How Memory in Optimization Algorithms Implicitly Modifies the Loss

Matias D. Cattaneo[*]
Princeton University
cattaneo@princeton.edu

Boris Shigida[*]
Princeton University
bs1624@princeton.edu

Abstract

In modern optimization methods used in deep learning, each update depends on the history of previous iterations, often referred to as *memory*, and this dependence decays fast as the iterates go further into the past. For example, gradient descent with momentum has exponentially decaying memory through exponentially averaged past gradients. We introduce a general technique for identifying a memoryless algorithm that approximates an optimization algorithm with memory. It is obtained by replacing all past iterates in the update by the current one, and then adding a correction term arising from memory (also a function of the current iterate). This correction term can be interpreted as a perturbation of the loss, and the nature of this perturbation can inform how memory implicitly (anti-)regularizes the optimization dynamics. As an application of our theory, we find that Lion does not have the kind of implicit anti-regularization induced by memory that AdamW does, providing a theory-based explanation for Lion's better generalization performance recently documented [13]. Empirical evaluations confirm our theoretical findings.

1 Introduction

Many optimization methods used in deep learning are first-order methods with exponentially decaying memory. For example, adding "momentum" to gradient descent (GD) is a well-established practice to make training smoother and convergence faster (e. g. Krizhevsky et al. [39]). Adaptive methods such as Adam [36], RMSProp [61], AdamW [46], and AdaFactor [57], which are commonly used to train large language models [27, 21, 15], all have exponentially decaying memory. Despite the popularity of such optimization methods, there is little theoretical knowledge about the implicit bias memory introduces to them (potentially informing what regions of the loss space the method takes the iterates to, what minima they converge to, how such minima influence the generalization of the trained model, and so on). In this article, we introduce a general framework for identifying such biases.

We study a general class of optimization algorithms described by the following iteration

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n)} - h\boldsymbol{F}^{(n)}(\boldsymbol{\theta}^{(n)}, \dots, \boldsymbol{\theta}^{(0)}), \tag{1}$$

where $\boldsymbol{\theta}^{(n)} \in \mathbb{R}^d$ are the (evolving) parameters of the machine learning model, $\boldsymbol{\theta}^{(0)}$ is some initial condition, h is the step size or learning rate, and the functions $\boldsymbol{F}^{(n)}$ map from (some subset of) $(\mathbb{R}^d)^{n+1}$ to $\mathbb{R}^d$ and are allowed to be different at each iteration. The right-hand side in Equation (1) depends on the whole history of previous iterates, which means the algorithm has memory.

[*]Authors are listed alphabetically by last name.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

For many algorithms used in practice, dependence on the history comes in one specific form: by using what we call "*momentum variables*", that is, exponential averages of some functions of the iterate $\boldsymbol{\theta}^{(n)}$ (usually, more specifically, functions of the loss gradient). We present five leading examples to illustrate this point.

Example 1.1 (Heavy-ball momentum gradient descent; Polyak [54]). This optimizer can be written in the form (1) with

$$\begin{aligned} \boldsymbol{F}^{(n)}(\boldsymbol{\theta}^{(n)}, \dots, \boldsymbol{\theta}^{(0)}) &= \boldsymbol{m}_1^{(n+1)}, \\ \text{where} \quad \boldsymbol{m}_1^{(n+1)} &= \sum_{k=0}^{n} \beta^{n-k} \nabla \mathcal{L}(\boldsymbol{\theta}^{(k)}), \end{aligned} \tag{2}$$

for some initial condition $\boldsymbol{\theta}^{(0)}$, and where $\beta \in [0, 1)$ is the momentum parameter, $\mathcal{L}$ is the loss function to be optimized, and $\nabla \mathcal{L}$ is its gradient. □

The optimizer in Example 1.1 is often just referred to as GD with momentum, where the exponential sum $\boldsymbol{m}_1^{(n+1)}$ in Equation (2) is the "momentum variable": it exponentially averages past gradients. The aforementioned optimizer is well-known and often used for training recurrent neural networks and convolutional neural networks, but it underperforms adaptive optimizers when training other architectures such as transformers [68, 45, 2, 40]. The following modification is also commonly used (this formulation is taken from Choi et al. [14] and matches the standard PyTorch implementation).

Example 1.2 (Nesterov's accelerated gradient descent; Nesterov [53]). This optimizer can be written in the form (1) with

$$\begin{aligned} \boldsymbol{F}^{(n)}(\boldsymbol{\theta}^{(n)}, \dots, \boldsymbol{\theta}^{(0)}) &= \boldsymbol{m}_1^{(n+1)} + \boldsymbol{m}_2^{(n+1)}, \\ \text{where} \quad \boldsymbol{m}_1^{(n+1)} &= \beta \sum_{k=0}^{n} \beta^{n-k} \nabla \mathcal{L}(\boldsymbol{\theta}^{(k)}), \\ \boldsymbol{m}_2^{(n+1)} &= \nabla \mathcal{L}(\boldsymbol{\theta}^{(n)}), \end{aligned}$$

for some initial condition $\boldsymbol{\theta}^{(0)}$, and where $\beta \in [0, 1)$ is the momentum parameter, $\mathcal{L}$ is the loss function to be optimized, and $\nabla \mathcal{L}$ is its gradient. □

The next example presents the most prominent adaptive optimizer, nowadays commonly used for training large language models [27, 21].

Example 1.3 (AdamW; Loshchilov and Hutter [46]). The optimizer can be written in the form (1) with

$$\begin{aligned} \boldsymbol{F}^{(n)}(\boldsymbol{\theta}^{(n)}, \dots, \boldsymbol{\theta}^{(0)}) &= \frac{\boldsymbol{m}_1^{(n+1)}}{\sqrt{\boldsymbol{m}_2^{(n+1)} + \varepsilon}} + \boldsymbol{m}_3^{(n+1)}, \\ \text{where} \quad \boldsymbol{m}_1^{(n+1)} &= \frac{1 - \beta_1}{1 - \beta_1^{n+1}} \sum_{k=0}^{n} \beta_1^{n-k} \nabla \mathcal{L}(\boldsymbol{\theta}^{(k)}), \\ \boldsymbol{m}_2^{(n+1)} &= \frac{1 - \beta_2}{1 - \beta_2^{n+1}} \sum_{k=0}^{n} \beta_2^{n-k} \big(\nabla \mathcal{L}(\boldsymbol{\theta}^{(k)})\big)^2, \\ \boldsymbol{m}_3^{(n+1)} &= \lambda \boldsymbol{\theta}^{(n)}, \end{aligned}$$

for some initial condition $\boldsymbol{\theta}^{(0)}$, and where $0 \leqslant \beta_1, \beta_2 < 1$ are momentum parameters, $\varepsilon > 0$ is a numerical stability parameter, $0 < \lambda < 1$ is a weight decay parameter, and the squares and square roots are taken component-wise. □

In Example 1.3, $\boldsymbol{m}_1^{(n+1)}$ and $\boldsymbol{m}_2^{(n+1)}$ are also "momentum variables": exponentially averaged gradients and exponentially averaged squared gradient components respectively, with coefficients in front of the sum, such as $(1 - \beta_1)(1 - \beta_1^{n+1})^{-1}$, providing "bias correction" [36]. The variable $\boldsymbol{m}_3^{(n+1)}$ here is a degenerate "momentum variable", with memory decaying infinitely fast.

The following modification incorporates Nesterov's momentum into AdamW. This formulation is taken from Choi et al. [14] (except here ε is inside the square root in the denominator).

Example 1.4 (NAdam with decoupled weight decay; Dozat [22]). The optimizer can be written in the form (1) with

$$\begin{aligned}
\boldsymbol{F}^{(n)}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(0)}) &= \frac{\beta_1 \boldsymbol{m}_1^{(n+1)} + (1-\beta_1)\boldsymbol{m}_4^{(n+1)}}{\sqrt{\boldsymbol{m}_2^{(n+1)}+\varepsilon}} + \boldsymbol{m}_3^{(n+1)}, \\
\text{where}\quad \boldsymbol{m}_1^{(n+1)} &= \frac{1-\beta_1}{1-\beta_1^{n+1}} \sum_{k=0}^{n} \beta_1^{n-k} \nabla \mathcal{L}(\boldsymbol{\theta}^{(k)}), \\
\boldsymbol{m}_2^{(n+1)} &= \frac{1-\beta_2}{1-\beta_2^{n+1}} \sum_{k=0}^{n} \beta_2^{n-k} \big(\nabla \mathcal{L}(\boldsymbol{\theta}^{(k)})\big)^2, \\
\boldsymbol{m}_3^{(n+1)} &= \lambda \boldsymbol{\theta}^{(n)}, \\
\boldsymbol{m}_4^{(n+1)} &= \nabla \mathcal{L}(\boldsymbol{\theta}^{(n)}),
\end{aligned}$$

for some initial condition $\boldsymbol{\theta}^{(0)}$, and where $0 \leqslant \beta_1, \beta_2 < 1$ are momentum parameters, $\varepsilon > 0$ is a numerical stability parameter, $0 < \lambda < 1$ is a weight decay parameter, and the squares and square roots are taken component-wise. □

As a final example, consider a new optimizer called Lion (Evo**L**ved S**i**gn M**o**me**n**tum), which was recently discovered by an evolutionary search, and then verified to generalize better than AdamW on a variety of tasks [13]. We consider a generalized version of the Lion algorithm.

Example 1.5 (Lion-$\mathcal{K}$; Chen et al. [12]). The optimizer can be written in the form of (1) with

$$\begin{aligned}
\boldsymbol{F}^{(n)}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(0)}) &= -\nabla \mathcal{K}(\boldsymbol{m}_1^{(n+1)} + \boldsymbol{m}_2^{(n+1)}) + \boldsymbol{m}_3^{(n+1)}, \\
\text{where}\quad \boldsymbol{m}_1^{(n+1)} &= -(1-\rho_2)\frac{\rho_1}{\rho_2} \sum_{k=0}^{n} \rho_2^{n-k} \nabla \mathcal{L}(\boldsymbol{\theta}^{(k)}), \\
\boldsymbol{m}_2^{(n+1)} &= -\left(1 - \frac{\rho_1}{\rho_2}\right) \nabla \mathcal{L}(\boldsymbol{\theta}^{(n)}), \\
\boldsymbol{m}_3^{(n+1)} &= \lambda \boldsymbol{\theta}^{(n)},
\end{aligned} \tag{3}$$

for some initial condition $\boldsymbol{\theta}^{(0)}$, and where $0 \leqslant \rho_1, \rho_2 < 1$ are Lion's momentum parameters, $\lambda > 0$ is a weight decay parameter, $\mathcal{K}\colon \mathbb{R}^d \to \mathbb{R}$ is some convex function, and $\nabla \mathcal{K}$ is its subgradient. □

We choose the letter ρ rather than β for Lion's momentum parameters because they are not precisely parameters controlling the speed of exponential decay in "momentum variables", as explained in Section C. Ordinary Lion corresponds to $\mathcal{K}(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$ and $\nabla \mathcal{K}(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{x})$ in Example 1.5, where the sign function is understood component-wise. We consider the generalized Lion-$\mathcal{K}$ algorithm because it covers a few known algorithms as special cases: see Table 1 and Section 3.1 in Chen et al. [12]. In fact, it also includes Example 1.1 as a special case by taking $\mathcal{K}(\boldsymbol{x}) = \|\boldsymbol{x}\|^2/2$, $\rho_1 = \rho_2$, and $\lambda = 0$, but we will deal with that important specific example separately for clarity.

It is reasonable to expect that adding exponentially decaying memory to an algorithm in such a way as described above (for example, replacing the gradient with exponentially averaged past gradients) changes the optimization dynamics, thereby affecting the performance of the trained model. The technique we introduce identifies *how* the iteration evolution changes when memory is added. This technique starts with an iteration having memory, and replaces it by a memoryless iteration that approximates the original one, provided a correction term is added. Specifically, we start with algorithm (1), and then construct a corresponding new memoryless iteration:

$$\boldsymbol{\theta}^{(n+1)} = \underbrace{\boldsymbol{\theta}^{(n)} - h\boldsymbol{F}^{(n)}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(0)})}_{\text{depends on the whole history } \boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(0)}}$$

$$\tilde{\boldsymbol{\theta}}^{(n+1)} = \underbrace{\tilde{\boldsymbol{\theta}}^{(n)} - h\big[\boldsymbol{F}^{(n)}(\tilde{\boldsymbol{\theta}}^{(n)}) + \overbrace{\boldsymbol{M}^{(n)}(\tilde{\boldsymbol{\theta}}^{(n)})}^{\text{correction}}\big]}_{\text{only depends on } \tilde{\boldsymbol{\theta}}^{(n)} \text{ (no memory)}}, \tag{4}$$

where we slightly abuse notation and put

$$\boldsymbol{F}^{(n)}(\tilde{\boldsymbol{\theta}}^{(n)}) \equiv \boldsymbol{F}^{(n)}(\underbrace{\tilde{\boldsymbol{\theta}}^{(n)}, \ldots, \tilde{\boldsymbol{\theta}}^{(n)}}_{n+1 \text{ times}}),$$

and where the function $\boldsymbol{M}^{(n)}(\boldsymbol{\theta})$ captures a correction due to the presence of memory. We then prove an explicit bound on the approximation error $\|\boldsymbol{\theta}^{(n)} - \tilde{\boldsymbol{\theta}}^{(n)}\|$, as a function of the learning rate h. Interpreting the correction term can sometimes generate predictions on whether memory helps or hurts generalization of first-order methods with momentum.

Our theory only relies on memory decaying sufficiently fast, not necessarily in the form of momentum variables, and thus covers all the examples listed above and many others, while also allowing for both full-batch and mini-batch training. Section 2 first presents a heuristic discussion of our proposed technique focusing on the simplest possible case for clarity: GD with momentum (Example 1.1). Then, Section 3 presents our main theoretical contribution, which we specialize and apply to all the listed examples in Sections C and D.

Depending on specific optimization algorithm considered, our general result can lead to different practical conclusions. As a substantive application, Section 4 studies AdamW (Example 1.3) and Lion-$\mathcal{K}$ (Example 1.5), and demonstrates that Lion does not suffer from the anti-regularization effect that AdamW's memory has, which predicts better generalization of (full-batch) Lion. Section 5 (with details in Section E) discusses further implications of our main theoretical result: constructing modified equations, and identifying implicit regularization by noise in mini-batch training. Section 6 is devoted to limitations and future directions.

Due to compute constraints, we consider a large-scale empirical testing of this theory out of scope of this paper. However, we provide a preliminary empirical illustration in Section J (with some details, compute resources and licenses in Section K).

1.1 Notation

We use standard notations for the ℓ_p norm of a vector $\|\boldsymbol{v}\|_p = \left(\sum_i |v_i|^p\right)^{1/p}$; the infinity-norm is defined as $\|\boldsymbol{v}\|_\infty = \max_i |v_i|$; finally, the norm without indices is by default Euclidean: $\|\boldsymbol{v}\| \equiv \|\boldsymbol{v}\|_2$. When we write $\boldsymbol{u}_{n,h} = O(g(h))$, where $g(h)$ is some fixed function of h and $\boldsymbol{u}_{n,h}$ is some sequence of vectors possibly depending on h, we mean that there is a constant C not depending on h or n such that $\|\boldsymbol{u}_{n,h}\| \leqslant Cg(h)$. We will contract repeating arguments when convenient, e. g. instead of $\boldsymbol{F}^{(n)}(\boldsymbol{\theta}, \ldots, \boldsymbol{\theta})$ we will write just $\boldsymbol{F}^{(n)}(\boldsymbol{\theta})$. We will use notation $\mathcal{L}(\cdot)$ for the loss and $\nabla\mathcal{L}(\cdot)$ for its gradient, h for the learning rate.

2 Building Intuition: Memory Regularizes GD with Momentum

We provide a heuristic explanation of our technique, considering the simplest algorithm with exponentially decaying memory: heavy-ball momentum GD (Example 1.1). As explained above, we would like to remove the dependence of the right-hand side in

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n)} - h\sum_{k=0}^{n} \beta^{n-k} \nabla\mathcal{L}(\boldsymbol{\theta}^{(k)}) \tag{5}$$

on the "past" iterates $\boldsymbol{\theta}^{(n-1)}, \ldots, \boldsymbol{\theta}^{(0)}$, leaving only the dependence on the "current" iterate $\boldsymbol{\theta}^{(n)}$. Let us represent "past" iterates through the "current" one. First, write

$$\boldsymbol{\theta}^{(n-1)} = \boldsymbol{\theta}^{(n)} + h\sum_{b=0}^{n-1} \beta^b \nabla\mathcal{L}(\boldsymbol{\theta}^{(n-1-b)}) = \boldsymbol{\theta}^{(n)} + h\sum_{b=0}^{n-1} \beta^b \nabla\mathcal{L}(\boldsymbol{\theta}^{(n)}) + O(h^2),$$

where the second equality relies on exponential averaging to replace historical iterates with $\boldsymbol{\theta}^{(n)}$, influencing only higher-order terms. Similarly,

$$\boldsymbol{\theta}^{(n-2)} = \boldsymbol{\theta}^{(n-1)} + h\sum_{b=0}^{n-2} \beta^b \nabla\mathcal{L}(\boldsymbol{\theta}^{(n-2-b)})$$

$$= \boldsymbol{\theta}^{(n-1)} + h\sum_{b=0}^{n-2}\beta^b\nabla\mathcal{L}(\boldsymbol{\theta}^{(n)}) + O(h^2),$$
$$= \boldsymbol{\theta}^{(n)} + h\Big\{\sum_{b=0}^{n-1}\beta^b + \sum_{b=0}^{n-2}\beta^b\Big\}\nabla\mathcal{L}(\boldsymbol{\theta}^{(n)}) + O(h^2),$$

where the last equality follows by inserting the expression for $\boldsymbol{\theta}^{(n-1)}$. Continue like this up to

$$\boldsymbol{\theta}^{(n-k)} = \boldsymbol{\theta}^{(n)} + h\sum_{l=1}^{k}\sum_{b=0}^{n-l}\beta^b\nabla\mathcal{L}(\boldsymbol{\theta}^{(n)}) + O(k^2h^2),$$

where the k^2 provides an estimate on the accumulation of error terms of order h^2.

We have now represented all the historical iterates through the current one. Combining it with Taylor expansion around $\boldsymbol{\theta}^{(n)}$ in Equation (5), we obtain

$$\begin{aligned}\boldsymbol{\theta}^{(n+1)} &= \boldsymbol{\theta}^{(n)} - h\sum_{k=0}^{n}\beta^k\Big\{\nabla\mathcal{L}(\boldsymbol{\theta}^{(n)}) + h\nabla^2\mathcal{L}(\boldsymbol{\theta}^{(n)})\sum_{l=1}^{k}\sum_{b=0}^{n-l}\beta^b\nabla\mathcal{L}(\boldsymbol{\theta}^{(n)}) + O(k^2h^2)\Big\}\\ &= \boldsymbol{\theta}^{(n)} - h\frac{1+o_n(1)}{1-\beta}\nabla\mathcal{L}(\boldsymbol{\theta}^{(n)}) - h^2\frac{\beta[1+o_n(1)]}{(1-\beta)^3}\nabla^2\mathcal{L}(\boldsymbol{\theta}^{(n)})\nabla\mathcal{L}(\boldsymbol{\theta}^{(n)}) + O(h^3),\end{aligned}$$

where $o_n(1)$ terms go to zero exponentially fast as $n\to\infty$. Now using $\nabla^2\mathcal{L}(\boldsymbol{\theta})\nabla\mathcal{L}(\boldsymbol{\theta}) = (1/2)\nabla\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2$, we obtain that heavy-ball momentum GD is close to ordinary GD (no momentum) with a different step size and different loss, given by

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n)} - \frac{h}{1-\beta}\nabla\tilde{\mathcal{L}}(\boldsymbol{\theta}), \quad \text{where} \quad \tilde{\mathcal{L}}(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}) + \frac{h\beta}{2(1-\beta)^2}\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2. \tag{6}$$

The term $\frac{h\beta}{2(1-\beta)^2}\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2$ that is added implicitly to the loss by the momentum can be interpreted as implicit regularization. Since β is usually taken to be close to one, the term strongly penalizes the squared norm of the gradient. There is empirical evidence that such penalization improves generalization [6, 59, 26]. In fact, this term (up to coefficients) can be interpreted as a first-order approximation of ℓ_2 sharpness [26], which suggests that it moves the trajectory towards "flatter" minima; this is often thought to improve generalization [25].

A much more fine-grained analysis of this algorithm is provided in [10] where, in particular, the modified loss is described to arbitrary precision rather than just to first order in h.

3 General Theory: The Effect of Memory

The general form of an optimization algorithm with memory is given by Equation (1). The only property of memory we use is that it (uniformly in n) decays exponentially fast, as made precise by Assumption 3.1 below. Openness and convexity of the domain of optimization $\mathcal{D}$, that is, where all $\{\boldsymbol{\theta}^{(n)}\}$ will be assumed to lie, are innocuous assumptions (e.g., $\mathbb{R}^d$ is open and convex); we impose them to avoid technicalities with differentiation and Taylor expansion.

Assumption 3.1 (Memory Decay)**.** *Let $\mathcal{D}$ be an open convex domain in $\mathbb{R}^d$. Let $\{\boldsymbol{F}^{(n)}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(0)})\}_{n=0}^{\infty}$ be a family of functions $\mathcal{D}^{n+1}\to\mathbb{R}^d$, two times continuously differentiable on their respective domains, such that for any $n\in\mathbb{Z}_{\geqslant 0}$, $k_1,k_2\in\{0,\dots,n\}$, $r,i,j\in\{1,\dots,d\}$,*

$$|F_r^{(n)}| \leqslant \gamma_{-1}, \qquad \left|\frac{\partial F_r^{(n)}}{\partial\theta_i^{(n-k_1)}}\right| \leqslant \gamma_{k_1}, \qquad \left|\frac{\partial^2 F_r^{(n)}}{\partial\theta_i^{(n-k_1)}\partial\theta_j^{(n-k_2)}}\right| \leqslant \gamma_{k_1,k_2},$$

where $\boldsymbol{F}^{(n)} = (F_1^{(n)},\dots,F_d^{(n)})^{\mathsf{T}}$, and γ_{-1}, γ_{k_1} and γ_{k_1,k_2} are families of positive reals (not depending on n) satisfying

$$\sum_{k_1=1}^{\infty}\gamma_{k_1}k_1^2 + \sum_{k_1,k_2=1}^{\infty}\gamma_{k_1,k_2}k_1k_2 < \infty. \tag{7}$$

3.1 Deriving the Memoryless Approximation

By Taylor expansion with the Lagrange remainder,

$$\begin{aligned}
&F_r^{(n)}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(0)}) - F_r^{(n)}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(n)}) \\
&\quad= \sum_{k=1}^{n} \big(\boldsymbol{\theta}^{(n-k)} - \boldsymbol{\theta}^{(n)}\big)^{\mathsf{T}} \frac{\partial F_r^{(n)}}{\partial \boldsymbol{\theta}^{(n-k)}}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(n)}) \\
&\qquad+ \frac{1}{2} \sum_{k_1,k_2=1}^{n} \big(\boldsymbol{\theta}^{(n-k_1)} - \boldsymbol{\theta}^{(n)}\big)^{\mathsf{T}} \frac{\partial^2 F_r^{(n)}}{\partial \boldsymbol{\theta}^{(n-k_1)} \partial \boldsymbol{\theta}^{(n-k_2)}}(\boldsymbol{\zeta}) \big(\boldsymbol{\theta}^{(n-k_2)} - \boldsymbol{\theta}^{(n)}\big) \\
&\quad= \sum_{k=1}^{n} \big(\boldsymbol{\theta}^{(n-k)} - \boldsymbol{\theta}^{(n)}\big)^{\mathsf{T}} \frac{\partial F_r^{(n)}}{\partial \boldsymbol{\theta}^{(n-k)}}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(n)}) + O(h^2),
\end{aligned} \tag{8}$$

where $\boldsymbol{\zeta}$ is some point on the segment between $\big(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(0)}\big)$ and $\big(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(n)}\big)$; in the last step we used Assumption 3.1, $\boldsymbol{\theta}^{(n-k_1)} - \boldsymbol{\theta}^{(n)} = O(k_1 h)$, and $\boldsymbol{\theta}^{(n-k_2)} - \boldsymbol{\theta}^{(n)} = O(k_2 h)$.

Next, write

$$\begin{aligned}
\boldsymbol{\theta}^{(n-k)} - \boldsymbol{\theta}^{(n)} &= \sum_{s=n-k}^{n-1} \big(\boldsymbol{\theta}^{(s)} - \boldsymbol{\theta}^{(s+1)}\big) \\
&= h \sum_{s=n-k}^{n-1} \boldsymbol{F}^{(s)}(\boldsymbol{\theta}^{(s)},\dots,\boldsymbol{\theta}^{(0)}) = h \sum_{s=n-k}^{n-1} \boldsymbol{F}^{(s)}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(n)}) + O(k^2 h^2),
\end{aligned}$$

where in the last step we used $\boldsymbol{F}^{(s)}(\boldsymbol{\theta}^{(s)},\dots,\boldsymbol{\theta}^{(0)}) - \boldsymbol{F}^{(s)}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(n)}) = O((n-s)h)$, which follows from Taylor expansion and Assumption 3.1. Insert this into Equation (8) and use Assumption 3.1 again to continue:

$$\begin{aligned}
&F_r^{(n)}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(0)}) - F_r^{(n)}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(n)}) \\
&\quad= h \sum_{k=1}^{n} \frac{\partial F_r^{(n)}}{\partial \boldsymbol{\theta}^{(n-k)}}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(n)})^{\mathsf{T}} \sum_{s=n-k}^{n-1} \boldsymbol{F}^{(s)}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(n)}) + O(h^2).
\end{aligned}$$

We conclude that the original numerical iteration can be rewritten in the form (4)*,* where the linear in h correction function is defined as $\boldsymbol{M}^{(n)} = (M_1^{(n)},\dots,M_d^{(n)})^{\mathsf{T}}$ with

$$\boxed{M_r^{(n)}(\boldsymbol{\theta}) := h \sum_{k=1}^{n} \frac{\partial F_r^{(n)}}{\partial \boldsymbol{\theta}^{(n-k)}}(\boldsymbol{\theta})^{\mathsf{T}} \sum_{s=n-k}^{n-1} \boldsymbol{F}^{(s)}(\boldsymbol{\theta}).} \tag{9}$$

The derivation of the memoryless iteration is now complete. Although not a proof yet, it is the first step towards the approximation bound constituting our main theoretical result.

3.2 Approximation Bound

An argument similar to the derivation in Section 3.1 can be made to obtain the following result.

Theorem 3.2 (Memoryless approximation: 1-step error bound)**.** *Under Assumption 3.1, there exists a discrete memoryless iteration* $\{\tilde{\boldsymbol{\theta}}^{(n)}\}_{n=0}^{\infty}$ *satisfying* (4) *with initial condition* $\tilde{\boldsymbol{\theta}}^{(0)} = \boldsymbol{\theta}^{(0)}$*, correction function defined in Equation* (9)*, and a constant* C_1 *not depending on* h*, such that*

$$\sup_{n \in \mathbb{Z}_{\geqslant 0}} \big\| \tilde{\boldsymbol{\theta}}^{(n+1)} - \tilde{\boldsymbol{\theta}}^{(n)} + h \boldsymbol{F}^{(n)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(0)}) \big\|_{\infty} \leqslant C_1 h^3.$$

The proof is available in Section F.

The importance of this one-step approximation result is that it allows to bound the global error between the memoryfull iteration $\boldsymbol{\theta}^{(n)}$ and memoryless iteration $\tilde{\boldsymbol{\theta}}^{(n)}$ on a finite time horizon.

Corollary 3.3 (Global error bound on a finite "time" horizon)**.** *In the setting of Theorem 3.2, let* $\{\boldsymbol{\theta}^{(n)}\}_{n\in\mathbb{Z}_{\geqslant 0}}$ *be the sequence of vectors generated by the iteration in Equation (1) with initial condition* $\boldsymbol{\theta}^{(0)}$*. Let* $T \geqslant 0$ *be a fixed "time" horizon. (The number of iterations considered is not* T *but* $\lfloor T/h \rfloor$*.) Then there exists a constant* C_2*, depending on* T *but independent of* h*, such that* $\max_{n\in[0:\lfloor T/h\rfloor]} \big\|\boldsymbol{\theta}^{(n)} - \tilde{\boldsymbol{\theta}}^{(n)}\big\|_\infty \leqslant C_2 h^2$.

The proof is in Section G.

4 Application: the Effect of Memory on AdamW, Lion and Signum

We first study AdamW with memory by an application of Theorem 3.2 and Corollary 3.3. Neglecting coefficients decaying to zero exponentially fast, we have

$$\boldsymbol{\theta}^{(n+1)} = (1-\lambda h)\boldsymbol{\theta}^{(n)} - h\Bigg(\underbrace{\frac{\nabla\mathcal{L}(\boldsymbol{\theta}^{(n)})}{\sqrt{\big(\nabla\mathcal{L}(\boldsymbol{\theta}^{(n)})\big)^2+\varepsilon}}}_{\approx\,\mathrm{sign}(\nabla\mathcal{L}(\boldsymbol{\theta}^{(n)}))} + \boldsymbol{M}^{(n)}(\boldsymbol{\theta}^{(n)})\Bigg),$$

where $\boldsymbol{M}^{(n)}(\boldsymbol{\theta})$ is given by

$$h\Bigg(\frac{\beta_1(1-\beta_1)^{-1}-\beta_2(1-\beta_2)^{-1}}{\big(\big|\nabla\mathcal{L}(\boldsymbol{\theta})\big|^2+\varepsilon\big)^{1/2}} + \varepsilon\frac{\beta_2(1-\beta_2)^{-1}}{\big(\big|\nabla\mathcal{L}(\boldsymbol{\theta})\big|^2+\varepsilon\big)^{3/2}}\Bigg)\big(\nabla\|\nabla\mathcal{L}(\boldsymbol{\theta})\|_{1,\varepsilon} + \lambda\nabla^2\mathcal{L}(\boldsymbol{\theta})\boldsymbol{\theta}\big).$$

Here $\|\cdot\|_{1,\varepsilon}$ is the perturbed one-norm defined as $\|\boldsymbol{v}\|_{1,\varepsilon} := \sum_{i=1}^d \sqrt{v_i^2+\varepsilon}$. Taking ε to zero, we can write this in the form of preconditioned gradient descent (with decoupled weight decay):

$$\boldsymbol{\theta}^{(n+1)} = (1-\lambda h)\boldsymbol{\theta}^{(n)} - h\frac{\nabla\tilde{\mathcal{L}}(\boldsymbol{\theta}^{(n)})}{\big|\big(\nabla\mathcal{L}(\boldsymbol{\theta}^{(n)})\big)\big|},$$

where

$$\tilde{\mathcal{L}}(\boldsymbol{\theta}) = \overbrace{\big[1+\lambda\big(\tfrac{\beta_2}{1-\beta_2}-\tfrac{\beta_1}{1-\beta_1}\big)h\big]\mathcal{L}(\boldsymbol{\theta})}^{\text{rescaled loss}} - h\big(\tfrac{\beta_2}{1-\beta_2}-\tfrac{\beta_1}{1-\beta_1}\big)\overbrace{\big(\|\nabla\mathcal{L}(\boldsymbol{\theta})\|_1 + \lambda\nabla\mathcal{L}(\boldsymbol{\theta})^\mathsf{T}\boldsymbol{\theta}\big)}^{(*)}$$

is the modified loss. Assuming $\beta_2 > \beta_1$, we see that $(*)$ is implicitly anti-penalized. By Theorem 1.1 in [66], full-batch AdamW converges to a KKT point of the constrained optimization $\min_{\|\boldsymbol{\theta}\|_\infty\leqslant 1/\lambda}\mathcal{L}(\boldsymbol{\theta})$. If $\|\boldsymbol{\theta}\|_\infty \leqslant 1/\lambda$, then the norm $\|\nabla\mathcal{L}(\boldsymbol{\theta})\|_1$ dominates the term $\lambda\nabla\mathcal{L}(\boldsymbol{\theta})^\mathsf{T}\boldsymbol{\theta}$ in absolute value, so the main effect of memory is anti-penalizing the one-norm of the gradient. Thus, *if weight decay is sufficiently small, memory anti-regularizes (large-batch) AdamW*. Incidentally, by Lemma 3.8 in that work, $\boldsymbol{\theta}$ is a KKT point of this optimization problem if and only if the constraint is satisfied and $(*)=0$. This is a generalization of the observation that the correction term is zero if and only if the point is stationary, true of simpler full-batch algorithms (for Adam with $\lambda=0$ it follows from the above; for full-batch GD with momentum it is clear from (6)).

Consider now Lion-$\mathcal{K}$ (Example 1.5). Neglecting terms going to zero exponentially fast as $n\to\infty$, the memoryless iteration is

$$\boldsymbol{\theta}^{(n+1)} = (1-h\lambda)\boldsymbol{\theta}^{(n)} - h\big[-\nabla\mathcal{K}\big(-\nabla\mathcal{L}(\boldsymbol{\theta}^{(n)})\big) + \boldsymbol{M}^{(n)}(\boldsymbol{\theta}^{(n)})\big]$$
$$\text{with}\quad \boldsymbol{M}^{(n)}(\boldsymbol{\theta}) = -h\frac{\rho_1}{1-\rho_2}\nabla^2\mathcal{K}(-\nabla\mathcal{L}(\boldsymbol{\theta}))\nabla^2\mathcal{L}(\boldsymbol{\theta})\big[\nabla\mathcal{K}(-\nabla\mathcal{L}(\boldsymbol{\theta})) - \lambda\boldsymbol{\theta}\big].$$

As mentioned above, ordinary Lion is recovered by setting $\mathcal{K}(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$. This function is not differentiable, so let us replace it with the smooth convex approximation $\|\boldsymbol{x}\|_{1,\varepsilon}$, where ε is a small positive constant. The results of Section 3 can be applied, and the memoryless iteration is

$$\boldsymbol{\theta}^{(n+1)} = (1-\lambda h)\boldsymbol{\theta}^{(n)} - h\Bigg[\frac{\nabla\mathcal{L}(\boldsymbol{\theta}^{(n)})}{\big(|\nabla\mathcal{L}(\boldsymbol{\theta}^{(n)})|^2+\varepsilon\big)^{1/2}} + \boldsymbol{M}^{(n)}(\boldsymbol{\theta}^{(n)})\Bigg]$$
$$\text{with}\quad M_r^{(n)}(\boldsymbol{\theta}) = h\frac{\rho_1}{1-\rho_2}\frac{\varepsilon}{\big(|\nabla_r\mathcal{L}(\boldsymbol{\theta})|^2+\varepsilon\big)^{3/2}}\nabla_r\big[\|\nabla\mathcal{L}(\boldsymbol{\theta})\|_{1,\varepsilon} + \lambda(\nabla\mathcal{L}(\boldsymbol{\theta})^\mathsf{T}\boldsymbol{\theta} - \mathcal{L}(\boldsymbol{\theta}))\big].$$

This term is small as long as ε is small. Therefore, better generalization of Lion on a number of tasks [13] may be partially attributed to the fact that memory does *not* anti-regularize Lion. In addition, notice that the correction terms are exactly the same for Adam with $\beta_1 = \beta_2 =: \beta$ and Lion with $\rho_1 = \rho_2 = \beta$. Since Lion with $\rho_1 = \rho_2$ is Signum [8], we provide a novel perspective on the similarity between Adam with $\beta_1 \approx \beta_2$ and Signum, a point discussed and verified empirically in [69].

5 Further Implications

5.1 Modified Equations

We have taken a very general algorithm (1) and converted it (under Assumption 3.1) to a memoryless iteration (4) with $O(h^2)$ uniform error bound on a finite time horizon (Corollary 3.3). Since this iteration has no memory, standard methods can be used to derive an ordinary differential equation (ODE) in the form $\dot{\boldsymbol{\theta}} = \boldsymbol{G}_h(\boldsymbol{\theta}) =: \boldsymbol{G}_1(\boldsymbol{\theta}) + h\boldsymbol{G}_2(h)$ whose continuous solution approximates this iteration and hence the initial algorithm (with the same approximation guarantee). Similarly to Section 2, the derivation is heuristic (although the approximation bound is easily made rigorous) and proceeds as follows. Relating the iteration number n of a discrete iteration and the time point $t = nh$ on a continuous trajectory, we would like the continuous trajectory to satisfy the same one-step relation as the discrete iteration, up to $O(h^3)$:

$$\boldsymbol{\theta}((n+1)h) = \boldsymbol{\theta}(nh) - h\big[\boldsymbol{F}^{(n)}(\boldsymbol{\theta}(nh), \dots, \boldsymbol{\theta}(nh)) + \boldsymbol{M}^{(n)}(\boldsymbol{\theta}(nh))\big] + O(h^3).$$

In fact, we will ensure it is true for nh replaced by any t:

$$\boldsymbol{\theta}(t+h) = \boldsymbol{\theta}(t) - h\big[\boldsymbol{F}^{(n)}(\boldsymbol{\theta}(t), \dots, \boldsymbol{\theta}(t)) + \boldsymbol{M}^{(n)}(\boldsymbol{\theta}(t))\big] + O(h^3). \tag{10}$$

But, using a Taylor expansion, and recalling that we are finding the trajectory satisfying $\dot{\boldsymbol{\theta}}(t) = \boldsymbol{G}_h(\boldsymbol{\theta}(t))$, hence $\ddot{\boldsymbol{\theta}}(t) = \nabla \boldsymbol{G}_h(\boldsymbol{\theta}(t))\dot{\boldsymbol{\theta}}(t)$, we have

$$\begin{aligned}
\boldsymbol{\theta}(t+h) &= \boldsymbol{\theta}(t) + h\dot{\boldsymbol{\theta}}(t) + \frac{h^2}{2}\ddot{\boldsymbol{\theta}}(t) + O(h^3) \\
&= \boldsymbol{\theta}(t) + h\big\{\boldsymbol{G}_1(\boldsymbol{\theta}(t)) + h\boldsymbol{G}_2(\boldsymbol{\theta}(t))\big\} \\
&\quad + \frac{h^2}{2}\big\{\nabla \boldsymbol{G}_1(\boldsymbol{\theta}(t))\boldsymbol{G}_1(\boldsymbol{\theta}(t)) + O(h)\big\} + O(h^3) \\
&= \boldsymbol{\theta}(t) + h\boldsymbol{G}_1(\boldsymbol{\theta}(t)) h^2 \left\{ \boldsymbol{G}_2(\boldsymbol{\theta}(t)) + \frac{\nabla \boldsymbol{G}_1(\boldsymbol{\theta}(t))\boldsymbol{G}_1(\boldsymbol{\theta}(t))}{2} \right\} + O(h^3).
\end{aligned}$$

In order to match (10), we need

$$\begin{aligned}
\boldsymbol{G}_1(\boldsymbol{\theta}) &= -\boldsymbol{F}^{(n)}(\boldsymbol{\theta}, \dots, \boldsymbol{\theta}), \\
\boldsymbol{G}_2(\boldsymbol{\theta}) &= -\left(\boldsymbol{M}^{(n)}(\boldsymbol{\theta})/h + \frac{\nabla \boldsymbol{G}_1(\boldsymbol{\theta})\boldsymbol{G}_1(\boldsymbol{\theta})}{2} \right).
\end{aligned}$$

So, apart from the correction term coming from memory, the ODE $\dot{\boldsymbol{\theta}} = \boldsymbol{G}_1(\boldsymbol{\theta}) + h\boldsymbol{G}_2(\boldsymbol{\theta})$ derived has another term

$$h^2 \frac{\nabla \boldsymbol{G}_1(\boldsymbol{\theta})\boldsymbol{G}_1(\boldsymbol{\theta})}{2}$$

arising from the fact that the algorithm is discrete.

For the example of full-batch heavy-ball momentum GD as in Section 2, where $\boldsymbol{G}_1(\boldsymbol{\theta}) = -(1-\beta)^{-1}\nabla \mathcal{L}(\boldsymbol{\theta})$ (ignoring coefficients going to zero exponentially fast in n), this additional term is equal to $h^2(1-\beta)^{-2}\nabla \|\nabla \mathcal{L}(\boldsymbol{\theta})\|^2/4$, providing additional implicit regularization. We recover the ODE derived by Kovachki and Stuart [37], Ghosh et al. [26]:

$$\dot{\boldsymbol{\theta}} = -\frac{\nabla \mathcal{L}(\boldsymbol{\theta})}{1-\beta} - h\frac{1+\beta}{(1-\beta)^3}\frac{\nabla \|\nabla \mathcal{L}(\boldsymbol{\theta})\|^2}{4}.$$

5.2 Mini-Batch Training

In specific cases, it is possible to identify the additional implicit regularization that is introduced to the algorithm by noise, if mini-batch training is used as opposed to full-batch. Assume that the form of $\boldsymbol{F}^{(n)}(\boldsymbol{\theta}^{(n)}, \dots, \boldsymbol{\theta}^{(0)})$ is given by

$$\boldsymbol{F}^{(n)}(\boldsymbol{\theta}^{(n)}, \dots, \boldsymbol{\theta}^{(0)}) = \sum_{k=0}^{n} \beta^k \boldsymbol{g}^{(\pi(n-k))}(\boldsymbol{\theta}^{(n-k)}),$$

where the $\{\boldsymbol{g}^{(k)}(\cdot)\}_{k=0}^{n}$ functions are uniformly bounded along with two derivatives, and π is a random permutation of $(0, \dots, n)$ (chosen among all such permutations with equal probability). The interpretation is that n is a large number of mini-batches in one epoch, and mini-batches are sampled randomly without replacement.

The correction term introduced by memory (9) is

$$M_r^{(n)}(\boldsymbol{\theta}) = h\beta \sum_{k=0}^{n-1} \beta^k \nabla g_r^{(\pi(n-1-k))}(\boldsymbol{\theta})^{\mathsf{T}} \sum_{l=1}^{k+1} \sum_{b=0}^{n-l} \beta^b \boldsymbol{g}^{(\pi(n-l-b))}(\boldsymbol{\theta}).$$

We can take the average $\mathbb{E}$ over all permutations π (re-orderings of mini-batches). Deferring some details to Section E, the result is that for large n

$$\begin{aligned}\mathbb{E}[M_r^{(n)}(\boldsymbol{\theta})]/h \approx{}& \frac{\beta}{(1-\beta)^3} \nabla g_r(\boldsymbol{\theta})^{\mathsf{T}} \boldsymbol{g}(\boldsymbol{\theta}) \\ &+ \frac{\beta}{(1-\beta)^2(1+\beta)} \mathbb{E}[(\nabla g_r^{(\pi(1))}(\boldsymbol{\theta}) - \nabla g_r(\boldsymbol{\theta}))^{\mathsf{T}} (\boldsymbol{g}^{(\pi(1))}(\boldsymbol{\theta}) - \boldsymbol{g}(\boldsymbol{\theta}))].\end{aligned}$$

The second term can be interpreted as implicit regularization by noise. For clarity, $\pi(1)$ is a uniformly distributed random variable over $\{0, \dots, n\}$, so this expectation is an average over mini-batch indices.

For example, take $\boldsymbol{g}^{(k)}(\boldsymbol{\theta}) = \nabla \mathcal{L}^{(k)}(\boldsymbol{\theta})$ the kth minibatch loss. Then we obtained that "on average" mini-batch GD with momentum is given by the iteration like (6), except the modified loss has an additional regularization term:

$$\tilde{\mathcal{L}}(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}) + \frac{h\beta}{2(1-\beta)^2} \|\nabla \mathcal{L}(\boldsymbol{\theta})\|^2 + \underbrace{\frac{h\beta}{2(1-\beta)(1+\beta)} \mathbb{E}\|\nabla \mathcal{L}^{(\pi(1))}(\boldsymbol{\theta}) - \nabla \mathcal{L}(\boldsymbol{\theta})\|^2}_{\text{regularization by mini-batch noise}}.$$

6 Limitations and Future Directions

The approximation bounds in Section 3.2 are in terms of the learning rate h, which means that h has to be sufficiently small for (our approximations and hence) the optimization trajectories to be close. This is a standard limitation in the literature. Fortunately, practically relevant learning rates are often small enough (especially mid-training, if a learning rate decay schedule is used). Additionally, there is a non-negligible effect of mini-batch noise on the picture we are describing in Section J; in particular, Lion does not necessarily outperform Adam if batches are small [13]. We are able to characterize this effect using similar techniques, but this is out of scope of this article and is a work in progress.

Taking a broader view, one may question the effect of (explicit or implicit) regularization on training progress and outcomes in deep learning, which is an intricate question not easily amenable to theoretical analysis [67, 1, 34]. Between 2023 and 2025, the mainstream paradigm in deep learning (especially large language models) is best described as scaling-centric rather than generalization-centric [65] which decreased interest in implicit regularization. The main purpose of this work is to introduce a general framework for identifying correction terms, whether they are treated as implicit regularizers or not. In the future, it is likely possible to build on it to study implications for the training dynamics, including characterizing the properties of the loss landscape around the optimizer's trajectory. Additionally, it is reasonable to anticipate that because of limited high-quality data, overfitting is about to become a widely important issue again, if not already [65, Section 7], [62, 35].

Finally, we discuss some of the most popular optimizers in recent years, but other important algorithms like Shampoo [30, 58] or its versions are also amenable to this analysis, and the approximation results in Section 3.2 hold for them (assuming a typical choice of momentum schemes). However, interpreting the higher-order corrections is not trivial, and we leave that as additional future work.

Acknowledgments

We thank Boris Hanin for his comments. Cattaneo gratefully acknowledges financial support from the National Science Foundation through DMS-2210561 and SES-2241575. We acknowledge the Princeton Research Computing resources, coordinated by the Princeton Institute for Computational Science and Engineering (PICSciE) and the Office of Information Technology's Research Computing.

References

[1] M. Andriushchenko, F. Croce, M. Müller, M. Hein, and N. Flammarion. A modern look at the relationship between sharpness and generalization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 840–902. PMLR, 2023. URL https://proceedings.mlr.press/v202/andriushchenko23a.html.

[2] R. Anil, V. Gupta, T. Koren, and Y. Singer. Memory efficient adaptive optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.neurips.cc/paper_files/paper/2019/hash/8f1fa0193ca2b5d2fa0695827d8270e9-Abstract.html.

[3] S. Arora, N. Cohen, W. Hu, and Y. Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/c0c783b5fc0d7d808f1d14a6e9c8280d-Abstract.html.

[4] S. Arora, Z. Li, and A. Panigrahi. Understanding gradient descent on the edge of stability in deep learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 948–1024. PMLR, 2022. URL https://proceedings.mlr.press/v162/arora22a.html.

[5] A. Barakat and P. Bianchi. Convergence and dynamical behavior of the ADAM algorithm for nonconvex stochastic optimization. *SIAM Journal on Optimization*, 31(1):244–274, 2021. doi: 10.1137/19M1263443.

[6] D. Barrett and B. Dherin. Implicit gradient regularization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3q5IqUrkcF.

[7] J. Bernstein and L. Newhouse. Old optimizer, new norm: An anthology. In *OPT 2024: Optimization for Machine Learning*.

[8] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar. signSGD: Compressed optimisation for non-convex problems. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 560–569. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/bernstein18a.html.

[9] A. Betti, G. Ciravegna, M. Gori, S. Melacci, K. Mottin, and F. Precioso. A new perspective on optimizers: leveraging Moreau-Yosida approximation in gradient-based learning. *Intelligenza Artificiale*, 18(2):301–311, 2024.

[10] M. D. Cattaneo and B. Shigida. Modified loss of momentum gradient descent: Fine-grained analysis. *arXiv preprint arXiv:2509.08483*, 2025.

[11] M. D. Cattaneo, J. M. Klusowski, and B. Shigida. On the implicit bias of Adam. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 5862–5906. PMLR, 2024. URL https://proceedings.mlr.press/v235/cattaneo24a.html.

[12] L. Chen, B. Liu, K. Liang, and Q. Liu. Lion secretly solves a constrained optimization: As lyapunov predicts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=e4xS9ZarDr.

[13] X. Chen, C. Liang, D. Huang, E. Real, K. Wang, H. Pham, X. Dong, T. Luong, C.-J. Hsieh, Y. Lu, and Q. V. Le. Symbolic discovery of optimization algorithms. In *Advances in Neural Information Processing Systems*, volume 36, pages 49205–49233, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/9a39b4925e35cf447ccba8757137d84f-Abstract.html.

[14] D. Choi, C. J. Shallue, Z. Nado, J. Lee, C. J. Maddison, and G. E. Dahl. On empirical comparisons of optimizers for deep learning. *arXiv preprint arXiv:1910.05446*, 2020. URL https://arxiv.org/abs/1910.05446.

[15] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel. PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. URL http://jmlr.org/papers/v24/22-1144.html.

[16] J. M. Cohen, B. Ghorbani, S. Krishnan, N. Agarwal, S. Medapati, M. Badura, D. Suo, D. Cardoze, Z. Nado, G. E. Dahl, et al. Adaptive gradient methods at the edge of stability. *arXiv preprint arXiv:2207.14484*, 2022.

[17] E. M. Compagnoni, T. Liu, R. Islamov, F. N. Proske, A. Orvieto, and A. Lucchi. Adaptive methods through the lens of SDEs: Theoretical insights on the role of noise. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ww3CLRhF1v.

[18] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Annual Meeting of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusID:57759363.

[19] A. Damian, T. Ma, and J. D. Lee. Label noise SGD provably prefers flat global minimizers. In *Advances in Neural Information Processing Systems*, volume 34, pages 27449–27461, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/e6af401c28c1790eaef7d55c92ab6ab6-Abstract.html.

[20] A. Damian, E. Nichani, and J. D. Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=nhKHA59gXz.

[21] DeepSeek-AI, A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Guo, D. Yang, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Zhang, H. Ding, H. Xin, H. Gao, H. Li, H. Qu, J. L. Cai, J. Liang, J. Guo, J. Ni, J. Li, J. Wang, J. Chen, J. Chen, J. Yuan, J. Qiu, J. Li, J. Song, K. Dong, K. Hu, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Xu, L. Xia, L. Zhao, L. Wang, L. Zhang, M. Li, M. Wang, M. Zhang, M. Zhang, M. Tang, M. Li, N. Tian, P. Huang, P. Wang, P. Zhang, Q. Wang, Q. Zhu, Q. Chen, Q. Du, R. J. Chen, R. L. Jin, R. Ge, R. Zhang, R. Pan, R. Wang, R. Xu, R. Zhang, R. Chen, S. S. Li, S. Lu, S. Zhou, S. Chen, S. Wu, S. Ye, S. Ye, S. Ma, S. Wang, S. Zhou, S. Yu, S. Zhou, S. Pan, T. Wang, T. Yun, T. Pei, T. Sun, W. L. Xiao, W. Zeng, W. Zhao, W. An, W. Liu, W. Liang, W. Gao, W. Yu, W. Zhang, X. Q. Li, X. Jin, X. Wang, X. Bi, X. Liu, X. Wang, X. Shen, X. Chen, X. Zhang, X. Chen, X. Nie, X. Sun, X. Wang, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yu, X. Song, X. Shan, X. Zhou, X. Yang, X. Li, X. Su, X. Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Y. Zhang, Y. Xu, Y. Xu, Y. Huang,

Y. Li, Y. Zhao, Y. Sun, Y. Li, Y. Wang, Y. Yu, Y. Zheng, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Tang, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Wu, Y. Ou, Y. Zhu, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Zha, Y. Xiong, Y. Ma, Y. Yan, Y. Luo, Y. You, Y. Liu, Y. Zhou, Z. F. Wu, Z. Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Huang, Z. Zhang, Z. Xie, Z. Zhang, Z. Hao, Z. Gou, Z. Ma, Z. Yan, Z. Shao, Z. Xu, Z. Wu, Z. Zhang, Z. Li, Z. Gu, Z. Zhu, Z. Liu, Z. Li, Z. Xie, Z. Song, Z. Gao, and Z. Pan. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. URL https://arxiv.org/abs/2412.19437.

[22] T. Dozat. Incorporating Nesterov momentum into Adam. *ICLR Workshops*, 2016. URL https://openreview.net/forum?id=OM0jvwB8jIp57ZJjtNEZ.

[23] C. L. Ernst Hairer and G. Wanner. *Geometric numerical integration*. Springer-Verlag, Berlin, 2 edition, 2006. ISBN 3-540-30663-3.

[24] M. Farazmand. Multiscale analysis of accelerated gradient methods. *SIAM Journal on Optimization*, 30(3):2337–2354, 2020. doi: 10.1137/18M1203997.

[25] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

[26] A. Ghosh, H. Lyu, X. Zhang, and R. Wang. Implicit regularization in heavy-ball momentum accelerated stochastic gradient descent. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ZzdBhtEH9yB.

[27] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Roziere, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Wyatt, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Guzmán, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Thattai, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. Kloumann, I. Misra, I. Evtimov, J. Zhang, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. van der Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu, J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnstun, J. Saxe, J. Jia, K. V. Alwala, K. Prasad, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, K. El-Arini, K. Iyer, K. Malik, K. Chiu, K. Bhalla, K. Lakhotia, L. Rantala-Yeary, L. van der Maaten, L. Chen, L. Tan, L. Jenkins, L. Martin, L. Madaan, L. Malo, L. Blecher, L. Landzaat, L. de Oliveira, M. Muzzi, M. Pasupuleti, M. Singh, M. Paluri, M. Kardas, M. Tsimpoukelli, M. Oldham, M. Rita, M. Pavlova, M. Kambadur, M. Lewis, M. Si, M. K. Singh, M. Hassan, N. Goyal, N. Torabi, N. Bashlykov, N. Bogoychev, N. Chatterji, N. Zhang, O. Duchenne, O. Çelebi, P. Alrassy, P. Zhang, P. Li, P. Vasic, P. Weng, P. Bhargava, P. Dubal, P. Krishnan, P. S. Koura, P. Xu, Q. He, Q. Dong, R. Srinivasan, R. Ganapathy, R. Calderer, R. S. Cabral, R. Stojnic, R. Raileanu, R. Maheswari, R. Girdhar, R. Patel, R. Sauvestre, R. Polidoro, R. Sumbaly, R. Taylor, R. Silva, R. Hou, R. Wang, S. Hosseini, S. Chennabasappa, S. Singh, S. Bell, S. S. Kim, S. Edunov, S. Nie, S. Narang, S. Raparthy, S. Shen, S. Wan, S. Bhosale, S. Zhang, S. Vandenhende, S. Batra, S. Whitman, S. Sootla, S. Collot, S. Gururangan, S. Borodinsky, T. Herman, T. Fowler, T. Sheasha, T. Georgiou, T. Scialom, T. Speckbacher, T. Mihaylov, T. Xiao, U. Karn, V. Goswami, V. Gupta, V. Ramanathan, V. Kerkez, V. Gonguet, V. Do, V. Vogeti, V. Albiero, V. Petrovic, W. Chu, W. Xiong, W. Fu, W. Meers, X. Martinet, X. Wang, X. Wang, X. E. Tan, X. Xia, X. Xie, X. Jia, X. Wang, Y. Goldschlag, Y. Gaur, Y. Babaei, Y. Wen, Y. Song, Y. Zhang, Y. Li, Y. Mao, Z. D. Coudert, Z. Yan, Z. Chen, Z. Papakipos, A. Singh, A. Srivastava, A. Jain, A. Kelsey, A. Shajnfeld, A. Gangidi, A. Victoria, A. Goldstand, A. Menon, A. Sharma, A. Boesenberg, A. Baevski, A. Feinstein, A. Kallet, A. Sangani, A. Teo, A. Yunus, A. Lupu, A. Alvarado, A. Caples, A. Gu, A. Ho, A. Poulton, A. Ryan, A. Ramchandani, A. Dong, A. Franco, A. Goyal, A. Saraf, A. Chowdhury, A. Gabriel, A. Bharambe, A. Eisenman, A. Yazdan, B. James, B. Maurer, B. Leonhardi, B. Huang, B. Loyd, B. D. Paola, B. Paranjape, B. Liu, B. Wu, B. Ni, B. Hancock, B. Wasti, B. Spence, B. Stojkovic, B. Gamido, B. Montalvo, C. Parker, C. Burton, C. Mejia, C. Liu, C. Wang, C. Kim, C. Zhou, C. Hu, C.-H. Chu, C. Cai,

C. Tindal, C. Feichtenhofer, C. Gao, D. Civin, D. Beaty, D. Kreymer, D. Li, D. Adkins, D. Xu, D. Testuggine, D. David, D. Parikh, D. Liskovich, D. Foss, D. Wang, D. Le, D. Holland, E. Dowling, E. Jamil, E. Montgomery, E. Presani, E. Hahn, E. Wood, E.-T. Le, E. Brinkman, E. Arcaute, E. Dunbar, E. Smothers, F. Sun, F. Kreuk, F. Tian, F. Kokkinos, F. Ozgenel, F. Caggioni, F. Kanayet, F. Seide, G. M. Florez, G. Schwarz, G. Badeer, G. Swee, G. Halpern, G. Herman, G. Sizov, Guangyi, Zhang, G. Lakshminarayanan, H. Inan, H. Shojanazeri, H. Zou, H. Wang, H. Zha, H. Habeeb, H. Rudolph, H. Suk, H. Aspegren, H. Goldman, H. Zhan, I. Damlaj, I. Molybog, I. Tufanov, I. Leontiadis, I.-E. Veliche, I. Gat, J. Weissman, J. Geboski, J. Kohli, J. Lam, J. Asher, J.-B. Gaya, J. Marcus, J. Tang, J. Chan, J. Zhen, J. Reizenstein, J. Teboul, J. Zhong, J. Jin, J. Yang, J. Cummings, J. Carvill, J. Shepard, J. McPhie, J. Torres, J. Ginsburg, J. Wang, K. Wu, K. H. U, K. Saxena, K. Khandelwal, K. Zand, K. Matosich, K. Veeraraghavan, K. Michelena, K. Li, K. Jagadeesh, K. Huang, K. Chawla, K. Huang, L. Chen, L. Garg, L. A, L. Silva, L. Bell, L. Zhang, L. Guo, L. Yu, L. Moshkovich, L. Wehrstedt, M. Khabsa, M. Avalani, M. Bhatt, M. Mankus, M. Hasson, M. Lennie, M. Reso, M. Groshev, M. Naumov, M. Lathi, M. Keneally, M. Liu, M. L. Seltzer, M. Valko, M. Restrepo, M. Patel, M. Vyatskov, M. Samvelyan, M. Clark, M. Macey, M. Wang, M. J. Hermoso, M. Metanat, M. Rastegari, M. Bansal, N. Santhanam, N. Parks, N. White, N. Bawa, N. Singhal, N. Egebo, N. Usunier, N. Mehta, N. P. Laptev, N. Dong, N. Cheng, O. Chernoguz, O. Hart, O. Salpekar, O. Kalinli, P. Kent, P. Parekh, P. Saab, P. Balaji, P. Rittner, P. Bontrager, P. Roux, P. Dollar, P. Zvyagina, P. Ratanchandani, P. Yuvraj, Q. Liang, R. Alao, R. Rodriguez, R. Ayub, R. Murthy, R. Nayani, R. Mitra, R. Parthasarathy, R. Li, R. Hogan, R. Battey, R. Wang, R. Howes, R. Rinott, S. Mehta, S. Siby, S. J. Bondu, S. Datta, S. Chugh, S. Hunt, S. Dhillon, S. Sidorov, S. Pan, S. Mahajan, S. Verma, S. Yamamoto, S. Ramaswamy, S. Lindsay, S. Lindsay, S. Feng, S. Lin, S. C. Zha, S. Patil, S. Shankar, S. Zhang, S. Zhang, S. Wang, S. Agarwal, S. Sajuyigbe, S. Chintala, S. Max, S. Chen, S. Kehoe, S. Satterfield, S. Govindaprasad, S. Gupta, S. Deng, S. Cho, S. Virk, S. Subramanian, S. Choudhury, S. Goldman, T. Remez, T. Glaser, T. Best, T. Koehler, T. Robinson, T. Li, T. Zhang, T. Matthews, T. Chou, T. Shaked, V. Vontimitta, V. Ajayi, V. Montanez, V. Mohan, V. S. Kumar, V. Mangla, V. Ionescu, V. Poenaru, V. T. Mihailescu, V. Ivanov, W. Li, W. Wang, W. Jiang, W. Bouaziz, W. Constable, X. Tang, X. Wu, X. Wang, X. Wu, X. Gao, Y. Kleinman, Y. Chen, Y. Hu, Y. Jia, Y. Qi, Y. Li, Y. Zhang, Y. Zhang, Y. Adi, Y. Nam, Yu, Wang, Y. Zhao, Y. Hao, Y. Qian, Y. Li, Y. He, Z. Rait, Z. DeVito, Z. Rosnbrick, Z. Wen, Z. Yang, Z. Zhao, and Z. Ma. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

[28] S. Gunasekar, B. E. Woodworth, S. Bhojanapalli, B. Neyshabur, and N. Srebro. Implicit regularization in matrix factorization. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/58191d2a914c6dae66371c9dcdc91b41-Abstract.html.

[29] S. Gunasekar, J. Lee, D. Soudry, and N. Srebro. Characterizing implicit bias in terms of optimization geometry. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1832–1841. PMLR, 2018. URL https://proceedings.mlr.press/v80/gunasekar18a.html.

[30] V. Gupta, T. Koren, and Y. Singer. Shampoo: Preconditioned stochastic tensor optimization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1842–1850. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/gupta18a.html.

[31] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[32] Z. Ji and M. Telgarsky. Risk and parameter convergence of logistic regression. *arXiv preprint arXiv:1803.07300*, 2018. URL https://arxiv.org/abs/1803.07300.

[33] Z. Ji and M. Telgarsky. The implicit bias of gradient descent on nonseparable data. In *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 1772–1798. PMLR, 2019. URL https://proceedings.mlr.press/v99/ji19a.html.

[34] Y. Jiang*, B. Neyshabur*, H. Mobahi, D. Krishnan, and S. Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgIPJBFvH.

[35] K. Kim, S. Kotha, P. Liang, and T. Hashimoto. Pre-training under infinite compute, 2025. URL https://arxiv.org/abs/2509.14786.

[36] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017. URL https://arxiv.org/abs/1412.6980.

[37] N. B. Kovachki and A. M. Stuart. Continuous time analysis of momentum methods. *Journal of Machine Learning Research*, 22(17):1–40, 2021. URL http://jmlr.org/papers/v22/19-466.html.

[38] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images, 2009.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

[40] F. Kunstner, J. Chen, J. W. Lavington, and M. Schmidt. Noise is not the main factor behind the gap between sgd and adam on transformers, but sign descent might be. *arXiv preprint arXiv:2304.13960*, 2023. URL https://arxiv.org/abs/2304.13960.

[41] F. Kunstner, J. Chen, J. W. Lavington, and M. Schmidt. Noise is not the main factor behind the gap between SGD and Adam on Transformers, but sign descent might be. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=a65YK0cqH8g.

[42] Q. Li, C. Tai, and W. E. Stochastic modified equations and adaptive stochastic gradient algorithms. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2101–2110. PMLR, 2017. URL https://proceedings.mlr.press/v70/li17f.html.

[43] Y. Li, C. Wei, and T. Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/bce9abf229ffd7e570818476ee5d7dde-Abstract.html.

[44] Z. Li, T. Wang, and S. Arora. What happens after SGD reaches zero loss? –a mathematical framework. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=siCt4xZn5Ve.

[45] L. Liu, X. Liu, J. Gao, W. Chen, and J. Han. Understanding the difficulty of training Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5747–5763, 2020. doi: 10.18653/v1/2020.emnlp-main.463. URL https://aclanthology.org/2020.emnlp-main.463/.

[46] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[47] C. Ma, L. Wu, and W. E. A qualitative study of the dynamic behavior for adaptive gradient algorithms. In *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference*, volume 145 of *Proceedings of Machine Learning Research*, pages 671–692. PMLR, 2022. URL https://proceedings.mlr.press/v145/ma22a.html.

[48] S. Malladi, K. Lyu, A. Panigrahi, and S. Arora. On the SDEs and scaling rules for adaptive gradient algorithms. In *Advances in Neural Information Processing Systems*, volume 35, pages 7697–7711, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/32ac710102f0620d0f28d5d05a44fe08-Abstract.html.

[49] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

[50] T. Miyagawa. Toward equation of motion for deep neural networks: Continuous-time gradient descent and discretization error analysis. In *Advances in Neural Information Processing Systems*, volume 35, pages 37778–37791, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/f6499ab2a923fa691accdc0077af9677-Abstract-Conference.html.

[51] M. S. Nacson, J. Lee, S. Gunasekar, P. H. P. Savarese, N. Srebro, and D. Soudry. Convergence of gradient descent on separable data. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 3420–3428. PMLR, 2019. URL https://proceedings.mlr.press/v89/nacson19b.html.

[52] M. S. Nacson, N. Srebro, and D. Soudry. Stochastic gradient descent on separable data: Exact convergence with a fixed learning rate. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 3051–3059. PMLR, 2019. URL https://proceedings.mlr.press/v89/nacson19a.html.

[53] Y. Nesterov. A method for solving the convex programming problem with convergence rate $o\left(\frac{1}{k^2}\right)$. In *Dokl. Akad. Nauk SSSR*, volume 269, pages 543–547, 1983.

[54] B. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. ISSN 0041-5553. doi: https://doi.org/10.1016/0041-5553(64)90137-5. URL https://www.sciencedirect.com/science/article/pii/0041555364901375.

[55] Q. Qian and X. Qian. The implicit bias of AdaGrad on separable data. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/3335881e06d4d23091389226225e17c7-Abstract.html.

[56] M. Rosca, Y. Wu, C. Qin, and B. Dherin. On a continuous time model of gradient descent dynamics and instability in deep learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=EYrRzKPinA.

[57] N. Shazeer and M. Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR, 2018. URL https://proceedings.mlr.press/v80/shazeer18a.html.

[58] H.-J. M. Shi, T.-H. Lee, S. Iwasaki, J. Gallego-Posada, Z. Li, K. Rangadurai, D. Mudigere, and M. Rabbat. A distributed data-parallel pytorch implementation of the distributed Shampoo optimizer for training neural networks at-scale. *arXiv preprint arXiv:2309.06497*, 2023. URL https://arxiv.org/abs/2309.06497.

[59] S. L. Smith, B. Dherin, D. Barrett, and S. De. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=rq_Qr0c1Hyo.

[60] D. Soudry, E. Hoffer, and N. Srebro. The implicit bias of gradient descent on separable data. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1q7n9gAb.

[61] T. Tieleman, G. Hinton, et al. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2): 26–31, 2012. URL https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

[62] P. Villalobos, A. Ho, J. Sevilla, T. Besiroglu, L. Heim, and M. Hobbhahn. Position: Will we run out of data? limits of LLM scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=ViZcgDQjyG.

[63] B. Wang, Q. Meng, H. Zhang, R. Sun, W. Chen, Z.-M. Ma, and T.-Y. Liu. Does momentum change the implicit regularization on separable data? In *Advances in Neural Information Processing Systems*, volume 35, pages 26764–26776, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/ab3f6bbe121a8f7a0263a9b393000741-Abstract-Conference.html.

[64] K. Wen, T. Ma, and Z. Li. How does sharpness-aware minimization minimizes sharpness? In *OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)*, 2022. URL https://openreview.net/forum?id=7H4YznLAwAj.

[65] L. Xiao. Rethinking conventional wisdom in machine learning: From generalization to scaling. *arXiv preprint arXiv:2409.15156*, 2024.

[66] S. Xie and Z. Li. Implicit bias of AdamW: ℓ_∞-norm constrained optimization. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 54488–54510. PMLR, 2024. URL https://proceedings.mlr.press/v235/xie24e.html.

[67] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[68] J. Zhang, S. P. Karimireddy, A. Veit, S. Kim, S. Reddi, S. Kumar, and S. Sra. Why are adaptive methods good for attention models? In *Advances in Neural Information Processing Systems*, volume 33, pages 15383–15393, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b05b57f6add810d3b7490866d74c0053-Abstract.html.

[69] R. Zhao, D. Morwani, D. Brandfonbrener, N. Vyas, and S. M. Kakade. Deconstructing what makes a good optimizer for autoregressive language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=zfeso8ceqr.

A Related Literature

Approximating a memoryful iteration with a memoryless one is closely connected with the method of modified equations (sometimes called *backward error analysis*), where a discrete algorithm like (1) is approximated by a continuous solution of an ordinary differential equation or stochastic differential equation. Typically, this method can only be applied to an algorithm with no memory, in a possibly enlarged phase space as opposed to $\mathbb{R}^d$; for example, heavy-ball momentum GD has no memory when viewed as a discrete iteration $(\boldsymbol{\theta}^{(n)}, \boldsymbol{m}^{(n)})$ in $\mathbb{R}^{2d}$, where $\boldsymbol{m}^{(n)}$ is the "momentum variable". The general technique introduced in this paper can be used to derive a memoryless discrete iteration which can then be approximated by a continuous trajectory. Background on the method of modified equations can be found in Ernst Hairer and Wanner [23], Li et al. [42].

Works deriving modified equations for (S)GD with or without momentum include Barrett and Dherin [6], Smith et al. [59], Ghosh et al. [26], Farazmand [24], Kovachki and Stuart [37], Miyagawa [50], Rosca et al. [56], Li et al. [42]. In particular, Ghosh et al. [26] identified that momentum strongly regularizes the loss function in the case of GD, though their error bounds both have a different focus (continuous approximation rather than discrete one), and follow a different approach which appears hard to generalize to other algorithms. Our approach works for a wide class of algorithms, and we recover their main results in Section 5. Works approximating adaptive methods with continuous trajectories include Ma et al. [47], Malladi et al. [48], Barakat and Bianchi [5], Compagnoni et al. [17]. More recently, Cattaneo et al. [11] studied the special case of Adam / RMSProp. We built on this work to conduct empirical evaluations. Their focus is not on memory but on continuous approximations; in particular, they do not have approximation bounds between two discrete iterations like we do. In addition, their arguments are highly specialized to Adam, and they do not incorporate weight decay. Although we also discuss Adam (with weight decay) extensively, it is only because of its importance in practice, and our results cover a much broader class of optimizers.

This paper is also connected to the strand of the literature studying the implicit bias of optimization algorithms. For example, Xie and Li [66] and Chen et al. [12] prove that weight decay causes AdamW and Lion to solve an ℓ_∞-norm constrained optimization problem. In that, they behave asymptotically like (normalized) steepest descent with respect to ℓ_∞-norm. Bernstein and Newhouse [7] also view Adam and Lion as smoothed-out versions of steepest descent. This perspective is connected to the Moreau-Yosida approximation of the loss function [9]; the latter work provides a concrete way to write down popular optimizers (including SGD with momentum, RMSProp and Adam) as a sequence of optimization problems. In addition, a large body of work is devoted to the bias of optimization algorithms towards the direction of the max-margin vector [60, 51, 52, 55, 63, 29, 32, 33]. Similarly, Damian et al. [19], Li et al. [44], Arora et al. [4], Wen et al. [64], Damian et al. [20] explore the sharpness of regions SGD converges to. Gunasekar et al. [28], Arora et al. [3] study implicit regularization in matrix factorization. Li et al. [43] prove in a certain setting that a larger learning rate leads to better generalization.

B Broader Impacts

This paper presents a general framework for contrasting certain properties of optimization algorithms commonly used for training neural networks, and thus this work can lead to societal consequences as common of deep learning.

C Special Case: $\boldsymbol{F}^{(n)}$ as a Function of "Momentum Variables"

In the examples listed in the introduction, $\boldsymbol{F}^{(n)}$ satisfies a more specific form that can be used to give more primitive conditions for Assumption 3.1. The following assumption, which is a special case of Assumption 3.1 by Lemma H.1, may look a bit technical but allows for a simpler calculation of correction terms.

Assumption C.1 (Special case of Assumption 3.1: $\boldsymbol{F}^{(n)}$ is a function of "momentum variables")**.** *Let $\{\boldsymbol{g}_\ell^{(n)}\}_{\ell=1}^L$ be L two times continuously differentiable functions $\mathcal{D} \to \mathbb{R}^d$, uniformly bounded along with two derivatives. Let $\{\beta_\ell\}_{\ell=1}^L$ be fixed reals in $[0, 1)$, and $\{b_\ell^{(n+1)}\}_{\ell=1}^L$ be L bounded nonnegative*

sequences of reals (for $n \in \mathbb{Z}_{\geqslant 0}$). Assume the function $\boldsymbol{F}^{(n)}$ has the form

$$\boldsymbol{F}^{(n)}(\boldsymbol{\theta}^{(n)}, \dots, \boldsymbol{\theta}^{(0)}) := \boldsymbol{\Phi}(\boldsymbol{m}_1^{(n+1)}, \dots, \boldsymbol{m}_L^{(n+1)})$$
$$\text{with } \boldsymbol{m}_\ell^{(n+1)} := b_\ell^{(n+1)} \sum_{k=0}^{n} \beta_\ell^k \boldsymbol{g}_\ell^{(n-k)}(\boldsymbol{\theta}^{(n-k)}) \in \mathcal{M}, \tag{11}$$

where $\mathcal{M}$ is a bounded open region in $\mathbb{R}^d$ and $\boldsymbol{\Phi}(\boldsymbol{m}_1, \dots, \boldsymbol{m}_L) \colon \mathcal{M}^L \to \mathbb{R}^d$ is a fixed two times continuously differentiable function, uniformly bounded along with two derivatives. In the full-batch case, $\boldsymbol{g}_\ell^{(n)} \equiv \boldsymbol{g}_\ell$ are not allowed to depend on n.

For instance, in the case of AdamW (Example 1.3), Assumption C.1 applies with $L = 3$,

$$\boldsymbol{g}_1(\boldsymbol{\theta}) = \nabla \mathcal{L}(\boldsymbol{\theta}), \quad \boldsymbol{g}_2(\boldsymbol{\theta}) = \big(\nabla \mathcal{L}(\boldsymbol{\theta})\big)^2, \quad \boldsymbol{g}_3(\boldsymbol{\theta}) = \boldsymbol{\theta},$$
$$b_1^{(n+1)} = \frac{1-\beta_1}{1-\beta_1^{n+1}} \to b_1 = 1 - \beta_1,$$
$$b_2^{(n+1)} = \frac{1-\beta_2}{1-\beta_2^{n+1}} \to b_2 = 1 - \beta_2,$$
$$b_3^{(n+1)} \equiv b_3 = \lambda, \quad \beta_3 = 0.$$

In the case of Lion-$\mathcal{K}$ (Example 1.5), the assumption applies with $L = 3$,

$$\boldsymbol{g}_1(\boldsymbol{\theta}) = -\nabla \mathcal{L}(\boldsymbol{\theta}), \quad \boldsymbol{g}_2(\boldsymbol{\theta}) = -\nabla \mathcal{L}(\boldsymbol{\theta}), \quad \boldsymbol{g}_3(\boldsymbol{\theta}) = \boldsymbol{\theta},$$
$$\beta_1 = \rho_2, \quad \beta_2 = 0, \quad \beta_3 = 0,$$
$$b_1^{(n+1)} \equiv b_1 = (1-\rho_2)\frac{\rho_1}{\rho_2},$$
$$b_2^{(n+1)} \equiv b_2 = 1 - \frac{\rho_1}{\rho_2},$$
$$b_3^{(n+1)} \equiv b_3 = \lambda.$$

We used the letter ρ when defining the Lion iteration to avoid confusion with the β in the definition of "momentum variables".

Specializing to the setup of Assumption C.1, and for any $s, n \in \mathbb{Z}_{\geqslant 0}$, $\boldsymbol{F}^{(s)}(\boldsymbol{\theta}) = \boldsymbol{\Phi}\big(\bar{\boldsymbol{g}}_1^{(s+1)}(\boldsymbol{\theta}), \dots, \bar{\boldsymbol{g}}_L^{(s+1)}(\boldsymbol{\theta})\big)$, where $\bar{\boldsymbol{g}}_\ell^{(s+1)}(\boldsymbol{\theta}) := b_\ell^{(n+1)} \sum_{k=0}^{s} \beta_\ell^k \boldsymbol{g}_\ell^{(n-k)}(\boldsymbol{\theta})$, and

$$\frac{\partial F_r^{(n)}}{\partial \boldsymbol{\theta}^{(n-k)}}(\boldsymbol{\theta}) = \sum_{\ell=1}^{L} \sum_i b_\ell^{(n+1)} \beta_\ell^k \frac{\partial \Phi_r}{\partial m_{\ell;i}} \big(\bar{\boldsymbol{g}}_1^{(n+1)}(\boldsymbol{\theta}), \dots, \bar{\boldsymbol{g}}_L^{(n+1)}(\boldsymbol{\theta})\big)^{\mathsf{T}} \nabla g_{\ell;i}^{(n-k)}(\boldsymbol{\theta}).$$

Therefore, in this special case, the correction term in the memoryless iteration (4) is given by, for $r = 1, \dots, d$,

$$M_r^{(n)}(\boldsymbol{\theta}) = h \sum_{\ell=1}^{L} \sum_i b_\ell^{(n+1)} \frac{\partial \Phi_r}{\partial m_{\ell;i}} \big(\bar{\boldsymbol{g}}_1^{(n+1)}(\boldsymbol{\theta}), \dots, \bar{\boldsymbol{g}}_L^{(n+1)}(\boldsymbol{\theta})\big) \times$$
$$\times \sum_{k=1}^{n} \beta_\ell^k \nabla g_{\ell;i}^{(n-k)}(\boldsymbol{\theta})^{\mathsf{T}} \sum_{s=n-k}^{n-1} \boldsymbol{\Phi}\big(\bar{\boldsymbol{g}}_1^{(s+1)}(\boldsymbol{\theta}), \dots, \bar{\boldsymbol{g}}_L^{(s+1)}(\boldsymbol{\theta})\big).$$

In the full-batch case $\boldsymbol{g}_\ell^{(n)}(\boldsymbol{\theta}) \equiv \boldsymbol{g}_\ell(\boldsymbol{\theta})$, this can be simplified further. Let us assume $b_\ell^{(n+1)} \underset{n\to\infty}{\longrightarrow} b_\ell$, where b_ℓ is constant in n. Then $\bar{\boldsymbol{g}}_\ell^{(n+1)}(\boldsymbol{\theta})$ also become constant in n: specifically, they settle to $\bar{\boldsymbol{g}}_\ell(\boldsymbol{\theta}) := b_\ell(1-\beta_\ell)^{-1}\boldsymbol{g}_\ell(\boldsymbol{\theta})$. Lemma H.2 then implies that the iteration becomes close to

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n)} - h\big[\boldsymbol{\Phi}(\bar{\boldsymbol{g}}_1(\boldsymbol{\theta}^{(n)}), \dots, \bar{\boldsymbol{g}}_L(\boldsymbol{\theta}^{(n)})) + \boldsymbol{M}^{(n)}(\boldsymbol{\theta})\big]$$

with

$$M_r^{(n)}(\boldsymbol{\theta}) = h\sum_{\ell=1}^{L}\sum_{i}\frac{b_\ell\beta_\ell}{(1-\beta_\ell)^2}\frac{\partial\Phi_r}{\partial m_{\ell;i}}\big(\bar{\boldsymbol{g}}_1(\boldsymbol{\theta}^{(n)}),\dots,\bar{\boldsymbol{g}}_L(\boldsymbol{\theta}^{(n)})\big)\times \\ \times\nabla g_{\ell;i}(\boldsymbol{\theta}^{(n)})^{\mathsf{T}}\boldsymbol{\Phi}\big(\bar{\boldsymbol{g}}_1(\boldsymbol{\theta}^{(n)}),\dots,\bar{\boldsymbol{g}}_L(\boldsymbol{\theta}^{(n)})\big).$$

These formulae admittedly look complicated, but we can easily plug in the definitions and calculate correction terms for all examples with little additional algebra. We list these terms in Section D.

D Details for Examples: Correction Terms

For GD with momentum (Example 1.1):

$$\boldsymbol{M}^{(n)}(\boldsymbol{\theta}) = \frac{h\beta}{2(1-\beta)^3}\nabla\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2.$$

For Nesterov's accelerated GD (Example 1.2):

$$\boldsymbol{M}^{(n)}(\boldsymbol{\theta}) = \frac{h\beta^2}{2(1-\beta)^3}\nabla\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2.$$

For AdamW (Example 1.3, also discussed in Section 4):

$$\boldsymbol{M}^{(n)}(\boldsymbol{\theta}) = h\left(\frac{\beta_1(1-\beta_1)^{-1}-\beta_2(1-\beta_2)^{-1}}{\big(\big|\nabla\mathcal{L}(\boldsymbol{\theta})\big|^2+\varepsilon\big)^{1/2}}+\varepsilon\frac{\beta_2(1-\beta_2)^{-1}}{\big(\big|\nabla\mathcal{L}(\boldsymbol{\theta})\big|^2+\varepsilon\big)^{3/2}}\right)\big(\nabla\|\nabla\mathcal{L}(\boldsymbol{\theta})\|_{1,\varepsilon}+\lambda\nabla^2\mathcal{L}(\boldsymbol{\theta})\boldsymbol{\theta}\big).$$

For Nadam (Example 1.4):

$$\boldsymbol{M}^{(n)}(\boldsymbol{\theta}) = h\left(\frac{\beta_1^2(1-\beta_1)^{-1}-\beta_2(1-\beta_2)^{-1}}{\big(\big|\nabla\mathcal{L}(\boldsymbol{\theta})\big|^2+\varepsilon\big)^{1/2}}+\varepsilon\frac{\beta_2(1-\beta_2)^{-1}}{\big(\big|\nabla\mathcal{L}(\boldsymbol{\theta})\big|^2+\varepsilon\big)^{3/2}}\right)\big(\nabla\|\nabla\mathcal{L}(\boldsymbol{\theta})\|_{1,\varepsilon}+\lambda\nabla^2\mathcal{L}(\boldsymbol{\theta})\boldsymbol{\theta}\big).$$

For Lion-$\mathcal{K}$ (Example 1.5, also discussed in Section 4):

$$\boldsymbol{M}^{(n)}(\boldsymbol{\theta}) = -h\frac{\rho_1}{1-\rho_2}\nabla^2\mathcal{K}(-\nabla\mathcal{L}(\boldsymbol{\theta}))\nabla^2\mathcal{L}(\boldsymbol{\theta})\big[\nabla\mathcal{K}(-\nabla\mathcal{L}(\boldsymbol{\theta}))-\lambda\boldsymbol{\theta}\big].$$

E Mini-Batch Training: Details

Let us take the expectation of the correction term with respect to the random permutation of mini-batches, that is, take the average over all re-orderings π of $(\boldsymbol{g}^{(0)},\dots,\boldsymbol{g}^{(n)})$:

$$\mathbb{E}\sum_{k=0}^{n-1}\beta^k\nabla g_r^{(\pi(n-1-k))}(\boldsymbol{\theta})^{\mathsf{T}}\sum_{l=1}^{k+1}\sum_{b=0}^{n-l}\beta^b\boldsymbol{g}^{(\pi(n-l-b))}(\boldsymbol{\theta}) \\ := \frac{1}{(n+1)!}\sum_{\pi}\sum_{k=0}^{n-1}\beta^k\nabla g_r^{(\pi(n-1-k))}(\boldsymbol{\theta})^{\mathsf{T}}\sum_{l=1}^{k+1}\sum_{b=0}^{n-l}\beta^b\boldsymbol{g}^{(\pi(n-l-b))}(\boldsymbol{\theta}).$$

Note that $\mathbb{E}\nabla g_r^{(i)}(\boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{g}^{(j)}(\boldsymbol{\theta})$ depends only on whether $i=j$ or $i\neq j$. Therefore,

$$\mathbb{E}[M_r^{(n)}(\boldsymbol{\theta})]/h = C_3(\beta)\mathbb{E}[\nabla g_r^{(1)}(\boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{g}^{(1)}(\boldsymbol{\theta})]+C_4(\beta)\mathbb{E}[\nabla g_r^{(1)}(\boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{g}^{(2)}(\boldsymbol{\theta})],$$

where $C_3(\beta)$ and $C_4(\beta)$ are given by

$$C_3(\beta) := \beta\sum_{b=0}^{n-1}\beta^b\sum_{l=1}^{b+1}\beta^{b+1-l}\underset{n\to\infty}{\longrightarrow}\frac{\beta}{(1-\beta)^2(1+\beta)},$$

$$C_4(\beta) := \beta\sum_{k=0}^{n-1}\beta^k\sum_{l=1}^{k+1}\sum_{b=0}^{n-l}\beta^b - C_3(\beta)\underset{n\to\infty}{\longrightarrow}\frac{2\beta^2}{(1-\beta)^3(1+\beta)}.$$

We can simplify

$$\mathbb{E}[\nabla g_r^{(1)}(\boldsymbol{\theta})^\mathsf{T}\boldsymbol{g}^{(2)}(\boldsymbol{\theta})] = \frac{1}{(n+1)n}\sum_{i\neq j}\nabla g_r^{(i)}(\boldsymbol{\theta})^\mathsf{T}\boldsymbol{g}^{(j)}(\boldsymbol{\theta}) = \nabla g_r(\boldsymbol{\theta})^\mathsf{T}\boldsymbol{g}(\boldsymbol{\theta}) + o_n(1),$$

where $\boldsymbol{g}(\boldsymbol{\theta}) = \mathbb{E}\boldsymbol{g}^{(1)}(\boldsymbol{\theta}) = (n+1)^{-1}\sum_{k=0}^{n+1}\boldsymbol{g}^{(k)}(\boldsymbol{\theta})$ is the average of $\{\boldsymbol{g}^{(k)}(\boldsymbol{\theta})\}$, $o_n(1)$ tends to zero as $n\to\infty$.

So, for large n we can write

$$\begin{aligned}
\mathbb{E}[M_r^{(n)}(\boldsymbol{\theta})]/h &\approx C_3(\beta)\mathbb{E}[\nabla g_r^{(1)}(\boldsymbol{\theta})^\mathsf{T}\boldsymbol{g}^{(1)}(\boldsymbol{\theta})] + C_4(\beta)\nabla g_r(\boldsymbol{\theta})^\mathsf{T}\boldsymbol{g}(\boldsymbol{\theta}) \\
&= \big(C_3(\beta)+C_4(\beta)\big)\nabla g_r(\boldsymbol{\theta})^\mathsf{T}\boldsymbol{g}(\boldsymbol{\theta}) + C_3(\beta)\mathbb{E}[(\nabla g_r^{(1)}(\boldsymbol{\theta}) - \nabla g_r(\boldsymbol{\theta}))^\mathsf{T}(\boldsymbol{g}^{(1)}(\boldsymbol{\theta}) - \boldsymbol{g}(\boldsymbol{\theta}))] \\
&\approx \frac{\beta}{(1-\beta)^3}\nabla g_r(\boldsymbol{\theta})^\mathsf{T}\boldsymbol{g}(\boldsymbol{\theta}) + C_3(\beta)\mathbb{E}[(\nabla g_r^{(1)}(\boldsymbol{\theta}) - \nabla g_r(\boldsymbol{\theta}))^\mathsf{T}(\boldsymbol{g}^{(1)}(\boldsymbol{\theta}) - \boldsymbol{g}(\boldsymbol{\theta}))].
\end{aligned}$$

F Proof of Theorem 3.2

Since, by the assumptions of the theorem,

$$\begin{aligned}
\tilde\theta_r^{(n+1)} - \tilde\theta_r^{(n)} &= -hF_r^{(n)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)}) \\
&\quad - h^2\underline{\sum_{k=1}^{n}\frac{\partial F_r^{(n)}}{\partial\boldsymbol{\theta}^{(n-k)}}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)})^\mathsf{T}\sum_{s=n-k}^{n-1}\boldsymbol{F}^{(s)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)})},
\end{aligned} \tag{12}$$

we need to show that

$$\begin{aligned}
&F_r^{(n)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(0)}) - F_r^{(n)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)}) \\
&\quad = h\sum_{k=1}^{n}\frac{\partial F_r^{(n)}}{\partial\boldsymbol{\theta}^{(n-k)}}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)})^\mathsf{T}\sum_{s=n-k}^{n-1}\boldsymbol{F}^{(s)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)}) + O(h^2).
\end{aligned} \tag{13}$$

By Taylor expansion with the Lagrange remainder,

$$\begin{aligned}
&F_r^{(n)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(0)}) - F_r^{(n)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)}) \\
&\quad = \sum_{k=1}^{n}\frac{\partial F_r^{(n)}}{\partial\boldsymbol{\theta}^{(n-k)}}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)})^\mathsf{T}\big(\tilde{\boldsymbol{\theta}}^{(n-k)} - \tilde{\boldsymbol{\theta}}^{(n)}\big) \\
&\qquad + \frac{1}{2}\sum_{k_1,k_2=1}^{n}\big(\tilde{\boldsymbol{\theta}}^{(n-k_1)} - \tilde{\boldsymbol{\theta}}^{(n)}\big)^\mathsf{T}\frac{\partial^2 F_r^{(n)}}{\partial\boldsymbol{\theta}^{(n-k_1)}\partial\boldsymbol{\theta}^{(n-k_2)}}(\boldsymbol{\zeta})\big(\tilde{\boldsymbol{\theta}}^{(n-k_2)} - \tilde{\boldsymbol{\theta}}^{(n)}\big) \\
&\quad \overset{\text{(a)}}{=} \sum_{k=1}^{n}\frac{\partial F_r^{(n)}}{\partial\boldsymbol{\theta}^{(n-k)}}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)})^\mathsf{T}\big(\tilde{\boldsymbol{\theta}}^{(n-k)} - \tilde{\boldsymbol{\theta}}^{(n)}\big) + O(h^2),
\end{aligned} \tag{14}$$

where $\boldsymbol{\zeta}$ is some point on the segment between $\big(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(0)}\big)$ and $\big(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)}\big)$; in (a) we used Assumption 3.1 and $\tilde{\boldsymbol{\theta}}^{(n-k_1)} - \tilde{\boldsymbol{\theta}}^{(n)} = O(k_1h)$, $\tilde{\boldsymbol{\theta}}^{(n-k_2)} - \tilde{\boldsymbol{\theta}}^{(n)} = O(k_2h)$. Since the underlined term in Equation (12) is $O(1)$, we have

$$\begin{aligned}
\tilde{\boldsymbol{\theta}}^{(n-k)} - \tilde{\boldsymbol{\theta}}^{(n)} &= \sum_{s=n-k}^{n-1}\big(\tilde{\boldsymbol{\theta}}^{(s)} - \tilde{\boldsymbol{\theta}}^{(s+1)}\big) \\
&= h\sum_{s=n-k}^{n-1}\big\{\boldsymbol{F}^{(s)}(\tilde{\boldsymbol{\theta}}^{(s)},\dots,\tilde{\boldsymbol{\theta}}^{(s)}) + O(h)\big\} \\
&= h\sum_{s=n-k}^{n-1}\boldsymbol{F}^{(s)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)}) + O(k^2h^2),
\end{aligned}$$

where in the last step we used $\boldsymbol{F}^{(s)}(\tilde{\boldsymbol{\theta}}^{(s)},\dots,\tilde{\boldsymbol{\theta}}^{(s)}) - \boldsymbol{F}^{(s)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)}) = O((n-s)h)$, which follows from Taylor expansion, Assumption 3.1 and $\tilde{\boldsymbol{\theta}}^{(n+1)} - \tilde{\boldsymbol{\theta}}^{(n)} = O(h)$. Combine this with Equation (14) to get

$$\begin{aligned}
&F_r^{(n)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(0)}) - F_r^{(n)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)}) \\
&\quad= \sum_{k=1}^{n} \frac{\partial F_r^{(n)}}{\partial \boldsymbol{\theta}^{(n-k)}}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)})^{\mathsf{T}} \left\{ h \sum_{s=n-k}^{n-1} \boldsymbol{F}^{(s)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)}) + O(k^2h^2) \right\} + O(h^2) \\
&\quad= h \sum_{k=1}^{n} \frac{\partial F_r^{(n)}}{\partial \boldsymbol{\theta}^{(n-k)}}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)})^{\mathsf{T}} \sum_{s=n-k}^{n-1} \boldsymbol{F}^{(s)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(n)}) + O(h^2),
\end{aligned}$$

which is (13), and the proof is complete.

G Proof of Corollary 3.3

We follow a standard argument, e. g. Ghosh et al. [26], Cattaneo et al. [11]. We prove the following claim by induction over $n \in \mathbb{Z}_{\geqslant 0}$:

$$\|\boldsymbol{\theta}^{(n)} - \tilde{\boldsymbol{\theta}}^{(n)}\|_\infty \leqslant d_1 e^{d_2 nh} h^2, \quad \|\boldsymbol{\theta}^{(n+1)} - \tilde{\boldsymbol{\theta}}^{(n+1)} - \boldsymbol{\theta}^{(n)} + \tilde{\boldsymbol{\theta}}^{(n)}\|_\infty \leqslant d_3 e^{d_2 nh} h^3,$$

where

$$d_1 = C_1, \quad d_2 = 1 + d\sum_{k=0}^{\infty} \gamma_k, \quad d_3 = C_1 d_2.$$

Because $nh \leqslant T$, Corollary 3.3 will follow.

Base: $n = 0$. It is indeed true that $\|\boldsymbol{\theta}^{(0)} - \tilde{\boldsymbol{\theta}}^{(0)}\|_\infty \leqslant d_1 h^2$ because the left-hand side is zero. It is indeed true that $\|\boldsymbol{\theta}^{(1)} - \tilde{\boldsymbol{\theta}}^{(1)} - \boldsymbol{\theta}^{(0)} + \tilde{\boldsymbol{\theta}}^{(0)}\|_\infty \leqslant d_3 h^3$ for the same reason.

Assume $n \in \mathbb{Z}_{\geqslant 1}$ and it is true that

$$\|\boldsymbol{\theta}^{(n')} - \tilde{\boldsymbol{\theta}}^{(n')}\|_\infty \leqslant d_1 e^{d_2 n' h} h^2, \quad \|\boldsymbol{\theta}^{(n'+1)} - \tilde{\boldsymbol{\theta}}^{(n'+1)} - \boldsymbol{\theta}^{(n')} + \tilde{\boldsymbol{\theta}}^{(n')}\|_\infty \leqslant d_3 e^{d_2 n' h} h^3.$$

for all $0 \leqslant n' \leqslant n-1$. Then

$$\|\boldsymbol{\theta}^{(n)} - \tilde{\boldsymbol{\theta}}^{(n)}\|_\infty \leqslant \|\boldsymbol{\theta}^{(n-1)} - \tilde{\boldsymbol{\theta}}^{(n-1)}\|_\infty + \|\boldsymbol{\theta}^{(n)} - \tilde{\boldsymbol{\theta}}^{(n)} - \boldsymbol{\theta}^{(n-1)} + \tilde{\boldsymbol{\theta}}^{(n-1)}\|_\infty$$

by the triangle inequality,

$$\leqslant d_1 e^{d_2(n-1)h} h^2 + d_3 e^{d_2(n-1)h} h^3$$

by the induction hypothesis,

$$= d_1 \left(1 + \frac{d_3}{d_1} h\right) e^{d_2(n-1)h} h^2 \leqslant d_1 (1 + d_2 h) e^{d_2(n-1)h} h^2$$

by $d_3 \leqslant d_1 d_2$,

$$\leqslant d_1 e^{d_2 nh} h^2$$

by the inequality $1 + x \leqslant e^x$ for all $x \geqslant 0$.

Next, write

$$\begin{aligned}
&\boldsymbol{\theta}^{(n+1)} - \boldsymbol{\theta}^{(n)} - \tilde{\boldsymbol{\theta}}^{(n+1)} + \tilde{\boldsymbol{\theta}}^{(n)} \\
&\quad= -h\boldsymbol{F}^{(n)}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(0)}) - \left\{\tilde{\boldsymbol{\theta}}^{(n+1)} - \tilde{\boldsymbol{\theta}}^{(n)}\right\}
\end{aligned}$$

$$= h\big[\boldsymbol{F}^{(n)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(0)}) - \boldsymbol{F}^{(n)}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(0)})\big] - \big\{\tilde{\boldsymbol{\theta}}^{(n+1)} - \tilde{\boldsymbol{\theta}}^{(n)} + h\boldsymbol{F}^{(n)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(0)})\big\}$$

Then

$$\begin{aligned}
&|\theta_r^{(n+1)} - \theta_r^{(n)} - \tilde{\theta}_r^{(n+1)} + \tilde{\theta}_r^{(n)}| \\
&\quad\leqslant h\big|F_r^{(n)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(0)}) - F_r^{(n)}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(0)})\big| + \big|\tilde{\theta}_r^{(n+1)} - \tilde{\theta}_r^{(n)} + hF_r^{(n)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(0)})\big| \\
&\quad\leqslant h\big|F_r^{(n)}(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(0)}) - F_r^{(n)}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(0)})\big| + C_1h^3
\end{aligned}$$

by Theorem 3.2,

$$= h\left|\sum_{k=0}^{n}\frac{\partial F_r^{(n)}}{\partial\boldsymbol{\theta}^{(n-k)}}(\boldsymbol{\zeta})^{\mathsf{T}}\big(\tilde{\boldsymbol{\theta}}^{(n-k)} - \boldsymbol{\theta}^{(n-k)}\big)\right| + C_1h^3,$$

where $\boldsymbol{\zeta}$ is a point on the segment between $\big(\tilde{\boldsymbol{\theta}}^{(n)},\dots,\tilde{\boldsymbol{\theta}}^{(0)}\big)$ and $\big(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(0)}\big)$,

$$\leqslant hd\sum_{k=0}^{n}\gamma_k\|\tilde{\boldsymbol{\theta}}^{(n-k)} - \boldsymbol{\theta}^{(n-k)}\|_\infty + C_1h^3$$

by (3.1) (recall that d is the dimension of $\boldsymbol{\theta}$),

$$\leqslant d_1h^3d\sum_{k=0}^{\infty}\gamma_ke^{d_2(n-k)h} + C_1h^3$$

by the induction hypothesis and the bound on $\|\tilde{\boldsymbol{\theta}}^{(n)} - \boldsymbol{\theta}^{(n)}\|_\infty$ already proven

$$\begin{aligned}
&\leqslant \underbrace{\left(d_1d\sum_{k=0}^{\infty}\gamma_k + C_1\right)}_{\leqslant d_3}e^{d_2nh}h^3 \\
&\leqslant d_3e^{d_2nh}h^3.
\end{aligned}$$

H Auxiliary Results

Lemma H.1 (Memory decays exponentially fast)**.** *If $\boldsymbol{F}^{(n)}$ is a function of "momentum variables" as described in Assumption C.1, then for any n and $k \leqslant n$*

$$\max_{r,i}\left|\frac{\partial F_r^{(n)}}{\partial\theta_i^{(n-k)}}\right| \leqslant \gamma_k, \tag{15}$$

and similarly for any n and $k_1, k_2 \leqslant n$

$$\max_{r,i,j}\left|\frac{\partial^2 F_r^{(n)}}{\partial\theta_i^{(n-k_1)}\partial\theta_j^{(n-k_2)}}\right| \leqslant \gamma_{k_1,k_2}, \tag{16}$$

where $\{\gamma_k\}$ and $\{\gamma_{k_1,k_2}\}$ are sequences decaying exponentially fast: specifically,

$$\gamma_k := C_\gamma\{\max_\ell\beta_\ell\}^k, \quad \gamma_{k_1,k_2} := C_\gamma\{\max_\ell\beta_\ell\}^{k_1+k_2}$$

for some constant $C_\gamma > 0$.

Proof. It is easy to see (15) by taking the derivative:

$$\frac{\partial}{\partial\boldsymbol{\theta}^{(n-k)}}F_r^{(n)}(\boldsymbol{\theta}^{(n)},\dots,\boldsymbol{\theta}^{(0)})$$

$$= \sum_{\ell=1}^{L} \sum_{i} \frac{\partial \Phi_r}{\partial m_{\ell;i}}(\boldsymbol{m}_1^{(n+1)}, \dots, \boldsymbol{m}_L^{(n+1)}) \frac{\partial m_{\ell;i}^{(n+1)}}{\boldsymbol{\theta}^{(n-k)}}$$
$$= \sum_{\ell=1}^{L} \sum_{i} b_\ell^{(n+1)} \beta_\ell^k \frac{\partial \Phi_r}{\partial m_{\ell;i}}(\boldsymbol{m}_1^{(n+1)}, \dots, \boldsymbol{m}_L^{(n+1)})$$
$$\times \nabla g_{\ell;i}^{(n-k)}(\boldsymbol{\theta}^{(n-k)}),$$

and using the uniform boundedness of derivatives of $g_{\ell;i}^{(n-k)}$ and Φ_r. Equation (16) is proven similarly. □

Lemma H.2. *Let $\{a_k\}_{k=1}^{\infty}$ and $\{b_k\}_{k=1}^{\infty}$ be sequences of reals such that $\sum_{k=1}^{\infty}(|a_k| + |b_k|) < \infty$. Then*

$$\sum_{k=1}^{n} a_k \sum_{s=n-k}^{n-1} b_s \underset{n\to\infty}{\longrightarrow} 0.$$

Proof. Fix $\varepsilon > 0$. Take such positive integer n_0 that for any $n_0 \leqslant n_1 \leqslant n_2$ we have $\sum_{s=n_1}^{n_2}(|a_k| + |b_k|) < \varepsilon$. Then for any $n \geqslant 2n_0 - 1$ the following holds:

$$\sum_{k=1}^{n}|a_k| \sum_{s=n-k}^{n-1} |b_s| = \sum_{k=1}^{n-n_0} |a_k| \underbrace{\sum_{s=n-k}^{n-1} |b_s|}_{<\varepsilon} + \underbrace{\sum_{k=n-n_0+1}^{n} |a_k|}_{<\varepsilon} \sum_{s=n-k}^{n-1} |b_s| < \varepsilon \sum_{k=1}^{\infty}(|a_k| + |b_k|).$$

Since ε is arbitrary and $\sum_{k=1}^{\infty}(|a_k| + |b_k|)$ is a finite constant, the statement follows. □

I Corollaries for Special Cases

Lemma I.1 (Application of Corollary 3.3 to Example 1.1)**.** *Let $\{\boldsymbol{\theta}^{(n)}\}_{n\in\mathbb{Z}_{\geqslant 0}}$ be the sequence of vectors generated by the iteration in Equation (1) with initial condition $\boldsymbol{\theta}^{(0)}$, where $\boldsymbol{F}^{(n)}(\cdot)$ is as defined in Example 1.1, and the loss function $\mathcal{L}(\cdot)$ defined in an open convex bounded domain $\mathcal{D} \subset \mathbb{R}^d$ is three times continuously differentiable with bounded derivatives; also, let $T \geqslant 0$ be a fixed "time" horizon. Then Assumption 3.1 holds; in particular, by Corollary 3.3 the inequality $\max_{n\in[0:\lfloor T/h\rfloor]} \big\|\boldsymbol{\theta}^{(n)} - \tilde{\boldsymbol{\theta}}^{(n)}\big\|_\infty \leqslant C_2 h^2$ holds, where*

$$\tilde{\theta}_r^{(n+1)} = \tilde{\theta}_r^{(n)} - h\bigg(\frac{1-\beta^{n+1}}{1-\beta}\nabla_r \mathcal{L}\big(\tilde{\boldsymbol{\theta}}^{(n)}\big) + M_r^{(n)}\big(\tilde{\boldsymbol{\theta}}^{(n)}\big)\bigg),$$
$$M_r^{(n)}\big(\boldsymbol{\theta}\big) = h\frac{\beta[1-(2n+1)\beta^n(1-\beta)-\beta^{2n+1}]}{2(1-\beta)^3}\nabla_r\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2, \quad r \in [1:d].$$

Proof. The fact that Assumption 3.1 holds is already verified in Section C.

Next, in this case

$$F_r^{(n)}(\boldsymbol{\theta}) = \frac{1-\beta^{n+1}}{1-\beta}\nabla_r\mathcal{L}(\boldsymbol{\theta}),$$
$$\frac{\partial F_r^{(n)}}{\partial \theta_i^{(n-k)}}(\boldsymbol{\theta}) = \beta^k \nabla_{ir}\mathcal{L}(\boldsymbol{\theta}).$$

Therefore,

$$M_r^{(n)}\boldsymbol{\theta} = h\sum_{k=1}^{n}\sum_{i=1}^{d}\frac{\partial F_r^{(n)}}{\partial\theta_i^{(n-k)}}(\boldsymbol{\theta})\sum_{s=n-k}^{n-1} F_i^{(s)}(\boldsymbol{\theta})$$
$$= h\sum_{k=1}^{n}\beta^k\sum_{s=n-k}^{n-1}\frac{1-\beta^{s+1}}{1-\beta}\sum_{i=1}^{d}\nabla_{ir}\mathcal{L}(\boldsymbol{\theta})\nabla_i\mathcal{L}(\boldsymbol{\theta})$$

$$= h\frac{\beta[1-(2n+1)\beta^n(1-\beta)-\beta^{2n+1}]}{(1-\beta)^3}\sum_{i=1}^{d}\nabla_{ir}\mathcal{L}(\boldsymbol{\theta})\nabla_i\mathcal{L}(\boldsymbol{\theta}).$$

□

Lemma I.2 (Application of Corollary 3.3 to Example 1.3)**.** *Let $\{\boldsymbol{\theta}^{(n)}\}_{n\in\mathbb{Z}_{\geqslant 0}}$ be the sequence of vectors generated by the iteration in Equation (1) with initial condition $\boldsymbol{\theta}^{(0)}$, where $\boldsymbol{F}^{(n)}(\cdot)$ is as defined in Example 1.3, and the loss function $\mathcal{L}(\cdot)$ defined in an open convex bounded domain $\mathcal{D}\subset\mathbb{R}^d$ is three times continuously differentiable with bounded derivatives; also, let $T\geqslant 0$ be a fixed "time" horizon. Then Assumption 3.1 holds; in particular, by Corollary 3.3 the inequality $\max_{n\in[0:\lfloor T/h\rfloor]}\big\|\boldsymbol{\theta}^{(n)}-\tilde{\boldsymbol{\theta}}^{(n)}\big\|_\infty\leqslant C_2h^2$ holds, where*

$$\tilde{\theta}_r^{(n+1)} = (1-\lambda h)\tilde{\theta}_r^{(n)} - h\bigg(\frac{\partial_r\mathcal{L}\big(\tilde{\boldsymbol{\theta}}^{(n)}\big)}{\big(\big|\partial_r\mathcal{L}\big(\tilde{\boldsymbol{\theta}}^{(n)}\big)\big|^2+\varepsilon\big)^{1/2}} + M_r^{(n)}\big(\tilde{\boldsymbol{\theta}}^{(n)}\big)\bigg),$$

$$M_r^{(n)}\big(\boldsymbol{\theta}\big) = h\bigg(\frac{\beta_1}{1-\beta_1}-\frac{(n+1)\beta_1^{n+1}}{1-\beta_1^{n+1}}-\frac{\beta_2}{1-\beta_2}+\frac{(n+1)\beta_2^{n+1}}{1-\beta_2^{n+1}}+\varepsilon\frac{\frac{\beta_2}{1-\beta_2}-\frac{(n+1)\beta_2^{n+1}}{1-\beta_2^{n+1}}}{|\partial_r\mathcal{L}(\boldsymbol{\theta})|^2+\varepsilon}\bigg)$$
$$\times\frac{\big(\partial_r\|\nabla\mathcal{L}(\boldsymbol{\theta})\|_{1,\varepsilon}+\lambda[\nabla^2\mathcal{L}(\boldsymbol{\theta})\boldsymbol{\theta}]_r\big)}{(|\partial_r\mathcal{L}(\boldsymbol{\theta})|^2+\varepsilon)^{1/2}},\quad r\in[1:d].$$

Proof. The fact that Assumption 3.1 holds is already verified in Section C.

Next, in this case

$$F_r^{(n)}(\boldsymbol{\theta}) = \frac{\partial_r\mathcal{L}(\boldsymbol{\theta})}{\sqrt{\partial_r\mathcal{L}(\boldsymbol{\theta})^2+\varepsilon}}+\lambda\theta_r,$$
$$\frac{\partial F_r^{(n)}}{\partial\theta_i^{(n-k)}}(\boldsymbol{\theta}) = \frac{\frac{1-\beta_1}{1-\beta_1^{n+1}}\beta_1^k\partial_{ir}\mathcal{L}(\boldsymbol{\theta})}{(|\partial_r\mathcal{L}(\boldsymbol{\theta})|^2+\varepsilon)^{1/2}} - \frac{\frac{1-\beta_2}{1-\beta_2^{n+1}}\beta_2^k|\partial_r\mathcal{L}(\boldsymbol{\theta})|^2\partial_{ir}\mathcal{L}(\boldsymbol{\theta})}{(|\partial_r\mathcal{L}(\boldsymbol{\theta})|^2+\varepsilon)^{3/2}}$$

Therefore,

$$\begin{aligned}
M_r^{(n)}(\boldsymbol{\theta}) &= h\sum_{k=1}^{n}\sum_{i=1}^{d}\frac{\partial F_r^{(n)}}{\partial\theta_i^{(n-k)}}(\boldsymbol{\theta})\sum_{s=n-k}^{n-1}F_i^{(s)}(\boldsymbol{\theta})\\
&= h\sum_{i=1}^{d}\frac{\partial_{ir}\mathcal{L}(\boldsymbol{\theta})}{(|\partial_r\mathcal{L}(\boldsymbol{\theta})|^2+\varepsilon)^{1/2}}\bigg[\frac{\partial_i\mathcal{L}(\boldsymbol{\theta})}{(|\partial_i\mathcal{L}(\boldsymbol{\theta})|^2+\varepsilon)^{1/2}}+\lambda\theta_i\bigg]\frac{1-\beta_1}{1-\beta_1^{n+1}}\sum_{k=1}^{n}k\beta_1^k\\
&\quad - h|\partial_r\mathcal{L}(\boldsymbol{\theta})|^2\sum_{i=1}^{d}\frac{\partial_{ir}\mathcal{L}(\boldsymbol{\theta})}{(|\partial_r\mathcal{L}(\boldsymbol{\theta})|^2+\varepsilon)^{3/2}}\bigg[\frac{\partial_i\mathcal{L}(\boldsymbol{\theta})}{(|\partial_i\mathcal{L}(\boldsymbol{\theta})|^2+\varepsilon)^{1/2}}+\lambda\theta_i\bigg]\frac{1-\beta_2}{1-\beta_2^{n+1}}\sum_{k=1}^{n}k\beta_2^k\\
&= h\bigg(\frac{\beta_1}{1-\beta_1}-\frac{(n+1)\beta_1^{n+1}}{1-\beta_1^{n+1}}\bigg)\sum_{i=1}^{d}\frac{\partial_{ir}\mathcal{L}(\boldsymbol{\theta})}{(|\partial_r\mathcal{L}(\boldsymbol{\theta})|^2+\varepsilon)^{1/2}}\bigg[\frac{\partial_i\mathcal{L}(\boldsymbol{\theta})}{(|\partial_i\mathcal{L}(\boldsymbol{\theta})|^2+\varepsilon)^{1/2}}+\lambda\theta_i\bigg]\\
&\quad - h\bigg(\frac{\beta_2}{1-\beta_2}-\frac{(n+1)\beta_2^{n+1}}{1-\beta_2^{n+1}}\bigg)\sum_{i=1}^{d}\frac{|\partial_r\mathcal{L}(\boldsymbol{\theta})|^2\partial_{ir}\mathcal{L}(\boldsymbol{\theta})}{(|\partial_r\mathcal{L}(\boldsymbol{\theta})|^2+\varepsilon)^{3/2}}\bigg[\frac{\partial_i\mathcal{L}(\boldsymbol{\theta})}{(|\partial_i\mathcal{L}(\boldsymbol{\theta})|^2+\varepsilon)^{1/2}}+\lambda\theta_i\bigg]\\
&= h\bigg(\frac{\beta_1}{1-\beta_1}-\frac{(n+1)\beta_1^{n+1}}{1-\beta_1^{n+1}}\bigg)\frac{\big(\partial_r\|\nabla\mathcal{L}(\boldsymbol{\theta})\|_{1,\varepsilon}+\lambda[\nabla^2\mathcal{L}(\boldsymbol{\theta})\boldsymbol{\theta}]_r\big)}{(|\partial_r\mathcal{L}(\boldsymbol{\theta})|^2+\varepsilon)^{1/2}}\\
&\quad - h\bigg(\frac{\beta_2}{1-\beta_2}-\frac{(n+1)\beta_2^{n+1}}{1-\beta_2^{n+1}}\bigg)\frac{|\partial_r\mathcal{L}(\boldsymbol{\theta})|^2\big(\partial_r\|\nabla\mathcal{L}(\boldsymbol{\theta})\|_{1,\varepsilon}+\lambda[\nabla^2\mathcal{L}(\boldsymbol{\theta})\boldsymbol{\theta}]_r\big)}{(|\partial_r\mathcal{L}(\boldsymbol{\theta})|^2+\varepsilon)^{3/2}}\\
&= h\bigg(\frac{\beta_1}{1-\beta_1}-\frac{(n+1)\beta_1^{n+1}}{1-\beta_1^{n+1}}-\frac{\beta_2}{1-\beta_2}+\frac{(n+1)\beta_2^{n+1}}{1-\beta_2^{n+1}}+\varepsilon\frac{\frac{\beta_2}{1-\beta_2}-\frac{(n+1)\beta_2^{n+1}}{1-\beta_2^{n+1}}}{|\partial_r\mathcal{L}(\boldsymbol{\theta})|^2+\varepsilon}\bigg)
\end{aligned}$$

$$\times \frac{\big(\partial_r \|\nabla \mathcal{L}(\boldsymbol{\theta})\|_{1,\varepsilon} + \lambda [\nabla^2 \mathcal{L}(\boldsymbol{\theta})\boldsymbol{\theta}]_r\big)}{(|\partial_r \mathcal{L}(\boldsymbol{\theta})|^2 + \varepsilon)^{1/2}}$$

as desired. □

Lemma I.3 (Application of Corollary 3.3 to Example 1.5)**.** *Let $\{\boldsymbol{\theta}^{(n)}\}_{n\in\mathbb{Z}_{\geqslant 0}}$ be the sequence of vectors generated by the iteration in Equation (1) with initial condition $\boldsymbol{\theta}^{(0)}$, where $\boldsymbol{F}^{(n)}(\cdot)$ is as defined in Example 1.5, the function $\mathcal{K}(\cdot) : \mathcal{M} \to \mathbb{R}$ defined in a open bounded region $\mathcal{M} \subset \mathbb{R}^d$ is three times continuously differentiable with bounded derivatives, and the loss function $\mathcal{L}(\cdot)$ defined in an open convex bounded domain $\mathcal{D} \subset \mathbb{R}^d$ is three times continuously differentiable with bounded derivatives; also, let $T \geqslant 0$ be a fixed "time" horizon. Then Assumption 3.1 holds; in particular, by Corollary 3.3 the inequality $\max_{n\in[0:\lfloor T/h \rfloor]} \big\|\boldsymbol{\theta}^{(n)} - \tilde{\boldsymbol{\theta}}^{(n)}\big\|_\infty \leqslant C_2 h^2$ holds, where*

$$\begin{aligned}
\tilde{\theta}_r^{(n+1)} &= (1-\lambda h)\tilde{\theta}_r^{(n)} - h\big(-\partial_r \mathcal{K}\big(-(1-\rho_1\rho_2^n)\nabla\mathcal{L}(\tilde{\boldsymbol{\theta}}^{(n)})\big) + M_r^{(n)}\big(\tilde{\boldsymbol{\theta}}^{(n)}\big)\big),\\
M_r^{(n)}\big(\boldsymbol{\theta}\big) &= h\rho_1(1-\rho_2)\sum_{i=1}^d\sum_{j=1}^d \partial_{jr}\mathcal{K}\big(-(1-\rho_1\rho_2^n)\nabla\mathcal{L}(\boldsymbol{\theta})\big)\partial_{ij}\mathcal{L}(\boldsymbol{\theta})\\
&\quad\times \sum_{k=1}^n \rho_2^{k-1}\sum_{s=n-k}^{n-1}\big(-\partial_i\mathcal{K}\big(-(1-\rho_1\rho_2^s)\nabla\mathcal{L}(\boldsymbol{\theta})\big) + \lambda\theta_i\big)\\
&= h\frac{\rho_1}{1-\rho_2}\sum_{i=1}^d\sum_{j=1}^d \partial_{jr}\mathcal{K}\big(-\nabla\mathcal{L}(\boldsymbol{\theta})\big)\partial_{ij}\mathcal{L}(\boldsymbol{\theta})\big(-\partial_i\mathcal{K}\big(-\nabla\mathcal{L}(\boldsymbol{\theta})\big) + \lambda\theta_i\big) + o_n(1),
\end{aligned}$$

where $o_n(1)$ is a function of $\boldsymbol{\theta}$ converging to zero uniformly in $\boldsymbol{\theta} \in \mathcal{D}$.

Proof. The fact that Assumption 3.1 holds is already verified in Section C.

Next, in this case

$$\begin{aligned}
F_r^{(n)}(\boldsymbol{\theta}) &= -\partial_r\mathcal{K}\big(-(1-\rho_1\rho_2^n)\nabla\mathcal{L}(\boldsymbol{\theta})\big) + \lambda\theta_r,\\
\frac{\partial F_r^{(n)}}{\partial\theta_i^{(n-k)}}(\boldsymbol{\theta}) &= \sum_{j=1}^d \partial_{jr}\mathcal{K}\big(-(1-\rho_1\rho_2^n)\nabla\mathcal{L}(\boldsymbol{\theta})\big)(1-\rho_2)\rho_1\rho_2^{k-1}\partial_{ij}\mathcal{L}(\boldsymbol{\theta}).
\end{aligned}$$

Therefore,

$$\begin{aligned}
M_r^{(n)}(\boldsymbol{\theta}) &= h\sum_{k=1}^n\sum_{i=1}^d \frac{\partial F_r^{(n)}}{\partial\theta_i^{(n-k)}}(\boldsymbol{\theta})\sum_{s=n-k}^{n-1} F_i^{(s)}(\boldsymbol{\theta})\\
&= h\rho_1(1-\rho_2)\sum_{i=1}^d\sum_{j=1}^d \partial_{jr}\mathcal{K}\big(-(1-\rho_1\rho_2^n)\nabla\mathcal{L}(\boldsymbol{\theta})\big)\partial_{ij}\mathcal{L}(\boldsymbol{\theta})\\
&\quad\times\sum_{k=1}^n\rho_2^{k-1}\sum_{s=n-k}^{n-1}\big(-\partial_i\mathcal{K}\big(-(1-\rho_1\rho_2^s)\nabla\mathcal{L}(\boldsymbol{\theta})\big) + \lambda\theta_i\big)
\end{aligned}$$

Note that

$$\partial_i\mathcal{K}(-\nabla\mathcal{L}(\boldsymbol{\theta})) - \partial_i\mathcal{K}\big(-(1-\rho_1\rho_2^s)\nabla\mathcal{L}(\boldsymbol{\theta})\big) = -\big[\nabla^2\mathcal{K}(\boldsymbol{\zeta})\nabla\mathcal{L}(\boldsymbol{\theta})\big]_i\rho_1\rho_2^s,$$

where $\boldsymbol{\zeta}$ is on the segment between $-(1-\rho_1\rho_2^s)\nabla\mathcal{L}(\boldsymbol{\theta})$ and $-\nabla\mathcal{L}(\boldsymbol{\theta})$. Applying Lemma H.2 with $a_k = \rho_2^{k-1}$ and $b_s = \rho_1\rho_2^s$ we see that

$$\begin{aligned}
&\sum_{k=1}^n\rho_2^{k-1}\sum_{s=n-k}^{n-1}\big(-\partial_i\mathcal{K}\big(-(1-\rho_1\rho_2^s)\nabla\mathcal{L}(\boldsymbol{\theta})\big) + \lambda\theta_i\big)\\
&\quad= \sum_{k=1}^n\rho_2^{k-1}k\big(-\partial_i\mathcal{K}\big(-\nabla\mathcal{L}(\boldsymbol{\theta})\big) + \lambda\theta_i\big) + o_n(1) = \frac{1}{(1-\rho_2)^2}\big(-\partial_i\mathcal{K}\big(-\nabla\mathcal{L}(\boldsymbol{\theta})\big) + \lambda\theta_i\big) + o_n(1),
\end{aligned}$$

where $o_n(1) \to 0$ as $n \to \infty$ uniformly in $\boldsymbol{\theta} \in \mathcal{D}$. Using the boundedness of derivatives again, we also have

$$h\rho_1(1-\rho_2)\sum_{i=1}^{d}\sum_{j=1}^{d}\partial_{jr}\mathcal{K}\big(-(1-\rho_1\rho_2^n)\nabla\mathcal{L}(\boldsymbol{\theta})\big)\partial_{ij}\mathcal{L}(\boldsymbol{\theta})$$

$$= h\rho_1(1-\rho_2)\sum_{i=1}^{d}\sum_{j=1}^{d}\partial_{jr}\mathcal{K}\big(-\nabla\mathcal{L}(\boldsymbol{\theta})\big)\partial_{ij}\mathcal{L}(\boldsymbol{\theta}) + o_n(1)$$

and

$$M_r^{(n)}(\boldsymbol{\theta}) = h\frac{\rho_1}{1-\rho_2}\sum_{i=1}^{d}\sum_{j=1}^{d}\partial_{jr}\mathcal{K}\big(-\nabla\mathcal{L}(\boldsymbol{\theta})\big)\partial_{ij}\mathcal{L}(\boldsymbol{\theta})\big(-\partial_i\mathcal{K}\big(-\nabla\mathcal{L}(\boldsymbol{\theta})\big) + \lambda\theta_i\big) + o_n(1)$$

as desired. □

J Empirical Illustrations

As a sanity check, we verify that the memoryless iteration (4) that we find is closer than the first-order approximation (the one with no correction from memory: for example, the first-order approximation of Adam is sign gradient descent [47]). We see in Figure 1 that the ℓ_∞ (maximal) norm of the weight difference is smaller with the correction term. Note that the learning rates are realistic and weight decay is present.

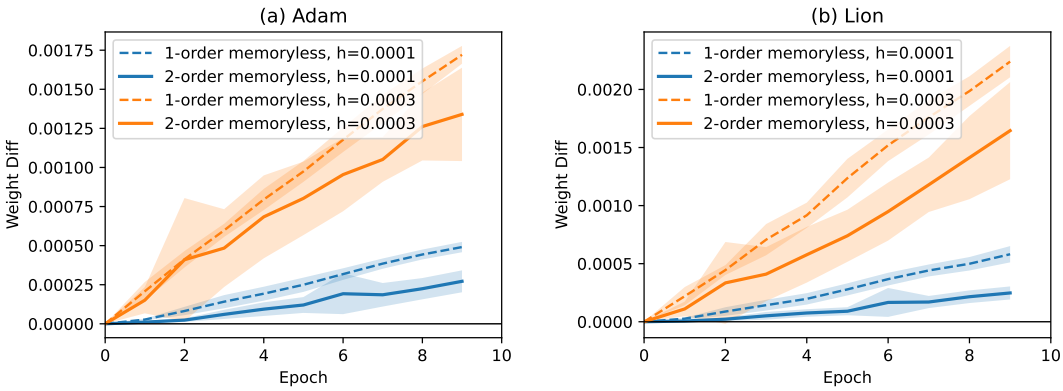

Figure 1: Comparison of the trajectories: ℓ_∞-norm of weight difference between the second-order memoryless method from Theorem 3.2 and the first-order approximation (signGD): **(a)** Adam, **(b)** Lion (perturbed by ε and with bias correction). MLP with GELU activation on MNIST-10K, learning rates $h \in \{10^{-4}, 3\times10^{-4}\}$, weight decay $10^{-3}/h$, $\varepsilon = 10^{-6}$.

As a preliminary empirical illustration, we train ResNet-50 [31] on CIFAR-10 [38] using Adam (with decoupled weight decay) and Lion. We keep the β_1 parameter of Adam at 0.99 (for stable training on CIFAR-10 [47, 11]) and sweep the β_2 parameter. We plot in Figure 2(a) the test accuracy at a fixed small training loss threshold (controlling for training speed). As predicted in Section 4 and confirming the observation from [11] for pure Adam without weight decay, the test accuracy drops as β_2 approaches one. We see that Adam with lower values of β_2 can sometimes outperform Lion with default hyperparameters and thus close the generalization gap between these two algorithms, consistent with the theory. We also observe this phenomenon on a language task by training Transformer-XL [18] on WikiText-2 [49]. We fix the default $\beta_1 = 0.9$ for Adam and sweep β_2, plotting the minimal validation perplexity achieved before overfitting; as in the vision task, we compare with Lion whose hyperparameters are set at default values. We observe in Figure 2(b) the same trends as above (in large-batch training, higher β_2 increases the best validation perplexity, that is, hurts generalization; sometimes, taking lower β_2 can close the gap between Adam and Lion).

We provide some additional sweeps with different learning rates in Figures 3 and 4.

The code is available at `https://github.com/borshigida/how-memory-modifies-loss`.

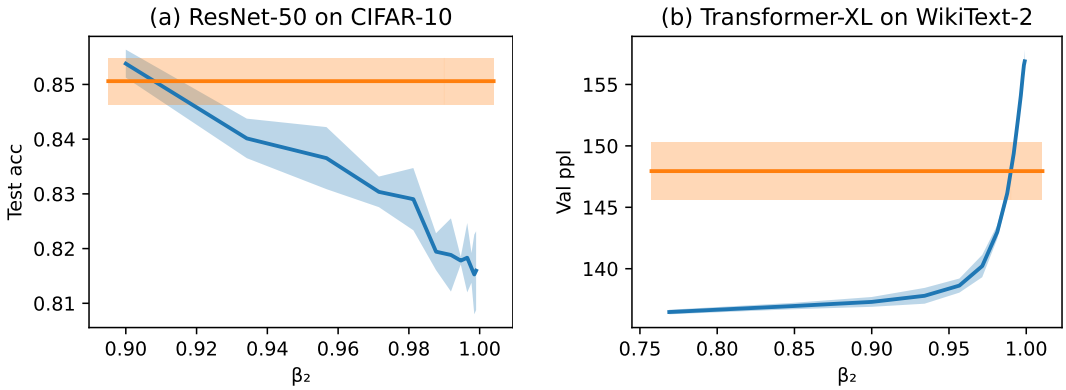

Figure 2: **(a)** ResNet-50 on CIFAR-10: test accuracy at training loss threshold 0.05. Full-batch Adam, learning rate $h = 10^{-3.5}$, $\beta_1 = 0.99$, $\varepsilon = 10^{-6}$, weight decay 0.005. For comparison, we also show Lion with the same learning rate and weight decay (with default $\rho_1 = 0.9$, $\rho_2 = 0.99$). **(b)** Minimal validation perplexity (before overfitting) of Transformer-XL trained with full-batch Adam on WikiText-2 with learning rate 10^{-4}, $\beta_1 = 0.9$, $\varepsilon = 10^{-6}$. For comparison, we also show Lion (with default $\rho_1 = 0.9$, $\rho_2 = 0.99$). All results are averaged over three iterations.

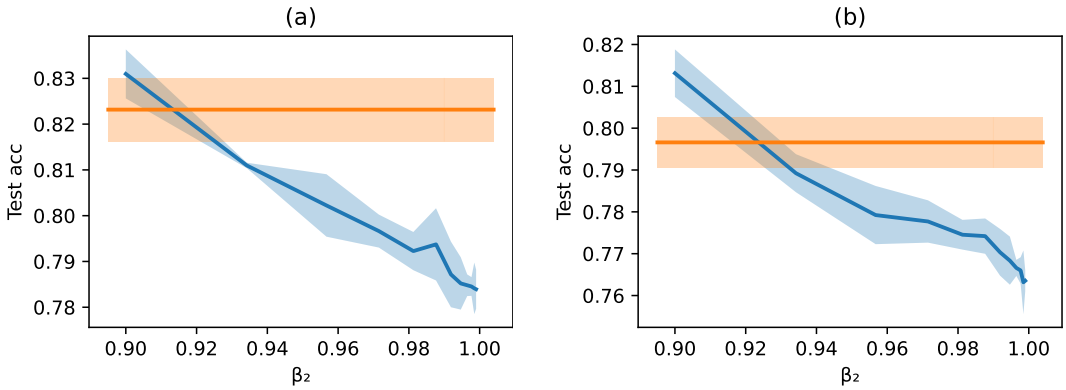

Figure 3: ResNet-50 on CIFAR-10: test accuracy at training loss threshold 0.05. Full-batch Adam, learning rates **(a)** $h = 10^{-4}$; **(b)** $h = 5 \times 10^{-5}$, $\beta_1 = 0.99$, $\varepsilon = 10^{-6}$, weight decay $5 \times 10^{-6.5}/h$. For comparison, we also show Lion with the same learning rates and weight decay (with default $\rho_1 = 0.9$, $\rho_2 = 0.99$). All results are averaged over three iterations.

J.1 A Note on the Edge of Stability and Comparisons with Lion on Vision Tasks

Cohen et al. [16] notice that in a sense Adam trains at the edge of stability. They view Adam as momentum gradient descent with evolving preconditioner

$$\mathbf{P}_{t+1} = (1 - \beta_1^{t+1}) \left[\operatorname{diag} \left(\sqrt{\frac{\boldsymbol{\nu}_{t+1}}{1 - \beta_2^{t+1}}} \right) + \epsilon \mathbf{I} \right].$$

They define "preconditioned sharpness" to be the top eigenvalue of the preconditioned Hessian $\lambda_1(\mathbf{P}_t^{-1}\mathbf{H}_t)$, where $\mathbf{H}_t$ is the Hessian of the loss, and observe that this quantity often oscillates around the stability threshold $\frac{2+2\beta_1}{(1-\beta_1)\eta}$, where η is the learning rate. (This fraction comes from the fact that if the preconditioner were constant, Adam would become a form of preconditioned gradient descent with EMA-style momentum, and this is the ordinary stability threshold of EMA-style heavy-ball momentum on the quadratic Taylor approximation of the loss; we refer to Cohen et al. [16] for details.) They use large-batch training on CIFAR-10/100. We train a CNN on CIFAR-10 as well, and reproduce this result in Figure 5. We also plot ordinary sharpness $\lambda_1(\mathbf{H}_t)$ (top hessian eigenvalue), which first increases and then decreases. Recall that this is an extremely unstable regime of training [47].

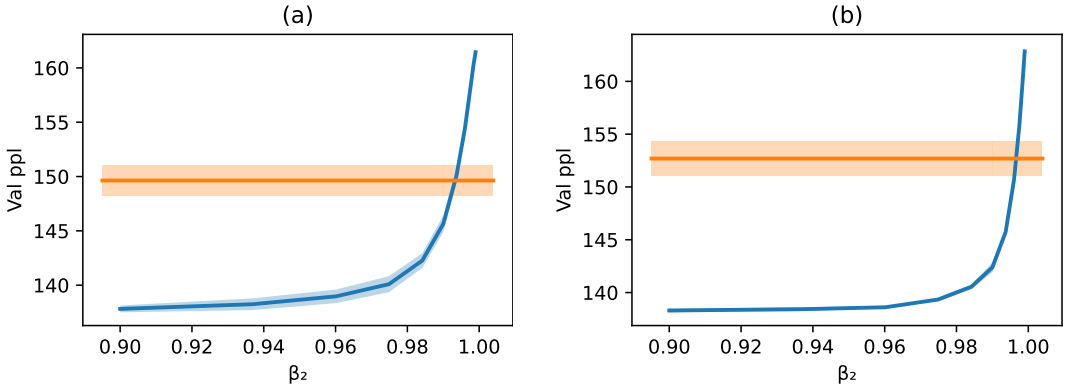

Figure 4: Minimal validation perplexity (before overfitting) of Transformer-XL trained with full-batch Adam on WikiText-2 with learning rates **(a)** $h = 5 \times 10^{-5}$; **(b)** $h = 2.5 \times 10^{-5}$, weight decay $10^{-8}/h$, $\beta_1 = 0.9$, $\varepsilon = 10^{-6}$. For comparison, we also show Lion with the same learning rates and weight decay (with default $\rho_1 = 0.9$, $\rho_2 = 0.99$). All results are averaged over three iterations.

Note that the parameter controlling the exponential forgetting of gradients β_1 corresponds to the ρ_2 parameter of Lion, so the default $\rho_2 = 0.99$ in Lion would match $\beta_1 = 0.99$ rather than $\beta_1 = 0.9$ in Adam. If we take $\beta_1 = 0.99$ which is the "smooth" regime of training, preconditioned sharpness does not reach the stability threshold (Figure 6). Note also that ordinary sharpness $\lambda_1(\mathbf{H}_t)$ (top hessian eigenvalue) is much lower for small β_2 (especially noticeable for $\beta_1 \leqslant 0.9$). This suggests that in large-batch training on vision tasks, taking $\beta_2 < \beta_1 = 0.99$ strongly regularizes training, moving the model parameters to flatter regions of the loss space. It is a promising direction to investigate the limits of such regularization: for example, taking β_2 near the threshold of divergence may regularize training so much that the default Lion will not be able to match in terms of generalization error.

K Experiment Details and Licenses

Our implementation of ResNet-50 follows the one from [11] (small modification of the standard `torchvision` implementation to allow for training on CIFAR-10 rather than ImageNet). The `torchvision` repository has the BSD 3-Clause license. CIFAR-10 is released without an explicit license. MNIST has the CC BY-SA 3.0 license.

Our implementation of training Transformer-XL on WikiText-2 follows the one from [41] which is a small modification of the codebase[2] for [18], licensed under the Apache-2.0 License. The WikiText-2 dataset is released under the CC BY-SA 3.0 license.

For Adam (with decoupled weight decay), we use the standard implementation from `pytorch.optim`; Lion is taken from the `google/automl` repository[3]. This repository is licensed under the Apache License 2.0. The implementations of the optimizers used for comparing the trajectories in Figure 1 of the paper are custom and match exactly our analytical formulas (in particular, Lion has bias correction and the soft-sign function $x \mapsto x/\sqrt{x^2+\varepsilon}$ instead of the sign function), given below.

AdamW: memoryless update The (full-batch) memoryless AdamW approximation is

$$\theta_j^{(n+1)} = \theta_j^{(n)} - hF_j^{(n)}(\boldsymbol{\theta}^{(n)}) - hM_j^{(n)}(\boldsymbol{\theta}^{(n)}),$$

where

$$F_j^{(n)}(\boldsymbol{\theta}) = \frac{\nabla_j \mathcal{L}(\boldsymbol{\theta})}{\sqrt{|\nabla_j \mathcal{L}(\boldsymbol{\theta})|^2 + \varepsilon}} + \lambda\theta_j,$$

$$M_j^{(n)}(\boldsymbol{\theta}) = -h\left(\frac{\beta_2}{1-\beta_2} - \frac{(n+1)\beta_2^{n+1}}{1-\beta_2^{n+1}}\right) \frac{|\nabla_j \mathcal{L}(\boldsymbol{\theta})|^2 \big(\nabla_j \|\nabla \mathcal{L}(\boldsymbol{\theta})\|_{1,\varepsilon} + \lambda[\nabla^2 \mathcal{L}(\boldsymbol{\theta})\boldsymbol{\theta}]_j\big)}{(|\nabla_j \mathcal{L}(\boldsymbol{\theta})|^2 + \varepsilon)^{3/2}}$$

[2] https://github.com/kimiyoung/transformer-xl
[3] https://github.com/google/automl/tree/master/lion

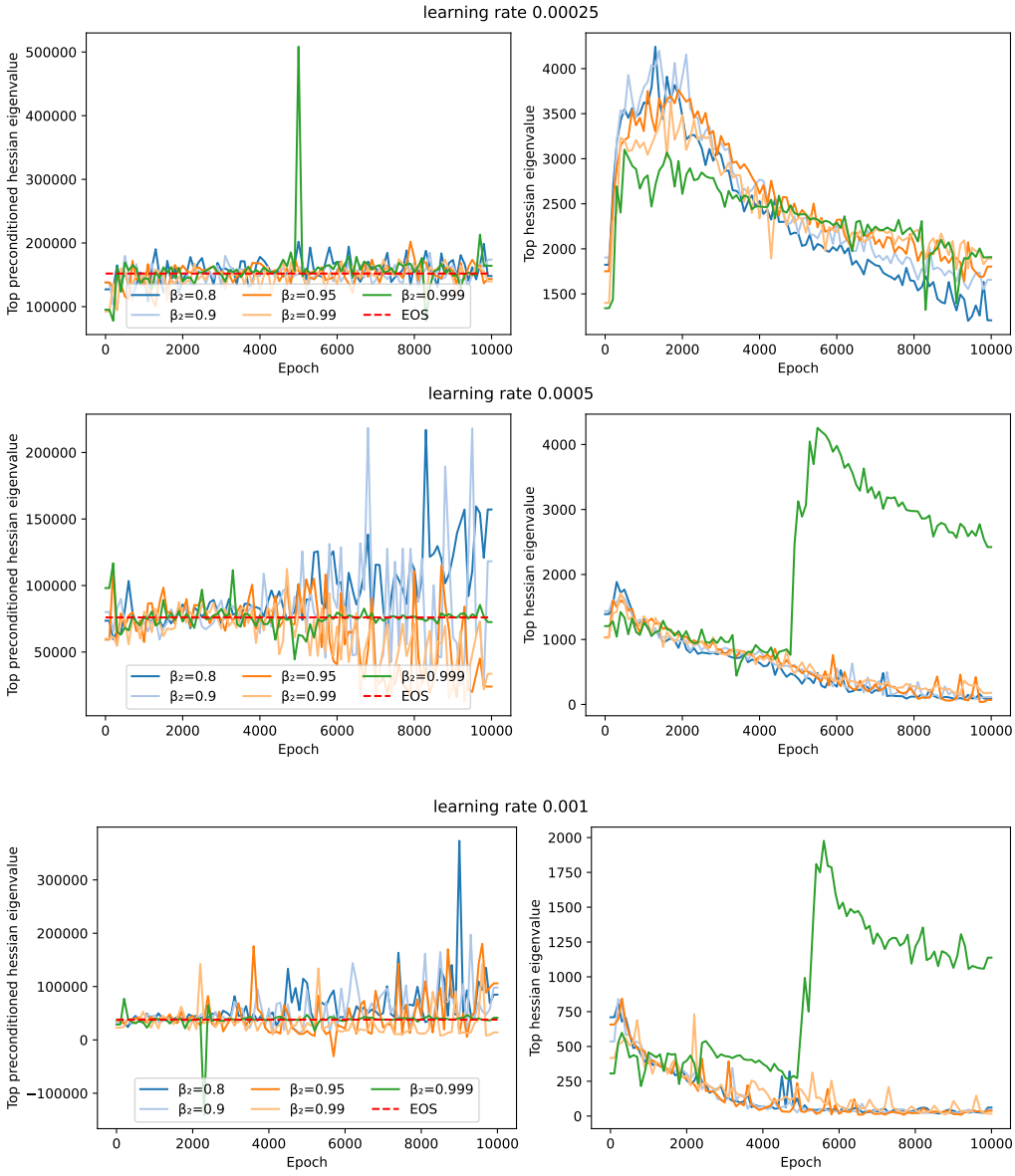

Figure 5: CNN trained on CIFAR-10 with full-batch Adam, $\beta_1 = 0.9$, $\epsilon = 10^{-6}$, learning rate 0.001. Left: preconditioned sharpness $\lambda_1(\mathbf{P}_t^{-1}\mathbf{H}_t)$ oscillates around the stability threshold. Right: the plots of ordinary sharpness $\lambda_1(\mathbf{H}_t)$.

$$+ h\left(\frac{\beta_1}{1-\beta_1} - \frac{(n+1)\beta_1^{n+1}}{1-\beta_1^{n+1}}\right)\frac{\big(\nabla_j \|\nabla \mathcal{L}(\boldsymbol{\theta})\|_{1,\varepsilon} + \lambda[\nabla^2 \mathcal{L}(\boldsymbol{\theta})\boldsymbol{\theta}]_j\big)}{\sqrt{|\nabla_j \mathcal{L}(\boldsymbol{\theta})|^2 + \varepsilon}}.$$

Lion-$\mathcal{K}$ with bias correction The Lion-$\mathcal{K}$ algorithm with bias correction is defined as in Example 1.5 of the paper except bias correction is added:

$$\boldsymbol{F}^{(n)}(\boldsymbol{\theta}^{(n)}, \dots, \boldsymbol{\theta}^{(0)}) = -\nabla \mathcal{K}(\boldsymbol{m}_1^{(n+1)} + \boldsymbol{m}_2^{(n+1)}) + \boldsymbol{m}_3^{(n+1)},$$

$$\text{where} \quad \boldsymbol{m}_1^{(n+1)} = -\frac{1-\rho_2}{1-\rho_2^{n+1}}\frac{\rho_1}{\rho_2}\sum_{k=0}^{n}\rho_2^{n-k}\nabla \mathcal{L}(\boldsymbol{\theta}^{(k)}),$$

$$\boldsymbol{m}_2^{(n+1)} = -\left(1 - \frac{\rho_1}{\rho_2}\right)\nabla \mathcal{L}(\boldsymbol{\theta}^{(n)}),$$

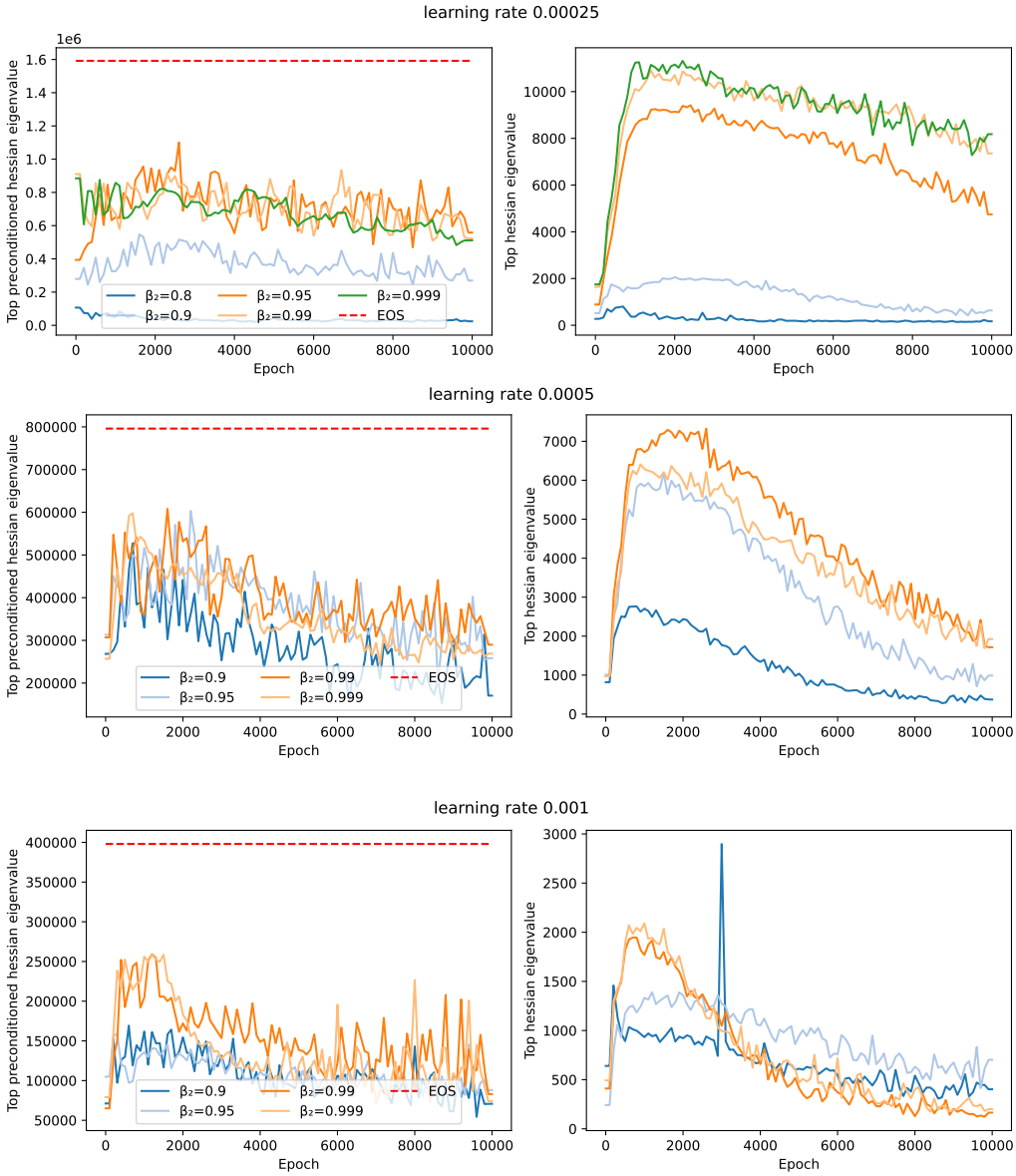

Figure 6: CNN trained on CIFAR-10 with full-batch Adam, $\beta_1 = 0.99$, $\epsilon = 10^{-6}$, learning rate 0.001. Left: preconditioned sharpness $\lambda_1(\mathbf{P}_t^{-1}\mathbf{H}_t)$ does not reach the stability threshold. Right: the plots of ordinary sharpness $\lambda_1(\mathbf{H}_t)$.

$$\boldsymbol{m}_3^{(n+1)} = \lambda \boldsymbol{\theta}^{(n)}.$$

Lion (perturbed by ε): memoryless update The memoryless iteration is given by

$$\theta_j^{(n+1)} = \theta_j^{(n)} - hF_j^{(n)}(\boldsymbol{\theta}^{(n)}) - hM_j^{(n)}(\boldsymbol{\theta}^{(n)}),$$

where

$$F_j^{(n)}(\boldsymbol{\theta}) = \frac{\nabla_j \mathcal{L}(\boldsymbol{\theta})}{\sqrt{|\nabla_j \mathcal{L}(\boldsymbol{\theta})|^2 + \varepsilon}} + \lambda\theta_j,$$
$$M_j^{(n)}(\boldsymbol{\theta}) = h\left[\frac{\rho_1}{1-\rho_2} - \frac{(n+1)\rho_2^n \rho_1}{1-\rho_2^{n+1}}\right]$$

$$\times \frac{\varepsilon}{\left(|\nabla_j \mathcal{L}(\boldsymbol{\theta})|^2 + \varepsilon\right)^{3/2}} \nabla_j \left[\|\nabla \mathcal{L}(\boldsymbol{\theta})\|_{1,\varepsilon} + \lambda(\nabla \mathcal{L}(\boldsymbol{\theta})^\top \boldsymbol{\theta} - \mathcal{L}(\boldsymbol{\theta})) \right].$$

K.1 Compute Resources

One sweep of hyperparameter β_2 contained about 12 runs, with each run repeated for 3 iterations. Each run took about 10 hours on average on one machine with a devoted 40 GB NVIDIA A100 GPU (though the training horizon was longer than necessary). This puts compute resources at around $12 \times 10 \times 3 = 360$ A100-GPU-hours per sweep. In Figure 2, two sweeps were conducted. The experiments on truncated MNIST conducted to produce Figure 1 used negligible resources compared to the sweeps described (less than 1 GPU-hour). Additional compute was used for preliminary experimentation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We substantiate all claims made in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We include the proofs of our theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experiment details are discussed in figure captions and relevant appendices.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to eproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code, along with instructions to reproduce the results, is released publicly and referenced in the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We discuss experimental details in figure captions and the appendices.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We include the error bars in our figures.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide compute resources in the appendices.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have checked that we conform to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We include a short note on broader impacts in Section B.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper does not pose any non-trivial risks of misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We provide the license information in Section K.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not conduct crowdsourcing or experiments with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not conduct crowdsourcing or experiments with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you sho ld clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We only used LLMs for code completion, reformatting assistance and brainstorming.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.