# REEVALUATING THEORETICAL ANALYSIS METHODS FOR OPTIMIZATION IN DEEP LEARNING

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

## ABSTRACT

There is a significant gap between our theoretical understanding of optimization algorithms used in deep learning and their practical performance. Theoretical development usually focuses on proving convergence guarantees under a variety of different assumptions, which are themselves often chosen based on a rough combination of intuitive match to practice and analytical convenience. In this paper, we carefully measure the degree to which the standard optimization analyses are capable of explaining modern algorithms. To do this, we develop new empirical metrics that compare real optimization behavior with analytically predicted behavior. Our investigation is notable for its tight integration with modern optimization analysis: rather than simply checking high-level assumptions made in the analysis (e.g. smoothness), we also verify key low-level identities used by the analysis to explain optimization behavior that might hold even if the high-level motivating assumptions do not. In general, we find that real optimizers often make progress even when typical optimization analysis suggests that they should not. This highlights a need for developing new theoretical frameworks that are better aligned with practice.

## 1 INTRODUCTION

In optimization theory, algorithmic development and analysis requires a set of assumptions about the functions we aim to optimize. These assumptions fundamentally influence the behavior of optimization algorithms and their efficacy in practice. For example, Adagrad (Duchi et al., 2011; McMahan & Streeter, 2010) (which later inspired Adam (Kingma & Ba, 2014)) classically relies on the convexity assumption to provide a theoretical convergence guarantee. When the loss is non-convex, a variety of alternate assumptions are deployed, such as smoothness (e.g. a bounded Hessian) (Ghadimi & Lan, 2013; Carmon et al., 2017; Li & Orabona, 2019; Ward et al., 2020; Wang et al., 2023) or "weak convexity" (Davis & Drusvyatskiy, 2019; Mai & Johansson, 2020; Liu et al., 2023b). If these conditions are not met, the convergence analyses of these algorithms may longer hold. In this paper, we systematically verify these assumptions and related optimization analyses across various deep learning tasks using simple, computationally feasible methods. We hope that our findings will serve as a guideline for future research, helping to develop theoretical frameworks that are both analytically tractable and practically applicable.

Importantly, we do not want to just ask "do current assumptions apply to deep neural networks". Instead, we wish to ask whether the *analyses* based on currently prevalent techniques can predict current practical performance. This is a subtly different question: it turns out that most modern analyses actually rely on a few key identities. These identities are usually *empirically measurable* from the iterates of an optimizer. In theoretical analysis, these identities are controlled via various global assumptions (such as convexity or smoothness), but we instead measure directly these identities. This has a significant advantage: not only can it falsify the global assumptions, it can tell if any *different* assumptions can be made that would "rescue" the analysis.

We propose simple, on-the-fly measures that capture how well modern analyses describe practice. We measure these on a wide range of tasks, including basic convex optimization problems, image classification tasks using deep residual networks, and pre-training large language models (LLMs). Overall, our results suggest that most analytical techniques do *not* describe practical performance. Our work fits into a recent trend of challenging and moving past classical optimization assumptions

Simsekli et al. (2019); Zhang et al. (2020b;a); Davis et al. (2021; 2020). However, our focus is not on algorithm development. Instead, we simply want to promote empirical verification of optimization analysis.

Of independent interest, we develop a new smoothness measure closely approximating the sharpness measure. This is an exciting finding, as our measure is computationally feasible even for very deep networks, where computing sharpness is infeasible. This allows for the use of our smoothness measure in studying flat/sharp minima and their implications for generalization in much larger networks. Finally, we offer alternative theoretical analyses for cases where common theoretical assumptions do not hold.

Overall, we feel that our findings motivate two actions in the research community: first, it is important to develop new assumptions and analytical techniques to understand modern optimization. Second, we advocate for verifying any new assumptions by carefully measuring quantities that *actually appear in the optimization analysis* rather than attempting to verify global assumptions.

## 2 Background and Experiment Setup

In typical optimization analysis for machine learning, the goal is to minimize objective $F$ given by $F(\mathbf{x}) = \mathbb{E}_{z\sim P_z}[f(\mathbf{x}, z)]$, where $f(\mathbf{x}, z) : \mathbb{R}^d \times \mathcal{Z} \mapsto \mathbb{R}$ is a differentiable function of $\mathbf{x} \in \mathbb{R}^d$. $\mathbf{x}$ indicates the model parameters, $z \in \mathcal{Z}$ indicates an example data point or minibatch of examples, and $P_z$ is some data distribution. The function $F$ represents either a train loss or a population loss depending on various details of the problem setup.

The most common paradigm in optimization analysis is the following three-step strategy: first, identify a "convergence criterion" of interest - for example the loss of some weights output by an algorithm minus the loss of the optimal weights. Second, identify an algebraic expression that can be related to this convergence criterion (often through use of some assumption on the loss landscape). Finally, establish that a given algorithm can guarantee a bound on this algebraic expression (often again using some assumption on the loss landscape):

$$\underbrace{\text{Convergence Criterion}}_{\text{e.g. } \frac{1}{T}\sum_{t=1}^{T} F(\mathbf{x}_t) - F(\mathbf{x}_\star)} \leq \underbrace{\text{Algebraic Expression}}_{\text{e.g. } \frac{1}{T}\sum_{t=1}^{T} \langle \nabla F(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_\star \rangle} \leq \underbrace{\text{Upper Bound}}_{\text{e.g. } O(1/\sqrt{T})} \tag{1}$$

The example values for the three terms above are typical of analysis of SGD for convex objectives, in which $\mathbf{x}_\star = \operatorname{argmin} F$, and the middle "algebraic expression" is often termed the *regret* (see Orabona (2019); Hazan (2022) for details).

This paradigm is used in two different ways: first, from a *scientific* perspective, one can try to prove convergence properties for well-known algorithms such as AdamW to explain why these algorithms work well in practice (see e.g. Li & Orabona (2019); Faw et al. (2022); Ward et al. (2020); Zaheer et al. (2018b); Reddi et al. (2019)). Second, from an *engineering* perspective, one can try to design better optimizers from first principles. For this second use-case, the typical approach is to identify a large class algorithms, such as SGD parametrized by the learning rate, and then choose the member of this class that analytically minimizes the upper bound (see e.g. Duchi et al. (2010); McMahan & Streeter (2010); Hazan et al. (2007); Ghadimi & Lan (2013)). This exact approach is how the AdaGrad family of algorithms (which was the intellectual precursor to Adam) was developed.

In order for this paradigm to provide meaningful answers, we must believe that the inequalities in equation (1) hold at least approximately. We can investigate this from two angles: first, we can ask whether the original assumptions that motivated the analysis hold. Second, we can often *empirically measure* expressions related to those appearing in (1), and check the degree to which the desired inequalities hold. These are more likely to hold than the underlying assumptions, because the assumptions imply the inequalities, but the reverse may not be true.

Empirical verification of these inequalities is made especially attractive for two reasons. First, many optimization analyses actually use only a few options for the "algebraic expression" in (1): the only thing that changes is the analysis of the algorithm leading to improved upper bounds. Thus, by empirically measuring the degree to which the *first* inequality in (1) holds, we can interrogate whether popular analyses strategies can explain optimization success in deep learning in way that is less tightly coupled to whether particular global assumptions hold or not.

Two very popular assumptions about the loss landscape and the optimization process are smoothness and convexity. Formally, a differentiable function $f(\cdot, \cdot) : \mathbb{R}^d \times \mathcal{Z} \mapsto \mathbb{R}$ is convex if it satisfies:

$$f(\mathbf{y}, z) \geq f(\mathbf{x}, z) + \langle \nabla f(\mathbf{x}, z), \mathbf{y} - \mathbf{x} \rangle \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, z \in \mathcal{Z}$$

Further, $f(\cdot, \cdot)$ is $L-$smooth if it satisfies:

$$\|\nabla f(\mathbf{x}, z) - \nabla f(\mathbf{y}, z)\| \leq L\|\mathbf{x} - \mathbf{y}\| \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, z \in \mathcal{Z}$$

These are some of the most common assumptions in optimization theory (Zinkevich, 2003; Duchi et al., 2010; Ghadimi & Lan, 2013; Bubeck et al., 2015; Carmon et al., 2017; Zhao et al., 2020; Hu et al., 2019; Hazan, 2022; Cutkosky & Orabona, 2019). We would like to quantify them in our experiments. Since computing the global smoothness constant as well as the convexity of the true loss functions $F(\mathbf{x})$ is infeasible, we instead measure proxies that we call the *instantaneous convexity gap*, denoted by inst_gap, and *instantaneous smoothness*, denoted by inst_smooth, to estimate the levels of convexity and smoothness of the true loss function. Formally, the instantaneous convexity gap with respect to a reference point $\mathbf{y}_t$ of the function $f(\cdot, z_t)$ (stochastic loss function computed at iteration $t$ using datapoint $z_t$) is defined as:

$$\text{inst\_gap}_t(\mathbf{y}_t) \coloneqq f(\mathbf{x}_t, z_t) - f(\mathbf{y}_t, z_t) - \langle \nabla f(\mathbf{x}_t, z_t), \mathbf{x}_t - \mathbf{y}_t \rangle \tag{2}$$

In our measurements, we use two settings for $\mathbf{y}_t$. First, we consider $\mathbf{y}_t = \mathbf{x}_{t-1}$ to analyze the properties of consecutive points and their impact on the optimization path. Next, we use the constant value $\mathbf{y}_t = \mathbf{x}^\star$, where $\mathbf{x}^\star$ is the *final* iterate produced by a previous training run. This setting provides a more *global* view of the loss landscape. If $f$ is convex, then the convexity gap defined in eq. (2) should be *non-positive*. We also compute the average convexity gap and the exponential moving average of the convexity gaps with respect to a sequence of reference points $\mathbf{y}_1, \ldots, \mathbf{y}_t$ (denoted as $\mathbf{y}_{1:t}$ for short), respectively defined as

$$\begin{aligned} &\text{avg\_gap}_t(\mathbf{y}_{1:t}) = \tfrac{1}{t}\textstyle\sum_{i=1}^{t} \text{inst\_gap}_i(\mathbf{y}_i), \\ &\text{exp\_gap}_t(\mathbf{y}_{1:t}) = \beta \cdot \text{exp\_gap}_{t-1}(\mathbf{y}_{1:t-1}) + (1-\beta) \cdot \text{inst\_gap}_t(\mathbf{y}_t). \end{aligned} \tag{3}$$

where $\beta \in (0, 1)$ (we choose $\beta = 0.99$ for our measurements).

Next, we define the instantaneous smoothness at iteration $t$ with respect to $\mathbf{y}_t$ as:

$$\text{inst\_smooth}_t(\mathbf{y}_t) = \tfrac{\|\nabla f(\mathbf{x}_t, z_t) - \nabla f(\mathbf{y}_t, z_t)\|}{\|\mathbf{x}_t - \mathbf{y}_t\|} \tag{4}$$

If the loss function is $L$-smooth, then $\text{inst\_smooth}_t \leq L$ for all $t \in [T]$. Thus, if this instantaneous smoothness quantity is uniformly bounded by a constant, it could indicate that our loss landscape is smooth. Similar to the convexity gap, we also keep track of other forms of the smoothness measure such as the maximum smoothness and the exponential average smoothness, respectively defined as

$$\begin{aligned} &\text{max\_smooth}_t(\mathbf{y}_{1:t}) = \max_{i \leq t} \text{inst\_smooth}_i(\mathbf{y}_i), \\ &\text{exp\_smooth}_t(\mathbf{y}_{1:t}) = \beta \cdot \text{exp\_smooth}_{t-1}(\mathbf{y}_{1:t-1}) + (1-\beta) \cdot \text{inst\_smooth}_t(\mathbf{y}_t). \end{aligned} \tag{5}$$

Our maximum smoothness is similar to the smoothness metric proposed in (Santurkar et al., 2018; Zhang et al., 2019). However, instead of tracking the largest smoothness value along the line of the update difference $\mathbf{x}_t - \mathbf{x}_{t-1}$, we keep track of the largest value across all iterations.

Most of our training runs involve multiple epochs. In this case, for the non-instantaneous metrics, we "reset" the averages at the start of each epoch so that the averages contain only iterates from the current epoch. The only exceptions are our pre-training tasks for BERT and GPT-2. Due to the large size of the datasets used in these tasks, we completed the training without traversing the entire dataset. Hence, we do not reset our metrics in these experiments. Beyond smoothness and convexity, we also track many other key properties. We defer these results to the Appendix. These metrics collectively offer deeper insights into the dynamic behavior of the loss function throughout the optimization process.

We conduct experiments across a diverse array of tasks, ranging from simple convex problems to complex NLP tasks involving models with hundreds of millions of parameters. For convex tasks, we run gradient descent on a synthetic dataset using squared loss and also perform logistic regression on various OpenML datasets (Aloi, Connect-4, Covertype, Poker). In the realm of non-convex tasks,

we address both Image Classification and NLP benchmarks. For Image Classification tasks, we train popular benchmark datasets Cifar10 and Imagenet (Deng et al., 2009) on Resnet18 (He et al., 2016) using SGD with momentum (SGDM) and Adamw. we use the configurations reported in (Yao et al., 2020; Tran & Cutkosky, 2022a). For NLP tasks, we pre-train Bert (Devlin et al., 2018b) using the C4 dataset (Raffel et al., 2019) and GPT2 (Radford et al., 2019) using the Pile dataset (Gao et al., 2020). Both tasks are trained using SGDM and AdamW. The learning rates for each optimizer are fine-tuned through a grid search in the range $[10^{-6}, 0.1]$.

## 3 Measuring Convexity

Convexity is a fundamental assumption in optimization theory since convex functions have many pleasant theoretical guarantees. For instance, every local minimum of a convex function is also a global minimum, which allows us to derive bounds on the suboptimality gap (Bottou & Bousquet, 2007; Defazio et al., 2014; Cutkosky, 2019). Unfortunately, the landscape of deep learning training is known to be non-convex (Jain et al., 2017; Li et al., 2018; Garipov et al., 2018; Choromanska et al., 2015) due to the complex architectures of deep learning models and the nonlinearity of the activation functions. However, the degree of non-convexity in practical scenarios still remains a bit of a mystery. In this section, we aim to quantify the level of convexity across various machine learning tasks. As a sanity check, we first examine the instantaneous convexity gaps with respect to the previous iterate in simple tasks for which the objective is indeed convex to verify that they align with our theoretical expectations. Results are presented in Fig.1. As we can see from Fig.1, the convexity gap is always non-positive as expected. Now, let us turn our attention to more complex deep learning tasks.

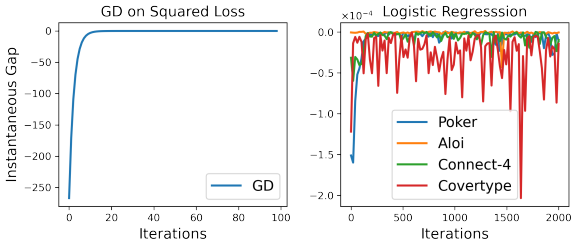


Figure 1: Instantaneous convexity gap w.r.t. $\mathbf{y}_t = \mathbf{x}_{t-1}$ of Gradient Descent (GD) on the squared loss (left) and Logistic Regression on OpenML datasets (Vanschoren et al., 2013) (right).

### 3.1 Are Deep Learning Loss Landscapes convex along optimization paths?

In this section, we aim to examine the convexity along the paths taken by popular optimizers such as Adam and SGD. To achieve this, we compute both the average and the exponential average convexity gaps with respect to the previous iterates, i.e., $\mathbf{y}_t = \mathbf{x}_{t-1}$, across various deep learning benchmarks. Setting $\mathbf{y}_t = \mathbf{x}_{t-1}$, allows us to measure convexity on a "small scale" along the optimization path, rather than as a global property. The presence of any positive gap would indicates non-convexity.

By measuring average gaps (both avg_gap and exp_gap), we gain insight into whether the optimization path could be in some sense "mostly" convex - i.e. whether instantaneous non-convexity is essentially a "rare event". Stochastic optimization analysis typically involves summing or averaging identities derived from convexity, and so one might hope that it is possible to exploit a non-positive average convexity gap.We also provide the instantaneous gap results in Section D in the Appendix.

Surprisingly, the convexity gap along the optimization trajectories of non-convex tasks is not consistently negative or positive, as demonstrated in fig. 2. For instance, while the convexity gap remains uniformly positive (indicating non-convnexity) during the training of ImageNet on ResNet18, the optimization trajectory in the training of Bert frequently shifts between convex and non-convex regions. Notably, in experiments involving CIFAR-10 and GPT-2, the convexity gap consistently exhibits negative values. Similar phenomenon is also observed in (Xing et al., 2018), where they demonstrated that the loss interpolation $F(\alpha\mathbf{x}_t + (1-\alpha)\mathbf{x}_{t+1})$ of deep neural networks trained on CIFAR-10 by SGD is locally convex.

Negative convexity gaps in our experiments do not necessarily indicate convex loss landscapes (since we only check the convexity gap at the points along the optimization trajectory) but rather suggests that effective optimizers like SGD and ADAM can navigate these landscapes by finding paths that are in some sense "locally convex". Further, as illustrated in Figure 2, the dataset plays a significant

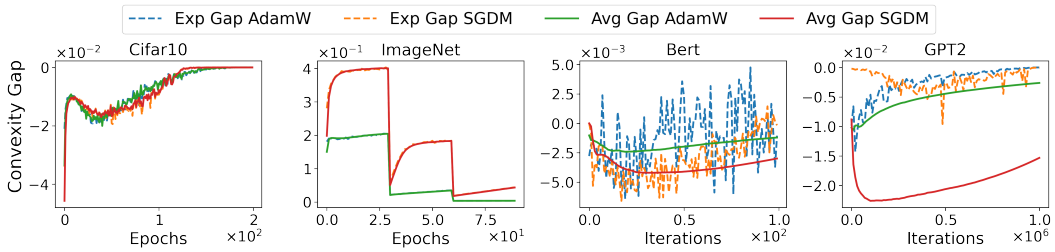


Figure 2: Average convexity gap and exponential average convexity gap w.r.t. $\mathbf{y}_t = \mathbf{x}_{t-1}$ of deep learning benchmarks. In most cases, the gaps are negative, indicating local convexity along training.

role in shaping the loss landscape. Despite using the same optimizer settings and the ResNet-18 architecture, the loss landscapes for the ImageNet and CIFAR-10 datasets show markedly different levels of convexity.

### 3.2 Can Convexity-based Analysis Explain Optimization Success?

Though the results in Section 3.1 suggest convexity along the optimization path often occurs, we might care more about global convexity, as this is useful to prove global convergence guarantees. Moreover, while the convexity gap can be used to falsify convexity or give intuition about the local properties of the loss landscape, this quantity does not appear in an obvious way in most optimization analyses. So, in this section, we measure a different quantity called *convexity ratio*, which allows us to probe more directly the degree to which analyses based on convexity apply to real problems.

$$\text{convexity\_ratio}_T = \frac{\sum_{t=1}^{T} \langle \nabla F(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle}{\sum_{t=1}^{T} F(\mathbf{x}_t) - F(\mathbf{x}^\star)} \tag{6}$$

Here we use a large batch loss to approximate $F$ in cases where it is computationally infeasible to compute $F$ exactly (more details on this computation are in the Appendix). $\mathbf{x}^\star$ is an approximate stationary point given by the output of a previous training run. When $F$ is convex, we should expect the convexity ratio to be larger than 1 so that we have the following important inequality:

$$\sum_{t=1}^{T} F(\mathbf{x}_t) - F(\mathbf{x}^\star) \leq \sum_{t=1}^{T} \langle \nabla F(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle \tag{7}$$

Equation (7) is the essential ingredient in many optimization analyses based on convexity. In fact, many analyses of SGD and related methods actually prove convergence by upper-bounding the RHS of the above equation - it is the standard instantiation of eq. (1) for convex analysis (Duchi et al., 2010; McMahan & Streeter, 2010; Zinkevich, 2003; Reddi et al., 2018; Hazan et al., 2007; 2006). For example, a typical analysis of SGD (e.g. (Zinkevich, 2003)) would show that the RHS is bounded by $O(\sqrt{T})$, from which one can then conclude that $\frac{1}{T}\sum_{t=1}^{T} F(\mathbf{x}_t) - F(\mathbf{x}^\star) \leq O(1/\sqrt{T})$: that is, the loss values of the iterates are "on average" approaching the loss of $F(x_\star)$. This holds for all possible values of $x_\star$, even though we will only evaluate it for one particular point.

As a result, even if our function does not satisfy eq. (7), it is still possible to derive global convergence. Assume that the convexity ratio is larger than $K$ for $0 < K < 1$ instead (this condition would be implied by "weak quasi-convexity" studied by Orabona & Tommasi (2017)). Then we still have:

$$\sum_{t=1}^{T} F(\mathbf{x}_t) - F(\mathbf{x}^\star) \leq \sum_{t=1}^{T} \frac{1}{K} \langle \nabla F(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle$$

Since our analysis typically bounds the RHS of this equation, the convergence bound degrades by only a factor of $1/K$. Therefore, as long as $K \geq \Omega(1/\sqrt{T})$, popular algorithms like SGD can still ensure global convergence. Interestingly, our experiments on CIFAR-10 and Bert (Figure 3) suggest that this property may hold for certain deep learning tasks. Despite the fact that CIFAR-10 and Bert losses are *not* globally convex, the standard *analysis* used with convexity assumptions may still explain optimization success in these tasks.

For the CIFAR-10 experiments, AdamW's convexity ratio suggests the optimization trajectory remains globally convex relative to the stationary point. While SGDM shows slight non-convexity

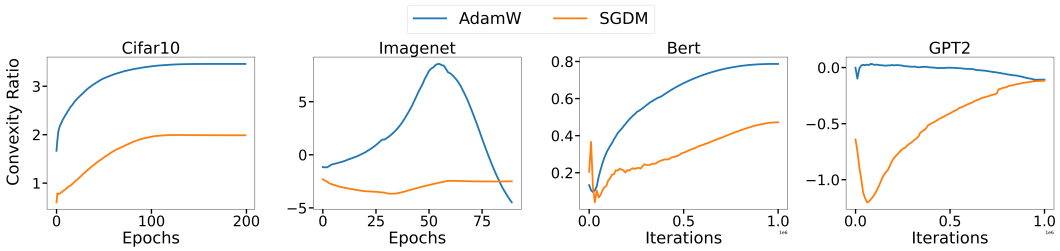


Figure 3: Convexity ratios of deep learning benchmarks. A convexity ratio greater than 1 indicates a convex function. Ratios between 0 and 1 suggest slight non-convexity, still permitting the application of classic convex optimization arguments. Ratios less than 0 denote strong non-convexity.

initially, its convexity ratio consistently exceeds 0.5, allowing for the application of classical convex analysis arguments. In the BERT experiments, both AdamW and SGDM exhibit convexity ratios below 1, indicating a globally non-convex trajectory. However, since the ratios are above 0.1, classical convex analysis remains applicable, though with a 10x degradation in convergence bounds.

Unfortunately, since the convexity ratios of both optimizers are negative in the GPT2 experiments, the convex analysis argument seems to be invalid. A similar lack of convexity is observed in the ImageNet experiments. Interestingly, AdamW seems to often find a "more convex" optimization path compared to SGDM. Nevertheless, these data suggest that significant alterations to classical analysis based on convexity would be needed to adequately explain optimization success for deep learning in general.

## 4 Measuring Smoothness

Smoothness assumptions plays a pivotal role in optimization theory. In convex optimization, smoothness can help accelerate the training process and achieve superlinear convergence rate if the loss is strictly convex or strongly convex (Nesterov et al., 2018). In non-convex optimization, smoothness is the key ingredient that makes many convergence analyses possible (Ghadimi & Lan, 2013; Allen-Zhu & Hazan, 2016; Jain et al., 2017; Reddi et al., 2019). Although smoothness is assumed for the majority of non-convex optimization results, it is unclear how well these smoothness conditions are satisfied in practice.

In fact, from a purely theoretical point of view, it may seem unlikely that the objective could be truly smooth: common activation functions such as the ReLU, and common layers such as MaxPools are not globally differentiable and so cannot possibly be smooth. However, one might hope that such issues are essentially pathological problems that do not affect practice. In this section, we attempt to measure smoothness along the real optimization trajectory in an efficient way analogous to our investigation of convexity in Section 3.

We will focus on the exponential average smoothness and the max smoothness defined in eq. (5) since they provide insights into the smoothness level of local and global loss landscape respectively.

First, we compute these measures using the optimally tuned learning rate and schedule in each deep learning experiment. As we can see from fig. 4 (top), in all experiments, the smoothness constants appear to be upper-bounded. However, in many cases these constants are quite large ($10^3$ to $10^6$), making it hard to consider the loss landscapes in these experiments to be smooth in practice. Furthermore, we note that smoothness correlates with changes in the learning rate scheduler. For example, as the learning rate approaches zero at the end of training, the smoothness value increases, as observed in Cifar10 with cosine decay and BERT with linear decay. Similarly, for Imagenet, where we used a piecewise linear scheduler, smoothness increases whenever the learning rate decreases. This observation suggests that smaller learning rates tend to result in larger smoothness values.

To better understand the loss landscapes, we reran all experiments with a constant learning rate (fig. 4 bottom). With constant learning rates, the loss landscape appeared smoother and more stable. Both the max and exponential average smoothness followed a similar pattern: a rapid drop initially (except for SGDM on ImageNet), followed by a consistent rise until reaching a boundary, then stabilizing. Adam typically achieved smaller (i.e., smoother) measures with a learning rate scheduler, while

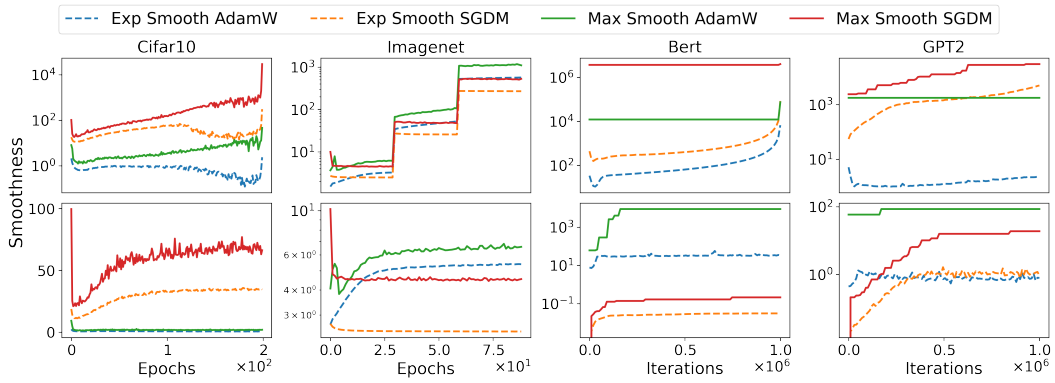


Figure 4: Smoothness measures w.r.t. $\mathbf{y}_t = \mathbf{x}_{t-1}$ of deep learning benchmarks using the optimal configurations. (Top) are the experiments with optimal learning rate scheduler, and (bottom) are the experiments with constant learning rate. Details of experiment setup can be found in Appendix B.

SGD found smaller measures with a constant rate. We conjecture that this phenomenon suggests that SGD's optimization path is more sensitive to changes in the learning rate, while Adam remains robust across different learning rate settings.

### 4.1 Smoothness measures as proxies for sharpness

As shown in fig. 4, the smoothness measured in most experiments exhibit similar behaviors. This pattern closely resembles the edge-of-stability phenomenon observed by (Cohen et al., 2020; 2022) in full-batch SGD and full-batch Adam for smaller tasks. Specifically, Cohen et al. (2020) defines the "sharpness" as the operator norm of the Hessian $\nabla^2 F(\mathbf{x}_t)$. They observe that when training with full-batch gradient descent on CIFAR-10, the sharpness increases until it reaches a value inversely proportional to the learning rate, and then stabilizes.

Our measurements track different quantities than the sharpness, but are faster to compute. Thus, these observations pose an interesting question: *Can our new metrics, max_smooth and exp_smooth, be used as proxies for the sharpness*? If this is true, our approach could substantially expedite the evaluation of sharpness. Our method also makes evaluating the sharpness of much larger models possible (for which computing Hessian information is prohibitively expensive).

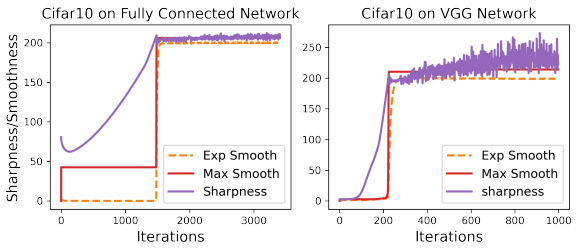


Figure 5: Sharpness (maximum eigenvalue of the training loss Hessian Matrix) v.s. Smoothness.

As discussed above, we notice that a smaller learning rate results in a larger smoothness value. We can potentially explain this using the edge-of-stability phenomenon. (Cohen et al., 2020; 2022) observe that the sharpness is oscillating at the value $c/\eta$ for some constant $c > 0$ and $\eta$ is the learning rate at the edge of stability. Thus, when the learning rate scheduler is applied, any time the learning drops, this boundary increases and causes the smoothness/sharpness level to increase. This phenomenon is also observed in (Cohen et al., 2022). To verify our conjecture, we replicate the experiments in (Cohen et al., 2020) where we train Cifar10 on a simple linear network with tanh activation and on a VGG-11 network (Simonyan & Zisserman, 2014) in Fig.5.

Our new smooth metrics track the actual sharpness value very closely (Fig.5). One possible justification for this is when we measure $\frac{\|\nabla f(\mathbf{x}_t,z)-\nabla f(\mathbf{x}_{t-1},z)\|}{\|\mathbf{x}_t-\mathbf{x}_{t-1}\|}$, we are effectively estimating how quickly the gradient of the function changes, which is bounded by the Hessian's spectral norm in smooth functions. A higher value indicates a steeper change in the gradient, implying a larger maximum eigenvalue of the Hessian matrix, hence a higher "sharpness". Thus, this metric and the sharpness are inherently related to characterizing the function's smoothness and curvature.

### 4.2 Can Smoothness-based Analysis Explain Optimization Success?

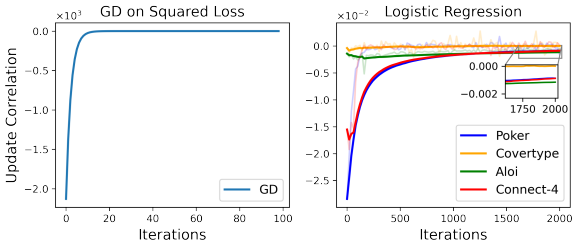


Figure 6: Update correlation of GD on the squared loss (left) and logistic regression on OpenML datasets (right). The blurred lines are the actual update correlations, and the thick lines are the varage.

The smoothness measurements discussed above are not actually the best criterion for judging the applicability of smooth non-convex optimization analysis. This is because they only capture gradient behavior rather than linking gradients o function values. In typical smoothness-based analysis, one encounters the quantity $\langle \nabla f(\mathbf{x}_{t+1}, z_{t+1}), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle$. In almost all analyses of non-convex optimization algorithms, this quantity usually plays the role of the "algebraic expression" in (1) (Khaled & Richtárik, 2020; Li et al., 2024; Zaheer et al., 2018a; Carmon et al., 2018; Li & Orabona, 2019; Faw et al., 2022; Reddi et al., 2019). To illustrate, consider an optimizer with update $\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta_t$, and assume that $F$ is $L$-smooth, $\mathbb{E}[\Delta_t] = -\eta \nabla F(\mathbf{x}_t)$ and $\mathbb{E}[\|\Delta_t\|^2] \leq \eta^2 G^2$. Then:

$$\begin{aligned}\mathbb{E}[F(\mathbf{x}_{t+1}) - F(\mathbf{x}_t)] &\leq \mathbb{E}[\langle \nabla F(\mathbf{x}_t), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle + \tfrac{L}{2}\|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2] \\ &\leq -\eta\, \mathbb{E}\left[\|\nabla F(\mathbf{x}_t)\|^2\right] + \tfrac{L\eta^2 G^2}{2}. \end{aligned} \tag{8}$$

Typical analyses show that $-\eta\, \mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2]$ dominates $\frac{L\eta^2 G^2}{2}$ so that $F(\mathbf{x}_t)$ decreases over time. Intuitively, this holds if we make $\eta$ sufficiently small because the negative term is linear in $\eta$ while the positive term is quadratic in $\eta$. Note that this high-level idea is used even for analyses based on less classical smoothness assumptions such as (L0,L1) smoothness (Zhang et al., 2019).

To check whether this analysis technique can explain the success of practical optimizers, we would like to measure the inner-product $\langle \nabla F(\mathbf{x}_t), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle$ and see if it is negative. This would directly capture the optimization analysis because in the typical analysis, all of the provable decrease in the function value is caused negative inner-products.

Unfortunately, this inner-product is difficult to estimate empirically because we do not know $\nabla F(\mathbf{x}_t)$. One might consider instead estimating it using $\langle \nabla f(\mathbf{x}_t, z_t), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle$. However, this approach is flawed because $\mathbf{x}_{t+1} - \mathbf{x}_t$ is not independent of $z_t$, giving the correlation a negative bias. Instead, we measure a quantity that we call the *update correlation*, which is defined as

$$\text{update\_corr}_t \coloneqq \langle \nabla f(\mathbf{x}_{t+1}, z_{t+1}), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle. \tag{9}$$

Since $\mathbf{x}_{t+1} - \mathbf{x}_t$ is independent of $z_{t+1}$, the update correlation is an unbiased estimator of $\langle \nabla F(\mathbf{x}_{t+1}), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle$. Moreover, it turns out that update correlation still captures the same notion of "function" progress measured by typical analysis. Here's a brief reasoning. Consider the update $\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta_t$ and assume $F$ is $L$-smooth (but this time we don't make assumptions on $\Delta_t$). By smoothness,

$$F(\mathbf{x}_{t+1}) - F(\mathbf{x}_t) \geq \langle \nabla F(\mathbf{x}_{t+1}), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle - \tfrac{L}{2}\|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 \tag{10}$$

$$F(\mathbf{x}_{t+1}) - F(\mathbf{x}_t) \leq \langle \nabla F(\mathbf{x}_{t+1}), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle + \tfrac{L}{2}\|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 \tag{11}$$

Consequently, if $\langle \nabla F(\mathbf{x}_{t+1}), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle$ is negative, for small enough learning rates $\eta$ the global loss decreases and the optimizer is consistently making progress. On the other hand, a positive update correlation $\langle \nabla F(\mathbf{x}_{t+1}), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle$ appears to be disastrous since this analysis would suggest that the loss should increase. In particular, we are not aware of any analysis based upon negative values of $\langle \nabla F(\mathbf{x}_t), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle$ that does not also predict negative values for the update correlation. Therefore, if the standard analysis of smooth non-convex optimization can explain optimization success in deep learning, then in every experiment we should expect that the update correlation $\langle \nabla f(\mathbf{x}_{t+1}, z_{t+1}), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle$ should be negative on average.

First, we check if this is the case for simple convex experiments (fig. 6). In all of these experiments, the update correlations are negative on average, which agrees with our intuition that a negative update correlation indicates progress in the training.

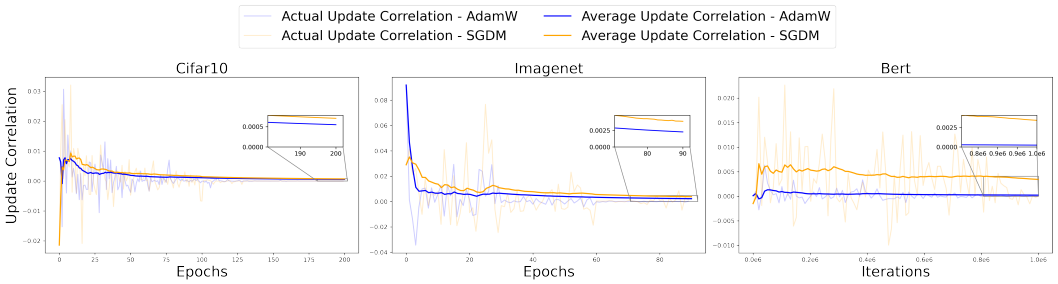


Figure 7: Update correlation on various Deep Learning tasks.

However, surprisingly, the update correlation is positive on average in almost every other Deep Learning experiment (fig. 7). This is a fascinating phenomenon because it indicates that the optimizer changes direction very often, and yet it still effectively minimizes the loss. This suggests that the classic smooth non-convex analysis that relies on the descent lemma is problematic in practice. The only case of negative correlations is GPT-2 on the Pile dataset, but they turn positive when the dataset is shuffled or replaced with the C4 dataset. It would be interesting to find out exactly the cause of this behavior.

The observation that $\nabla F(\mathbf{x}_{t+1})$ is positively correlated with $\mathbf{x}_{t+1} - \mathbf{x}_t$ suggests that the objective may be "poorly conditioned", so that the optimizer is bouncing back-and-forth along the walls of a narrow ravine in the optimization landscape. Previous empirical studies have also suggested similar dynamics (Rosenfeld & Risteski, 2023). The classical mitigations for poorly conditioned objectives in the *deterministic or convex* settings are preconditioning, including via second-order algorithms, as well as accelerated gradient descent (e.g. Gupta et al. (2018); Liu et al. (2023a); Yao et al. (2021); Nesterov et al. (2018); Dozat (2016)). However, the advantages of such techniques are poorly understood in the stochastic setting (indeed, there is no advantage in the worst-case (Arjevani et al., 2020)). Instead, most current analyses we are aware of in the stochastic setting appear to rely on negative update correlations.

### 4.3 Alternatives for smooth non-convex optimization

In previous sections, we observed that some common assumptions or identities used in analysis, such as convexity, smoothness, or negative update correlation, might not hold in practice. In this section, we will discuss alternative frameworks that do not rely on these assumptions.

The first direction focuses on a family of weakly convex objectives (Davis & Drusvyatskiy, 2019; Mai & Johansson, 2020), where the goal is to minimize a proxy of the objectives called the Moreau envelope (Moreau, 1965) . In a different direction, Zhang et al. (2020b) propose employing the Goldstein stationary point (Goldstein, 1977) as a convergence criterion that is tractable for non-smooth objectives. Later, Cutkosky et al. (2023) proposes an online-to-non-convex conversion (O2NC) technique that later inspires other works on non-smooth non-convex optimization (Ahn et al., 2024; Zhang & Cutkosky, 2024). The key idea of their technique is the use of random scaling: suppose $s_t$ is sampled i.i.d. from $\mathrm{Exp}(1)$, then $\mathbf{x}_{t+1} = \mathbf{x}_t + s_t\Delta_t$ satisfies

$$\mathbb{E}_{s_t}[F(\mathbf{x}_{t+1}) - F(\mathbf{x}_t)] = \mathbb{E}_{s_t}\langle \nabla F(\mathbf{x}_{t+1}), \Delta_t\rangle. \tag{12}$$

We refer to the update form $\mathbf{x}_{t+1} = \mathbf{x}_t + s_t\Delta_t$ where $s_t \sim \mathrm{Exp}(1)$ i.i.d. as the *update with random scaling (RS)*, and the update with $s_t \equiv 1$ as the *update without RS*. Unlike the lower bound in equation 10, the equality in equation 12 suggests that $\langle \nabla F(\mathbf{x}_{t+1}), \Delta_t\rangle$, which we referred to as *update correlation with RS*, is an unbiased estimator of function progress $F(\mathbf{x}_{t+1}) - F(\mathbf{x}_t)$ and a good indicator of the training progress: we should expect $F(\mathbf{x}_t)$ to decrease as long as $\langle \nabla F(\mathbf{x}_{t+1}), \Delta_t\rangle$ is negative in average.

To verify if the theory holds in practice, we test SGDM and AdamW with random scaling updates and compare them to their counterparts without RS. Specifically, we measure the following three properties: update correlation, update correlation with random scaling, and instantaneous loss difference, where the first is defined in eq. (9) and the latter two are respectively defined as

$$\text{update\_corr\_RS}_t = \langle \nabla f(\mathbf{x}_t, z_t), \Delta_{t-1}\rangle, \quad \text{loss\_diff}_t = f(\mathbf{x}_t, z_t) - f_t(\mathbf{x}_{t-1}, z_t). \tag{13}$$

Note that if the update does *not* have random scaling applied, then $\text{update\_corr\_RS}_t = \text{update\_corr}_t$.

In fig. 8 we plot the cumulative sum of these quantities. The sum of update correlation always increases, regardless of whether random scaling is employed. However, for optimizers with random scaling, the sum of update_corr_RS$_t$ decreases and closely aligns with the sum of loss difference. This supports the theory that update_corr_RS$_t$ is an unbiased estimator of loss difference, even for complicated LLM models. Also, it motivates a guideline for developing empirically effective optimizers: keeping $\langle \nabla f(\mathbf{x}_t, z_t), \Delta_{t-1} \rangle$ as negative as possible while applying random scaling to the update.

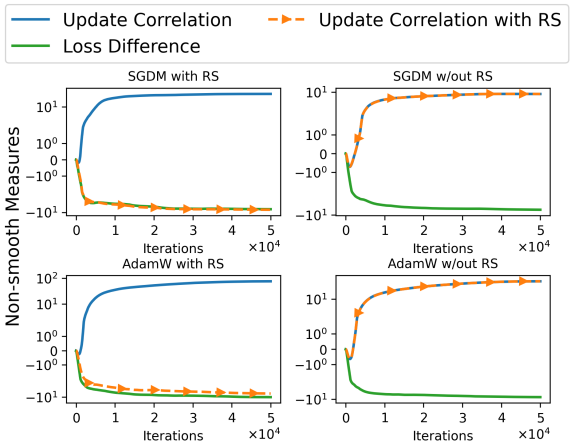


Figure 8: Cumulative sum (symmetric log scale) of update correlation, update correlation with RS, and loss difference of GPT2 model trained on Pile dataset. (Top) is SGDM and (bottom) is AdamW; (left) is update with RS and (right) is the benchmark without RS.

## 5 Related Works

There have been extensive studies on the empirical properties and the loss landscape of modern machine learning. Goodfellow & Vinyals (2015) proposed one-dimensional and two-dimensional visualization tools for the loss landscape of various neural networks, demonstrating that SGD rarely encounters local minima during training. Im et al. (2017) tested the training trajectories of different optimizers using the same visualization tools and observed that different optimizers exhibit distinct behaviors when encountering saddle points. Li et al. (2018) proposed more refined visualization techniques and showed that the smoothness of the loss landscape closely correlates with generalization performance. Nakkiran et al. (2019) studied the dynamics of SGD training, showing that SGD learns simple classifiers at early training stages and learns more complex classifiers at later stages. Power et al. (2022) reported the grokking phenomenon on a synthesized dataset such that after a long period of severe overfitting, validation score suddenly increases to almost perfect generalization. Thilak et al. (2022) revealed the slingshot effect of training neural networks with adaptive optimizers, which is a cyclic behavior between stable and unstable regimes during training process. While these results provide general insight into neural network landscapes, we focus on validating common assumptions and key identities fundamental to the analysis of optimization theory.

There are several studies that align more closely with our work. Xing et al. (2018) demonstrated that loss interpolation between consecutive iterates is locally convex, which agrees with our observations in Sec 3.1. While their experiments focus on SGD and image classification tasks, we extended the scope of our convexity measures to include AdamW and LLMs. Furthermore, we also tested a more global convexity measure in Sec 3.2. Cohen et al. (2020; 2022) observed the "edge of stability" phenomenon where the sharpness increases during early stage of training and then stabilizes. Our observations in Sec 4 align with their finding and extend beyond CIFAR-10 tasks. Rosenfeld & Risteski (2023) demonstrated the opposing signal phenomenon that there are groups of outliers such that decreasing loss over one group increases loss over other groups, which could explain our observation of positive update correlation in Sec 4.2. Unlike these works, our work does not only verify common assumptions but also directly measures key quantities in modern analyses.

## 6 Conclusions

We address the critical question of whether modern analyses in stochastic optimization theory align with practice. To this end, we empirically measure key quantities that are commonly used in theory across a diverse range of machine learning benchmarks. Our results indicate that, in most cases, these commonly assumed identities do not hold in practice. Further, we provide comparisons between the behaviors of SGD and Adam across various important properties. We hope that our experiments results can contribute to a better understanding of what enables practical optimization, as well as motivate more rigorous empirical verification of optimization analyses in the future.

## REFERENCES


Kwangjun Ahn, Zhiyu Zhang, Yunbum Kook, and Yan Dai. Understanding adam optimizer via online learning of updates: Adam is ftrl in disguise, 2024. URL https://arxiv.org/abs/2402.01567.

Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. In *International conference on machine learning*, pp. 699–707. PMLR, 2016.

Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Ayush Sekhari, and Karthik Sridharan. Second-order information in non-convex stochastic optimization: Power and limitations. In *Conference on Learning Theory*, pp. 242–299, 2020.

Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis (eds.), *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/0d3180d672e08b4c5312dcdafdf6ef36-Paper.pdf.

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. "convex until proven guilty": Dimension-free acceleration of gradient descent on non-convex functions. In *International Conference on Machine Learning*, pp. 654–663. PMLR, 2017.

Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018. doi: 10.1137/17M1114296. URL https://doi.org/10.1137/17M1114296.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pp. 192–204. PMLR, 2015.

Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2020.

Jeremy M Cohen, Behrooz Ghorbani, Shankar Krishnan, Naman Agarwal, Sourabh Medapati, Michal Badura, Daniel Suo, David Cardoze, Zachary Nado, George E Dahl, et al. Adaptive gradient methods at the edge of stability. *arXiv preprint arXiv:2207.14484*, 2022.

Ashok Cutkosky. Anytime online-to-batch conversions, optimism, and acceleration. *arXiv preprint arXiv:1903.00974*, 2019.

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *Advances in neural information processing systems*, 32, 2019.

Ashok Cutkosky, Harsh Mehta, and Francesco Orabona. Optimal stochastic non-smooth non-convex optimization through online-to-non-convex conversion. In *International Conference on Machine Learning (ICML)*, 2023.

Damek Davis and Dmitriy Drusvyatskiy. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019. doi: 10.1137/18M1178244.

Damek Davis, Dmitriy Drusvyatskiy, Sham Kakade, and Jason D Lee. Stochastic subgradient method converges on tame functions. *Foundations of computational mathematics*, 20(1):119–154, 2020.

Damek Davis, Dmitriy Drusvyatskiy, Yin Tat Lee, Swati Padmanabhan, and Guanghao Ye. A gradient sampling method with complexity guarantees for lipschitz functions in high and low dimensions. *arXiv preprint arXiv:2112.06969*, 2021.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018a. URL http://arxiv.org/abs/1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018b.

Timothy Dozat. Incorporating nesterov momentum into adam. *ICLR Workshop*, 2016.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *Conference on Learning Theory (COLT)*, pp. 257–269, 2010.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011. URL http://jmlr.org/papers/v12/duchi11a.html.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pp. 689–699, 2018.

Matthew Faw, Isidoros Tziotis, Constantine Caramanis, Aryan Mokhtari, Sanjay Shakkottai, and Rachel Ward. The power of adaptivity in sgd: Self-tuning step sizes with unbounded gradients and affine variance. In *Conference on Learning Theory*, pp. 313–355. PMLR, 2022.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

A. A. Goldstein. Optimization of lipschitz continuous functions. *Math. Program.*, 13(1):14–22, dec 1977. ISSN 0025-5610. doi: 10.1007/BF01584320. URL https://doi.org/10.1007/BF01584320.

Ian J. Goodfellow and Oriol Vinyals. Qualitatively characterizing neural network optimization problems. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pp. 1842–1850. PMLR, 2018.

Elad Hazan. *Introduction to online convex optimization*. MIT Press, 2022.

Elad Hazan, Adam Kalai, Satyen Kale, and Amit Agarwal. Logarithmic regret algorithms for online convex optimization. In *International Conference on Computational Learning Theory*, pp. 499–513. Springer, 2006.

Elad Hazan, Alexander Rakhlin, and Peter Bartlett. Adaptive online gradient descent. *Advances in neural information processing systems*, 20, 2007.

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

Wenqing Hu, Chris Junchi Li, Xiangru Lian, Ji Liu, and Huizhuo Yuan. Efficient smooth non-convex stochastic compositional optimization via stochastic recursive gradient descent. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/21ce689121e39821d07d04faab328370-Paper.pdf.

Daniel Jiwoong Im, Michael Tao, and Kristin Branson. An empirical analysis of the optimization of deep network loss surfaces, 2017.

Prateek Jain, Purushottam Kar, et al. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142–363, 2017.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/ac1dd209cbcc5e5d1c6e28598e8cbbe8-Paper.pdf.

Ahmed Khaled and Peter Richtárik. Better theory for sgd in the nonconvex world. *arXiv preprint arXiv:2002.03329*, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

Haochuan Li, Jian Qian, Yi Tian, Alexander Rakhlin, and Ali Jadbabaie. Convex and non-convex optimization under generalized smoothness. *Advances in Neural Information Processing Systems*, 36, 2024.

Xiaoyu Li and Francesco Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. In *The 22nd international conference on artificial intelligence and statistics*, pp. 983–992. PMLR, 2019.

Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. *arXiv preprint arXiv:2305.14342*, 2023a.

Zijian Liu, Ta Duy Nguyen, Alina Ene, and Huy Nguyen. On the convergence of adagrad (norm) on r^ d: Beyond convexity, non-asymptotic rate and acceleration. In *International Conference on Learning Representations*. International Conference on Learning Representations, 2023b.

Vien Mai and Mikael Johansson. Convergence of a stochastic gradient method with momentum for non-smooth non-convex optimization. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6630–6639. PMLR, 13–18 Jul 2020.

H Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. *arXiv preprint arXiv:1002.4908*, 2010.

J.J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965. URL http://eudml.org/doc/87067.

Preetum Nakkiran, Gal Kaplun, Dimitris Kalimeris, Tristan Yang, Benjamin L. Edelman, Fred Zhang, and Boaz Barak. Sgd on neural networks learns functions of increasing complexity, 2019.

Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

Francesco Orabona and Tatiana Tommasi. Training deep networks without learning rates through coin betting. *Advances in Neural Information Processing Systems*, 30, 2017.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019.

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

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

Elan Rosenfeld and Andrej Risteski. Outliers with opposing signals have an outsized effect on neural network optimization. *arXiv preprint arXiv:2311.04163*, 2023.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

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

Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In *International Conference on Machine Learning*, pp. 5827–5837. PMLR, 2019.

Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, Roni Paiss, and Joshua Susskind. The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon, 2022.

Hoang Tran and Ashok Cutkosky. Better sgd using second-order momentum. *Advances in Neural Information Processing Systems*, 35:3530–3541, 2022a.

Hoang Tran and Ashok Cutkosky. Momentum aggregation for private non-convex erm. *Advances in Neural Information Processing Systems*, 35:10996–11008, 2022b.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. doi: 10.1145/2641190.2641198. URL `http://doi.acm.org/10.1145/2641190.2641198`.

Bohan Wang, Huishuai Zhang, Zhiming Ma, and Wei Chen. Convergence of adagrad for non-convex objectives: Simple proofs and relaxed assumptions. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 161–190. PMLR, 2023.

Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. *Journal of Machine Learning Research*, 21(219):1–30, 2020.

Chen Xing, Devansh Arpit, Christos Tsirigotis, and Yoshua Bengio. A walk with sgd, 2018.

Zhewei Yao, Amir Gholami, Sheng Shen, Kurt Keutzer, and Michael W Mahoney. Adahessian: An adaptive second order optimizer for machine learning. *arXiv preprint arXiv:2006.00719*, 2020.

Zhewei Yao, Amir Gholami, Sheng Shen, Mustafa Mustafa, Kurt Keutzer, and Michael Mahoney. Adahessian: An adaptive second order optimizer for machine learning. In *proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 10665–10673, 2021.

Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018a. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/90365351ccc7437a1309dc64e4db32a3-Paper.pdf.

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

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. *arXiv preprint arXiv:1905.11881*, 2019.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33:15383–15393, 2020a.

Jingzhao Zhang, Hongzhou Lin, Stefanie Jegelka, Ali Jadbabaie, and Suvrit Sra. Complexity of finding stationary points of nonsmooth nonconvex functions. 2020b.

Qinzi Zhang and Ashok Cutkosky. Random scaling and momentum for non-smooth non-convex optimization, 2024. URL https://arxiv.org/abs/2405.09742.

Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Dynamic regret of convex and smooth functions. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12510–12520. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/939314105ce8701e67489642ef4d49e8-Paper.pdf.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 928–936, 2003.