1, allowing $\mathcal{H} \succeq 0$ to be possibly degenerate and not necessarily commuting with the noise covariance. The result is summarized in Proposition 2, showing rigorously that GNI $\approx 2/\eta$ arises under *any* stable constant-stepsize mini-batch SGD orbit. This "$\frac{2}{\eta}$-law" is precisely the high-dimensional extension of the toy one-dimensional phenomenon above.

*In particular, we show here that the appearance of some quantity being $2/\eta$ means that the system is oscillating but does not mean in principle that the landscape or the curvature **adapted** to the hyper parameters.*

D Proof of Proposition 1

Setup and Notation. Let

$$L(x) \;=\; \tfrac{1}{n}\sum_{i=1}^{n}\ell_i(x)$$

be twice–continuously differentiable with a (possibly non–isolated) minimiser $x^\star$. Denote by $\mathcal{H} := \nabla^2 L(x^\star) \succeq 0$ the positive–semidefinite Hessian at $x^\star$. Decompose the ambient space $\mathbb{R}^d$ into

$$\mathbb{R}^d \;=\; E_+ \oplus E_0, \quad E_+ := \mathrm{Im}(\mathcal{H}), \quad E_0 := \ker(\mathcal{H}),$$

with corresponding orthogonal projectors P_+, P_0. Let $\mathcal{H}^\dagger$ denote the Moore–Penrose pseudoinverse of $\mathcal{H}$. It will be convenient to define the *Kronecker–sum* operator

$$\mathcal{K} : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}, \quad \mathcal{K}(X) \;=\; \mathcal{H}\,X \;+\; X\,\mathcal{H}^{11}.$$

Assumptions Near $x^\star$. We work under the following assumptions in a neighborhood of $x^\star$.

(A1) Local linearity. Each ℓ_i is (locally) twice differentiable, and in a sufficiently small neighborhood of $x^\star$ we have

$$\nabla\ell_i(x) \;=\; \nabla\ell_i\big(x^\star\big) \;+\; \mathcal{H}\,\big(x - x^\star\big) \;+\; \mathcal{O}\big(\|x - x^\star\|^2\big), \quad \mathcal{H} \;=\; \nabla^2 L\big(x^\star\big) \succeq 0.$$

(A2) Finite, compatible noise. Define

$$\Sigma_g \;=\; \mathbb{E}_i\Big[\nabla\ell_i(x^\star)\,\nabla\ell_i(x^\star)^\top\Big] \quad \text{(finite matrix)}.$$

We assume that in the flat subspace $\ker(\mathcal{H})$, there is *no large* random forcing. Formally, either

$$P_0\,\Sigma_g\,P_0 \;=\; 0, \quad \text{where } P_0 \text{ is the orthogonal projector onto } \ker(\mathcal{H}),$$

or else $\|P_0\,\Sigma_g\,P_0\| \lesssim \eta$ so that the random walk does not blow up in flat directions.

(A3) Stepsize stability. Let

$$\lambda^+_{\max} \;:=\; \max\big\{\lambda > 0 : \lambda \in \sigma(\mathcal{H})\big\}.$$

We take a constant stepsize η such that

$$0 \;<\; \eta \;<\; \frac{2}{\lambda^+_{\max}}.$$

This guarantees mean–square stability on the subspace $\mathrm{Im}(\mathcal{H})$, since $\rho\big(I - \eta\,\mathcal{H}\big) \le 1 - \eta\,\lambda^+_{\min}$.

Remark 1 (Remarks on the assumptions).

Exact vs. Lipschitz Hessian (on (A1)) When each ℓ_i is strictly quadratic, the local linearity $\nabla\ell_i(x) = \nabla\ell_i(x^\star) + \mathcal{H}\,(x - x^\star)$ holds *exactly*. In general, if $\nabla^2\ell_i$ is L_2–Lipschitz, the second–order expansion yields a small $\mathcal{O}(\|x - x^*\|^2)$ remainder. For small η, typical SGD iterates remain within $\mathcal{O}(\sqrt{\eta}\,)$ of $x^\star$, causing only $\mathcal{O}(\eta^2)$ corrections in the Lyapunov equation—less than the main $\mathcal{O}(\eta)$ term.

[11] When $\mathcal{H}$ is diagonalizable (e.g. symmetric PSD), it admits an eigenbasis $\{v_i\}$. Then $\mathcal{K}$ is diagonalizable in the basis $\{v_i \otimes v_j\}$ with eigenvalues $\lambda_i + \lambda_j$. Thus $\mathcal{K}^\dagger$ acts as $\frac{1}{\lambda_i+\lambda_j}$ on $v_i \otimes v_j$, and zero if $\lambda_i + \lambda_j = 0$.

Small drift in flat directions (on (A2)) The condition $P_0\,\Sigma_g\,P_0 = 0$ can be relaxed to $\|P_0\,\Sigma_g\,P_0\| \le \delta$. A discrete–Lyapunov analysis shows that Σ_x remains finite provided $\delta = \mathcal{O}(\eta)$. Concretely, if $\|P_0\,\Sigma_g\,P_0\| \le \frac{1}{2}\,b\,\lambda^+_{\min}\,\eta$, then the null–space covariance Σ_x^{00} does not diverge, matching the same $\mathcal{O}(\eta)$ scale in $\mathrm{Im}(\mathcal{H})$.

Near–critical slow mixing (on (A3)) As $\eta \to 2/\lambda^+_{\max}$ from below, the *mixing time* $\tau_{\mathrm{mix}} \sim 1/(\eta\,\lambda^+_{\min})$ can become quite large. Reviewers may appreciate a remark that we assume $\eta\lambda^+_{\min} \ll 1$, so that the chain quickly enters stationarity and the covariance arguments hold.

Mini–Batch SGD and Gradient Noise. For each iteration t, sample a mini–batch B_t of size b (either i.i.d. or well–shuffled from a finite dataset) and define

$$g_t \;:=\; \frac{1}{b}\sum_{i\in B_t}\nabla\ell_i(x_t), \qquad x_{t+1} \;:=\; x_t \;-\; \eta\,g_t.$$

Let

$$\Delta_t \;:=\; x_t - x^\star, \quad \xi_t \;:=\; g_t - \nabla L(x^\star).$$

Then $\mathbb{E}[\xi_t] = 0$ and $\mathbb{E}[\xi_t\,\xi_t^\top] = \frac{1}{b}\,\Sigma_g$. In particular, for *with–replacement* (i.i.d.) sampling we have $\mathbb{E}[\xi_t \mid x_t] = 0$, so the cross–term $\mathbb{E}[\xi_t\,\Delta_t^\top] = 0$. For *without–replacement* sampling, one can show $\mathbb{E}[\xi_t\,\Delta_t^\top]$ remains $\mathcal{O}(\eta)$ (Smith et al., 2021; Beneventano, 2023; Mishchenko et al., 2020), hence appearing only at order $\mathcal{O}(\eta^2)$ in the final covariance.

Proposition 2 (General Gradient–Noise Interaction)**.** *Under the above Setup/Notation and assumptions **(A1)**–**(A3)**, consider the mini–batch SGD updates*

$$x_{t+1} \;=\; x_t - \eta\,g_t, \quad g_t \;=\; \tfrac{1}{b}\sum_{i\in B_t}\nabla\ell_i(x_t), \quad B_t \overset{\text{i.i.d.}}{\sim} \mathcal{P}_b.$$

Then the linearized *error process $\Delta_t := x_t - x^\star$ admits a unique stationary covariance Σ_x on $E_+ = \mathrm{Im}(\mathcal{H})$. In particular,*

$$\Sigma_x \;=\; \frac{\eta}{b}\,\mathcal{K}^\dagger\big(\Sigma_g\big) \;+\; \mathcal{O}(\eta^2), \qquad \mathcal{K}(X) \;=\; \mathcal{H}\,X + X\,\mathcal{H}, \tag{10}$$

and, under stationarity,

$$\frac{\mathbb{E}_\pi\mathbb{E}_B\big[\nabla L_B(x_t)^\top\,\mathcal{H}\,\nabla L_B(x_t)\big]}{\mathbb{E}_\pi\big[\|\nabla L(x_t)\|^2\big]} \;=\; \frac{2}{\eta}\Big[1 + \mathcal{O}(\eta)\Big]. \tag{11}$$

Furthermore, all hidden constants scale at most linearly in $\|\Sigma_g\|$ and inversely in $\lambda^+_{\min}$ and one can make the constant in $\mathcal{O}(\eta)$ explicit by Lipschitz bounds on $\mathcal{H}$.

In particular, this implies Proposition 1

$$\mathrm{GNI}(\mathbf{x}) \;:=\; \frac{\mathbb{E}_B\big[\,\nabla L_B(\mathbf{x})^\top\,\mathcal{H}\,\nabla L_B(\mathbf{x})\big]}{\|\nabla L(\mathbf{x})\|^2} \;\approx\; \frac{2}{\eta}.$$

Proof of Proposition 2.

Linearized Dynamics and Lyapunov Equation. By (A1) (or its Lipschitz extension) we have $\nabla L(x_t) = \mathcal{H}\,\Delta_t + \mathcal{O}(\|\Delta_t\|^2)$. Restricting to the leading linear term, the recursion is

$$\Delta_{t+1} \;=\; \big(I - \eta\,\mathcal{H}\big)\,\Delta_t \;-\; \eta\,\xi_t.$$

At stationarity, let $\Sigma_x := \mathbb{E}[\Delta_t\,\Delta_t^\top]$. Taking outer products in the linear approximation gives the discrete Lyapunov equation

$$\Sigma_x \;=\; \big(I - \eta\,\mathcal{H}\big)\,\Sigma_x\,\big(I - \eta\,\mathcal{H}\big)^\top \;+\; \frac{\eta^2}{b}\,\Sigma_g \;+\; \mathcal{O}(\eta^3), \tag{12}$$

where higher–order terms come from the $\mathcal{O}(\|\Delta_t\|^2)$ nonlinearity and possible small correlation $\mathbb{E}[\xi_t\,\Delta_t^\top]$. These can be shown to contribute only at order $\mathcal{O}(\eta^2)$ or higher in Σ_x.

Vectorization and Solving for Σ_x. Define $\mathrm{vec}(\cdot)$ so that $\mathrm{vec}(A\,X\,B) = (B^\top \otimes A)\,\mathrm{vec}(X)$. Then (12) rewrites as

$$\Big[I \;-\; \big(I-\eta\,\mathcal{H}\big)\otimes\big(I-\eta\,\mathcal{H}\big)\Big]\,\mathrm{vec}(\Sigma_x) \;=\; \frac{\eta^2}{b}\,\mathrm{vec}\big(\Sigma_g\big) \;+\; \mathcal{O}(\eta^3).$$

Since $(I-\eta\,\mathcal{H})^{\otimes 2} = I-\eta\,\mathcal{K}+\mathcal{O}(\eta^2)$, the bracket equals $\eta\,\mathcal{K}+\mathcal{O}(\eta^2)$. Restricting to $E_+\otimes E_+$, the operator $\mathcal{K}$ is positive–definite, so its Moore–Penrose inverse $\mathcal{K}^\dagger$ exists there (while $\mathcal{K}=0$ on null directions). Hence

$$\mathrm{vec}(\Sigma_x) \;=\; \eta\,\mathcal{K}^\dagger\Big(\tfrac{1}{b}\,\mathrm{vec}(\Sigma_g)\Big) \;+\; \mathcal{O}(\eta^2).$$

Reverting to matrix form proves the key statement $\Sigma_x = \frac{\eta}{b}\,\mathcal{K}^\dagger(\Sigma_g) + \mathcal{O}(\eta^2)$.

Full–Batch vs. Mini–Batch Gradients. Next, we compare the second–moment of $\nabla L(x_t)$ versus $\nabla L_B(x_t)$. By definition,

$$\nabla L_B(x_t) \;=\; \xi_t \;+\; \nabla L(x_t).$$

Under i.i.d. sampling, ξ_t is conditionally uncorrelated with Δ_t, hence also with $\nabla L(x_t) = \mathcal{H}\,\Delta_t$. We then get:

$$\mathbb{E}_B\big[\nabla L_B(x_t)^\top\,\mathcal{H}\,\nabla L_B(x_t)\big] \;=\; \tfrac{1}{b}\,\mathrm{tr}\big(\mathcal{H}\,\Sigma_g\big) \;+\; \mathrm{tr}\big(\mathcal{H}\,\Sigma_x\,\mathcal{H}\big).$$

Meanwhile,

$$\mathbb{E}\big[\|\nabla L(x_t)\|^2\big] \;=\; \mathrm{tr}\big(\mathcal{H}\,\Sigma_x\,\mathcal{H}\big).$$

Substituting $\Sigma_x \approx \frac{\eta}{b}\,\mathcal{K}^\dagger(\Sigma_g)$ makes both terms proportional to $\mathrm{tr}(\mathcal{H}\,\Sigma_g)$, and one finds

$$\frac{\mathbb{E}_B\big[\nabla L_B^\top\,\mathcal{H}\,\nabla L_B\big]}{\mathbb{E}\big[\|\nabla L(x_t)\|^2\big]} \;=\; \frac{2}{\eta}\,\Big(1 \;+\; \mathcal{O}(\eta)\Big).$$

This completes the proof of (11) and Proposition 2. □

Remark 2 (Discussion and Constants).

- **Explicit constants in $\mathcal{O}(\eta)$.** If $\|\mathcal{H}\| \le L_2$ and $\lambda_{\min}^+ > 0$, then $\|\mathcal{K}^\dagger\| \lesssim (\lambda_{\min}^+)^{-1}$. Consequently, the $\mathcal{O}(\eta^2)$ terms in (10) scale linearly with $\|\Sigma_g\|$ and at most quadratically with $(\lambda_{\min}^+)^{-1}$. Concretely,

$$\big\|\Sigma_x - \tfrac{\eta}{b}\,\mathcal{K}^\dagger(\Sigma_g)\big\|_2 \;\le\; C_{\mathrm{nonlin}}\,\eta^2\,\|\Sigma_g\|_2, \quad \text{where} \quad C_{\mathrm{nonlin}} \;=\; \frac{L_2}{\big(2\,\lambda_{\min}^+\big)^2},$$

  and thus $\|\Sigma_x\|_2 \;\le\; \frac{\eta}{2\,b\,\lambda_{\min}^+}\|\Sigma_g\|_2 + \mathcal{O}(\eta^2)$.

- **Without–replacement sampling.** For large n or for well–shuffled datasets, the correlation introduced by sampling without replacement typically appears only in the cross–term $\mathbb{E}[\xi_t\,\Delta_t^\top]$ at order $\mathcal{O}(\frac{b}{n}\,\eta)$, which is again absorbed into $\mathcal{O}(\eta^2)$ at the level of Σ_x. Hence all conclusions remain valid to leading order $\mathcal{O}(\eta)$.

- **Flat directions.** If $\|P_0\,\Sigma_g\,P_0\|$ is nonzero but $\lesssim \eta$, the same discrete Lyapunov analysis shows there is still a finite stationary Σ_x with $\mathcal{O}(\eta)$ scale in E_+ and at most $\mathcal{O}(\eta)$ in E_0. If $\|P_0\,\Sigma_g\,P_0\| \gg \eta$, the iterates execute an unbounded random walk in the null space and the stationary covariance diverges in those directions.

Remark 3.

1. **Universality.** The factor $2/\eta$ emerges *without assuming* $[\mathcal{H},\Sigma_g]=0$ or $\mathcal{H}\succ 0$; commutativity affects only the size of the $\mathcal{O}(\eta)$ remainder. Thus *every* stable constant–stepsize SGD orbit satisfies the Gradient–Noise Interaction (GNI) rule.
2. **Flat directions.** Condition $P_0\Sigma_g P_0 = 0$ in (A2) is essential: if violated, the iterates execute an uncontrolled random walk in E_0, for which no finite stationary covariance exists and the ratio is undefined.

3. **Edge of Stability.** Observing GNI $\approx 2/\eta$ in practice is therefore *necessary but not sufficient* for curvature–driven ("Edge-of-Stability") dynamics. One must additionally check that the top Hessian eigenvalue $\lambda^+_{\max}$ itself approaches $2/\eta$.
4. **Higher–order corrections.** Retaining the $\eta^2(H \otimes H)$ term in the Neumann expansion of $(I - \eta\mathcal{H})^{\otimes 2}$ refines Σ_x and hence the GNI ratio; see (Mandt et al., 2016; Bottou et al., 2018) for explicit bounds.

Discussion of Key Assumptions & Broader Context. (1) Local linearity and neighborhood size. We rely on the *locally linear* approximation $\nabla\ell_i(x) \approx \nabla\ell_i(x^\star) + \mathcal{H}\,(x - x^\star)$ from Assumption (A1), which is exact in the strictly quadratic case and otherwise follows from a second-order Taylor expansion with Lipschitz Hessians. In practice, as long as the step size η is small enough, the SGD iterates remain close to $x^\star$. Consequently, the higher-order ($\mathcal{O}(\|x - x^\star\|^2)$) terms contribute only $\mathcal{O}(\eta^2)$ to the discrete Lyapunov equation, negligible compared to our main $\mathcal{O}(\eta)$ term. A related subtlety is that our stationarity result is *exact* for the linearized process; transferring it to the fully nonlinear dynamics requires "small-noise ergodicity" arguments (e.g. Rhee and Glynn, 2017; Meyn and Tweedie, 2009), ensuring that for sufficiently small η, the nonlinear SGD inherits nearly the same stationary covariance.

(2) Flat directions and step-size stability. Assumption (A2) states $P_0\,\Sigma_g\,P_0 = 0$ (or at least $\|P_0\,\Sigma_g\,P_0\| \lesssim \eta$) to prevent unbounded drift along $\ker(\mathcal{H})$. If there is too much noise in flat directions, the covariance Σ_x diverges, and the notion of a stable "oscillation" no longer holds. Assumption (A3) with $\eta < 2/\lambda^+_{\max}$ ensures Δ_t is *mean-square stable* in the curved subspace $E_+ = \mathrm{Im}(\mathcal{H})$. In changing or "progressively sharpening" landscapes, one must still check that η remains safely below $2/\lambda^+_{\max}$. Another practical concern is whether ξ_t is truly independent of x_t. For i.i.d. mini-batches, the tower property $\mathbb{E}[\xi_t \mid x_t] = 0$ justifies the cross-term in the Lyapunov approach; without-replacement sampling, one obtains a small correlation bounded by $\mathcal{O}(\frac{b}{n}\,\eta)$, still absorbed by our $\mathcal{O}(\eta^2)$ remainder.

(3) Universal ratio vs. Edge-of-Stability. A key outcome is that the ratio

$$\frac{\mathbb{E}_B\big[\nabla L_B(x)^\top\,\mathcal{H}\,\nabla L_B(x)\big]}{\|\nabla L(x)\|^2} \;\approx\; \frac{2}{\eta}$$

holds *broadly* once the system is in a stable constant-stepsize regime. However, by itself it does not guarantee that the top Hessian eigenvalue is exactly $2/\eta$. Both curvature-driven and noise-driven oscillations can produce the same measured ratio. Thus, to detect a genuine "Edge-of-Stability" (EoS), one must also verify that $\lambda^+_{\max}(\mathcal{H}) \approx 2/\eta$. Otherwise, the $2/\eta$ ratio simply reflects noise-dominated plateaus.

Historical note on the $2/\eta$ law and Lyapunov analysis. The observation that fixed–stepsize SGD does *not* converge to a point but rather equilibrates at a noise–controlled "temperature" originates in classical stochastic approximation. Already in the seminal paper of Robbins and Monro (1951) a diminishing step was prescribed precisely to avoid this residual variance. In the neural–network community, Bottou (1991) drew the first explicit analogy between a constant step size and the temperature of a thermodynamic system, noting that the algorithm must reach a stationary distribution whose spread scales with η. This qualitative picture foreshadows the

$$\frac{\mathbb{E}[\nabla L_B^\top \mathcal{H}\,\nabla L_B]}{\|\nabla L\|^2} \;\approx\; \frac{2}{\eta}$$

identity proved in Proposition 1.

Control–theoretic formalization. A rigorous derivation of the stationary covariance appeared with the discrete–time Lyapunov treatment of Mandt et al. (2016), who viewed constant–stepsize SGD on a local quadratic as a Markov chain converging to an Ornstein–Uhlenbeck law. Solving the Lyapunov equation yields $\Sigma_x = \frac{\eta}{b}\mathcal{K}^\dagger(\Sigma_g) + \mathcal{O}(\eta^2)$, and, after a short calculation, the gradient–noise interaction plateaus at $\approx 2/\eta$. For strongly convex objectives this covariance was independently derived in the comprehensive optimisation survey

of Bottou et al. (2018, Thm. 4.6), who emphasised that the full–batch gradient cannot decay below a floor proportional to $\eta \operatorname{Tr}(H\Sigma_g)$. The commutativity and invertibility constraints of these early analyses were removed by Yaida (2018), who solved the *exact* discrete Lyapunov equation in the general non–normal, possibly degenerate case. His fluctuation–dissipation relation shows that even without $[\mathcal{H}, \Sigma_g] = 0$ (or $\mathcal{H} \succ 0$) the first–order term in η still enforces the same $2/\eta$ ratio proved in our Lemma.

D.1 Full- vs. mini-batch gradient norms

Lemma 2 (Bias–variance identity.)**.** *For every $\theta \in \mathbb{R}^d$*

$$\mathbb{E}_B\big[\|\nabla L_B\|^2\big] \;=\; \|\nabla L\|^2 \;+\; \mathbb{E}_B\big[\|\nabla L_B - \nabla L\|^2\big]. \tag{13}$$

Proof. Expand $\|\nabla L_B\|^2 = \|\nabla L + (\nabla L_B - \nabla L)\|^2$ and take $\mathbb{E}_B[\cdot]$. The cross term vanishes because $\mathbb{E}_B[\nabla L_B] = \nabla L$. □

Explicit variance. Let $\Sigma(\theta) := \operatorname{Cov}_{(x,y)\sim\mathcal{D}}[\,\nabla L_B(\theta)\,]$, $B \sim \mathcal{P}_b$, $b = 1$, be the covariance of a *single sample* gradient. Then, for i.i.d. batches of size b,

$$\mathbb{E}_B\big[\|\nabla L_B - \nabla L\|^2\big] = \frac{1}{b}\,\operatorname{tr}\big(\Sigma(\theta)\big).$$

For a finite dataset of size n sampled *without* replacement the factor $1/b$ is replaced by $(n-b)/\big(b\,(n-1)\big)$.

Stationary, small-step regime. In Appendix D we solved the discrete Lyapunov equation and obtained (first order in the step size η):

$$\|\nabla L(\theta)\|^2 = \frac{\eta}{b}\,\operatorname{tr}\big(\mathcal{H}\mathcal{K}^\dagger(\Sigma)\mathcal{H}\big) + \mathcal{O}(\eta^2)$$

where $\mathcal{K}(X) = \mathcal{H}X + X\mathcal{H}$ and $\mathcal{K}^\dagger$ is its Moore–Penrose inverse on $\operatorname{Im}\mathcal{H} \otimes \operatorname{Im}\mathcal{H}$. This implies that

$$\frac{\|\nabla L\|^2}{\mathbb{E}_B[\|\nabla L_B\|^2]} \;=\; \eta\,\frac{\operatorname{tr}(\mathcal{H}\mathcal{K}^\dagger(\Sigma)\mathcal{H})}{\operatorname{tr}(\Sigma) + \eta\operatorname{tr}(\mathcal{H}\mathcal{K}^\dagger(\Sigma)\mathcal{H})} + \mathcal{O}(\eta^2). \tag{14}$$

Spectral bounds on the bias–variance ratio. Let the eigen–decomposition of the full-batch Hessian be $\mathcal{H} = \sum_{i=1}^r \lambda_i\, v_i v_i^\top$ with strictly positive eigen-values $0 < \mu := \lambda_1 \le \cdots \le \lambda_r = \lambda_{\max}$ and let $P_+ := \sum_{i=1}^r v_i v_i^\top$ be the projector onto $\operatorname{Im}\mathcal{H}$. Define the curved-subspace covariance $\Sigma_+ := P_+\Sigma P_+$. Then, up to the $\mathcal{O}(\eta^2)$ term already displayed,

$$\frac{\mu}{2}\,\operatorname{tr}(\Sigma_+) \;\le\; \operatorname{tr}\big(\mathcal{H}\,\mathcal{K}^\dagger(\Sigma)\,\mathcal{H}\big) \;\le\; \frac{\lambda_{\max}}{2}\,\operatorname{tr}(\Sigma_+) \;\le\; \frac{\lambda_{\max}}{2}\,\operatorname{tr}(\Sigma). \tag{15}$$

Consequently

Lemma 3 (Bound on the ratio)**.** *In the oscillatory regime, we have*

$$\frac{\|\nabla L\|^2}{\mathbb{E}_B[\|\nabla L_B\|^2]} \;=\; \eta\,\frac{\operatorname{tr}(\mathcal{H}\mathcal{K}^\dagger(\Sigma)\mathcal{H})}{\operatorname{tr}(\Sigma) + \eta\operatorname{tr}(\mathcal{H}\mathcal{K}^\dagger(\Sigma)\mathcal{H})} + \mathcal{O}(\eta^2)$$

and

$$\frac{\eta\,\mu}{2}\,\frac{\operatorname{tr}(\Sigma_+)}{\operatorname{tr}(\Sigma) + \frac{\eta\lambda_{\max}}{2}\operatorname{tr}(\Sigma_+)} \;\le\; \frac{\|\nabla L\|^2}{\mathbb{E}_B[\|\nabla L_B\|^2]} \;\le\; \frac{\eta\,\lambda_{\max}}{2}\,\frac{\operatorname{tr}(\Sigma_+)}{\operatorname{tr}(\Sigma) + \frac{\eta\mu}{2}\operatorname{tr}(\Sigma_+)}.$$

If the gradient noise has no component in the null-space of $\mathcal{H}$ ($\Sigma_+ = \Sigma$),

$$\frac{\eta\mu}{2+\eta\mu} \;\le\; \varepsilon^2(\theta) \;\le\; \frac{\eta\lambda_{\max}}{2+\eta\lambda_{\max}}, \qquad \frac{\|\nabla L\|^2}{\mathbb{E}_B[\|\nabla L_B\|^2]}) \in \big[\tfrac{\eta\mu}{2}, \tfrac{\eta\lambda_{\max}}{2}\big] + \mathcal{O}(\eta^2).$$

Proof. Write $\Sigma = \sum_{i,j} s_{ij}\, v_i v_j^\top$. Because $\mathcal{K}^\dagger(v_i v_j^\top) = (\lambda_i + \lambda_j)^{-1} v_i v_j^\top$ for $\lambda_i + \lambda_j > 0$,

$$\mathcal{H}\,\mathcal{K}^\dagger(\Sigma)\,\mathcal{H} = \sum_{i,j} \frac{\lambda_i \lambda_j}{\lambda_i + \lambda_j}\, s_{ij}\; v_i v_j^\top.$$

Taking the trace removes all off-diagonal terms and gives

$$\operatorname{tr}\big(\mathcal{H}\,\mathcal{K}^\dagger(\Sigma)\,\mathcal{H}\big) = \sum_{i=1}^{r} \frac{\lambda_i^2}{2\lambda_i}\, s_{ii} = \frac{1}{2} \sum_{i=1}^{r} \lambda_i\, s_{ii}.$$

Each coefficient λ_i lies between μ and $\lambda_{\max}$, while $\sum_i s_{ii} = \operatorname{tr}(\Sigma_+)$. This yields the two inequalities in (15). Insert them into the fraction $\eta \operatorname{tr}(\mathcal{H}\mathcal{K}^\dagger(\Sigma)\mathcal{H}) / \big[\operatorname{tr}(\Sigma) + \eta \operatorname{tr}(\mathcal{H}\mathcal{K}^\dagger(\Sigma)\mathcal{H})\big]$ and simplify. □

E On the Two Types of Oscillations in NNs

Differentiating Oscillations in Neural Network Optimization Our analytical treatment of SGD on one-dimensional quadratic objectives in Appendix C.1 leverages the simplicity of having a single curvature measure–the second derivative–which facilitates a precise landscape characterization and explicit stability conditions. However, extending this analysis to multidimensional quadratics already introduces significantly more intricate dynamics, necessitating advanced analytical frameworks as developed by (Wu et al., 2018; Ma and Ying, 2021; Mulayoff and Michaeli, 2024). Transitioning further to neural network optimization increases this complexity dramatically, since training predominantly occurs away from the manifold of minima, including the EoS-like instabilities themselves (as evidenced by the continuous reduction in loss)—and therefore requires to go beyond linear stability of quadratics near the manifold of minima.

Given the current absence of robust theoretical tools to comprehensively analyze such dynamics, distinguishing between curvature-driven and noise-driven oscillations necessitates empirical experimentation. Specifically, we probe the dynamics by systematically varying hyperparameters (e.g., step size or batch size), as illustrated in Figure 3, allowing us to differentiate curvature-induced (Type-2) oscillations from purely noise-induced (Type-1) oscillations (Figure 5).

Type-2 Oscillations Are Unique to NN Optimization This complexity inherent in neural network optimization is not merely an analytical inconvenience; rather, it is intrinsically tied to the emergence and significance of Type-2 oscillations and EoS-style phenomena. Notably, Type-2 oscillations emerge naturally[12] only in the case of neural network optimization, but not in the case of quadratic objectives. In the one-dimensional quadratic scenario analyzed previously, curvature-driven oscillations require the step size to precisely match the stability threshold $2/\lambda_{\max}$, or exceed it, in which case we have divergence—in either case, it means that optimization of quadratics does naturally enter a regime of instability. In contrast, neural network optimization uniquely exhibits *progressive sharpening*, a third-order derivative phenomenon (Damian et al., 2023), where curvature naturally increases during training. This progressive increase in curvature means that training with a fixed step size can transition into an EoS-like regime of instability without any explicit adjustment of the hyperparameters, and stay there due to self-stabilization effects (Damian et al., 2023). Hence, Type-2 oscillations emerge naturally and robustly within neural network training dynamics due to this intrinsic change of the loss landscape. Consequently, Type-2 oscillations and EoS-like regimes are fundamentally driven by progressive sharpening, which does not happen in quadratics, making it a purely neural network optimization phenomena.

[12] We define an as emerging *naturally* if it arises inherently from the training dynamics, and not a result of precisely-selected hyperparameters or initializations, reflecting a fundamental characteristic of the optimization process itself. Formally, it needs to happen over a range of hyperparameter choices and initializations.

E.1 On the Importance of Type-2 Oscillations Compared to Type-1

Noise-induced (Type-1) oscillations are not unstable when introducing slight perturbations (increase step size or decrease batch size), as showcased in Figure 9 and 10. Therefore, they do not constitute an EoS-type phenomena, where slight perturbations do cause divergence ("complete" divergence as long as we consider just the quadratic terms and can ignore higher terms — the fact that it doesn't fully diverge is exactly the higher-terms effect). Instead, after a perturbation, noise-induced oscillations quickly re-stabilize at a higher level.

Crucially, a lack of such divergence means that noise-induced oscillations wouldn't exhibit the self-stabilization mechanism of Damian et al. (2023) characteristic of EoS (differing it from classical convex optimization). Moreover, as shown in the quadratic example and in the proofs, noise-induced oscillations happen for any quadratic, for a wide range of step sizes, making them inherently "unsurprising", while EoS is a beyond-quadratic phenomena (and, as far as we know, a deep-learning-specific phenomena), as it relies on both progressive sharpening and the aforementioned self-stabilization, both being an effects of higher order terms. And the reason why we care specifically about effects of beyond-quadratic terms is specifically the adaptation of the landscape to the hyper-parameters, which is, by definition, an effect of higher order terms. That is the reason we specifically care about curvature-driven oscillations.

Now, with all of the above, GNI, being an indicator of those noise-induced oscillations, is therefore not an indicator of EoS-like regime. This is despite the fact that GNI in SGD comes from the same place as $\lambda_{\max}$/Rayleigh quotient in GD — i.e. from the descent lemma; yet, it does not mean that the two quantities serve the same role. Instead, it is the presence of the natural noise in SGD that makes the analysis much more complex. Instead, GNI has its usefulness as a measure of the level of noise coming from SGD. That is, noise-induced oscillations are influenced by the Hessian, but are also strongly influenced by the ratio between the noise covariance and the norm of full batch gradient, with the latter being the leading cause of change. In particular, GNI is decoupled from the Hessian, and can change drastically without any change of landscape sharpness, as showcased in our experiments. Lastly, another important consequence of EoS is that the landscape adapts to the hyper-parameters (rather than the other way around in classical optimization). With GNI being decoupled from the Hessian, GNI being at 2/eta is not an indication of landscape adopting to the hyper-parameters, as is the case with $\lambda_{\max}$ being at $2/\eta$ during GD.

F Quadratic Setting: *Batch Sharpness* and *GNI*

This appendix serves as a set-up of the setting for the proof of Theorem 1 in Appendix G.

F.1 Setting: Data and risk.

Intuitively, we minimise a *random quadratic* loss whose curvature changes with each sample. Let $(H_i, x_i)_{i\geq 1}$ be i.i.d. with

$$H_i \in \mathbb{R}^{d\times d},\ H_i \succeq 0, \quad x_i \in \mathbb{R}^d, \quad \mathbb{E}\big[\|H_i\|_{\mathrm{F}}^4 + \|x_i\|^4\big] < \infty.$$

Define[13]

$$H := \mathbb{E}[H_i], \quad G := \mathbb{E}[H_i x_i], \quad \theta^\star := H^{-1}G.$$

The population risk is $L(\theta) := \mathbb{E}_i\big[(\theta - x_i)^\top H_i(\theta - x_i)\big]$.

F.2 SGD dynamics.

With *constant stepsize* η satisfying

$$0 < \eta < \frac{2}{\lambda_{\max}(H)}, \tag{16}$$

[13] All eigenvalues of $H := \mathbb{E}[H_i]$ are positive because $\lambda_{\min}(H) > 0$ is assumed; the '+' superscript in earlier drafts is therefore redundant.

the update

$$\theta_{t+1} = \theta_t - 2\eta\, H_{i_t}\big(\theta_t - x_{i_t}\big), \tag{17}$$

where $i_t \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}\{1,\dots,n\}$ and $i_t \perp\!\!\!\perp \theta_t$, generates a Markov chain that is geometrically ergodic (Meyn and Tweedie, 2009). Let π_η denote its unique stationary law and write $m_\eta := \mathbb{E}_{\pi_\eta}[\theta]$, $\Sigma_\eta := \mathrm{Cov}_{\pi_\eta}(\theta)$.

Key intuition: the stochastic gradient $g(\theta, i) := 2H_i(\theta - x_i)$ is unbiased, $\mathbb{E}_i[g(\theta,i)] = \nabla L(\theta) = 2H(\theta - \theta^\star)$, so the mean iterate should coincide with $\theta^\star$.

F.3 Stationary mean and covariance.

The discussion in Appendix C implies that in this setting

Lemma 4 (Lyapunov solution)**.** *For any stepsize satisfying* (16)*,*

$$m_\eta = \theta^\star, \tag{18}$$

$$\Sigma_\eta = \frac{\eta}{2} H^{-1}\Sigma_g H^{-1} + \mathcal{O}(\eta^2), \qquad \Sigma_g := 4\ \mathrm{Var}_i\big[H_i(\theta^\star - x_i)\big], \tag{19}$$

where the $\mathcal{O}(\eta^2)$ constant depends polynomially on $\lambda_{\min}(H)^{-1}$ and on the 4^{th}-moment bound above.

Sketch. Set $\Delta_t := \theta_t - \theta^\star$ and $\xi_t := 2H_{i_t}(\theta^\star - x_{i_t})$ (mean 0, variance Σ_g). Subtracting (17) at $\theta^\star$ gives

$$\Delta_{t+1} = (I - 2\eta H)\Delta_t - \eta\,\xi_t - 2\eta\big(H_{i_t} - H\big)\Delta_t. \tag{20}$$

Mean. Taking expectations and using $i_t \perp\!\!\!\perp \theta_t$, $\mathbb{E}[\Delta_{t+1}] = (I - 2\eta H)\,\mathbb{E}[\Delta_t]$, whose unique fixed point is 0, proving $m_\eta = \theta^\star$ (18).

Covariance. Let $S_t := \mathbb{E}[\Delta_t \Delta_t^\top]$. Multiplying (20) by its transpose and exploiting $\mathbb{E}[\Delta_t] = 0$, one obtains the discrete Lyapunov recursion $S_{t+1} = (I - 2\eta H)S_t(I - 2\eta H)^\top + \eta^2\Sigma_g + R_t$ with $\|R_t\| = O(\eta^3)$ thanks to $\mathbb{E}\|H_i\|_{\mathrm{F}}^4 < \infty$. Passing to the limit and inverting $\big[I - (I - 2\eta H)^{\otimes 2}\big]$ by a Neumann series yields (19). □

Take-away. *Constant-stepsize SGD is **unbiased in the mean** but has an $\mathcal{O}(\eta)$ stationary variance that blows up if η is too large; the $2/\lambda_{\max}(H)$ is thus necessary for avoiding the blowup, although not sufficient.*

F.4 Per-sample residuals.

Define, for any $\theta \in \mathbb{R}^d$,

$$Y_i(\theta) := H_i(\theta - x_i), \quad \mu(\theta) := \mathbb{E}_i[Y_i(\theta)] = H\theta - G.$$

At stationarity write $Y := Y_i(\theta)$, $\mu := \mu(\theta)$, and set

$$\widetilde{Y} \;:=\; Y - \mathbb{E}_i[Y \mid \theta] \;=\; Y - \mu \quad\Longrightarrow\quad \mathbb{E}_i[\widetilde{Y} \mid \theta] = 0.$$

Finally define the scalars[14]

$$\begin{aligned} \mathcal{A} &:= \mathbb{E}_{i,\pi}\big[Y^\top H_i Y_i\big], & \mathcal{B} &:= \mathbb{E}_{i,\pi}\big[Y^\top H Y_i\big], & \mathcal{C} &:= \mathbb{E}_{i,\pi}\big[\|Y_i\|^2\big], \\ C_0 &:= \mathbb{E}_{i,\pi}\big[\big\|\widetilde{Y}_i\big\|^2\big], & \mathcal{D} &:= \mu^\top \mathbb{E}_{i,\pi}\big[H_i Y_i\big]. \end{aligned} \tag{21}$$

Why these symbols? $\mathcal{A}$ is the *average loss at stationarity*; $\mathcal{B}$ and $\mathcal{C}$ act as *mixed-moment controls* that will upper-bound $\mathcal{A}$ via a trace inequality.

[14] Notation: $\mathbb{E}_{i,\pi}$ integrates i *and* $\theta \sim \pi_\eta$.

F.5 Bounding curvature–fluctuation interaction.

Lemma 5 (Cross term). *Let* $\widetilde{\Delta} := \mathbb{E}_{i,\pi}\big[\widetilde{Y}^\top (H_i - H)\widetilde{Y}\big]$ *and* $\Delta := \mathcal{A} - \mathcal{B} = \mathbb{E}_{i,\pi}\big[Y^\top (H_i - H)Y\big]$. *Then* $\Delta := \mathcal{A} - \mathcal{B} = \widetilde{\Delta} + 2\mathcal{D}$,

$$\widetilde{\Delta} \;\le\; \sqrt{\mathbb{E}_i[\|H_i - H\|_2^2]}\,\sqrt{\mathbb{E}_\pi[\|\widetilde{Y}\|^4]}, \tag{22}$$

and

$$\Delta \;\le\; \sqrt{\mathbb{E}_i[\|H_i - H\|_2^2]}\,\sqrt{\mathbb{E}_\pi[\|\widetilde{Y}\|^4] + 2C_0\|\mu\|^2 + \|\mu\|^4}. \tag{23}$$

Proof. In the following two steps we establish the first inequality.
Step 1 (Cauchy - Schwarz). For any matrix A and vector v, $|v^\top A v| \le \|A\|_2\,\|v\|^2$. With $A := H_i - H$ and $v := Y$:

$$|\,\widetilde{Y}^\top (H_i - H)\widetilde{Y}| \le \|H_i - H\|_2\,\|\widetilde{Y}\|^2.$$

Note that this is sharp if and only if $\widetilde{Y}$ aligns with the eigenvectors of the maximal eigenvalues of $H_i - H$.

Step 2 (Average and separate: Jensen). Independence of i and θ at stationarity implies, taking the expectation that

$$|\Delta| \le \sqrt{\mathbb{E}_i[\|H_i - H\|_2^2]}\,\sqrt{\mathbb{E}_\pi[\|\widetilde{Y}\|^4]}.$$

Second inequality. Note that

$$Y^\top (H_i - H)Y \;-\; \widetilde{Y}^\top (H_i - H)\widetilde{Y} \;=\; 2\mu^\top (H_i - H)\widetilde{Y} + \underbrace{\mu^\top (H_i - H)\mu}_{centered}. \tag{24}$$

Note that the fact that $\mu^\top (H_i - H)\mu$ is centered in zero, implies that

$$\mathcal{A} \;=\; \mathcal{B} \;+\; \widetilde{\Delta} \;+\; 2\mathcal{D}. \tag{25}$$

This establish the first thesis of the lemma and implies that

$$|Y^\top (H_i - H)Y| \;\le\; \|H_i - H\|_2 \left(\|\widetilde{Y}\|^2 + 2|\mu^\top \widetilde{Y}| + \|\mu\|^2\right).$$

Taking the expectation we have

$$|\Delta| \le \sqrt{\mathbb{E}_i[\|H_i - H\|_2^2]}\,\sqrt{\mathbb{E}_\pi[\|\widetilde{Y}\|^4] + 2\|\mu\|_2^2 \cdot \mathbb{E}_\pi[\|\widetilde{Y}\|^2] + \|\mu\|_2^4}.$$

This concludes the proof of the lemma. □

Δ measures how *curvature noise* $(H_i - H)$ correlates with noise Y. Lemma 5 shows this interaction cannot exceed the *root product* of the moments of two natural quantities.

F.6 Main trace inequality

Proposition 3. *We have that Batch Sharpness is smaller than GNI if and only if*

$$\mathcal{B}C_0 \;\ge\; \|\mu\|_2^2 \cdot (\tilde{\Delta} + 2\mathcal{D}). \tag{26}$$

In particular,

$$\mathcal{B}\,\mathcal{C} - \mathcal{A}\,\|\mu\|^2 \;\ge\; \mathcal{B}C_0 - \sqrt{\mathbb{E}_i[\|H_i - H\|_2^2]}\,\sqrt{\mathbb{E}_\pi[\|\widetilde{Y}\|^4] + 2C_0\|\mu\|^2 + \|\mu\|^4}\|\mu\|^2. \tag{27}$$

This implies that Batch Sharpness is smaller than GNI up to $O(\eta)$ at the stationary distribution when

$$C_0 \ge \frac{\eta}{2}\sqrt{\mathbb{E}_i[\|H_i - H\|_2^2]}\,\sqrt{\mathbb{E}_\pi[\|\widetilde{Y}\|^4] + 2C_0\|\mu\|^2 + \|\mu\|^4}. \tag{28}$$

Proof. Note that $\mathcal{A}\|\mu\|^2 \le \mathcal{B}\|\mu\|^2 + |\Delta|\|\mu\|^2$. Applying Lemma 5 and noticing that $\mathcal{C} = C_0 + \|\mu\|^2$ concludes the proof of (27). Next note that up to $\mathcal{O}(\eta)$ we have that *GNI* $= \mathcal{B}/\|\mu\|^2 = 2/\eta$ at the stationary distribution. This proves the second part (28) and establishes Proposition 3. □

F.7 The moments of the oscillations

1. Centering first simplifies the ratio Because C_0 is defined with the centred residual $\widetilde{Y}$, we compare

$$\frac{\mathbb{E}_\pi[\|\widetilde{Y}\|^4] + 2C_0\|\mu\|^2 + \|\mu\|^4}{C_0^2} \;=\; \frac{\mathbb{E}[\|\widetilde{Y}\|^4]}{C_0^2} + 2\,\frac{\|\mu\|^2}{C_0} + \frac{\|\mu\|^4}{C_0^2}.$$

The first term is the *kurtosis-type* ratio $\kappa_Y \;:=\; \mathbb{E}[\|\widetilde{Y}\|^4]/C_0^2$.

2. Exact formula in the Gaussian quadratic setting Assume:

- $x_i \,|\, H_i \sim \mathcal{N}(\theta^\star, \Sigma_x)$ – an *isotropic* sampling model for simplicity;
- H_i is *deterministic*, i.e. $H_i \equiv H \succeq 0$.

Then $\xi := Y = H(\theta^\star - x_i) \sim \mathcal{N}\big(0,\ S\big)$ with $S := H\Sigma_x H$ and $\widetilde{Y} \equiv Y$ because $\mu = 0$. By Isserlis' theorem

$$\mathbb{E}[\|Y\|^4] \;=\; (\operatorname{tr} S)^2 + 2\operatorname{tr}(S^2), \qquad C_0 \;=\; \operatorname{tr} S.$$

Therefore

$$\boxed{\kappa_Y^{\text{Gauss}} \;=\; 1 + 2\,\frac{\operatorname{tr}(S^2)}{(\operatorname{tr} S)^2} \;\in\; \big[\,1 + \tfrac{2}{d},\, 3\,\big].} \tag{29}$$

- **Isotropic case** $S = \sigma^2 I_d \Longrightarrow \kappa = 1 + \frac{2}{d}$, e.g. $= 3$ in $d = 1$, $= 1.4$ in $d = 5$, $\to 1$ for large d.
- **Rank-one noise** $S = \sigma^2 uu^\top$ $(\|u\| = 1) \Longrightarrow \kappa = 3$ – the maximal value compatible with Gaussianity.
- The interval bounds follow from $\operatorname{tr}(S^2) \in [\frac{1}{d}(\operatorname{tr} S)^2,\, (\operatorname{tr} S)^2]$.

3. Effect of random curvature H_i If H_i varies but is *independent* of x_i and of θ a priori, the second moment becomes

$$C_0 \;=\; \mathbb{E}_H\big[\operatorname{tr}(H\Sigma_x H)\big],$$

while

$$\mathbb{E}[\|Y\|^4] \;=\; \mathbb{E}_H\Big[(\operatorname{tr}(H\Sigma_x H))^2 + 2\,\operatorname{tr}\big((H\Sigma_x H)^2\big)\Big].$$

Define $a := \mathbb{E}_H[\operatorname{tr}(H\Sigma_x H)],\ b := \mathbb{E}_H[\operatorname{tr}(H\Sigma_x H)^2],\ c := \mathbb{E}_H[\operatorname{tr}((H\Sigma_x H)^2)]$. Then

$$\kappa_Y \;=\; 1 \;+\; 2\,\frac{c}{a^2} \;+\; \frac{b - a^2}{a^2}. \tag{30}$$

The last term is a *curvature-variance correction* that vanishes when H_i is deterministic. By Cauchy-Schwarz, $0 \le b - a^2 \le \operatorname{Var}\big(\operatorname{tr}(H\Sigma_x H)\big)$, so $\kappa_Y \in [1 + \frac{2}{d},\, 3 + \tilde{\kappa}_H]$ where $\tilde{\kappa}_H := \frac{b - a^2}{a^2} \ge 0$ depends solely on the spread of $\operatorname{tr}(H\Sigma_x H)$.

F.8 Closing up

In the setting above, thus, we can transform (28) in the following inequality

$$1 \ge \frac{\eta}{2}\sqrt{\mathbb{E}_i[\|H_i - H\|_2^2]}\,\sqrt{\frac{\mathbb{E}_\pi[\|\widetilde{Y}\|^4] + 2C_0\|\mu\|^2 + \|\mu\|^4}{C_0^2}} \tag{31}$$

and the previous subsection tells us that there exists a universal constant κ_Y which is a form of kurtosis of our steps (which depends on our distribution) such that *Batch Sharpness* is smaller than *GNI* whenever

$$\sqrt{\mathbb{E}_i[\|H_i - H\|_2^2]} \;\le\; \frac{2}{\eta(\kappa_Y + \mathcal{O}(\eta))}. \tag{32}$$

Proposition 4 (Stationary trace inequality)**.** *For every η such that $0 < \eta < 2/\lambda_{\max}$ we have*

$$\mathcal{A}\,\|\mu\|^2 \;\le\; \mathcal{B}\,\mathcal{C} \tag{33}$$

up to $\mathcal{O}(\eta)$ *when*

$$\sqrt{\mathbb{E}_i[\|H_i - H\|_2^2]} \;\leq\; \frac{2}{\eta\kappa_Y} + \mathcal{O}(\eta). \tag{34}$$

In particular, at the beginning of the training for randomly initialized weights this is the case as the expectation of the LHS is $\mathcal{O}(\log(d))$ *where* d *is the number of parameters.*

Proof. The theorem follows from (32) and (30). □

Interpretation. Inequality (33) says: *"the loss under random curvature cannot align* too strongly *with the squared prediction bias."* When either (i) the stationary iterate is unbiased ($\mu = 0$) or (ii) the curvature is deterministic, the bound is tight.

G Gradients explode above the EoSS: Proof of Theorem 1

We now compute the update of the norm of the gradients $\mathbb{E}_i[\|Y_i\|_2^2]$ after one step in the setting in which we are at the EoSS. Precisely we are computing here the value of $\mathbb{E}_t\mathbb{E}_i[\|Y_i^{t+1}\|_2^2]$ so the average over the iterations of the update to the quantity $\mathcal{C}$ above. Precisely we here prove the following Proposition.

Proposition 5. *In the setting and notations of Appendix F.1 and F.4. Assume* $\eta \leq 2/\lambda_{\max}(\mathcal{H})$. *Then there exists an absolute constant* $c > 0$ *such that when Batch Sharpness* $> 2/\eta + c\eta$, *then* $\mathcal{C}$ *increases in size exponentially and the trajectory diverges (is quadratically unstable, see Definition 1). Note that assumptions on GNI are not necessary.*

Proof. In the proof we use the notations of Appendix F.

Step 1: One step on the gradient's second moment . Remind that the SGD iterate satisfy

$$\theta_{t+1} \;=\; \theta_t - \eta\, Y_{j_t}(\theta_t), \quad i_t \overset{\text{i.i.d.}}{\sim} \mathcal{D},$$

and define a *fresh*, independent index j used only for the outer expectation in $\mathcal{C}^{t+1}$. Because $j \perp i_t$ we may write

$$Y_i(\theta_{t+1}) \;=\; H_i\big(\theta_{t+1} - x_i\big) = Y_i(\theta_t) \;-\; \eta\, H_i Y_{j_t}(\theta_t).$$

Squaring, expanding, and averaging over j gives

$$\begin{aligned}\mathcal{C}^{t+1} &= \mathbb{E}_i\big\|Y_i(\theta_t) - \eta H_i Y_{j_t}(\theta_t)\big\|^2 \\ &= \mathcal{C}^t - 2\eta \underbrace{\mathbb{E}_{i,j_t}\big[Y_i(\theta_t)^\top H_i Y_{j_t}(\theta_t)\big]}_{\text{cross term}} + \eta^2 \underbrace{\mathbb{E}_{i,j_t}\big[Y_{j_t}(\theta_t)^\top H_i^2 Y_{j_t}(\theta_t)\big]}_{\text{variance term}}.\end{aligned} \tag{35}$$

Step 2: Decoupling the indices. Note that (25) in the proof of Lemma 5 establishes that

$$2\mathbb{E}_{i,j_t}\big[Y_i(\theta_t)^\top H_i Y_{j_t}(\theta_t)\big] \;=\; \mathcal{A} - \mathcal{B} - \widetilde{\Delta}. \tag{36}$$

This implies that we can rewrite

$$\mathcal{C}^{t+1} = \mathcal{C}^t - \eta\,(\mathcal{A} - \mathcal{B} - \widetilde{\Delta}) + \eta^2\,(\text{variance term}). \tag{37}$$

Next note that if we are at the EoSS, then $\mathcal{A} \approx \frac{2}{\eta}(1+\delta)\mathcal{C}$ for some $\delta \in \mathbb{R}$. This implies that we can rewrite the term above as

$$\mathcal{C}^{t+1} \;\approx\; -(1+2\delta)\,\mathcal{C}^t \;+\; \underbrace{\eta\mathcal{B} + \eta\widetilde{\Delta} + \eta^2\,(\text{variance term})}_{\text{rest}}. \tag{38}$$

Let us know understand the size of the rest, the trajectory diverges if and only if:

$$\eta\mathcal{B} + \eta\widetilde{\Delta} + \eta^2\mathbb{E}_{i,j_t}\big[Y_{j_t}(\theta_t)^\top H_i^2 Y_{j_t}(\theta_t)\big] \;>\; 2(1+\delta)\,\mathcal{C}^t. \tag{39}$$

Next note that by applying Jensen inequality to the term multiplied by η^2 we obtain that

$$\sqrt{\underbrace{\mathbb{E}_{i,j_t}\big[Y_{j_t}(\theta_t)^\top H_i^2 Y_{j_t}(\theta_t)\big]}_{\text{variance term}} \cdot \underbrace{\mathbb{E}_i\big[Y_i(\theta_t)^\top Y_i(\theta_t)\big]}_{\mathcal{C}}} \;\geq\; \underbrace{\mathbb{E}_{i,j_t}\big[Y_{j_t}(\theta_t)^\top H_i \cdot Y_i(\theta_t)\big]}_{\mathcal{D}}. \tag{40}$$

Step 3: Final algebra. Plugging this above, we obtain that the trajectory diverges when

$$\eta\mathcal{B} + \eta\widetilde{\Delta} + \eta^2\frac{\mathcal{D}^2}{\mathcal{C}} \;>\; 2(1+\delta)\,\mathcal{C}. \tag{41}$$

Again applying (36) we obtain that this is equivalent to

$$\eta\mathcal{B} + \eta\widetilde{\Delta} + \eta^2\frac{\left(\mathcal{A}-\mathcal{B}-\widetilde{\Delta}\right)^2}{4\mathcal{C}} \;>\; 2(1+\delta)\,\mathcal{C}. \tag{42}$$

Since $\eta\mathcal{A} = 2(1+\delta)\mathcal{C}$, then $\eta^2\mathcal{A}^2 = 4(1+\delta)^2\mathcal{C}^2$ to asking

$$\eta\mathcal{B} + \eta\widetilde{\Delta} + \eta^2\frac{\mathcal{B}^2 + \widetilde{\Delta}^2 - 2\mathcal{A}\widetilde{\Delta} - 2\mathcal{A}\mathcal{B} + 2\mathcal{B}\widetilde{\Delta}}{4\mathcal{C}} \;>\; 2(1+\delta)\,\mathcal{C} - \frac{4(1+\delta)^2\mathcal{C}^2}{4\mathcal{C}}. \tag{43}$$

Furthermore, equivalent to asking

$$\eta\mathcal{B} + \eta\widetilde{\Delta} - \frac{2(1+\delta)}{2}\eta\widetilde{\Delta} - \frac{2(1+\delta)}{2}\eta\mathcal{B} + \eta^2\frac{\mathcal{B}^2 + \widetilde{\Delta}^2 + 2\mathcal{B}\widetilde{\Delta}}{4\mathcal{C}} \;>\; (1-\delta+\delta^2)\,\mathcal{C} \tag{44}$$

or, even further simplified

$$\eta\delta(\mathcal{B} + \widetilde{\Delta}) + \eta^2\frac{(\mathcal{B}+\widetilde{\Delta})^2}{4\mathcal{C}} \;>\; (1-\delta+\delta^2)\,\mathcal{C}. \tag{45}$$

Here we plug in (25) again and we can rewrite this as

$$\eta\delta(\mathcal{A} - 2\mathcal{D}) + \eta^2\frac{(\mathcal{A}-2\mathcal{D})^2}{4\mathcal{C}} \;>\; (1-\delta+\delta^2)\,\mathcal{C}. \tag{46}$$

By plugging, as before, $\eta\mathcal{A} = 2(1+\delta)\mathcal{C}$ we obtain

$$2\delta(1+\delta)\mathcal{C} - 2\eta\delta 2\mathcal{D} - 2\eta(1+\delta)\mathcal{D} + \eta^2\frac{\mathcal{D}^2}{\mathcal{C}} \;>\; \left(1-\delta+\delta^2-(1+\delta)^2\right)\mathcal{C} \tag{47}$$

which simplifies as

$$\underbrace{2\eta(1+2\delta)\mathcal{D}}_{\mathcal{O}(\eta^2)} - \underbrace{\eta^2\frac{\mathcal{D}^2}{\mathcal{C}}}_{\mathcal{O}(\eta^4)} \;<\; \delta(5+2\delta)\;\underbrace{\mathcal{C}}_{\mathcal{O}_\eta(1)}. \tag{48}$$

Thus there exists a constant $c > 0$, such that if $\delta > c\eta^2$ the trajectory diverges exponentially, if $\delta < c\eta^2$ the trajectory is stable.

□

H When $H_i \equiv H$: Pure Gradient–Noise Oscillations

Set–up. Assume in the Setting of Appendix F every sample shares the *same* curvature: $H_i \equiv H \succeq 0$ for all i. SGD still sees noisy gradients $g_t = H\Delta_t + \xi_t$ with $\mathbb{E}_{i_t}[\xi_t] = 0$, $\mathrm{Var}_{i_t}(\xi_t) = \Sigma_g$, but now there is **no curvature noise**.

Batch Sharpness is capped by $\lambda_{\max}(H)$. For any vector v, $v^\top Hv \le \lambda_{\max}(H)\,\|v\|^2$. Averaging over the mini–batch therefore yields

$$\textit{Batch Sharpness}(\theta) = \frac{\mathbb{E}_i[g^\top Hg]}{\mathbb{E}_i[\|g\|^2]} \;\le\; \lambda_{\max}(H) \quad \forall\theta. \tag{49}$$

Hence, as long as the classical stability condition $\eta < 2/\lambda_{\max}(H)$ holds,

$$\boxed{\textit{Batch Sharpness} \;<\; \frac{2}{\eta}, \qquad \text{no Type–2 oscillations arise.}}$$

Gradient–Noise Interaction still plateaus at $\frac{2}{\eta}$. The derivation in Appendix C did *not* use $H_i - H$ fluctuations, only the fixed stepsize. Consequently the stationary covariance satisfies $\Sigma_x = \eta\,\mathcal{K}^\dagger(\Sigma_g) + \mathcal{O}(\eta^2)$, and

$$\mathrm{GNI} = \frac{\mathbb{E}_i[g^\top H g]}{\|\nabla L\|^2} = \frac{2}{\eta}\big[1 + \mathcal{O}(\eta)\big]. \tag{50}$$

Thus SGD still wiggles with the universal *Type–1* ratio $2/\eta$, even though the curvature never adapts.

Instability threshold reverts to $\lambda_{\max}(H)$. Because (49) enforces *Batch Sharpness* $\leq \lambda_{\max}(H)$, the only way to reach the critical value $2/\eta$ is to push the stepsize past the quadratic limit:

$$\eta \;>\; \frac{2}{\lambda_{\max}(H)} \quad\Longrightarrow\quad \text{divergence exactly as in full-batch GD.}$$

Summary. With $H_i \equiv H$ the dynamics exhibits *only* noise–driven (Type–1) oscillations:

- GNI$\to 2/\eta$ at stationarity (same as the general case);
- *Batch Sharpness* remains *below* $2/\eta$, bounded by $\lambda_{\max}(H)$;
- the classical GD threshold $\eta = 2/\lambda_{\max}(H)$ once again marks the onset of true instability.

In short, removing curvature variability collapses the Edge-of-*Stochastic* Stability back to the familiar quadratic picture.

I Mini-Batch *Without* Replacement

The main text and Appendix C—G treated SGD with i.i.d. sampling. Here we show how the two key quantities

$$\mathrm{GNI} \;=\; \frac{\mathbb{E}_B[g^\top \mathcal{H} g]}{\|\nabla L\|^2}, \qquad \textit{Batch Sharpness} \;=\; \frac{\mathbb{E}_B[g^\top H_B g]}{\mathbb{E}_B[\|g\|^2]},$$

behave when *each epoch* is a random permutation of the n samples (batch size b, $k := n/b$ steps per epoch).

We argue that without replacement sampling does not shift the EoSS in any practically relevant way. The only visible effect is a $1 - \frac{b-1}{n-1}$ reduction in gradient-noise variance, which cancels in the GNI and *Batch Sharpness* and appears only as a second-order $\mathcal{O}(b/n)$ correction. The differences in effect between with and without replacements are thus about speed of convergence (Mishchenko et al., 2020; Gürbüzbalaban et al., 2021) or in terms of traveling of the manifold of minima (Smith et al., 2021; Beneventano, 2023), not in terms of EoSS.

Noise statistics under permutation sampling. Fix θ and set $\bar{g} := \nabla L(\theta) = \frac{1}{n}\sum_{i=1}^n \nabla \ell_i(\theta)$. For a single batch B drawn *without* replacement

$$g_B := \frac{1}{b}\sum_{i\in B} \nabla \ell_i(\theta) = \bar{g} + \xi_B, \qquad \mathbb{E}[\xi_B \mid \theta] = 0,$$

and the conditional covariance is (Mishchenko et al., 2020; Beneventano, 2023)

$$\mathrm{Var}[\xi_B \mid \theta] = \frac{n-b}{b(n-1)}\,\Sigma_g(\theta), \quad \Sigma_g(\theta) := \frac{1}{n}\sum_{i=1}^n \big(\nabla \ell_i - \bar{g}\big)\big(\nabla \ell_i - \bar{g}\big)^\top.$$

Key point: compared to i.i.d. sampling the variance is simply scaled by $\alpha_{b,n} := \frac{n-b}{n-1} \in (0, 1]$.

Updated Lyapunov solution. Denote $\Delta_t := \theta_t - \theta^\star$ and keep the stepsize fixed. The error recursion for a *batch* update is

$$\Delta_{t+1} = (I - \eta\mathcal{H})\Delta_t - \eta\xi_{B_t}, \qquad \xi_{B_t} \perp\!\!\!\perp \theta_t.$$

Exactly as in Appendix C, the discrete Lyapunov equation now reads

$$\Sigma_x = (I - \eta\mathcal{H})\Sigma_x(I - \eta\mathcal{H}) + \eta^2\,\alpha_{b,n}\,\Sigma_g + \mathcal{O}(\eta^3),$$

so that

$$\boxed{\Sigma_x = \eta\,\alpha_{b,n}\,\mathcal{K}^\dagger(\Sigma_g) + \mathcal{O}(\eta^2).}$$

Correlation between ξ_{B_t} and Δ_t enters only at $\mathcal{O}\big(\frac{b}{n}\eta\big)$ and is absorbed into the $\mathcal{O}(\eta^2)$ remainder (see Beneventano, 2023, App. E).

Gradient–Noise Interaction is mostly unchanged. Plugging the equation above into the ratio $\mathbb{E}_B[g^\top\mathcal{H}g]/\|\nabla L\|^2$ shows that the common factor $\alpha_{b,n}$ *cancels*:

$$\boxed{\mathrm{GNI}_{\text{w/o repl.}} = \frac{2}{\eta}\Big[1 + \mathcal{O}(\eta) + \mathcal{O}\big(\tfrac{b}{n}\big)\Big].}$$

Hence the plateau $2/\eta$ persists up to tiny $\mathcal{O}(b/n)$ corrections.

Batch Sharpness and the instability edge.

1. **Variable curvature ($H_i \not\equiv H$).** The numerator and the denominator of *BatchSharpness* contain the *same* prefactor $\alpha_{b,n}$, so their ratio—like GNI—is unchanged at first order:

$$\mathit{BatchSharpness}_{\text{w/o repl.}} = \frac{2}{\eta}\Big[1 + \mathcal{O}(\eta) + \mathcal{O}\big(\tfrac{b}{n}\big)\Big].$$

   Consequently EoSS still appears once the directional curvature meets $2/\eta$; the $\mathcal{O}(b/n)$ shift is negligible whenever $n \gg b$.

2. **Constant curvature ($H_i \equiv H$).** The bound $\mathit{BatchSharpness} \le \lambda_{\max}(H)$ of Appendix H is unaltered, so the classical quadratic threshold $\eta = 2/\lambda_{\max}(H)$ remains the unique instability point.

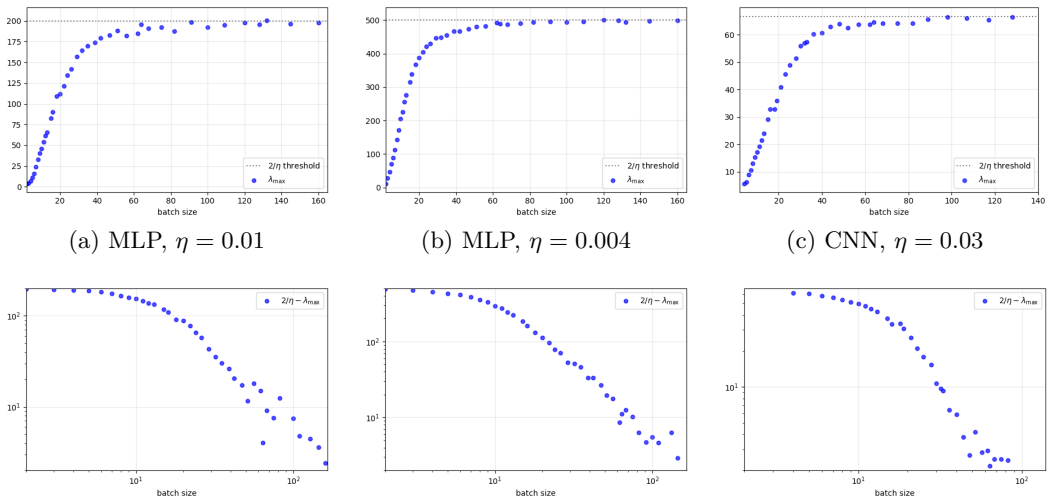
J On the fate of $\lambda_{\max}$

In this section we examine how $\lambda_{\max}$ behaves once EoSS is reached and clarify its relationship to *Batch Sharpness*. A key aspect of the original EoS analysis is, indeed, that the controlling quantity—the largest eigenvalue of the full-batch Hessian $\lambda_{\max}$—has an immediate geometric interpretation. There exists an extensive literature about $\lambda_{\max}$ size and role in neural networks, and it is a main ingredient of any proof of convergence. The EoSS picture replaces $\lambda_{\max}$ with *Batch Sharpness*, a statistic whose connection to generalization and role in optimization theory is largely unexplored.

J.1 Empirical facts

Below the phenomena we extensively observe in vision classification tasks trained with MSE, ablating on batch sizes, step sizes, architectures, datasets. See Figure 6 for a good reference of what generally goes on.

- **Fact 1: Progressive Sharpening.** $\lambda_{\max}$ increases at most as long as *Batch Sharpness* increases.
- **Fact 2: Phase Transition.** Once *Batch Sharpness* plateaus at $2/\eta$, $\lambda_{\max}$ stops increasing. If it moves, it only decreases from this time on.
- **Fact 3: Path-dependence.** If changes to hyper parameters are made, *Batch Sharpness* changes abruptly or restart growing and $\lambda_{\max}$ also changes. Stabilization of both happen as *Batch Sharpness* reaches $2/\eta$. The trajectory of $\lambda_{\max}$ is not fully determined by the size of hyper parameters (see Figure 7). That is, the level of $\lambda_{\max}$ is *path-dependent*: it inherits the history of progressive sharpening up to the moment EoSS is reached.



(a) MLP, $\eta = 0.01$ (b) MLP, $\eta = 0.004$ (c) CNN, $\eta = 0.03$

Figure 12: **Stabilisation level of $\lambda_{\max}$ across step sizes and architectures. Top:** final-epoch $\lambda_{\max}$ vs. batch size. **Bottom:** log–log plots of the gap $2/\eta - \lambda_{\max}$ for the same runs. All experiments use CIFAR-10 8k.

- **Fact 4: Smaller batches ⇒ flatter minima.** Across every setting we tested, reducing the batch size monotonically decreases the plateau level of $\lambda_{\max}$. This aligns with the long-standing empirical observation that smaller batches locate flatter minima see, e.g., Keskar et al. (2016); Jastrzębski et al. (2021)).
- **Fact 5: A critical batch size marks the SGD → GD crossover.** Each curve in Figure 12 exhibits a bend at $b \approx b_c(\eta)$: for $b < b_c$ the plateau falls rapidly with b, while for $b > b_c$ it flattens and approaches the full-batch value. This b_c corresponds to the regime in which the mini-batch landscapes approximate *well enough* the full-batch landscape, restoring GD-like dynamics (Appendix J.2).
- **Fact 6: No universal power law.** From *static* analysis, one would expect a scaling $2/\eta - \lambda_{\max} = O(b^{-\alpha})$ for some α, see Appendix K. The log–log plots (bottom row of Figure 12) show no robust straight-line behaviour, ruling out such law for any possible exponent $-\alpha$.

J.2 Critical Batch Size

We can characterize two regimes for the stabilization levels (see Figure 13):

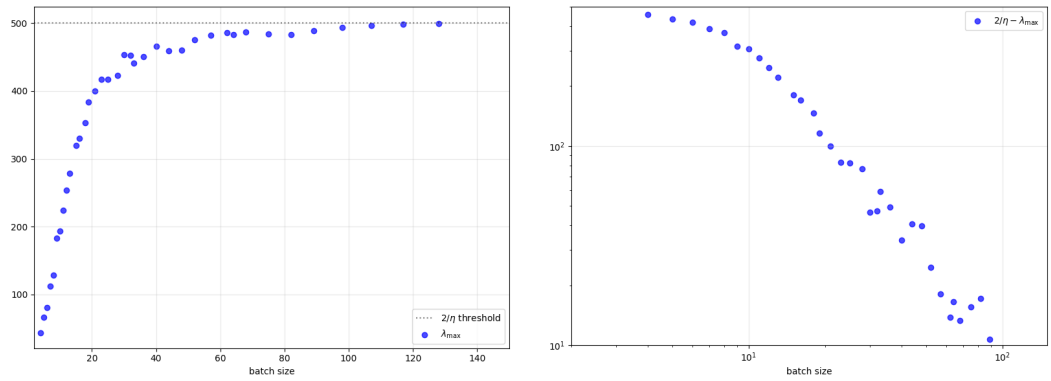
(i) Small-batch regime ($b \le b_c$): $\lambda_{\max}$ stabilizes well *below* the full-batch threshold $2/\eta$, signaling strong implicit regularization by SGD. The stabilization level rises steeply with batch size, so even modest changes in b materially affect the final curvature of the loss landscape of the solution

(ii) Large-batch regime ($b \ge b_c$): the growth of $\lambda_{\max}$ with b becomes much slower and the curve asymptotically approaches $2/\eta$ from below, mirroring full-batch gradient descent and reflecting weak implicit regularization.

The *critical batch size* b_c is therefore the point at which the training dynamics cross over into a full-batch-like regime. Works as Zhang et al. (2024) study the following notion of *critical batch size*: "the point beyond which increasing batch size may result in computational efficiency degradation". Likewise, works focusing on generalization performance depending on the batch size (Masters and Luschi, 2018) identify a cut-off batch sizes above which test performance degrades significantly. We conjecture there may be a relation between these quantities and leave a systematic investigation to future work.

The limited characterization of the level of stabilization of $\lambda_{\max}$ comes from analysis of $\lambda_{\max}^b$:

$$\lambda_{\max}^b := \mathbb{E}_{B\sim\mathcal{P}_b}\left[\lambda_{\max}(\mathcal{H}(L_{\mathbf{B}}))\right].$$



(a) $\lambda_{\max}$ stabilization level (linear) (b) gap to stability threshold (log–log)

Figure 13: **Baseline MLP: stabilization of $\lambda_{\max}$ as a function of batch size**. Baseline MLP (2 hidden layers, width 512) trained on an 8k-subset of CIFAR-10 with step size 0.004 until convergence. **(a)** Final $\lambda_{\max}$ (linear axes). Smaller batches settle to flatter minima. For batch sizes *below* the *critical batch size* b_c the level of stabilization is significantly below the $2/\eta$ level of full-batch, indicating strong *implicit regularization*. Moreover, the curve is steep, making the the final landscape sensitive to the choice of batch size. For *larger* batches ($b > b_c$) the slope flattens and $\lambda_{\max}$ plateaus close to $2/\eta$, so the dynamics resemble full-batch GD, implicit regularization is **weak**. **(b)** Log–log plot of the gap $2/\eta - \lambda_{\max}$, used to test for any power-law decay.

In particular, $\lambda^b_{\max}$ also stabilizes at a level $[2/\eta, 4/\eta]$, see Appendix T. We analyze the level of stabilization $\lambda_{\max}$ through the dependence of the difference $\lambda^b_{\max} - \lambda_{\max}$ on the batch size (see Appendix K and L).

J.3 Why $2/\eta - C/b^\alpha$ fails.

From linear stability analyses near the manifold of minima (Wu et al., 2018; Ma and Ying, 2021; Granziol et al., 2021; Mulayoff and Michaeli, 2024) or random matrix theory (see Appendix K) (together with the fact that we have *Batch Sharpness* stabilize at $2/\eta$) one would expect to have a law of the form $\lambda_{\max} \approx 2/\eta - O(1/b^\alpha)$. Log-log plots of the gap $2/\eta - \lambda_{\max}$ in Figure 13b shows no robust power law (for the lack of any linear dependency), invalidating this prediction (see also Figures 14-20). Importantly, this does not invalidate the findings of those theories, instead showcases the insufficiency of a *static* analysis. Indeed, those estimates are taken from changing the batch size *statically*, without making any training steps. In particular, linear stability analisys does accommodate virtually any law, as long as there is change in alignment between the mini-batch gradients. The fact that the static law does not apply means that there is a change to the alignment also happening. Therefore, as will be discussed further in detail, the fact that these estimates do not apply means that to give faithful description of the loss landscape at convergence one has to undertake an analysis that is path-dependent.

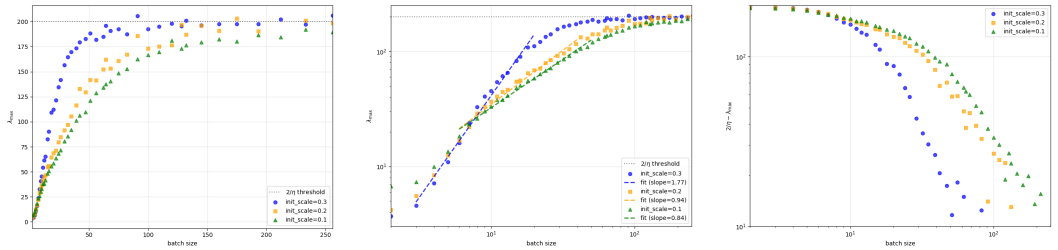
J.4 Conclusion & Outlook: Why Path-Dependence Matters

With all of the above, we arrive at a **negative answer** to the question posed at the start:

> *There is no single, path-independent law that fixes the stabilization level of $\lambda_{\max}$ from basic hyper-parameters alone.*

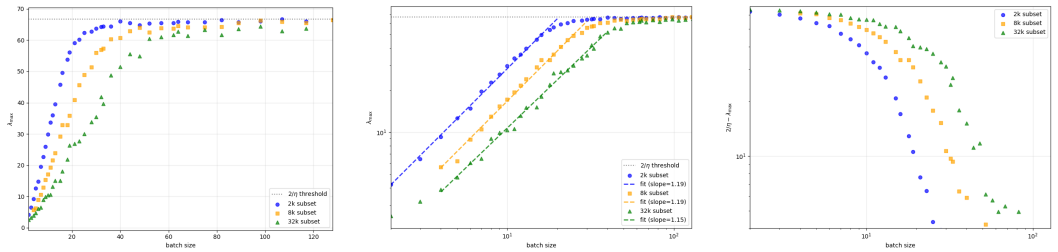
J.5 Implications and Open Questions

The findings above lead to the following main conclusions.



(a) $\lambda_{\max}$ stabilization level (linear) (b) $\lambda_{\max}$ stabilization level (log–log) (c) gap to stability threshold (log–log)

Figure 14: **Effect of weight-scale at initialization on the EoSS stabilization of $\lambda_{\max}$.** We train the same network and dataset under identical hyperparameters, varying only a global rescaling ($\times 0.1, 0.2, 0.3$ of He) of the initial weights. (a) Final-epoch $\lambda_{\max}$ as a function of batch size (linear axes). Smaller batches always converge to flatter solutions, yet the absolute level—and the critical batch size at which the curve begins to approach the full-batch limit $2/\eta$ (horizontal dashed line)—shift markedly with the initialization scale. This demonstrates that the landscape geometry at convergence is already seeded by early-training choices. (b) Same data in log-log scale. The three curves exhibit distinct slopes, ruling out a single power-law exponent and confirming strong path-dependence. Linear fit is provided to the linear portion (c) Log–log plot of the gap $, 2/\eta - \lambda_{\max},$. The absence of a straight line contradicts the prediction $2/\eta - \lambda_{\max} \propto b^{-\alpha}$ that follows from linear stability analyses near a minimum.



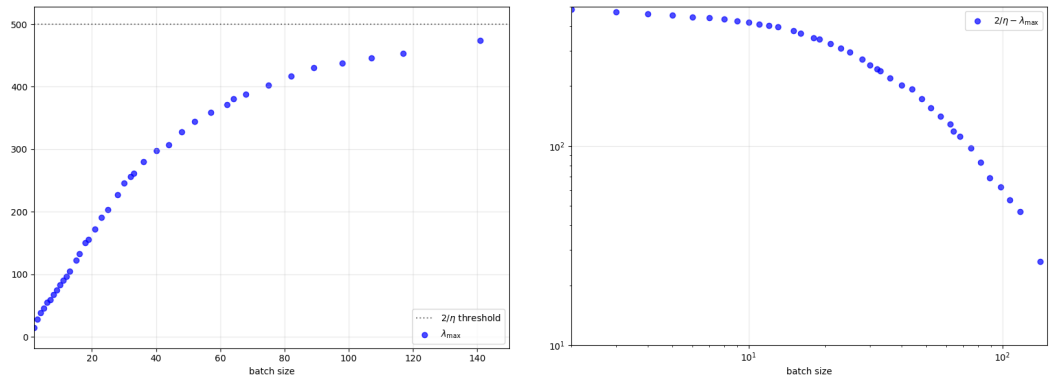
(a) $\lambda_{\max}$ stabilization level (linear) (b) $\lambda_{\max}$ stabilization level (log–log) (c) gap to stability threshold (log–log)

Figure 15: **Varying dataset size alters the EoSS plateau of $\lambda_{\max}$ for a CNN.** We use the same setup as Fig. 14 but instead varying the number of training examples (2k, 8k, 32k). Larger datasets drive $\lambda_{\max}$ to lower plateaus—i.e. flatter minima—and push the critical batch size (the knee toward the full-batch limit $2/\eta$) to higher b, as expected from b/N scaling. Plateau heights also differ from the MLP results in Fig. 14 or 13, highlighting architectural sensitivity. Panel order and axes mirror Fig. 14; see that caption for sub-plot details.

(C1) $\lambda_{\max}$ *is* <u>*not*</u> *the stability limiter for mini-batch training.* *Batch Sharpness* governs EoSS; $\lambda_{\max}$ follows. $\lambda_{\max}$ is capped from above by the value it reaches at the phase transition characterized by *Batch Sharpness* reaching $2/\eta$. This and Facts 1—3 above imply that:

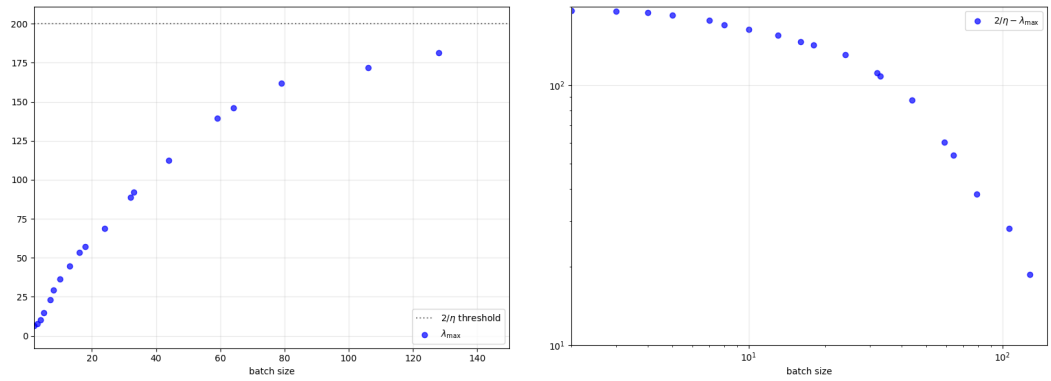
> The stabilization of $\lambda_{\max}$ is a *by-product* of EoSS, not the quantity that governs it.

(C2) *A theory of* $\lambda_{\max}$ *has to account for the* correct *progressive sharpening.* By fixing the model and changing batch size b, the gap between the maximal eigenvalue of $\mathbb{E}\big[\lambda_{\max}(\mathcal{H}(L_B))\big]$ and $\lambda_{\max} = \lambda_{\max}(\mathcal{H}) = \lambda_{\max}\mathbb{E}[\mathcal{H}(L_B)]$ scales as $1/b$, see a proof in Appendix K. Any theory that keeps the parameter vector fixed and only varies b, or anyways leads to a power law, misses the path-dependent descent that determines where training arrives and where $\lambda_{\max}$ stabilizes. Facts 3 and 6 thus imply that analysis of $\lambda_{\max}$ is insufficient *if* it does not account for **(1)** the precise and correct



(a) $\lambda_{\max}$ stabilization level (linear) (b) gap to stability threshold (log–log)

Figure 16: **Level of stabilization of $\lambda_{\max}$.** Same setup as Fig. 13 but the initial weights are rescaled by 1/3; see Fig. 14 for the broader effect of initialization. (See Fig. 13 for sub-plot explanations.)



(a) $\lambda_{\max}$ stabilization level (linear) (b) gap to stability threshold (log–log)

Figure 17: **Level of stabilization of $\lambda_{\max}$.** Identical to Fig. 13 except for a larger step size of 0.01. (See Fig. 13 for sub-plot explanations.)

effect of progressive sharpening on the higher moments of the Hessian and **(2)** the correct alignment between mini-batch steps and Hessians.

Quantifying the plateau of $\lambda_{\max}$ is thus still an (important) open problem. A complete account will require a dynamical theory *through* the progressive-sharpening phase and beyond. Not just properties at its endpoint as for full-batch methods.

Remark: Why not $\lambda_{\max}\mathbb{E}[\mathcal{H}(L_B)]$. One might hope that the largest eigenvalue of each mini-batch Hessian could act as a stochastic proxy for curvature. The reason is that the step is generally not aligned with the eigenvector of the largest eigenvalue and thus $\lambda_{\max}\mathbb{E}[\mathcal{H}(L_B)]$ stabilizes *right above* $2/\eta$, however, the level of stabilization slightly changes with b—see Appendix L—unlike for *Batch Sharpness*.

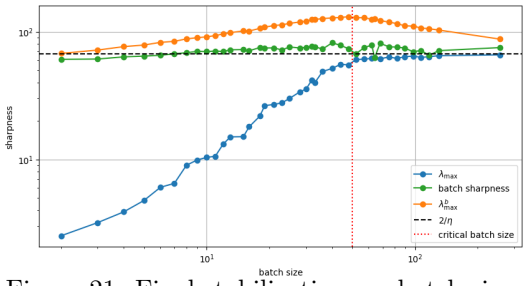
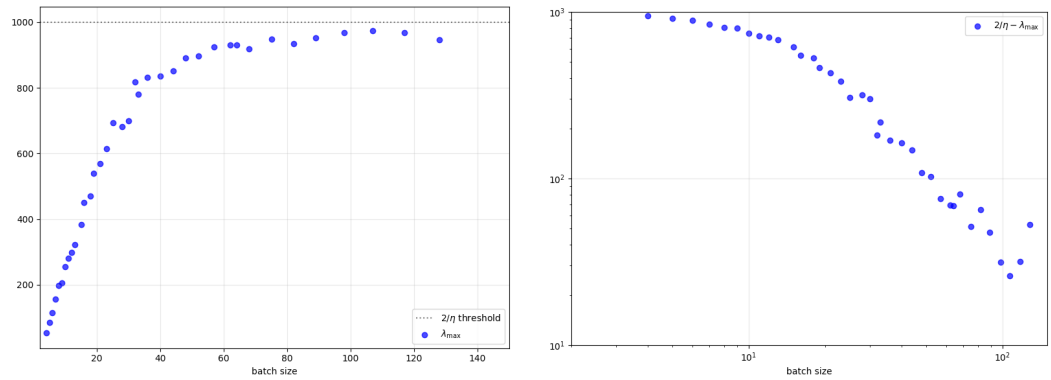
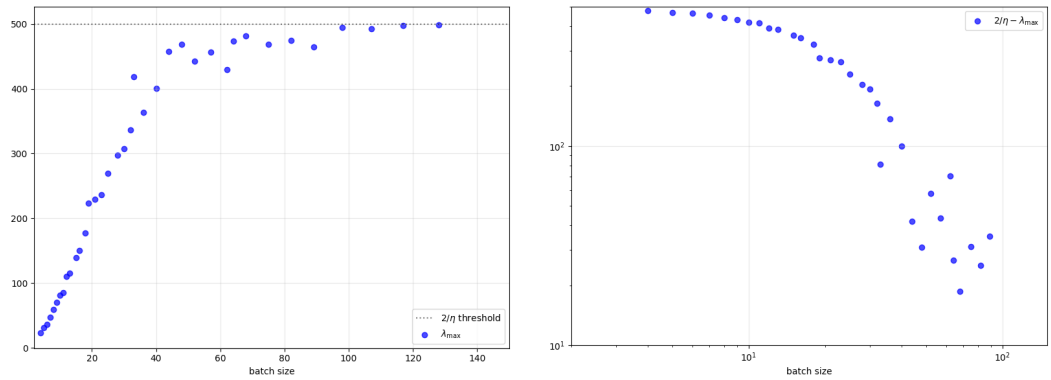


Figure 21: Final stabilizations *vs* batch size.



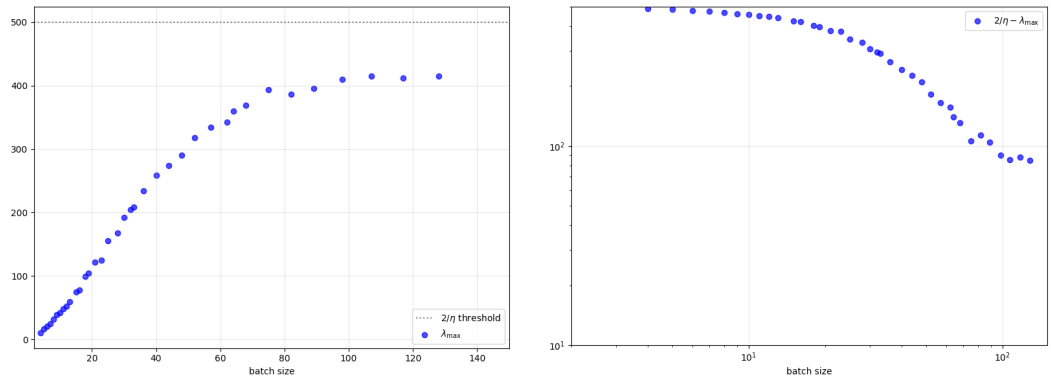
(a) $\lambda_{\max}$ stabilization level (linear) (b) gap to stability threshold (log–log)

Figure 18: **Level of stabilization of $\lambda_{\max}$.** Baseline network trained on a 32k-subset of CIFAR-10 subset with step size 0.002. (See Fig. 13 for sub-plot explanations.)



(a) $\lambda_{\max}$ stabilization level (linear) (b) gap to stability threshold (log–log)

Figure 19: **Level of stabilization of $\lambda_{\max}$.** Deeper MLP (the `mlp_l`: 4 hidden layers, width 512) on the 8k-subset, step size 0.004. (See Fig. 13 for sub-plot explanations.)



(a) $\lambda_{\max}$ stabilization level (linear) (b) gap to stability threshold (log–log)

Figure 20: **Level of stabilization of $\lambda_{\max}$.** Same deeper MLP as in Fig. 19 but trained on a 32k subset. (See Fig. 13 for sub-plot explanations.)

K On Largest Eigenvalues of Sums of Matrices

In this section we establish mathematically how the gap between $\lambda_{\max}^b$ and $\lambda_{\max}$ scales with the batch size. Precisely, what size we can expect from the $\lambda_{\max}^b - \lambda_{\max}$ gap for fixed network.

In particular, the following linear algebra results collectively enhance our understanding of the stability and scaling properties of the largest eigenvalues in the context of matrix sums.

K.1 Ordering the Largest Eigenvalues.

The largest singular value of the Hessian matrix derived from single data points is positive. This observation is crucial in establishing the following well-known property of matrix eigenvalues.

Lemma 6. *Let $m, b \in \mathbb{N}$ and consider m matrices $M_1, M_2, \ldots, M_b \in \mathbb{R}^{m\times m}$ satisfying $\lambda_{\max} > |\lambda_{\min}|$. Then, the largest eigenvalue of their sum satisfies*

$$\lambda_{\max}\left(\sum_{i=1}^{b} M_i\right) \;\leq\; \sum_{i=1}^{b} \lambda_{\max}\left(M_i\right) \tag{51}$$

with equality only if all M_i are identical.

This lemma is a direct consequence of the convexity of the operator norm in matrices and the fact that the largest eigenvalue is positive in our setting. In our setting, it implies that with non-identical matrices, the maximum eigenvalue of the sum is strictly less than the sum of the maximum eigenvalues of the individual matrices. To illustrate, consider eigenvalue sequences for batch sizes that are powers of four, though the result generalizes to any $b_1 < b_2$:

$$\lambda_{\max}^1 \;<\; \lambda_{\max}^4 \;<\; \lambda_{\max}^{16} \;<\; \lambda_{\max}^{64} \;<\; \lambda_{\max}^{256} \;<\; \ldots \tag{52}$$

K.2 Trends of $\lambda_{\max}$ given b

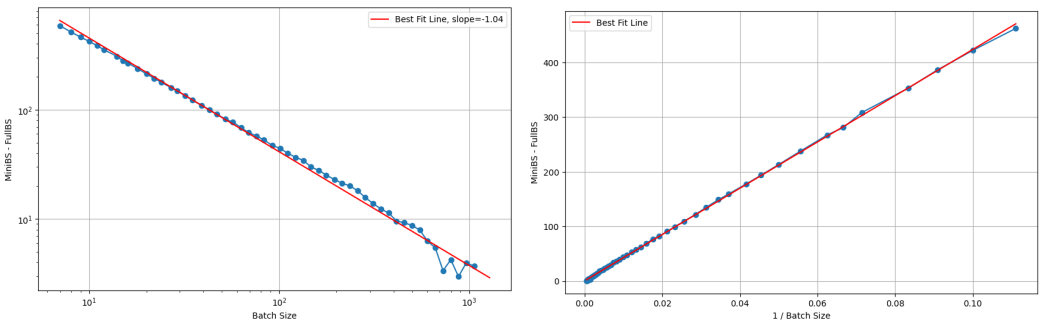


Figure 22: The *static* difference between $\lambda_{\max}^b$ and $\lambda_{\max}$ vs batch_size. The log-log plot is above, indicating the 1/batch_size dependence. The plot with 1/batch_size is below. We fix the parameters of the network at the end of training, and compute the $\lambda_{\max}^b$ using the definition 3. This means that the $\lambda_{\max}$ stays constant, and is subsracted for consistency.

As we discuss in Section L, while the behavior we observe for very small batches is not surprising, for bigger batch sizes it is. In computer vision tasks where there are way more parameters than datapoints, we observe that the *gap between* $\lambda_{\max}^b$ *and* $\lambda_{\max}$ *decreases linearly with* $1/b$ when evaluating the $\lambda_{\max}^b$ of any fixed model with different batch sizes, see Figure 22. This $1/b$ scaling is also what we expect from our mathematical analysis in Appendix K. We are indeed able to establish that for a fixed net, the gap scales as $1/b$ for

small batch sizes, when the worst case $\lambda^b_{\max}$ is very different from $\lambda_{\max}$, and with $1/\sqrt{b}$ for big batch sizes[15]. Indeed:

Proposition 6 (Expected Size of the Average of Matrices)**.** *In the notations of Lemma 6. Under the same assumptions as Lemma 7, the spectral norm of the deviation of the average* $\left\|\frac{1}{b}\sum_i M_i - M\right\|$ *from its expectation* M *satisfies:*

$$\left\|\frac{1}{b}\sum_i M_i - M\right\| \quad = \quad O_b\left(\sqrt{\frac{\sigma^2 \log m}{b}} + \frac{B \log m}{b}\right)$$

where $\sigma^2 = \frac{1}{b}\max\{\|\mathbb{E}[\sum_i M_i^\top M_i\|, \|\mathbb{E}[\sum_i M_i M_i^\top\|\}$ *is the expected second moment of the matrices and* $B \geq \|M_i - M\|$*, for all* i*, is a bound to the biggest random matrix* M_i*.*

K.3 Random Matrix Theory for Scaling Eigenvalues.

While Section J establishes mathematically the order of *Batch Sharpness*, it lacks of mathematical quantification of their magnitudes. Random matrix theory helps bridging this gap at least for big batch sizes b in the Online SGD case where instead of the full-batch Hessian we take as a reference its theoretical expectation.

Lemma 7 (Matrix Bernstein Inequality)**.** *Let* $n_1, n_2, b \in \mathbb{N}$*, let* $M_1, M_2, \ldots, M_b \in \mathbb{R}^{n_1 \times n_2}$ *be independent random matrices satisfying* $\mathbb{E}[M_i] = M$ *and* $\|M_i - M\| \leq B$ *for all* i*, let* $v = \max\{\|\mathbb{E}[\sum_i M_i^\top M_i\|, \|\mathbb{E}[\sum_i M_i M_i^\top\|\}$ *then for all* $t > 0$

$$\mathbb{P}\left(\|\textstyle\sum_i M_i - M\| \geq t\right) \quad \leq \quad (n_1 + n_2) \cdot \exp\left(-\frac{b^2 t^2/2}{v + Bbt/3}\right).$$

This lemma provides a probabilistic upper bound on the deviation of the largest eigenvalue as the batch size increases. We now state a proposition that quantifies the expected spectral norm of the average of b matrices M_i based on this inequality.

Proof of 6. The Matrix Bernstein inequality bounds the probability of deviation of $|\sum_{i=1}^b (M_i - M)|$ by t. Rescaling by $1/b$, we see that for the average $\overline{M}_b := \frac{1}{b}\sum_{i=1}^b M_i$ we have

$$\mathbb{P}\left(\left\|\overline{M}_b - M\right\| \geq t/b\right) \quad \leq \quad (n_1 + n_2) \cdot \exp\left(-\frac{b^2 t^2/2}{v + Bbt/3}\right).$$

To bound the expectation $\mathbb{E}[\|\overline{M}_b - M\|]$, we use the following general inequality for random variables X with tail bounds:

$$\mathbb{E}[X] \leq \int_0^\infty \mathbb{P}(X \geq t)\, dt.$$

For $X = \|\overline{M}_b - M\|$, substitute the tail bound:

$$\mathbb{E}[\|\overline{M}_b - M\|] \leq \int_0^\infty (n_1 + n_2) \cdot \exp\left(-\frac{b^2 t^2/2}{v + Bbt/3}\right) dt.$$

We now, introduce a substitution to handle the exponential term. Let:

$$z = \frac{b^2 t^2}{v}, \quad \text{so that} \quad t = \sqrt{\frac{vz}{b^2}} \quad \text{and} \quad dt = \frac{1}{2}\sqrt{\frac{v}{b^2 z}}\, dz.$$

Rewriting the integral in terms of z:

$$\mathbb{E}[\|\overline{M}_b - M\|] \leq (n_1 + n_2) \int_0^\infty \exp\left(-\frac{z}{2 + \frac{Bb}{3\sqrt{\frac{vz}{b^2}}}}\right) \cdot \frac{1}{2}\sqrt{\frac{v}{b^2 z}}\, dz.$$

While this integral is complex in its full form, we focus on the dominant terms by examining the asymptotics Large b:

[15] Although big b means such that the number of directions spanned in the parameter space by the vectors $\nabla_\theta f(\theta, x)$ are repeated multiple times, and that may be practically unrealistic with the current sizes of networks.

- **Variance Contribution:** The v-term dominates when z is small. This leads to a contribution proportional to:
$$O\left(\frac{\sqrt{v\log(n_1+n_2)}}{b}\right).$$
- **Max Norm Contribution**: The B-term dominates when z is large. This leads to a contribution proportional to:
$$O\left(\frac{B\log(n_1+n_2)}{b}\right).$$

Combining these contributions gives:

$$\mathbb{E}[\|\overline{M}_b - M\|] = O\left(\frac{\sqrt{v\log(n_1+n_2)}}{b} + \frac{B\log(n_1+n_2)}{b}\right).$$

Next note that $v = b\cdot\sigma^2$. This concludes the proof of Proposition 6.

□

The proposition indicates that as b increases, the expected deviation of $\overline{M}_b$ from M diminishes, with a leading-order term scaling as:

1. **Variance Decay:** The term $\sqrt{\sigma^2/b}$ reflects how the variance contribution decreases as b increases (similar to $1/\sqrt{b}$ scaling for scalar averages).
2. **Norm Bound Decay:** The term B/b reflects how the worst-case individual matrix norm affects the average.
3. **Logarithmic Dimension Dependence:** The $\log(n_1+n_2)$ factor accounts for the high-dimensional nature of the problem.

L Dependence of $\lambda^b_{\max}$-$\lambda_{\max}$ Gap on the Batch Size

L.1 Highest Eigenvalue of Mini-Batch Hessian

For a batch size b, denote by

$$\lambda^b_{\max} := \mathbb{E}_{B\sim\mathcal{P}_b}\left[\lambda_{\max}(\mathcal{H}(L_{\mathbf{B}}))\right].$$

Then, we establish here that:

- Also $\lambda^b_{\max}$ stabilizes.
- $\lambda^b_{\max}$ stabilizes at a level that ranges between $4/\eta$ and $2/\eta$. The level is lower for very small and very large batch sizes.
- $\lambda^b_{\max}$ is always greater than $\lambda_{\max}$.
- We find that the level at which $\lambda_{\max}$ stabilizes is characterized by two different regimes. The threshold is what we call *critical batch size*. This critical batch size depends on the complexity of the data-model.
- However, the gap between $\lambda^b_{\max}$ and $\lambda_{\max}$ typically goes as $1/b^\alpha$ generally with $\alpha = 0.70$. This is surprising, as when fixing the network, the gap goes as $1/b$ or $1/b^{1/2}$. See Appendices K, L

Average $\lambda^b_{\max}$ for the mini-batch Hessians We establish that $\lambda^b_{\max}$–the average over the possible batches sampled of the highest eigenvalue of the mini-batch Hessian–generally stabilizes at a value just bigger than $2/\eta$. This happens on a wide range or models and datasets. Refer to Appendix T

We observe that its stabilization levels is generally very close to $2/\eta$ for very small batch sizes, it increases until the critical batch size, then it decreases again.

In this appendix, we further discuss the dependence of the gap between $\lambda^b_{\max}$ and $\lambda_{\max}$ from the batch size. As mentioned earlier, there are two distinct cases - the *static* case, and the "trained" case. In the former, we fix a model at some point of the training, and vary the batch size. In particular, $\lambda_{\max}$ stays constant, and the only variation comes from $\lambda^b_{\max}$. In the latter, we fix a batch size at the beginning of the training, and look at the $\lambda^b_{\max}$-$\lambda_{\max}$ gap at the end of the training.

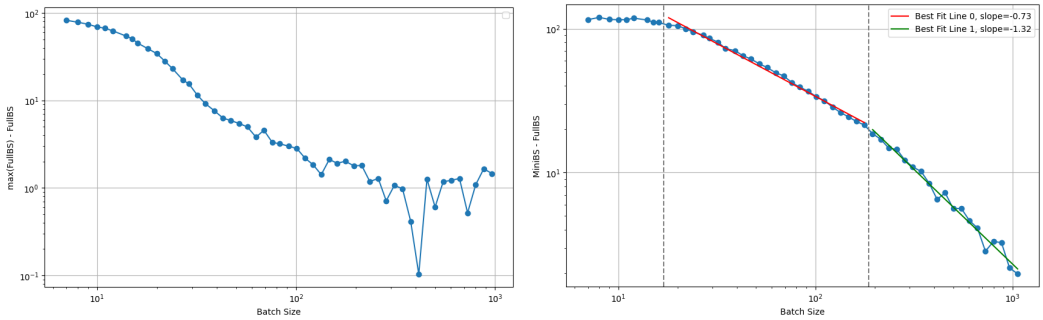


Figure 23: Log-log plots of **(LHS)** the gap between the maximum reached by the $\lambda_{\max}$ and the $\lambda_{\max}$ vs batch size; **(RHS)** $\lambda^b_{\max}$-$\lambda_{\max}$ gap vs batch size.

L.2 The *static* case

For the *static* case, we first look at a network at the end of training. The network is a preceptron with two hidden layers of dimension 512, trained with batch size of 256 on a 8k subset of CIFAR-10 to convergence. Log-log plot in Figure 24 confirms an approximate 1/batch_size dependence.

One can notice that at high batch sizes the observed slope is somewhat bigger (-1.1 if fitted to the $\lambda^b_{\max} - \lambda_{\max}$ computed for batch sizes larger than 1000). Now, for a dataset of size 8k, a batch size of 1000 constitutes 1/8 of dataset, and thus has the $\lambda^b_{\max}$ very close to the $\lambda_{\max}$. This might potentially reveal a different scaling regime for batch sizes that are closer to dataset size. On the other hand, since the difference between $\lambda^b_{\max}$ and $\lambda_{\max}$ becomes increasingly small when batch size approaches dataset size (especially in comparison to the value of each: $\lambda^b_{\max} - \lambda_{\max}$ being of order of 1, and each being around 500), the change in scaling might just be an effect of noise in estimating the highest eigenvalue of the hessian. Lastly, there is the effect of finiteness of the dataset size - that is, that the $1/b$ dependence would turn to 0 only when b is infinite, although in reality the gap would be 0 when b is equal to the dataset size. This dependence might effectively break the scaling. Answering the above questions necessitates further investigation. Nonetheless, the $1/b$ dependence appears to persist within the 'realistic' SGD regime, characterized by batch sizes that are substantially smaller than the dataset size.

The $1/b$ scaling appears to hold throughout the training. In particular, it also applies at initialization, as showcased in the log-log in Figure 25.

Moreover, the $1/b$ dependence is also architecture-independent. As illustrated in the log-log plot in Figure 26, it is also the case for a CNN architecture at convergence.

L.3 The *trained* case

As illustrated in Figure 27, the 1/batch_size dependence breaks down in the trained case, holding only within specific ranges of batch sizes. Specifically, for batch sizes in the range $[10, 100]$ the gap appears to scale as $1/b^{0.7}$. Meanwhile, for batch sizes in $[100, 1000]$, the gap scales as $1/b$. The corresponding regimes are depicted in Figure 28 and in Figure 29.

Similar to the static case, we again see that the anomalous region at batch sizes that are larger than 1/8, requiring further investigation. A distinct scaling regime emerges for very small batch sizes (< 10), differing from the patterns described above. In this regime, the

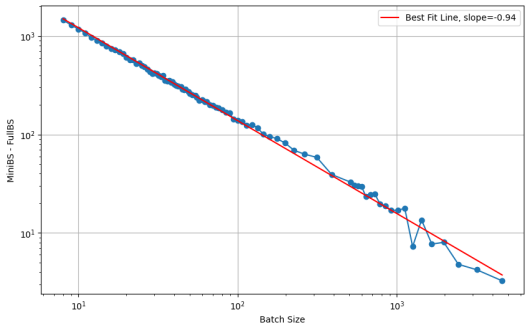


Figure 24: Log-log plot of the $\lambda^b_{\max}$-$\lambda_{\max}$ for *fixed* model at convergence.

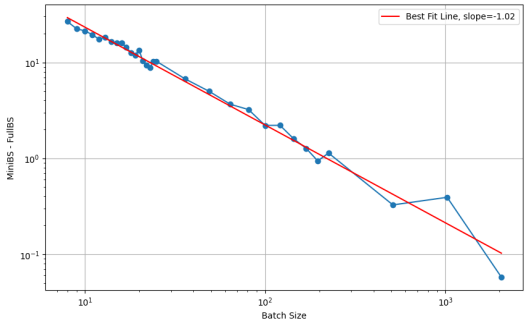


Figure 25: Log-log plot of the $\lambda^b_{\max}$-$\lambda_{\max}$ for a model at initialization

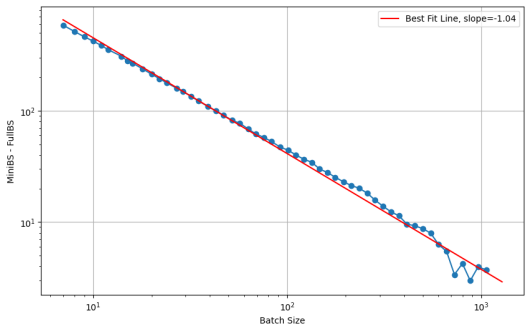


Figure 26: Log-log plot of the $\lambda^b_{\max}$-$\lambda_{\max}$ for *fixed* CNN model at convergence.

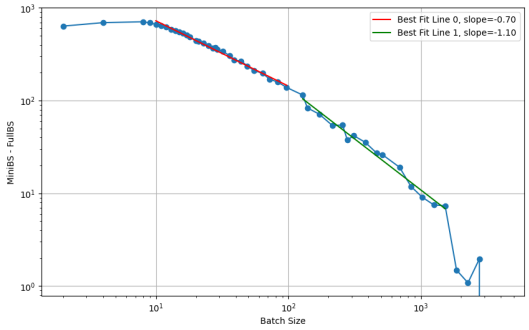


Figure 27: Log-log plot of $\lambda^b_{\max}$-$\lambda_{\max}$ gap vs batch size at the EoSS. Notice how the scaling breaks for very small and very large batch sizes

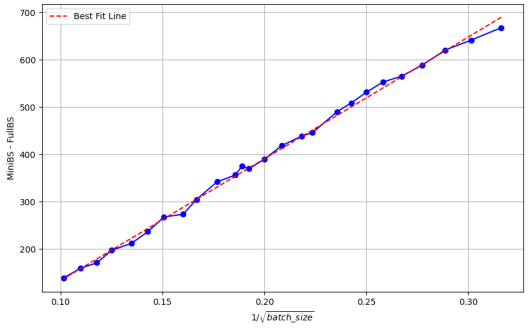


Figure 28: $\lambda^b_{\max}$-$\lambda_{\max}$ gap vs $1/\sqrt{\text{batch\_size}}$ at the EoSS, for batch sizes in $[10, 100]$.

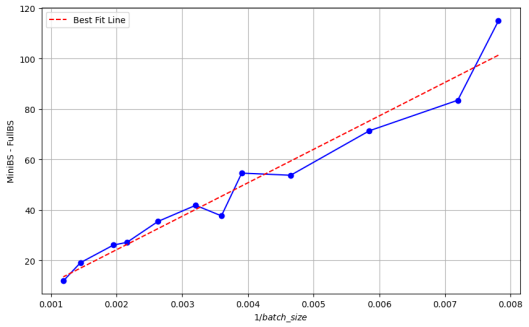


Figure 29: $\lambda^b_{\max}$-$\lambda_{\max}$ gap vs 1/batch_size at the EoSS, for batch sizes in $[100, 1000]$.

gap appears largely independent of the batch size. This anomaly might arise because, at such small batch sizes, the $\lambda^b_{\max}$ starts at levels at or beyond the EoSS level, bypassing the standard progressive sharpening phase and instead entering a regime where the $\lambda_{\max}$ decreases. Further investigation is necessary to rigorously characterize the scaling behavior in this regime.

M Implications: How Noise-Injected GD Differs from SGD

SGD vs. Noisy Gradient Descent. A common belief is that SGD's regularization stems from its "noisy" gradients, which find flatter minima. However, our analysis points to the "noisy" Hessians as crucial. To test this, we compare mini-batch SGD (batch size 16) against three noisy GD variants: (see details in Appendix M)

- *Anisotropic Sampling Noise:* Gaussian reweighting on the samples (Wu et al., 2020), which is different from SGD but maintains the mini-batch structure (and injects noise in the Hessians).

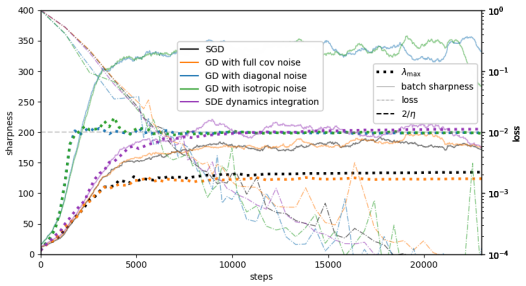


Figure 30: Version of Figure 8, with the loss curves added.

- *Diagonal Noise:* Gaussian noise restricted to the diagonal part of the SGD noise covariance (Zhu et al., 2019).
- *Isotropic Noise:* Gaussian noise with isotropic covariance (Zhu et al., 2019).
- *SDE dynamics integration* (Li et al., 2017)

As shown in Figure 8 and Appendix H, only noise which maintains the higher moments of the Hessian(s) (and thus preserves the mini-batch landscape structure) leads to an EoSS-like regime with $\lambda_{\max}$ stabilizing well below $2/\eta$. More generic (e.g., diagonal or isotropic) noise fails to reproduce this behavior. These experiments suggest that stability thresholds differ fundamentally between mini-batch SGD (governed by *Batch Sharpness*) and noise-injected GD (governed by $\lambda_{\max}$). Notably, these results are consistent with the findings of Zhu et al. (2019)—although their focus is on generalization. Unsurprisingly, in the case in which the noise affects only the gradients—not the Hessians—indeed, EoSS comes for $\lambda_{\max} = 2/\eta$ as for GD (Ma and Ying, 2021; Mulayoff and Michaeli, 2024). Even in the quadratic setting, the appearance of *Type-1* oscillations and *GNI* are not affected by the structure and distribution of the Hessian on the mini-batches, see Appendix C. The stability threshold, however, is affected. It depends on the Hessian's higher moments, see Theorem 1 or (Ma and Ying, 2021; Mulayoff and Michaeli, 2024).

Challenges for SDE Modeling. Classical analyses of neural network optimization often assume a single, static landscape: (*i*) **Online** perspective, modeling each step's gradient as a noisy unbiased estimator of the expected gradient, or (*ii*) **Offline** perspective, treating the dataset as fixed and SGD as noisy GD on the empirical loss. In both views, it is the *full-batch* Hessian that supposedly drives curvature. Our results instead highlight that each update sees a Hessian $\mathcal{H}(L_{\mathbf{B}})$ that generally differs significantly from $\mathcal{H}$, leading to *Batch Sharpness* stabilizing at $2/\eta$ when $\lambda_{\max}$ is smaller.

> Standard SDE—or analogous—approximations of SGD cannot thus describe the location of convergence of SGD or its behavior for neural networks under the assumption of progressive sharpening. Indeed, they typically ignore any statistics of the Hessians except for the mean.

Prior works already note limitations of SDE-based approaches for SGD implicit regularization: they may be mathematically ill-posed (Yaida, 2018), fail except under restrictive conditions (Li et al., 2021), converge to qualitatively different minima (HaoChen et al., 2020), or miss higher-order effects (Damian et al., 2021; Li et al., 2022). Recent discrete analyses (Smith et al., 2021; Beneventano, 2023; Roberts, 2021) attempt to address some of these issues. Nonetheless, our findings expose a deeper gap: when batch sizes are small, the *geometry of the mini-batch Hessian* differs markedly from that of the full-batch, altering both eigenvalues and eigenvector alignments. Conventional SDE models, which assume a static or average Hessian, cannot easily capture these rapid fluctuations.

M.1 Noisy GD

We are running a number of noisy GD implementations.

M.1.1 Noisy GD with Anisotropic Noise (Gaussian Resampling)

This version of noisy GD essentially preserves the mini-batch landscape structure by averaging the landscapes using Gaussian sampling noise. In particular, it takes a Gaussian sampling vector with the same first and second moments as the sampling vector of SGD. Now, this trivially forces the expectation of the mini-batch Hessians to be the same between SGD and Gaussian resampling (and essentially equal to the full-batch Hessians. Importantly, though, this also makes the covariance of the mini-batch Hessians to be the same between SGD and GD with Gaussian resampling noise (as per linearity of the mini-batch Hessians in the weights of the sampling vector). Together with the fact that GD with the Gaussian resampling behaves in the same manner as SGD from the point of view of stability, *Batch Sharpness*, and suppression of $\lambda_{\max}$—it is an indicator of the fact that it is the higher moments of the mini-batch Hessians that determine the dynamics SGD; and it is indeed the noise in the Hessians that creates the instability regime of EoSS and its consequences. As a weaker consequence, it also preserves the covariance of the noise of SGD.

For implementation details refer to Wu et al. (2020). In summary, we re-draw the sampling vector at each step with the corresponding covariance.

M.1.2 Noisy GD with Diagonal Noise

This implementation follows Zhu et al. (2019) — it recreates what they refer to as "GLD diagonal". This is essentially noisy GD with the noise covariance being equal to the diagonal of the covariance of the noise produced by SGD. This preserves each parameter's marginal variance while ignoring off-diagonal correlations. Conceptually, we are approximating SGD's noise by $\mathcal{N}\left(0, \frac{1}{b}\text{diag}(\Sigma(\theta)\right)$ and add it to the full-batch gradient before the optimizer step. Essentially, this is one step further from a true SGD then the aforementioned GD with anisotropic noise. In particular, it does not preserve the mini-batch landscape structure. As a result, the behavior of GD with diagonal noise differs from SGD from the point of view of $\lambda_{\max}$ stabilizing below $2/\eta$, and instead stabilizing at $2/\eta$. We refer the reader to (Zhu et al., 2019) for the details of implementation. In our implementation, we compute the diagonal of the covariance every 30 steps and reuse it on those 30 steps (as it is too computationally expensive to compute it at every step).

M.1.3 Noisy GD with Isotropic Noise

This implementation follows Zhu et al. (2019) — it recreates what they refer to as "GLD dynamic". This is essentially noisy GD with the noise covariance being identity (hence the "isotropic"), scaled such that the magnitude of the noise conincides with that of SGD. That is, this is isotropic gradient noise that matches the average variance of SGD noise but ignores both parameter-wise variability and correlations. Conceptually, we are approximating SGD's noise by $\mathcal{N}\left(0, \frac{\sigma^2}{b} I\right)$ add it to the full-batch gradient before the optimizer step, where $\sigma^2 = \frac{\text{tr}(\Sigma)}{d}$ is the mean per-parameter variance from the per-sample gradient covariance Σ, b is the target batch size, and d is the number of parameters. This is one step "further" from SGD then the noisy GD with diagonal noise. Consequently, this sort of noisy GD does not preserve the regularization effect of SGD on $\lambda_{\max}$ either.

M.2 SDE

We are taking the standard SDE approximation of SGD: (see e.g. Li et al. (2018))

$$d\theta_t \;=\; -\nabla f(\theta_t)\, dt \;+\; \sqrt{\eta}\; \Sigma^{1/2}(\theta_t)\, dW_t$$

where dW_t is the standard d-dimensional Wiener process, and Σ is the covariance matrix of mini-batch gradients.

To simulate its dynamics, we are using the Euler–Maruyama discretization with a step size of 0.0005, chosen to be sufficiently small compared to η (1/20th of $\eta = 0.01$ in this example). In Figure 31 we are showing a number of sample paths of the SDE trajectory illustrate the similarity in the properties of the solutions found by those dynamics – in particular, that

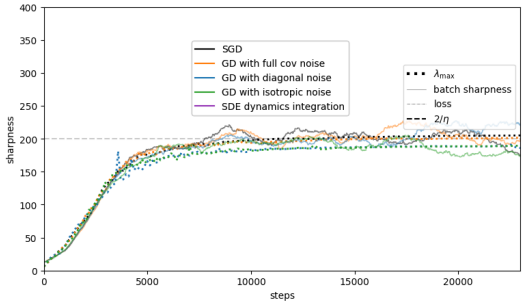


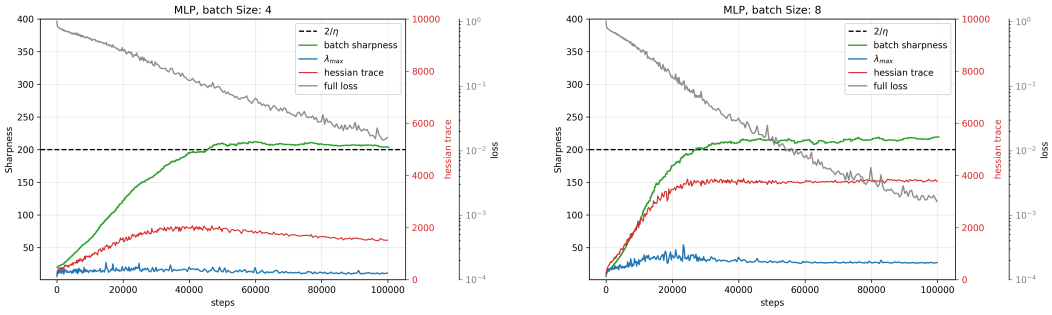
Figure 31: **SDE sample paths** Multiple realizations of SDE trajectory to showcase the similarity of the solutions found by SDE dynamics

$\lambda_{\max}$ stabilizes around $2/\eta$, rather than below as it does for SGD dynamics. In all the experiments, batch size is 16, and η is 0.01.

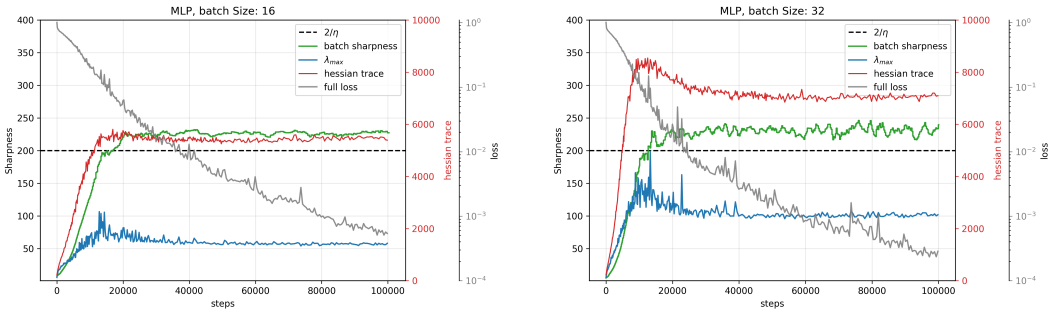
N EoSS and Trace of the Hessian

A number of works Ma and Ying (2021); Wu and Su (2023); Agarwala and Pennington (2024) have linked the trace of the Hessian to implicit regularization by SGD. We plot in Figure 32 and 33: $\lambda_{\max}$, *Batch Sharpness*, and the trace of the Hessian along the training for a variety of models and batch sizes. We observe here that trace of the Hessian behaves very similarly to the previously studies $\lambda_{\max}$. In particular, it doesn't have a consistent stabilization level, and depends significantly on the batch size—with smaller batch sizes leading to lower stabilization level of the trace (aka flatter solutions). Also analogous to $\lambda_{\max}$, it undergoes progressive sharpening, as long as *Batch Sharpness* is under $2/\eta$. Analogously, the stabilization of *Batch Sharpness* leads to stabilization of the trace. All of this showcases that trace of the Hessian is not the quantity that governs stability of the SGD dynamics.

It is noteworthy that, in the context of MSE loss combined with piecewise-linear activation functions (e.g., ReLU), the trace of the full-batch loss Hessian coincides with the trace of its Gauss–Newton approximation. Furthermore, under MSE loss, the trace of the Gauss–Newton matrix is euqal to the trace of the NTK. Consequently, evaluating the trace of the loss Hessian subsumes these cases.

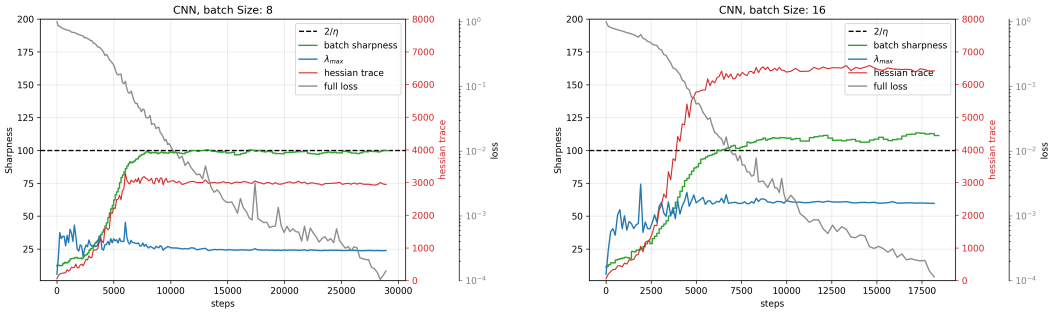


(a) batch size 4 (b) batch size 8



(c) batch size 16 (d) batch size 32

Figure 32: **Trace of the Hessian.** We plot the trace of the full-batch loss Hessian (red), together with the usual *Batch Sharpness* (green) and $\lambda_{\max}$ (blue). Notice that the scale of the trace of the Hessian is much bigger than the rest of the quantities, and it follows the axis on the right (in particular, has no particular relation to $2/\eta$. The plots showcase that trace behaves in a similar manner as $\lambda_{\max}$—its level of stabilization is highly dependent on the batch size, it raises as long as *Batch Sharpness* is rising, and it is stabilizes as batch sharpness stabilizes. Here, we are doing experiments with MLP on CIFAR-10-8k and $\eta = 0.01$



(a) batch size 8 (b) batch size 16

Figure 33: **Trace of the Hessian.** Similar to 32, but for CNN, and with $\eta = 0.02$

O Hardware & Compute Requirements

All experiments were executed on a single NVIDIA A100 GPU (80 GB) with 256 GB of host RAM. The software stack comprises Python 3.12 and PyTorch 2.5.1 (built with the default CUDA tool-chain supplied by the wheel).

Baseline MLP (2M parameters, Section R) Training for 100k steps on the 8 k-image CIFAR-10 subset finishes in ≈ 5 min wall-clock while computing step sharpness every 8 steps, batch sharpness every 128 steps and $\lambda_{\max}$ every 256 steps. Peak device memory

is 14 GB during ordinary training and $\approx$ 70 GB while estimating $\lambda_{\max}$ on a 32k subset, comfortably fitting the 80 GB card.

Algorithmic caveats. We rely on power iteration for $\lambda_{\max}$; while Lanczos would reduce the number of Hessian–vector products, the official PyTorch implementation remains CPU-only. To offset the extra memory incurred by double backward, we cache the first forward pass; batching $\lambda_{\max}$ is left to future work.

P The Hessian and the Fisher Information Matrix Overlap

We show here empirically that at EoSS generally $\lambda^b_{\max}$ generally overlaps with the largest eigenvalue of the averaged mini-batch NTK and $\frac{1}{b} J_B^\intercal J_B$, which corresponds with the FIM in vision classification tasks. See Figure 34.

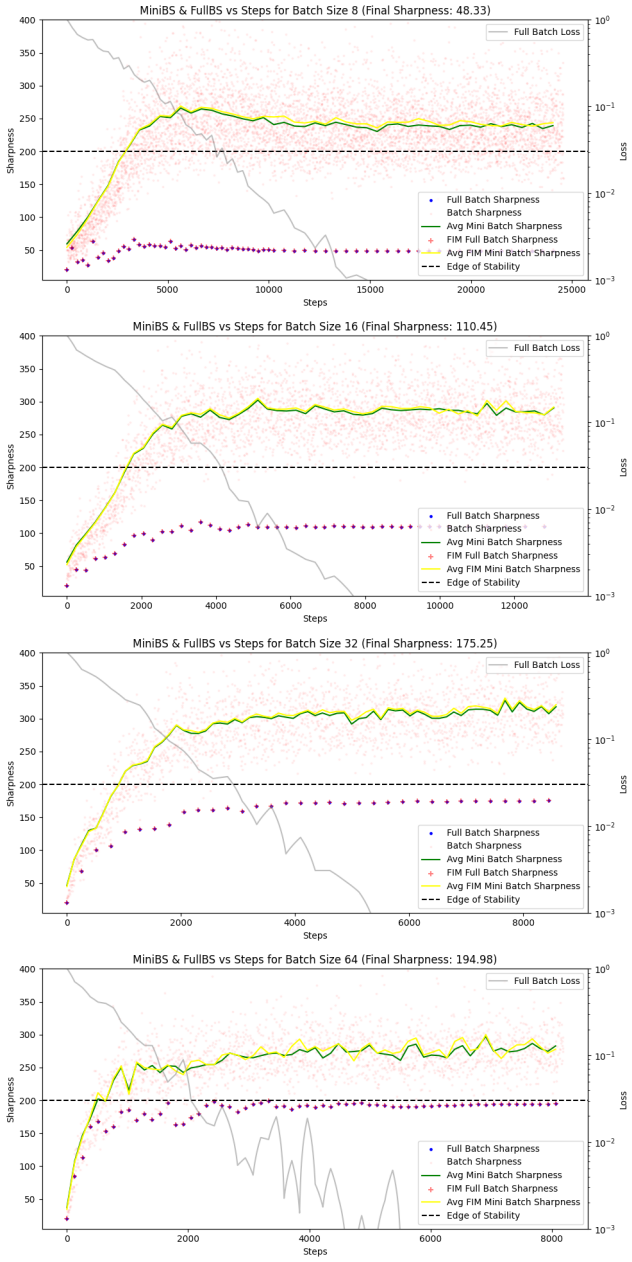


Figure 34: Ranging different batch sizes, the $\lambda_{\max}^{b}$ corresponds to the largest eigenvalue of the averaged mini-batch NTK and $\frac{1}{b} J_B^\top J_B$, which corresponds with the FIM in vision classification tasks.

Q Exemplification Through a Simplified Models

A natural implicit notion of preceived curvature. We give here a qualitative heuristic for which *Batch Sharpness* is a sensible notion of curvature along the step of SGD. When the optimization dynamics occur on a fixed loss landscape, as in GD or noise-injected GD, the stability criterion trivially reduces to examining a single Hessian. Stability in such cases hinges directly on properties of this single Hessian, typically characterized by $\lambda_{\max}$, or other norms. In contrast, mini-batch SGD inherently samples different landscapes–and consequently different Hessians–at every iteration. Thus, it is non-trivial to identify which

statistical properties of these sampled Hessians govern the stability dynamics. Contrary to initial intuition, the average Hessian does not adequately characterize stability[16].

A natural first guess for what statistic reaches the value of $2/\eta$ would be a statistic of the maximum eigenvalue of the mini-batch Hessians, and not the maximum eigenvalue of the averaged Hessian. Indeed, if one has a few outlier Hessians, those may induce divergence even if the average Hessian is small. We also establish that is not the quantity to look at—and is generally bigger than *Batch Sharpness*—in Appendix T. While the landscape on the batch matter, the reason is that the (mis-)alignment of gradients and Hessians plays a bigger role. $\lambda_{\max} = 2/\eta$ is the instability threshold if and only if the landscape is quadratic *or* the step is aligned with the eigenvector of the highest eigenvalue, here this is not the case. *Batch Sharpness* emerges naturally as the relevant measure because it explicitly captures the alignment between the gradient step direction and the curvature of the sampled mini-batch landscape at each iteration, effectively measuring the curvature perceived by SGD on each particular step.

Q.1 Stability cannot Depend on Full-Batch Quantities—Quadratics

We show here with an example that in the mini-batch setting the stability thresholds can not depend only on full-batch Hessian or gradients, but it *has to depend* on the higher moments of them over batch sampling, as we can always construct a counterexample which diverges otherwise.

Imagine on two data points we have $\mathbf{A}_1 = \alpha \cdot I + M$ and $\mathbf{A}_2 = \alpha \cdot I - M$, with $\alpha, \gamma > 0$ and $M = \begin{pmatrix} 0 & 0 \\ 0 & \gamma \end{pmatrix}$, and $\mathbf{b}_1 = \mathbf{b}_2 = 0$. Here $\mathbf{A}$ is αI. $\mathbf{A}_1$ has eigenvalues $\alpha, \alpha + \gamma$ and $\mathbf{A}_2$ has eigenvalues $\alpha, \alpha - \gamma$.

Let us now look at possible full-batch stability results, as developed by Cohen et al. (2021). If the right stability notion for mini-batch SGD depended on the full-batch Hessian or gradients, then it would be independent on γ, and this can not be the case.

Thus a study of the stability of the system can not depend on the full-batch Hessian but has to depend on how big the oscillations due to the size of γ are, i.e., on the higher moments of the distribution of the mini-batch Hessian. Note that this situation would be even more extreme if we had (as in the practice of deep learning) the top eigenvectors of the mini-batch Hessians to point in completely different directions, not just to have high variance. As a sanity check, any stability threshold dependent on the higher moments of the mini-batch Hessian or gradients–as *Batch Sharpness* or $\mathbb{E}_B[\lambda_{\max}(\mathcal{H}(L_{\mathbf{B}})]$–would induce $\eta_{\max}$ to depend on γ too.

Q.2 Diagonal Linear Networks

Consider a simplified scenario involving a diagonal linear network trained on data from two orthogonal classes. Assume $(x, y) \in \mathbb{R}^2 \times \mathbb{R}$ is either $z_1 = \big((1, 0),\ 1\big)$ or $z_2 = \big((0, 1),\ -1\big)$ with probability $1/2$. We learn this data with a diagonal linear network and MSE, precisely where

$$f(x) = a^\top B \cdot x, \qquad a \in \mathbb{R}^2, \quad B \in \mathbb{R}^{2\times 2}.$$

Then with a diagonal initialization, gradient descent will converge almost surely to a neural network of the following kind

$$f(x) = (a_1, a_2) \cdot \begin{pmatrix} b_1 & 0 \\ 0 & b_2 \end{pmatrix} \cdot x, \qquad \text{where} \quad |a_1 \cdot b_1| \ = \ |a_2 \cdot b_2| \ = \ 1.$$

At convergence, the spectrum of the Hessian on the data point z_1 is $\{\lambda_1, 0, 0, 0, 0, 0\}$, with $\lambda_1 := a_1^2 + b_1^2$, the Hessian on the data point z_2 is instead $\{\lambda_2, 0, 0, 0, 0, 0\}$, where $\lambda_2 := a_2^2 + b_2^2$, and the two eigenvectors for these two eigenvalues are orthogonal between each other. This

[16] In Appendices Q.1 and Q.2 we provide two toy examples where is clear that stability has to depend on higher moments or different statistics of the mini-batch Hessians and gradients and not on the full-batch (averaged) quantities.

implies that the Hessian of the full-batch loss has spectrum $\{\lambda_1/2, \lambda_2/2, 0, 0, 0, 0\}$, while the Hessian on the mini batches of size one has either one of the spectra above.

This implies that

$$\lambda_{\max} \quad = \quad \lambda_{\max}\left(\frac{1}{2}\mathcal{H}(z_1) + \frac{1}{2}\mathcal{H}(z_1)\right) \quad = \quad \max\left\{\frac{\lambda_1}{2}, \frac{\lambda_2}{2}\right\} \tag{53}$$

This is smaller than the average largest eigenvalue of the mini-batch Hessian which is

$$\lambda^1_{\max} \quad = \quad \frac{1}{2}\lambda_{\max}(\mathcal{H}(z_1)) + \frac{1}{2}\lambda_{\max}(\mathcal{H}(z_2)) \quad = \quad \frac{\lambda_1}{2} + \frac{\lambda_2}{2}. \tag{54}$$

- **Smaller size:** Thus setting $\lambda_{\max}$ equal to λ means that the max between λ_1 and λ_2 is exactly 2λ. Note that the fact that $a_1 \cdot b_1 = a_2 \cdot b_2 = 1$ and Cauchy-Schwarz imply that $\lambda_1, \lambda_2 \geq 2$. Setting $\lambda^1_{\max}$ to λ thus implies that the maximum between λ_1 and λ_2 is *at most* $2\lambda - 2$, generally smaller.
- **Higher alignment:** Moreover, we have that the gradient $\nabla f(z_i)$ on the data point z_i exactly aligns with the eigenvector v_i of the highest eigenvalue λ_i of the Hessian in z_i. On the full batch, we are averaging them differently, precisely we have that there exist two constants c_1, c_2 such that the gradient is $\frac{c_1}{2}v_1 + \frac{c_2}{2}v_2$. Thus, where WLOG $\lambda_1 > \lambda_2$ we have the alignments

$$\mathcal{H}(z_1) \cdot \nabla L(z_1) \;\sim\; c_1\lambda_1^2 v_1 \qquad \text{but} \qquad \mathcal{H} \cdot \nabla f \;\sim\; \frac{c_1}{2}\lambda_1^2 v_1 \tag{55}$$

  Thus one half of it (batch size divided by number of data points).

This shows that in the same point of the gradient, SGD perceives the largest eigenvalue of the Hessian bigger and more relevant to the gradient then GD.

R Illustration of EoSS in Variety of Settings: *Batch Sharpness*

In this appendix, we provide further empirical evidence that EoSS arises robustly across a variety of models, step sizes, and batch sizes. Consistent with our main observations, we find that *Batch Sharpness* invariably stabilizes around $2/\eta$.

MLP (2-Layer) Baseline. Figure 41 illustrates EoSS for our baseline network, an MLP with two hidden layers of dimension 512, trained on an 8192-sample subset of CIFAR-10 with step size $\eta = 0.004$. As the training proceeds, *Batch Sharpness* stabilizes around $2/\eta$, whereas $\lambda_{\max}$ plateaus strictly below *Batch Sharpness*. Decreasing the step size to $\eta = 0.002$ (see Figure 35) rescales the plateau of *Batch Sharpness* around the new threshold $2/\eta$, in line with the behavior discussed in the main text.

5-Layer CNN. We further confirm the EoSS regime in a five-layer CNN. As depicted in Figures 45 and 36, *Batch Sharpness* continues to plateau near the instability threshold for two distinct step sizes, while $\lambda_{\max}$ once again settles at a lower level. Notably, as we vary the batch size, the gap between *Batch Sharpness* and $\lambda_{\max}$ increases for smaller batches, mirroring the patterns described in Section J.

ResNet-14. Finally, we demonstrate that the EoSS regime also emerges for a canonical architecture commonly used in computer vision: ResNet-14. Note that we are using a version without Batc hNormalization. Figure 40 highlights the same qualitative behavior, with *Batch Sharpness* stabilizing at $2/\eta$.

Overall, these experiments provide further confirmation that EoSS is a robust phenomenon across different architectures, step sizes, and batch sizes.

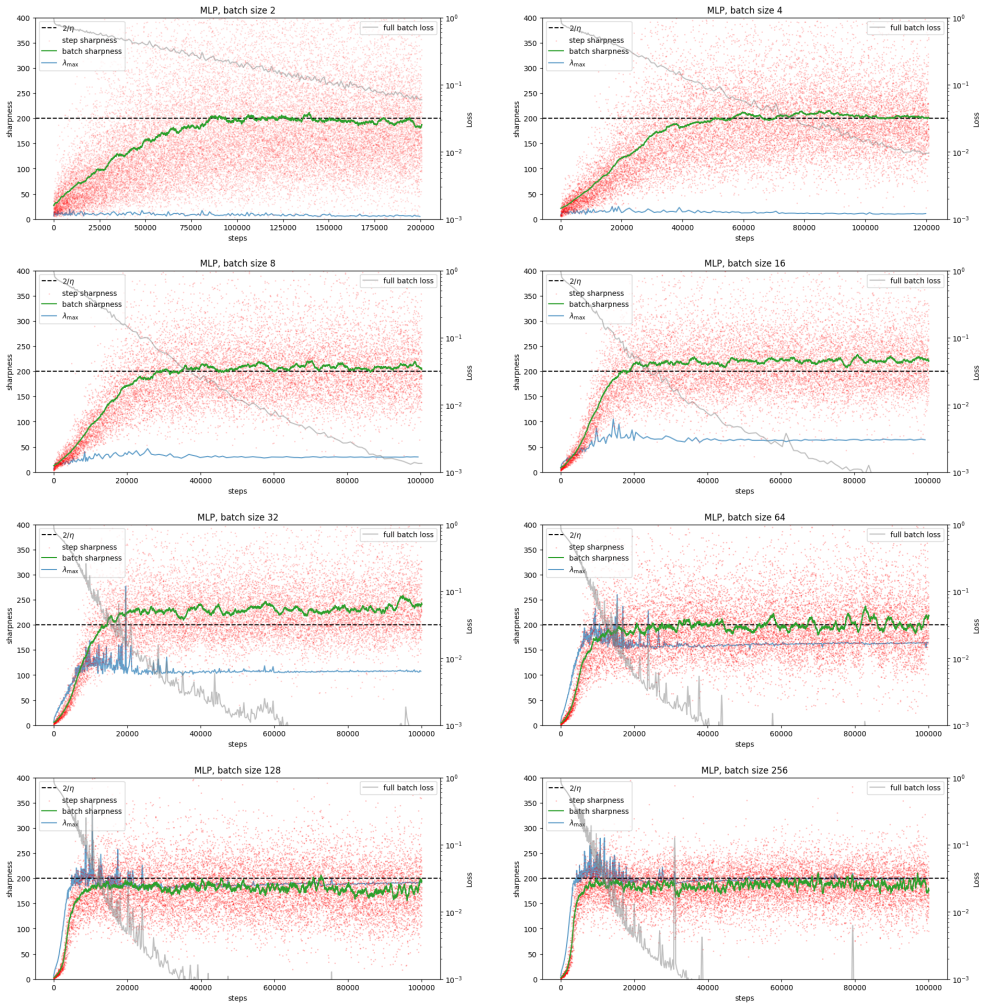


Figure 35: **MLP**: 2 hidden layers, hidden dimension 512; **step size 0.01**, 8k subset of CIFAR-10. Comparison between: the observed highest eigenvalue for the Hessian of the mini-batch loss (red dots), the empirical *Batch Sharpness* (green line), the $\lambda_{\max}$ (blue line).

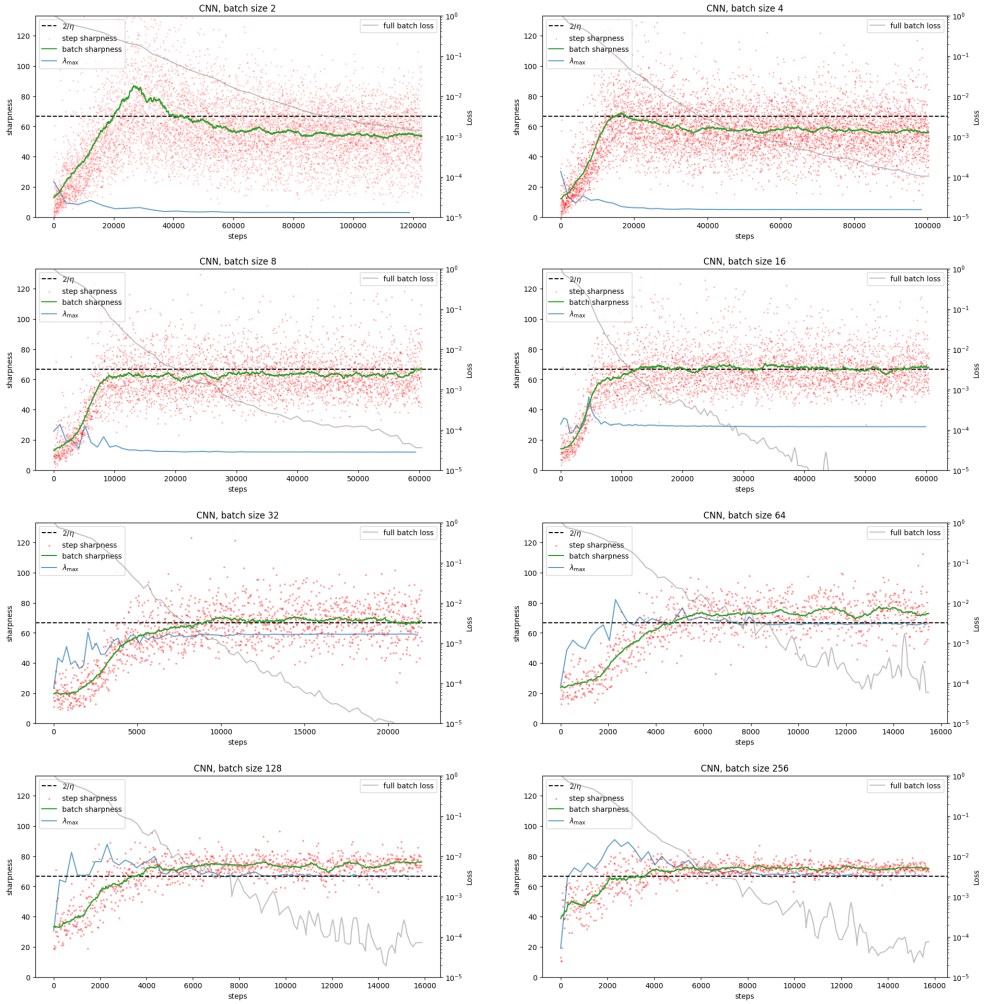


Figure 36: **CNN**: 5 layers (3 convolutional, 2 fully-connected), **step size 0.03**, 8k subset of CIFAR-10. Comparison between: the observed highest eigenvalue for the Hessian of the mini-batch loss (red dots), the empirical *Batch Sharpness* (green line), the $\lambda_{\max}$ (blue line).

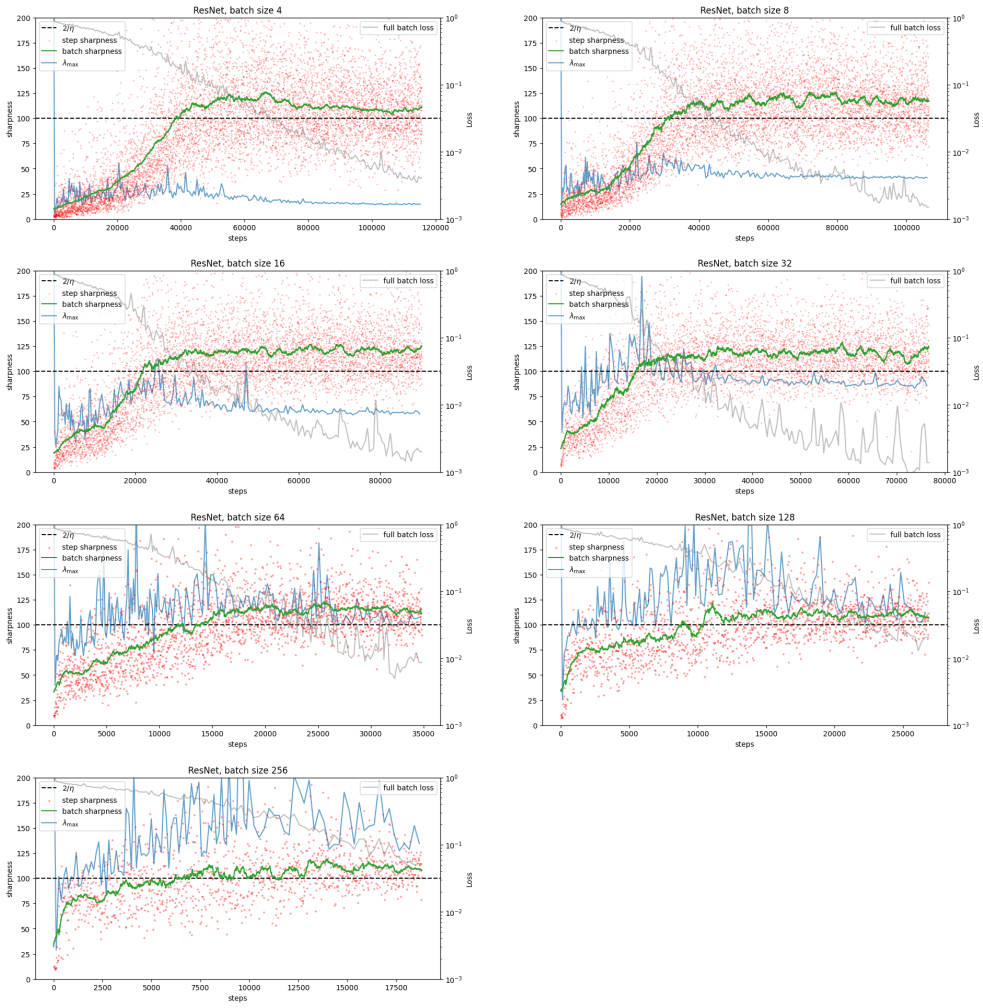


Figure 37: **ResNet-10**, step size 0.005, 8k subset of CIFAR-10. Comparison between: the observed highest eigenvalue for the Hessian of the mini-batch loss (red dots), the empirical *Batch Sharpness* (green line), the $\lambda_{\max}$ (blue line).

S Illustration of EoSS for the SVHN dataset

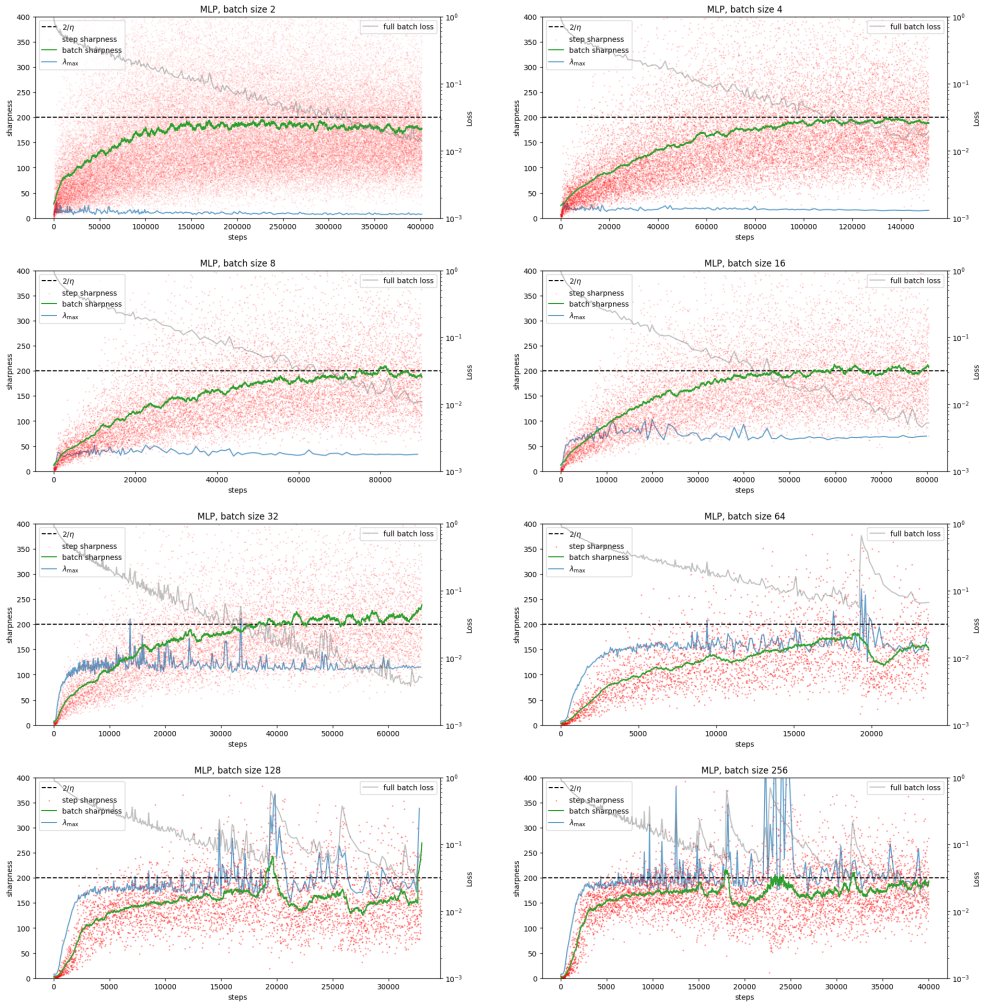


Figure 38: **MLP**: 2 hidden layers, hidden dimension 512; **step size 0.01**, 8k subset of **SVHN**. Comparison between: the observed highest eigenvalue for the Hessian of the mini-batch loss (red dots), the empirical *Batch Sharpness* (green line), the $\lambda_{\max}$ (blue line).

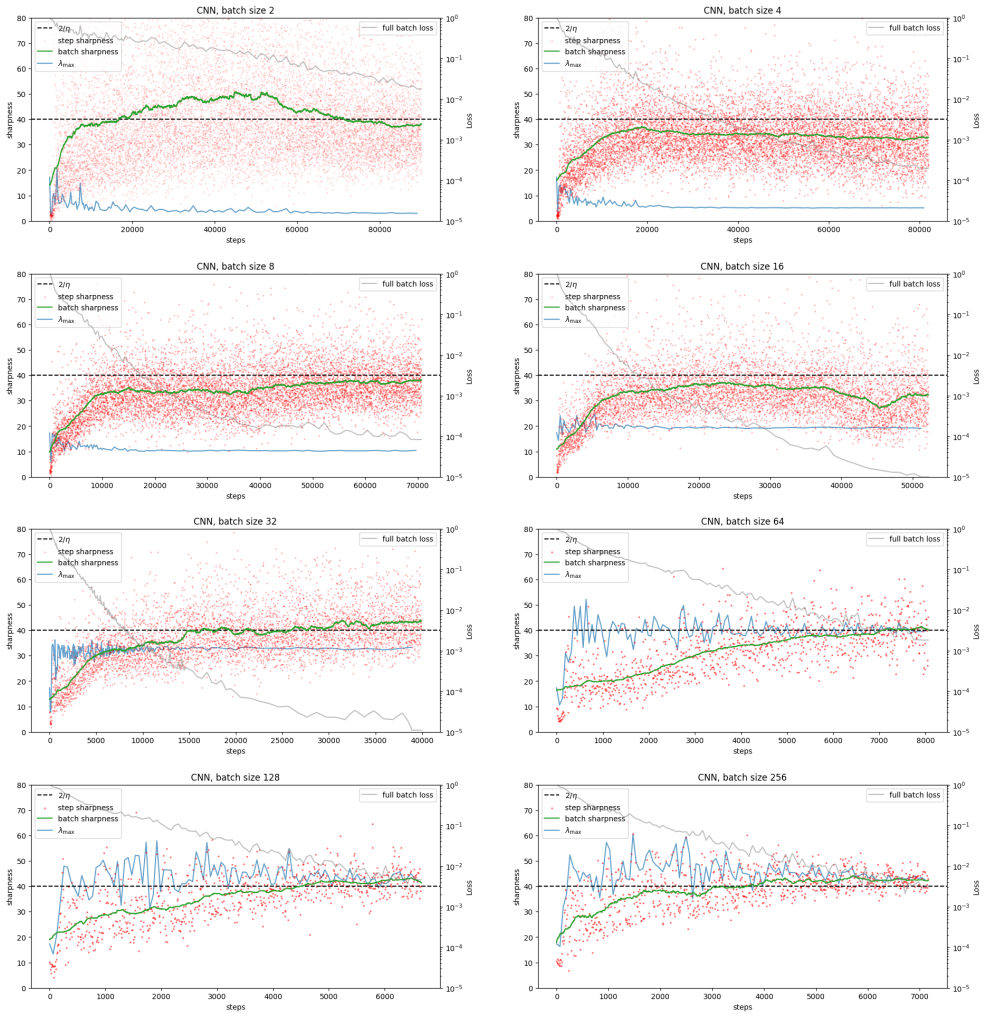


Figure 39: **CNN**: 5 layers (3 convolutional, 2 fully-connected), **step size 0.05**, 8k subset of **SVHN**. Comparison between: the observed highest eigenvalue for the Hessian of the mini-batch loss (red dots), the empirical *Batch Sharpness* (green line), the $\lambda_{\max}$ (blue line).

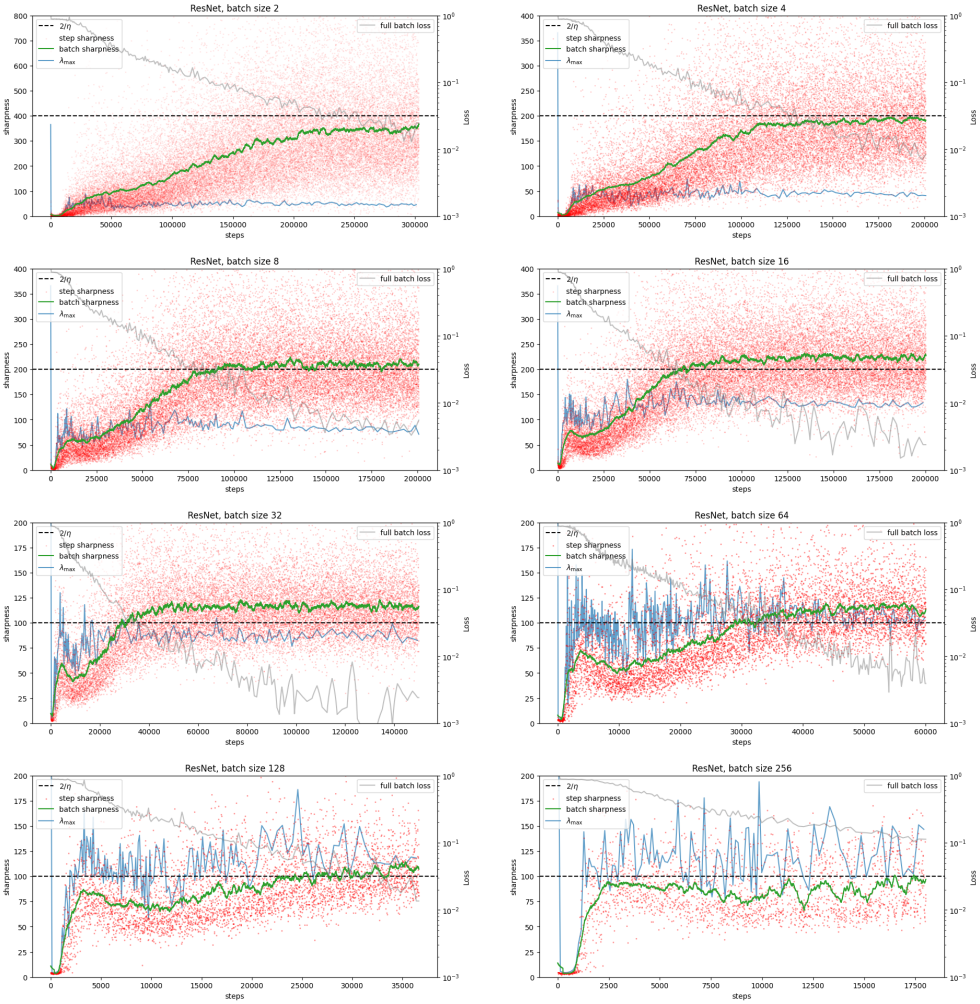


Figure 40: **ResNet-10**, step size 0.005, 8k subset of **SVHN**. Comparison between: the observed highest eigenvalue for the Hessian of the mini-batch loss (red dots), the empirical *Batch Sharpness* (green line), the $\lambda_{\max}$ (blue line).

T Illustration of EoSS in Variety of Settings: $\lambda^b_{\max}$

In this appendix, we provide additional empirical evidence that EoSS emerges robustly across a diverse range of models, step sizes, and batch sizes. Consistent with our primary findings, we observe that $\lambda^b_{\max}$ consistently stabilizes within the interval $\left[2/\eta,\ 2 \times 2/\eta\right]$. Furthermore, we note that the full-batch metric, $\lambda_{\max}$, remains strictly below $\lambda^b_{\max}$, with this gap expanding as the batch size decreases. Crucially, our findings demonstrate that $\lambda^b_{\max}$ lacks a consistent stabilization level, reinforcing that *Batch Sharpness* is the metric that reliably stabilizes around the $2/\eta$ threshold.

MLP (2-Layer) Baseline. Figure 41 illustrates EoSS for our baseline network, an MLP with two hidden layers of dimension 512, trained on an 8192-sample subset of CIFAR-10 with step size $\eta = 0.004$. As the training proceeds, $\lambda^b_{\max}$ stabilizes in the range $\left[2/\eta,\ 2\times 2/\eta\right]$, whereas $\lambda_{\max}$ plateaus strictly below $\lambda^b_{\max}$. Decreasing the step size to $\eta = 0.002$ (see Figure 42) rescales the plateau of $\lambda^b_{\max}$ around the new threshold $2/\eta$, in line with the behavior discussed in the main text.

Deeper MLP (4-Layer). To assess whether increased depth alters the phenomenon, we use a deeper MLP (MLP_L) with four hidden layers, training again on the same CIFAR-10 subset. Figures 43 and 44 show that $\lambda^b_{\max}$ exhibits the same EoSS behavior for two different step sizes, reinforcing that depth alone does not invalidate our findings.

5-Layer CNN. We further confirm the EoSS regime in a five-layer CNN. As depicted in Figures 45 and 46, $\lambda^b_{\max}$ continues to plateau near the instability threshold for two distinct step sizes, while $\lambda_{\max}$ once again settles at a lower level. Notably, as we vary the batch size, the gap between $\lambda^b_{\max}$ and $\lambda_{\max}$ increases for smaller batches, mirroring the patterns described in Section J.

ResNet-10. Finally, we demonstrate that the EoSS regime also emerges for a canonical architecture commonly used in computer vision: ResNet-10. Figure 47 highlights the same qualitative behavior, with $\lambda^b_{\max}$ stabilizing at $\left[2/\eta,\ 2\times 2/\eta\right]$ and $\lambda_{\max}$ remaining consistently below $\lambda^b_{\max}$.

Overall, these experiments provide further confirmation that EoSS is a robust phenomenon across different architectures, step sizes, and batch sizes. Although the specific magnitude of $\lambda_{\max}$ and the exact "hovering" value of $\lambda^b_{\max}$ can vary, the overarching pattern of $\lambda^b_{\max} \approx 2/\eta$ and $\lambda_{\max} < \lambda^b_{\max}$ persists in all our tested settings.

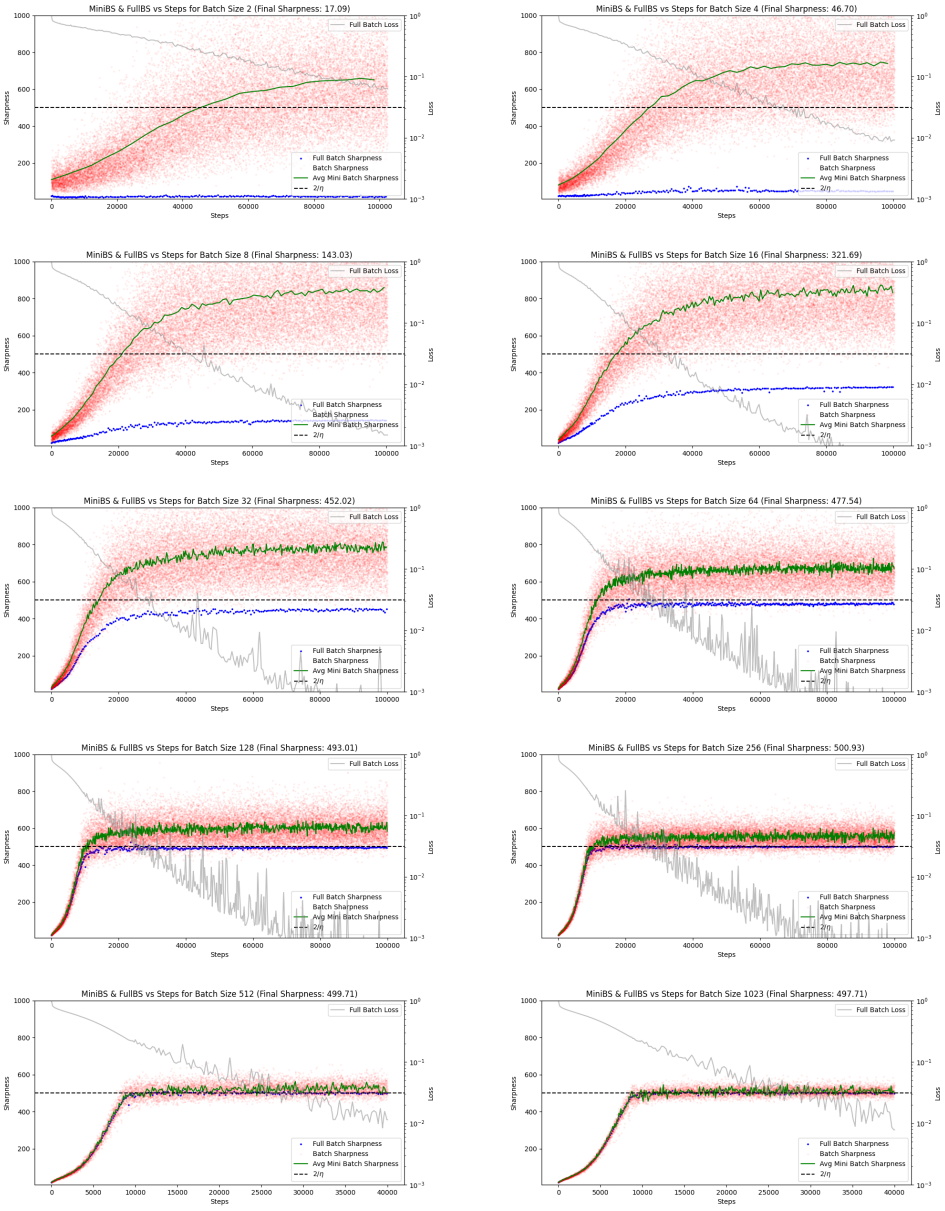


Figure 41: **MLP**, 2 hidden layers, hidden dimension 512, step size 0.004, 8k subset of CIFAR-10. Comparison between: the observed highest eigenvalue for the Hessian of the mini-batch loss (red dots), the empirical $\lambda_{\max}^{b}$ (green line), the $\lambda_{\max}$ (blue dots).

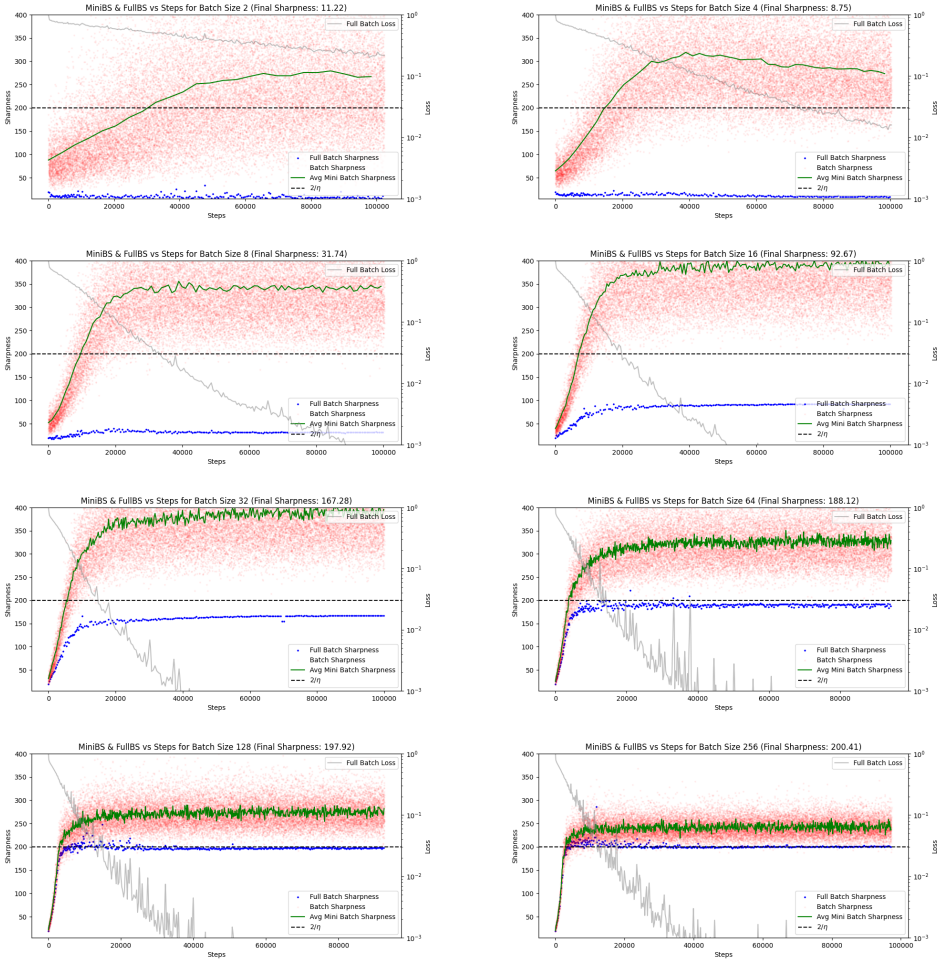


Figure 42: **MLP**: 2 hidden layers, hidden dimension 512; **step size 0.01**, 8k subset of CIFAR-10. Comparison between: the observed highest eigenvalue for the Hessian of the mini-batch loss (red dots), the empirical $\lambda_{\max}^{b}$ (green line), the $\lambda_{\max}$ (blue dots).

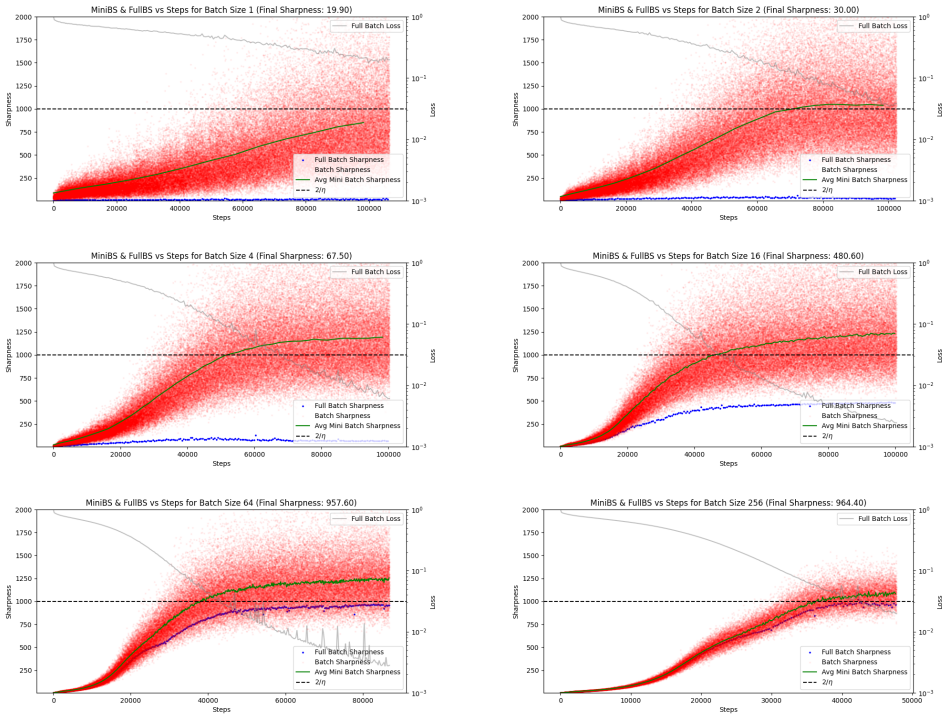


Figure 43: **MLP_L**: **4 hidden layers**, hidden dimension 512, **step size 0.002**, 8k subset of CIFAR-10. Comparison between: the observed highest eigenvalue for the Hessian of the mini-batch loss (red dots), the empirical $\lambda_{\max}^{b}$ (green line), the $\lambda_{\max}$ (blue dots).

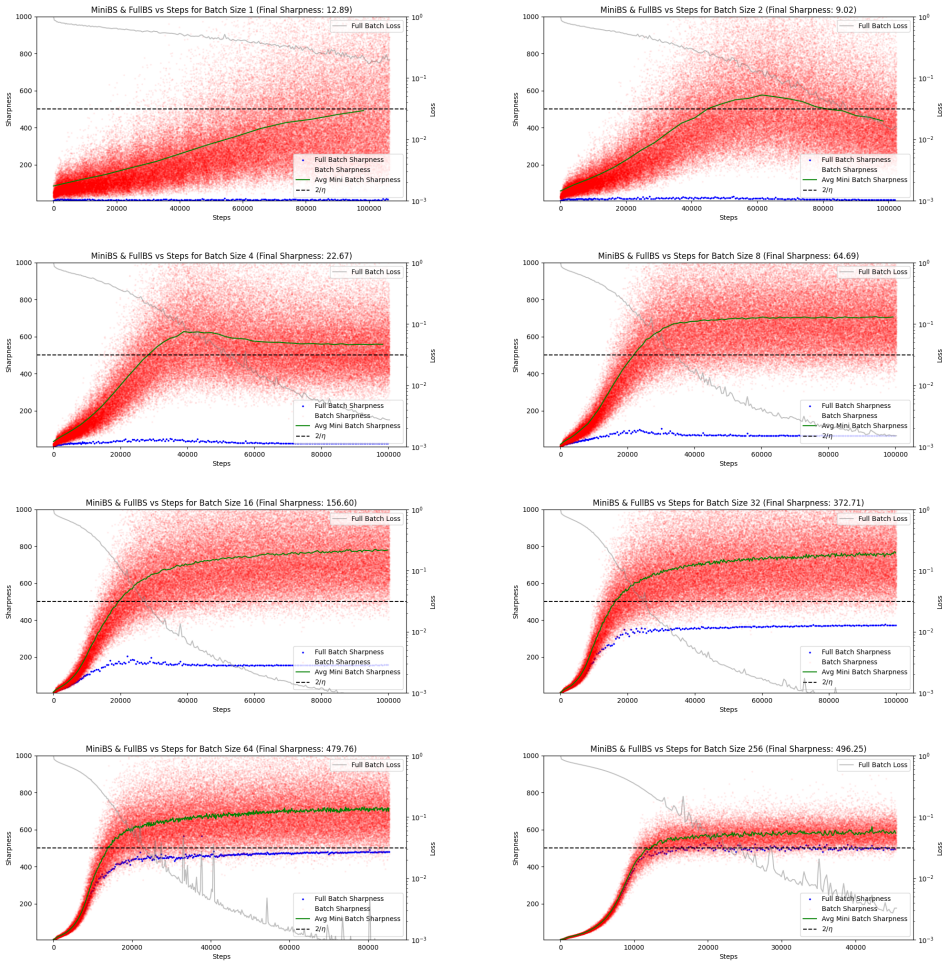


Figure 44: **MLP_L**, **4 hidden layers**, hidden dimension 512, **step size 0.004**, 8k subset of CIFAR-10. Comparison between: the observed highest eigenvalue for the Hessian of the mini-batch loss (red dots), the empirical $\lambda^b_{\max}$ (green line), the $\lambda_{\max}$ (blue dots).

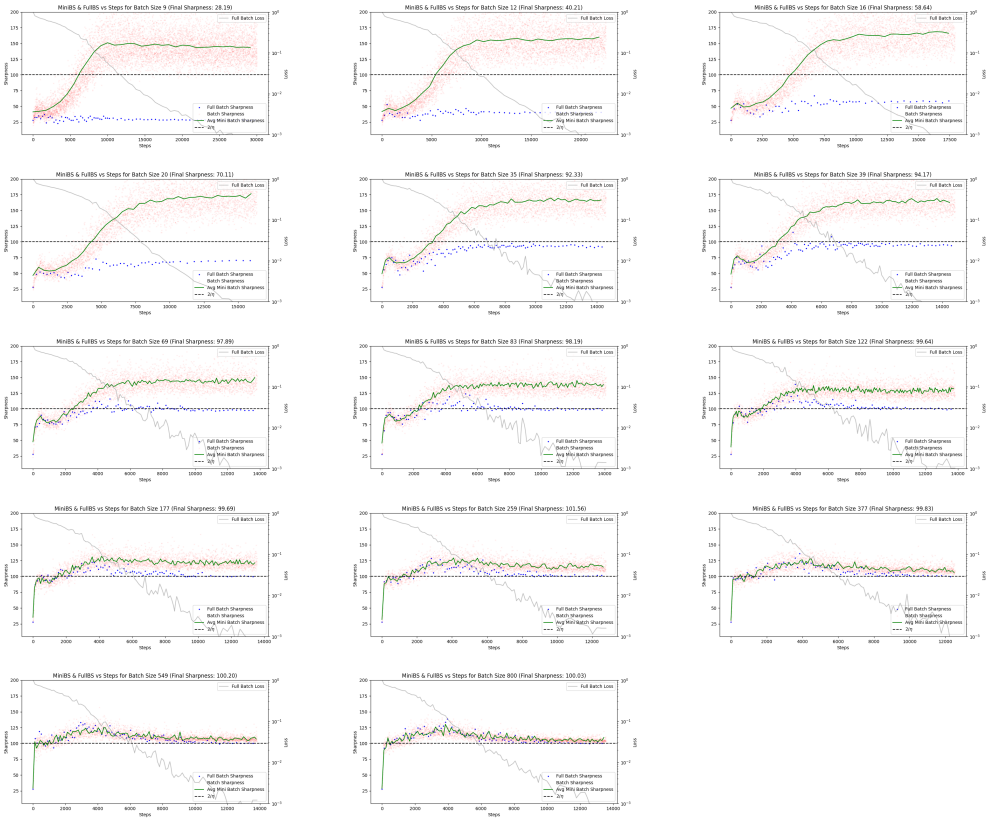


Figure 45: **CNN**, 5 layers (3 convolutional, 2 fully-connected), step size 0.02, 8k subset of CIFAR-10. Comparison between: the observed highest eigenvalue for the Hessian of the mini-batch loss (red dots), the empirical $\lambda_{\max}^{b}$ (green line), the $\lambda_{\max}$ (blue dots).

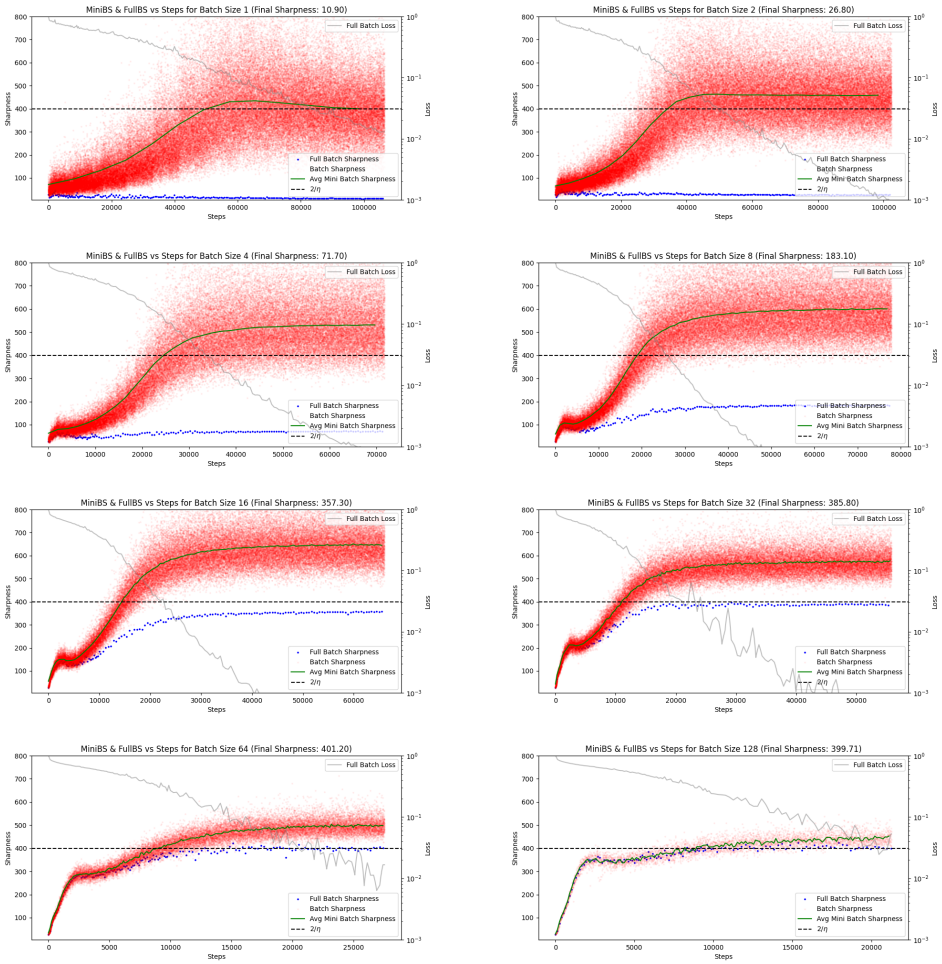


Figure 46: **CNN**, 5 layers (3 convolutional, 2 fully-connected), **step size 0.005**, 8k subset of CIFAR-10. Comparison between: the observed highest eigenvalue for the Hessian of the mini-batch loss (red dots), the empirical $\lambda_{\max}^{b}$ (green line), the $\lambda_{\max}$ (blue dots).

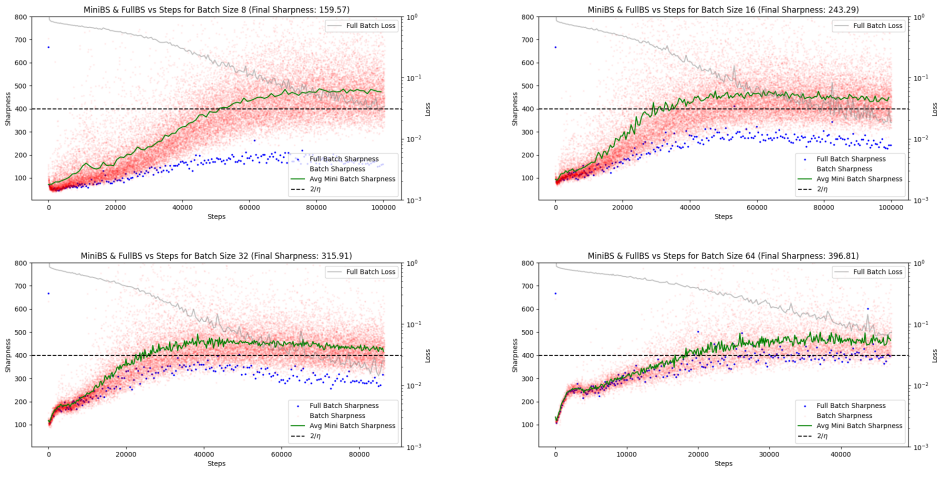


Figure 47: **ResNet-10**, step size 0.005, 8k subset of CIFAR-10. Comparison between: the observed highest eigenvalue for the Hessian of the mini-batch loss (red dots), the empirical $\lambda_{\max}^{b}$ (green line), the $\lambda_{\max}$ (blue dots).