Implicit Bias of AdamW: ℓ_∞-Norm Constrained Optimization

Shuo Xie [1] **Zhiyuan Li** [1]

Abstract

Adam with decoupled weight decay, also known as AdamW, is widely acclaimed for its superior performance in language modeling tasks, surpassing Adam with ℓ_2 regularization in terms of generalization and optimization. However, this advantage is not theoretically well-understood. One challenge here is that though intuitively Adam with ℓ_2 regularization optimizes the ℓ_2 regularized loss, it is not clear if AdamW optimizes a specific objective. In this work, we make progress toward understanding the benefit of AdamW by showing that it implicitly performs constrained optimization. More concretely, we show in the full-batch setting, if AdamW converges with any non-increasing learning rate schedule whose partial sum diverges, it must converge to a KKT point of the original loss under the constraint that the ℓ_∞ norm of the parameter is bounded by the inverse of the weight decay factor. This result is built on the observation that Adam can be viewed as a smoothed version of SignGD, which is the normalized steepest descent with respect to ℓ_∞ norm, and a surprising connection between normalized steepest descent with weight decay and Frank-Wolfe.

1. Introduction

`Adam` (Kingma & Ba, 2014) and its variant `AdamW` (Loshchilov & Hutter, 2018) have been the most successful and widely used optimization algorithms in deep learning, especially for large language models (LLMs), whose pretraining costs massively and cannot be done with SGD. Despite its tremendous empirical success, we lack a good theoretical understanding of `Adam`'s underlying work and the roles of its hyperparameters, in particular, *weight decay*. `AdamW` achieves better optimization and generalization over `Adam` and decouples the effect of the learning rate and the weight decay coefficient by using a different implementation of weight decay (Loshchilov & Hutter, 2018). While `Adam` implements weight decay as a ℓ_2 regularization of the training objective, `AdamW` directly shrinks its weight per step, known as the *decoupled weight decay* (see Algorithm 1).

However, the advantage of `AdamW` over `Adam` is mostly empirical and our theoretical understanding is quite limited. Zhuang et al. (2022) argues that one desirable property that `AdamW` has while `Adam` does not is scale-freeness, meaning `AdamW` yields the same optimization trajectory if loss is multiplied by any positive constant. Yet this property does not give us enough information to understand the difference regarding the optimization processes and the final learned solutions between `AdamW` and `Adam` with ℓ_2 regularization. Intuitively, if `Adam` with ℓ_2 regularization converges to some point, it converges to at least a stationary point of the regularized loss function, if not a minimizer. But for `AdamW`, it is even not clear if it is optimizing any (regularized) loss function. Thus, towards taking the first step of understanding the benefit of decoupled weight decay in `AdamW`, we ask the following question:

> Which solution does `AdamW` converge to, if it converges?

Our following main result Theorem 1.1 characterizes the implicit bias of `AdamW` in the *deterministic* case, where a full-batch loss is used:

Theorem 1.1. *For any continuously differentiable function $L : \mathbb{R}^d \to \mathbb{R}$, $\beta_1 \le \beta_2 < 1$, initialization $\boldsymbol{x}_0$ and non-increasing learning rate $\{\eta_t\}_{t=1}^\infty$ such that $\sum_{t=1}^\infty \eta_t = \infty$, if the iterates of* `AdamW` *$\{\boldsymbol{x}_t\}_{t=0}^\infty$ on L converges to some $\boldsymbol{x}_\infty$, then $\boldsymbol{x}_\infty$ is a KKT point (Definition 3.6) of the constrained optimization problem $\min_{\|\boldsymbol{x}\|_\infty \le \frac{1}{\lambda}} L(\boldsymbol{x})$.*

If L is additionally convex, then `AdamW` *converges to the constrained minimizer,* i.e., $\boldsymbol{x}_\infty \in \arg\min_{\|\boldsymbol{x}\|_\infty \le \frac{1}{\lambda}} L(\boldsymbol{x})$.

Despite being simplistic, the full-batch setting is still a very interesting and highly non-trivial regime, because the two main hypotheses of why `Adam` outperforms `SGD` got challenged recently in the deterministic regime (Kunstner et al., 2022). The first hypothesis is that `Adam` outperforms `SGD` by better handling heavy-tailed noise (Zhang et al., 2020).

[1]Toyota Technological Institute at Chicago, IL, the United States. Correspondence to: Shuo Xie <shuox@ttic.edu>, Zhiyuan Li <zhiyuanli@ttic.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Algorithm 1 Adam with ℓ_2 regularization and Adam with decoupled weight decay (`AdamW`)

Input: $\beta_1, \beta_2 > 0$, initialization $\boldsymbol{x}_0$, total steps T, learning rate schedule $\{\eta_t\}_{t=1}^T$, weight decay coefficient λ
$\boldsymbol{m}_0 \leftarrow \mathbf{0}, \boldsymbol{v}_0 \leftarrow \mathbf{0}$
for $t = 1, 2, \cdots, T$ **do**
  $\boldsymbol{g}_t \leftarrow \nabla L(\boldsymbol{x}_{t-1}) + \lambda \boldsymbol{x}_{t-1}$
  $\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t$
  $\boldsymbol{v}_t \leftarrow \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\boldsymbol{g}_t^2$
  $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_{t-1} - \eta_t \frac{\boldsymbol{m}_t}{\sqrt{\boldsymbol{v}_t}} - \lambda \eta_t \boldsymbol{x}_{t-1}$
end for
return $\boldsymbol{x}_T$

However, Kunstner et al. (2022) finds that `Adam` still outperforms `GD` for optimizing language tasks even in the full-batch setting. The second hypothesis is the smoothness of the training loss landscape can linearly increase as the gradient norm increases and thus clipping or normalization is necessary for gradient descent. Intriguingly, Kunstner et al. (2022) finds that normalizing each update of GD cannot close the gap towards `Adam` in the full-batch setting, but normalizing *each coordinate* to its sign (*i.e.*, `SignGD`) closes the gap. The theoretical results and analysis in this work support the empirical observation made by Kunstner et al. (2022). The way we prove Theorem 1.1 is first to prove that normalized steepest descent with weight decay (`NSD-WD`) for any norm $\|\cdot\|$ must converge to KKT points of the constrained optimization problem $\min_{\|\boldsymbol{x}\| \le \frac{1}{\lambda}} L(\boldsymbol{x})$ (Theorem 3.7). Then we show that `AdamW` asymptotically behaves just like `SignGD` with weight decay, which is the normalized steepest descent w.r.t. ℓ_∞ norm with weight decay and the same proof framework generalizes to `AdamW` if $\beta_1 \le \beta_2 < 1$. The condition $\beta_1 \le \beta_2$ is crucial, and we provide a counter-example where $1 > \beta_1 > \beta_2$ and `AdamW` converges somewhere else, instead of a constrained minimizer.

It remains interesting why `SignGD` beats `Normalized-GD` in the full batch setting. They are both normalized steepest descent but with respect to different norms – `Normalized-GD` picks the steepest direction under the geometry of ℓ_2 norm, while `SignGD` picks the steepest direction under the geometry of the ℓ_∞ norm. It is natural to make the following conjecture: `Adam` *outperforms* `GD` *due to its utilization of* ℓ_∞ *geometry, under which the loss function could have better properties, e.g., smaller smoothness*. Our main result Theorem 1.1 provides positive evidence for this conjecture. We also provide a convergence analysis for normalized steepest descent with weight decay for convex loss, where the suboptimality against the constrained minimizer in norm ball of radius $\frac{1}{\lambda}$ vanishes is $O(\frac{H}{T\lambda^2})$, where T is the total number of steps, λ is the weight decay factor, and H is the smoothness of loss w.r.t. the particular norm used for picking the steepest descent direction. Based on the convergence bound, we construct a concrete d-dimensional loss function in Section 3.1 whose minimizer $\boldsymbol{x}^*$ satisfies $\|\boldsymbol{x}^*\|_2 \approx \sqrt{d}\,\|\boldsymbol{x}^*\|_\infty$ and `SignGD` with weight decay converges much faster than `Normalized-GD` with weight decay because `SignGD` with weight decay can use a $\sqrt{d}$ times larger weight decay factor λ than `Normalized-GD`.

Contributions. Below we summarize our contributions:

1. In Section 3.1, we prove normalized steepest descent with weight decay optimizes convex functions under norm constraints (Theorem 3.5). In Section 3.2, we prove it must converge to KKT points of the norm-constrained optimization problem for general loss functions if it converges with a learning rate schedule whose partial sum diverges (Theorem 3.7).
2. In Section 4, we prove `AdamW` must converge to KKT points of the norm-constrained optimization problem for general loss functions if it converges with a non-increasing learning rate schedule whose partial sum diverges (Theorem 1.1).
3. Towards generalizing the proof of Theorem 3.7 to Theorem 1.1, we prove a novel and tight upper bound on average update size of `Adam` (Lemma 4.2), which holds even for non-deterministic settings as well and might be of independent interest to the community. We test various predictions made by our bound in experiments.

2. Preliminaries and Notations

Notations: We use $\|\cdot\|$ to denote a general norm and $\|\cdot\|_*$ to denote its dual norm. We say a function $L : \mathbb{R}^d \to \mathbb{R}$ has H-lipschitz gradient w.r.t. norm $\|\cdot\|$ for some $H > 0$ iff for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, $\|\nabla L(\boldsymbol{x}) - \nabla L(\boldsymbol{y})\|_* \le H\|\boldsymbol{x} - \boldsymbol{y}\|$. We define the *smoothness* of loss L as the smallest positive H w.r.t. $\|\cdot\|$ such that L has H-lipschitz gradient. We say a function $L : \mathbb{R}^d \to \mathbb{R}$ is convex iff for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ and $\theta \in [0, 1]$, it holds that $L(\theta\boldsymbol{x} + (1-\theta)\boldsymbol{y}) \le \theta L(\boldsymbol{x}) + (1-\theta)L(\boldsymbol{y})$. We define the *subgradients* of convex function L at point $\boldsymbol{x}$ as $\{\boldsymbol{g} \in \mathbb{R}^d \mid L(\boldsymbol{y}) \ge L(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \boldsymbol{g}\rangle, \forall \boldsymbol{y} \in \mathbb{R}^d\}$, which is denoted by $\partial L(\boldsymbol{x})$. When L is differentiable at $\boldsymbol{x}$, $\partial L(\boldsymbol{x})$ contains only one element, which is the gradient $\nabla L(\boldsymbol{x})$. In particular, all norms are convex functions and we have the following standard lemma for the subgradients of norms:

Lemma 2.1. *For any norm* $\|\cdot\|$ *and* $\boldsymbol{x} \in \mathbb{R}^d$,

$$\partial \|\boldsymbol{x}\| = \{\boldsymbol{\Delta} \in \mathbb{R}^d \mid \|\boldsymbol{\Delta}\|_* = 1, \langle \boldsymbol{\Delta}, \boldsymbol{x}\rangle = \|\boldsymbol{x}\|\}.$$

Steepest Descent: We say $\boldsymbol{v}$ is a *steepest descent direction* for objective function L at current iterate $\boldsymbol{x}$ w.r.t. norm $\|\cdot\|$

iff $\|\boldsymbol{v}\| = 1$ and $\langle \boldsymbol{v}, \nabla L(\boldsymbol{x})\rangle = \min_{\|\boldsymbol{v}'\|\le 1} \langle \boldsymbol{v}', \nabla L(\boldsymbol{x})\rangle$. Thus for all steepest descent direction $\boldsymbol{v}$, we have that $\langle \boldsymbol{v}, \nabla L(\boldsymbol{x})\rangle = -\|\nabla L(\boldsymbol{x})\|_*$.

Given initialization $\boldsymbol{x}_0$, learning rate schedule $\{\eta_t\}_{t=0}^{\infty}$ and weight decay factor λ, the tth iterate of *normalized steepest descent* w.r.t. $\|\cdot\|$ with decoupled weight decay is defined as

$$\begin{aligned} \boldsymbol{x}_t &= (1-\lambda\eta_t)\boldsymbol{x}_{t-1} - \eta_t \boldsymbol{\Delta}_t, \\ \text{where} \quad \boldsymbol{\Delta}_t &\in \arg\max_{\|\boldsymbol{\Delta}\|\le 1} \nabla L(\boldsymbol{x}_{t-1})^\top \boldsymbol{\Delta}. \end{aligned} \tag{1}$$

Because the dual norm of the dual norm is always equal to the original norm, by Lemma 2.1, we can also characterize the steepest descent directions as the subgradient of its dual norm.

Lemma 2.2. $\arg\max_{\|\boldsymbol{\Delta}\|\le 1} \nabla L(\boldsymbol{x})^\top \boldsymbol{\Delta} = \partial \|\boldsymbol{y}\|_*|_{\boldsymbol{y}=\nabla L(\boldsymbol{x})}$.

For completeness, we also define the *steepest descent* w.r.t. $\|\cdot\|$ with decoupled weight decay below, though we will not use it in our analysis. If we pick ℓ_2 norm, Equation 2 becomes standard GD.

$$\begin{aligned} \tilde{\boldsymbol{x}}_t &= (1-\lambda\eta_t)\tilde{\boldsymbol{x}}_{t-1} - \eta_t \tilde{\boldsymbol{\Delta}}_t, \\ \text{where } \tilde{\boldsymbol{\Delta}}_t &\in \arg\max_{\tilde{\boldsymbol{\Delta}}\in\mathbb{R}^d} \left(\nabla L(\boldsymbol{x}_{t-1})^\top \tilde{\boldsymbol{\Delta}} - \frac{1}{2}\left\|\tilde{\boldsymbol{\Delta}}\right\|^2 \right). \end{aligned} \tag{2}$$

It can be shown that for each steepest descent update $\tilde{\boldsymbol{\Delta}}$ for objective L at $\boldsymbol{x}$, there exists some normalized steepest descent update $\boldsymbol{\Delta}$ satisfying $\tilde{\boldsymbol{\Delta}} = \|\nabla L(\boldsymbol{x})\|_* \boldsymbol{\Delta}$.

3. Warm Up: Implicit Bias of Normalized Steepest Descent w. Weight Decay

In this section, we aim to present some high-level intuition about the constrained-minimization implicit bias of `AdamW` (Theorem 1.1), by showing the same implicit bias for `SignGD` with weight decay, or equivalently, normalized steepest descent w.r.t. ℓ_∞ norm. `AdamW` is arguably a smoothed version of `SignGD`, which reduces the correlation between its numerator and denominator by using past moving average and thus reduces the biasedness of the update direction in the presence of noise. But intuitively, their behaviors are similar when there is no noise and the learning rate is small.

Our analysis in this section holds for all norms, including the non-differentiable ones, like $\|\cdot\|_\infty$.

3.1. Convex Setting: Constrained Optimization

In this subsection, we give a simple non-asymptotic convergence analysis for normalized Steepest descent w. weight decay (`NSD-WD`) w.r.t. to general norms over smooth convex loss functions. If the norm of initialization is no larger than $\frac{1}{\lambda}$ where λ is the weight decay factor then surprisingly `NSD-WD` is exactly equivalent to a well-known optimization algorithm in literature, `Frank-Wolfe` (Frank et al., 1956), where the constraint set here is the norm ball with radius $\frac{1}{\lambda}$. If the norm of initialization is larger than $\frac{1}{\lambda}$, then the analysis contains an additional phase where the norm of iterates linearly converges to $\frac{1}{\lambda}$. In this case, the iterate of `NSD-WD` may always be outside the $\frac{1}{\lambda}$ norm ball, but still, the convergence analysis of `Frank-Wolfe` can be adopted (*e.g.*,Jaggi (2013)). First, we show that the norm of the iterates will shrink to $\frac{1}{\lambda}$ as long as the norm of each update is bounded by 1, i.e., $\|\boldsymbol{\Delta}_t\| \le 1$. Note this conclusion doesn't use the convexity of the function $L(\boldsymbol{x})$ nor the update $\boldsymbol{\Delta}_t$ being the steepest descent direction. It can hold under non-deterministic settings.

Lemma 3.1. *For any learning rate schedule $\{\eta_t\}_{t=1}^{\infty}$ and update $\{\boldsymbol{\Delta}_t\}_{t=1}^{\infty}$ such that $\lambda\eta_t < 1$ and $\|\boldsymbol{\Delta}_t\| \le 1$,*

$$\|\boldsymbol{x}_t\| - \frac{1}{\lambda} \le \max\left(e^{-\lambda\sum_{i=1}^{t}\eta_i} \left(\|\boldsymbol{x}_0\| - \frac{1}{\lambda} \right), 0 \right).$$

The proof is deferred to Appendix A.1. Lemma 3.1 shows that $\boldsymbol{x}_t$ is either always inside the norm ball with radius $\frac{1}{\lambda}$, or their distance shrinks exponentially as the sum of learning rates increases. Whenever $\boldsymbol{x}_t$ gets into the norm ball with radius $\frac{1}{\lambda}$, $\boldsymbol{x}_t$ will not leave it and the remaining trajectory of `NSD-WD` is exactly the same as `Frank-Wolfe`, as shown in the following theorem. We note the relationship between `Frank-Wolfe` and steepest descent algorithms is also observed very recently in the continuous case (Chen et al., 2023).

Theorem 3.2. *For any norm $\|\cdot\|$, weight decay λ, and $\|\boldsymbol{x}_{t-1}\| \le \frac{1}{\lambda}$,* `NSD-WD` *with learning rate $\eta_t < \frac{1}{\lambda}$ and* `Frank-Wolfe` *(Algorithm 2) with step size $\gamma_t = \eta_t\lambda$ and convex set $\mathcal{X} \triangleq \{\boldsymbol{y} \mid \|\boldsymbol{y}\| \le \frac{1}{\lambda}\}$ generate the same next iterate $\boldsymbol{x}_t$.*

Algorithm 2 `Frank-Wolfe`

Input: convex set $\mathcal{X}$, $\boldsymbol{x}_0 \in \mathcal{X}$, total steps T, step sizes $\{\gamma_t\}_{t=1}^{T}$
for $t = 1, 2, \cdots, T$ **do**
  $\boldsymbol{y}_t \leftarrow \arg\min_{\boldsymbol{y}\in\mathcal{X}} \nabla L(\boldsymbol{x}_{t-1})^\top \boldsymbol{y}$
  $\boldsymbol{x}_t \leftarrow (1-\gamma_t)\boldsymbol{x}_{t-1} + \gamma_t \boldsymbol{y}_t$
end for
return $\boldsymbol{x}_T$

Define $\boldsymbol{x}^* = \arg\min_{\|\boldsymbol{x}\|\le\frac{1}{\lambda}} L(\boldsymbol{x})$ to be the constrained minimizer of convex function $L(\boldsymbol{x})$. We first compute how much the gap between $L(\boldsymbol{x}_t)$ and $L(\boldsymbol{x}^*)$ can decrease in one normalized steepest descent step when the iterate $\boldsymbol{x}_t$ is bounded.

Lemma 3.3 (Descent Lemma for Smooth Convex Loss)**.** *Suppose loss function L is convex and has H-lipschitz gra-*

dient w.r.t. norm $\|\cdot\|$. *For iterates* $\{\boldsymbol{x}_t\}$ *in* NSD-WD *(Equation 1), we have that*

$$L(\boldsymbol{x}_t) - L(\boldsymbol{x}^*) \leq (1-\lambda\eta_t)(L(\boldsymbol{x}_{t-1}) - L(\boldsymbol{x}^*)) + \frac{H}{2}\eta_t^2 \left(1+\lambda \|\boldsymbol{x}_{t-1}\|\right)^2.$$

The proof of Lemma 3.3 is deferred to Appendix A.2. With Lemma 3.3, we can prove the convergence of $L(\boldsymbol{x}_t)$ for learning rate schedules with certain conditions. The proof is also deferred to Appendix A.2.

Theorem 3.4. *Assume that* $\eta_t \geq 0$, $\lim_{t\to\infty}\eta_t = 0$ *and* $\sum_{t=1}^{\infty}\eta_t = \infty$. *For any convex loss* L *with* H*-lipschitz gradient,* $\lim_{t\to\infty} L(\boldsymbol{x}_t) = L(\boldsymbol{x}^*)$.

We also provide a specific example of learning rates $\{\eta_t\}_{t=1}^{\infty}$ that can achieve $O(\frac{1}{t})$ convergence of $f(\boldsymbol{x}_t)$, which is the same as Frank-Wolfe over convex objectives (Jaggi, 2013) and the proof is standard. For completeness, we provide a proof of Theorem 3.5 in Appendix A.2.

Theorem 3.5. *Define* $B = \max\{\|\boldsymbol{x}_0\|, \frac{1}{\lambda}\}$. *For* NSD-WD *with learning rate schedule* $\eta_t = \frac{2}{\lambda(t+1)}$, *we have*

$$L(\boldsymbol{x}_t) - L(\boldsymbol{x}^*) \leq \frac{2H(1+\lambda B)^2}{(t+2)\lambda^2}$$

for $t \geq 1$.

Note that the descent rate highly depends on the smoothness coefficient H, which is determined by the selected norm. Therefore, we provide a synthetic example that may demonstrate the advantage of ℓ_∞ norm as mentioned in Section 1. For some constant $\boldsymbol{x}^* \in \mathbb{R}^d$, the loss function $g: \mathbb{R}^d \to \mathbb{R}$ is defined as

$$g(\boldsymbol{x}) = \sum_{i=1}^{d} \frac{(\boldsymbol{x}_i - \boldsymbol{x}_i^*)^2}{i^2}. \tag{3}$$

The Hessian matrix $\nabla^2 g$ is a diagonal matrix with diagonal entries $\{\frac{2}{i^2}\}_{i=1}^d$. For ℓ_2 norm, the smoothness coefficient is the largest eigenvalue of Hessian matrix, which is 2. For ℓ_∞ norm, the smoothness coefficient is the sum of the diagonal entries because of the diagonality, which is $\sum_{i=1}^d \frac{2}{i^2}$. It is upper bounded by $\sum_{i=1}^{\infty}\frac{2}{i^2} = \frac{\pi^2}{3}$ for all dimension d. However, if $\boldsymbol{x}^*$ is set to be in the unit ℓ_∞ norm ball, its ℓ_2 norm can be as large as $\sqrt{d}$, which makes the suboptimality bound in Theorem 3.5 for ℓ_2 norm normalized steepest descent with weight decay d times larger than its ℓ_∞ counterpart. We implement steepest descent with ℓ_∞ norm and ℓ_2 norm on a 100-dimension example in Section 5.2 and find that ℓ_∞ norm can indeed work better.

3.2. Non-convex Setting: Convergence to KKT Points

In this subsection, we study the implicit bias of SignGD (or more generally, NSD-WD) when the loss is non-convex. In such case, last-iterate parameter convergence is, in general, difficult to show[1], and thus we turn to study *what parameters* SignGD *and* NSD-WD *can converge to*. Our main results Theorem 3.7 show that such parameters must be the KKT points (see Definition 3.6) of the constrained optimization problems. In particular, if the objective is convex, since the norm ball constraint is always convex for all norm, all KKT points are constrained minimizers.

Definition 3.6 (KKT points)**.** *We say* $\boldsymbol{x}^* \in \mathbb{R}^d$ *is a KKT point of the constrained optimization problem* $\min_{\|\boldsymbol{x}\|\leq\frac{1}{\lambda}} L(\boldsymbol{x})$ *if and only if there exists* s^* *such that* $\boldsymbol{x}^*$ *and* s^* *satisfy the following KKT condition.*

(Stationarity)	$0 \in \nabla L(\boldsymbol{x}^*) + s^*\partial\|\boldsymbol{x}^*\|$.
(Primal feasibility)	$\|\boldsymbol{x}^*\| \leq \frac{1}{\lambda}$.
(Dual feasibility)	$s^* \geq 0$.
(Complementary slackness)	$s^*(\|\boldsymbol{x}^*\| - \frac{1}{\lambda}) = 0$.

For convex L, all KKT points $\boldsymbol{x}^*$ are optimal and the dual variable $s^* \geq 0$ is the certificate for the optimality. To see that, for any other $\|\boldsymbol{y}\| \leq \frac{1}{\lambda}$, it holds that

$$L(\boldsymbol{y}) \geq L(\boldsymbol{y}) + s^*(\|\boldsymbol{y}\| - \frac{1}{\lambda}) \geq L(\boldsymbol{x}^*) + s^*(\|\boldsymbol{x}^*\| - \frac{1}{\lambda}),$$

where the second inequality is because $L(\boldsymbol{x}) + s^*\|\boldsymbol{x}\|$ is also convex and 0 is its subgradient at $\boldsymbol{x}^*$. Thus we conclude

$$L(\boldsymbol{y}) \geq L(\boldsymbol{x}^*) + s^*(\|\boldsymbol{x}^*\| - \frac{1}{\lambda}) = L(\boldsymbol{x}^*).$$

Now we state the main result for this subsection.

Theorem 3.7 (Non-convex, KKT)**.** *For any continuously differentiable function* $L: \mathbb{R}^d \to \mathbb{R}$, *initialization* $\boldsymbol{x}_0$, *and learning rate schedule* $\{\eta_t\}_{t=1}^{\infty}$ *such that* $\sum_{t=1}^{\infty}\eta_t = \infty$, *if the iterates of* NSD-WD $\{\boldsymbol{x}_t\}_{t=0}^{\infty}$ *on* L *converges to some* $\boldsymbol{x}_\infty$, *then* $\boldsymbol{x}_\infty$ *is a KKT point (Definition 3.6) of the constrained optimization problem* $\min_{\|\boldsymbol{x}\|\leq\frac{1}{\lambda}} L(\boldsymbol{x})$.

To prove Theorem 3.7, we use the following alternative characterization for KKT points of $\min_{\|\boldsymbol{x}\|\leq\frac{1}{\lambda}} L(\boldsymbol{x})$ below based on Lemma 2.1.

Lemma 3.8. $\boldsymbol{x}$ *is a KKT point of* $\min_{\|\boldsymbol{x}\|\leq\frac{1}{\lambda}} L(\boldsymbol{x})$ *iff* $\|\boldsymbol{x}\| \leq \frac{1}{\lambda}$ *and* $\langle -\lambda\boldsymbol{x}, \nabla L(\boldsymbol{x})\rangle = \|\nabla L(\boldsymbol{x})\|_*$.

With Lemma 3.8, next we illustrate the intuition for Theorem 3.7 for the case where the dual norm $\|\cdot\|_*$ is continuously differentiable at $\nabla L(\boldsymbol{x}_\infty)$, *e.g.*, ℓ_2 norm and when $\nabla L(\boldsymbol{x}_\infty) \neq \mathbf{0}$. For sufficiently large t, when $\nabla L(\boldsymbol{x}_t)$ gets sufficiently close to $\nabla L(\boldsymbol{x}_\infty)$, the descent direction $-\boldsymbol{\Delta}_t$

[1] Indeed, even for convex case, Frank-Wolfe may not converge in parameter.(Bolte et al., 2023)

is unique, equal to $-\nabla \|\nabla L(\boldsymbol{x}_t)\|_*$ by Lemma 2.2, and satisfies $\langle \boldsymbol{\Delta}_t, \nabla L(\boldsymbol{x}_t)\rangle = \|\nabla L(\boldsymbol{x}_t)\|_*$.

Taking $t \to \infty$ we get

$$\langle -\nabla \|\nabla L(\boldsymbol{x}_\infty)\|_*, \nabla L(\boldsymbol{x}_\infty)\rangle = \|\nabla L(\boldsymbol{x}_\infty)\|_*.$$

Moreover, we must have

$$\nabla \|\nabla L(\boldsymbol{x}_\infty)\|_* + \lambda \boldsymbol{x}_\infty = \lim_{t\to\infty} (\nabla \|\nabla L(\boldsymbol{x}_t)\|_* + \lambda \boldsymbol{x}_t) = 0,$$

otherwise $\boldsymbol{x}_t$ keeps moving towards $\nabla \|\nabla L(\boldsymbol{x}_\infty)\|_* + \lambda \boldsymbol{x}_\infty$ and thus $\boldsymbol{x}_\infty$ cannot be the limit, since $\sum_{t=1}^\infty \eta_t = \infty$. This implies the second condition in Lemma 3.8. The first condition that $\|\boldsymbol{x}_\infty\| \le \frac{1}{\lambda}$ is immediate from Lemma 3.1 and that $\sum_{t=1}^\infty \eta_t = \infty$.

However, the above intuition no longer works when dual norm $\|\cdot\|_*$ is not differentiable at $\nabla L(\boldsymbol{x}_\infty)$. This could happen for ℓ_∞ norm where the dual norm is ℓ_1 norm and $\nabla L(\boldsymbol{x}_\infty)$ with coordinates of value 0, because the subgradient of absolute value function at 0, $\partial|0|$, could be anything between -1 and 1. And more generally, this could happen for any norm and $\nabla L(\boldsymbol{x}_\infty) = \mathbf{0}$. If the limit point $\boldsymbol{x}_\infty$ has zero gradient for L, then the steepest descent direction $-\boldsymbol{\Delta}$ is provably not continuous around $\boldsymbol{x}_\infty$.

The following lemma (Lemma 3.9) circumvents the above issue by considering the weighted average of past steepest descent directions, which provably converges, given the iterates $\{\boldsymbol{x}_t\}_{t=1}^\infty$ converge. Theorem 3.7 is a direct combination of Lemma 3.9 and Lemma 3.8 and we omit its proof. The proof of Lemma 3.9 is deferred into Appendix A.3.

Lemma 3.9. *For any learning rate schedule $\{\eta_t\}_{t=1}^\infty$ satisfying $\sum_{t=1}^\infty \eta_t = \infty$, if the iterates of* `NSD-WD` *$\{\boldsymbol{x}_t\}_{t=0}^\infty$ converges to some $\boldsymbol{x}_\infty$, we have that*

1. $\boldsymbol{\Delta}_\infty := \lim_{T\to\infty} \frac{\sum_{t=1}^T \eta_t \boldsymbol{\Delta}_t}{\sum_{t=1}^T \eta_t}$ *exists and* $\boldsymbol{\Delta}_\infty = -\lambda \boldsymbol{x}_\infty$.
2. $\langle \nabla L(\boldsymbol{x}_\infty), \boldsymbol{\Delta}_\infty\rangle = \|\nabla L(\boldsymbol{x}_\infty)\|_*$.
3. $\|\boldsymbol{\Delta}_\infty\| \le 1$.

4. Implicit Bias of AdamW

In this section, we extend the analysis on `NSD-WD` in Section 3 to `AdamW` to prove that the converged parameters of `AdamW` is the KKT point of the constrained optimization problem. The proof relies on an upper bound of average update size of `AdamW` and we find that the bound can also be used to guide hyperparameter tuning in empirical study.

We first state the analog of Lemma 3.9 for `AdamW` with the norm being ℓ_∞ norm since we treat `AdamW` as a smoothed version of `SignGD`, which is Lemma 4.1. Here we additionally assume that $\{\eta_t\}_{t=1}^\infty$ is non-increasing and $\boldsymbol{\Delta}_t$ is defined as $\frac{\boldsymbol{m}_t}{\sqrt{\boldsymbol{v}_t}}$ from Algorithm 1. Theorem 1.1 is again a direct combination of Lemma 3.8 and Lemma 4.1.

Lemma 4.1. *For non-increasing learning rate schedule $\{\eta_t\}_{t=0}^\infty$ satisfying $\sum_{t=1}^\infty \eta_t = \infty$ and $\beta_2 \ge \beta_1$, we get $\{\boldsymbol{x}_t\}_{t=1}^\infty$ by running AdamW with weight decay factor λ. If $\{\boldsymbol{x}_t\}_{t=0}^\infty$ converges to some $\boldsymbol{x}_\infty$, then it holds that*

1. $\boldsymbol{\Delta}_\infty \triangleq \lim_{T\to\infty} \frac{\sum_{t=1}^T \eta_t \boldsymbol{\Delta}_t}{\sum_{t=1}^T \eta_t}$ *exists and* $\boldsymbol{\Delta}_\infty = -\lambda \boldsymbol{x}_\infty$.
2. $\langle \nabla L(\boldsymbol{x}_\infty), \boldsymbol{\Delta}_\infty\rangle = \|\nabla L(\boldsymbol{x}_\infty)\|_1$.
3. $\|\boldsymbol{\Delta}_\infty\|_\infty \le 1$.

The condition $\beta_1 \le \beta_2$ is necessary for the conclusion to hold. Otherwise, the iterates can converge outside the ℓ_∞ norm ball with radius $\frac{1}{\lambda}$ as shown in Appendix B.3.

The first two properties in Lemma 4.1 are straightforward, and the main technical difficulty here lies in the proof of the third property. This is because for any single t, $\|\boldsymbol{\Delta}_t\|$ could be larger than 1, which is different from the case of `NSD-WD`. To prove the third property, we need a tight upper bound for the average update size of `Adam`-like update rule, which is Lemma 4.2. The proof of Lemma 4.1 is deferred to Appendix B.

4.1. Upper Bound for Average Update Size of `Adam`

As mentioned earlier, `Adam` updates $\left\|\frac{\boldsymbol{m}_t}{\sqrt{\boldsymbol{v}_t}}\right\|$ can easily go beyond 1 and thus we prove the following upper bound for the average update size of `Adam` (Lemma 4.2). The proof of Lemma 4.2 is deferred to Appendix B.1.

Lemma 4.2. *Given any $\beta_1 \le \beta_2 < 1$, suppose scalar sequences $\{v_t\}_{t=0}^\infty$ and $\{g_t\}_{t=1}^\infty$ satisfy that $v_0 \ge 0, v_1 > 0$ and $v_t - \beta_2 v_{t-1} \ge (1-\beta_2) g_t^2$ for $t \ge 1$. Given initial value $|m_0| \le \sqrt{v_0}$, define $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$ and $\Delta_t = \frac{m_t}{\sqrt{v_t}}$ for $t \ge 1$. For any coefficients $\{\eta_t\}_{t=1}^\infty$ and $T \in \mathbb{N}$, it always holds that*

$$\left(\frac{|\sum_{t=1}^T \eta_t \Delta_t|}{\sum_{t=1}^T \eta_t}\right)^2 \le 1 + \frac{\beta_2-\beta_1}{1-\beta_2} \frac{\sum_{t=1}^T \eta_t \beta_1^{t-1}}{\sum_{t=1}^T \eta_t} + \frac{(\beta_2-\beta_1)(1-\beta_1)}{(1-\beta_2)\sum_{t=1}^T \eta_t} \sum_{t=2}^T \alpha_t \ln \frac{v_t}{v_1} \tag{4}$$

where $\alpha_t = \eta_t \frac{1-\beta_1^{t-1}}{1-\beta_1} - \sum_{i=1}^{T-t} \eta_{t+i} \beta_1^{i-1}$. In particular, when $\beta_1 = \beta_2$, it even holds that $|\Delta_t| \le 1$.

Note $\{v_t\}_{t=0}^\infty$ here only needs to satisfy a more general condition rather than to be the exact moving average of g_t^2. It can be applied to the practical scenario where a small

positive constant ϵ is added to $\sqrt{\boldsymbol{v}_t}$ in the denominator to improve the numerical stability of Adam. It is easy to verify that for $\boldsymbol{v}_t$ in Algorithm 1, we have that

$$\begin{aligned}&(\sqrt{\boldsymbol{v}_t}+\epsilon)^2-\beta_2(\sqrt{\boldsymbol{v}_{t-1}}+\epsilon)^2\\ \geq&(\boldsymbol{v}_t-\beta_2\boldsymbol{v}_{t-1})+2\epsilon(\sqrt{\boldsymbol{v}_t}-\beta_2\sqrt{\boldsymbol{v}_{t-1}})\\ =&(1-\beta_2)\boldsymbol{g}_t^2+2\epsilon(\sqrt{\beta_2^2\boldsymbol{v}_{t-1}+(1-\beta_2)\boldsymbol{g}_t^2}-\sqrt{\beta_2^2\boldsymbol{v}_{t-1}})\\ \geq&(1-\beta_2)\boldsymbol{g}_t^2.\end{aligned}$$

Therefore, for Adam with ϵ, v_t in Equation 4 is always lower bounded, and if we further have an upper bound for gradients, then we can easily control the average update size of Adam. One nice property is that the upper bound only scales up logarithmically to $1/\epsilon$, instead of linearly, as the naive upper bound scales.

Relationship of ℓ_∞ norm and hyperparameters Another application of Lemma 4.2 is to provide a tight upper bound for the norm of iterates for any setting, *e.g.*, before convergence or even when the gradient is stochastic. In particular, when the learning rate does not change over steps, we have the following upper bound whose proof is in Appendix B.4.

Lemma 4.3. *For any coordinate $j\in[d]$, for* AdamW *with constant learning rate η and weight decay factor λ, with $C\triangleq\max_{1\leq t\leq T}\left|\ln\frac{\boldsymbol{v}_{t,j}}{\boldsymbol{v}_{1,j}}\right|$, it holds that*

$$\lambda\left|\boldsymbol{x}_{T,j}\right|-1\leq(1-\lambda\eta)^T\lambda\left|\boldsymbol{x}_{0,j}\right|+\frac{\lambda\eta(\beta_2-\beta_1)\left[2C+\beta_1^T+(1-\lambda\eta)^T\right]}{2(1-\beta_2)|1-\lambda\eta-\beta_1|}.$$

When $\beta_1=\beta_2$, we only need $T=\Omega\left(\frac{\log\|\boldsymbol{x}_0\|_\infty}{\lambda\eta}\right)$ to guarantee that $|\boldsymbol{x}_T,j|$ is no larger than $\frac{1}{\lambda}$ for any $\lambda\eta\leq1$. However, when $\beta_1<\beta_2$ and $\beta_1<1-\lambda\eta$, the dominating term on the right-hand side is $C\cdot\frac{\eta\lambda(\beta_2-\beta_1)}{(1-\beta_2)(1-\eta\lambda-\beta_1)}$. Assuming $C=O(1)$, it also requires $\lambda\eta\ll1-\beta_2<1-\beta_1$ or $\lambda\eta<1-\beta_2\approx1-\beta_1$ to ensure the remaining term is small.

5. Experiments

In this section, we run experiments to verify the theoretical claims. In Section 5.1, we show that the ℓ_∞ norm of iterates by AdamW can converge below $\frac{1}{\lambda}$ as shown in Theorem 1.1 even when the function is non-convex. In Section 5.2, we show that steepest descent w.r.t. ℓ_∞ norm works better than w.r.t. ℓ_2 norm for a specific function, which has better properties under ℓ_∞ geometry.

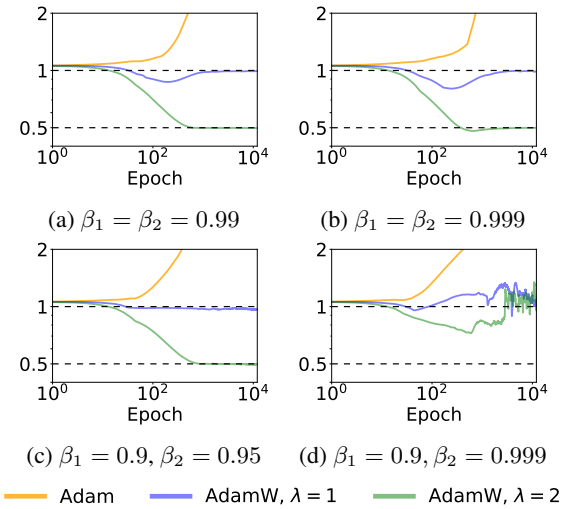


Figure 1: The ℓ_∞ norm of parameters during the training process of language modeling task on PTB. The complete results for Adam are in Figure 6. As predicted by Lemma 4.3, ℓ_∞ norm can be bounded by $\frac{1}{\lambda}$ when $\beta_1=\beta_2$ or $\lambda\eta\ll1-\beta_2<1-\beta_1$. However, for the default setting $\beta_1=0.9$ and $\beta_2=0.999$, the ℓ_∞ norm of AdamW may not be bounded by $\frac{1}{\lambda}$ because $1-\beta_2<\lambda\eta<1-\beta_1$.

5.1. Language Modeling Task on PTB

We train a small two-layer transformer for language modeling task on the Penn Treebank dataset (PTB) (Marcus et al., 1993) based on the implementation provided by Kunstner et al. (2022). We train the model in full batch without dropout in order to get deterministic gradients and follow the constant learning rate setting for the total 12800 epochs. The learning rate η is $\sqrt{10}\times10^{-3}$[2]. For each setting of β_1, β_2, we use Adam and AdamW with weight decay coefficient $\lambda=1,2$ to compare the ℓ_∞ norm for iterates in each optimizer. We employ the standard implementation in PyTorch but set ϵ to be 10^{-16} in Adam and AdamW rather than 0 to avoid division by zero error because the gradient of some coordinates is 0 in the first iteration. Each run is repeated for 4 random seeds to show the robustness of our claim. More details can be found in Appendix C.

From the discussion in Lemma 4.3, it requires either $\beta_1\approx\beta_2$ or $\lambda\eta\ll1-\beta_2$ when $\beta_1<\beta_2$ in order for $\|\boldsymbol{x}_T\|_\infty$ to be bounded by $\frac{1}{\lambda}$. To verify the first case, we employ the two hyperparamter settings where $\beta_1=\beta_2=0.99$ and $\beta_1=\beta_2=0.999$. To verify the second case, we employ the hyperparameter setting where $\beta_1=0.9$ and $\beta_2=0.95$. The ℓ_∞ norm of iterates for one random seed are shown in

[2]We follow the tuning process in Kunstner et al. (2022) for AdamW and $\sqrt{10}\times10^{-3}$ achieves the best training performance. $\sqrt{10}\times10^{-3}$ also achieves the best training performance for full-batch Adam in Kunstner et al. (2022).

Figures 1a to 1c. In order to show the details around 0.5 and 1, we truncate the range of y-axis and the full result is plotted in Figure 6 in Appendix C. In all these three settings, the ℓ_∞ norm of iterates in `Adam` keeps increasing while the ℓ_∞ norm of iterates in `AdamW` is constrained below 1 and $\frac{1}{2}$ for $\lambda = 1$ and 2 respectively. The results for another three random seeds show similar pattern and are plotted in Figures 3 to 5 and Figures 7 to 9.

We also show that the condition is necessary for empirical study by training with default $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Now $1 - \beta_2 < \lambda\eta$ and $\beta_1 \neq \beta_2$, which breaks the condition. The ℓ_∞ norm of iterates are shown in Figure 1d. The ℓ_∞ norm of `AdamW` can not be constrained by 1 and $\frac{1}{2}$ for $\lambda = 1$ and 2.

5.2. Synthetic Problem

As mentioned in Section 3.1, we construct the loss function Equation 3 for some $\boldsymbol{x}^* \in \mathbb{R}^{100}$. The first 10 coordinates of $\boldsymbol{x}^*$ is 1 while the rest 90 coordinates are uniformly sampled between $[-1, 1]$. With such initialization, the ℓ_∞ norm of $\boldsymbol{x}^*$ is always upper bounded by 1 but the ℓ_2 norm can be as large as 10.

We want to verify that an optimization algorithm can take advantage when it employs the norm that is more suitable for the loss function. So we implement normalized steepest descent and steepest descent with ℓ_∞ norm and ℓ_2 norm. In order to test the effect of weight decay, we also implement normalized steepest descent with weight decay for both norms. In order to be able to reach the global minimizer, the weight decay factor λ is set to be the lower bound of $\frac{1}{\|\boldsymbol{x}^*\|}$, which is 1 for ℓ_∞ norm and 0.1 for ℓ_2 norm. The learning rate can be set according to theoretical results. The learning rate for steepest descent is constant $\frac{1}{H}$ in which H is the smoothness coefficient. By Theorem 3.5, the learning rate for normalized steepest descent with weight decay factor λ should be set as $\eta_t = \frac{2}{\lambda(t+1)}$. We also use the same learning rate schedule for normalized steepest descent without weight decay.

All the algorithms receive the same initialization $\boldsymbol{x}_0$, whose coordinate is uniformly initialized from $[-5, 5]$. The results for the first 100 iterations is shown in Figure 2. The steepest descent w.r.t. ℓ_∞ norm always performs better than the steepest descent w.r.t. ℓ_2 norm no matter whether the update is normalized or not. For both norms, the performance of the normalized steepest descent is improved when weight decay is activated.

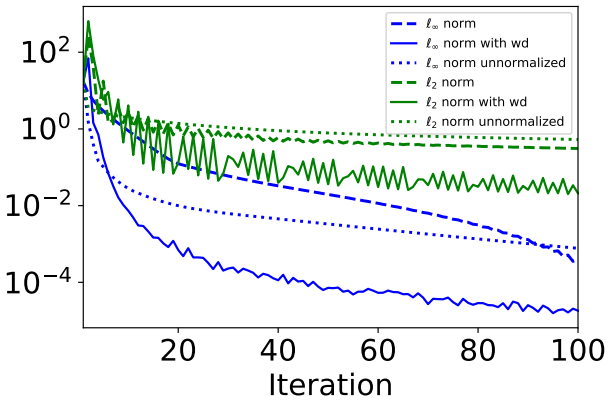


Figure 2: For both ℓ_2 and ℓ_∞ norm, we plot training loss of normalized steepest descent w. and w.o. weight decay and unnormalized steepest descent over the quadratic loss $g(\boldsymbol{x}) = \sum_{i=1}^{100} \frac{(\boldsymbol{x}_i - \boldsymbol{x}_i^*)^2}{i^2}$. When weight decay is turned on, it is set as $\frac{1}{\|\boldsymbol{x}^*\|}$ to preserve the optimal value even with the norm constraints Theorem 3.4. We find that **ℓ_∞ norm always outperforms ℓ_2 norm** regardless of the usage of weight decay and irrespective of whether the steepest descent method is normalized. The usage of **weight decay accelerates the optimization** for both ℓ_∞ norm and ℓ_2 norm.

6. Related Work

Adaptive Methods: While stochastic gradient descent (Robbins & Monro, 1951) remains popular for optimizing deep learning models like ResNet (He et al., 2016), only adaptive methods can efficiently train recently-emerged large language models (Zhang et al., 2020). There has been a fruitful amount of research on adaptive gradient method, including AdaGrad (Duchi et al., 2011), RMSProp (Tieleman & Hinton, 2012), AdaDelta (Zeiler, 2012), Adam (Kingma & Ba, 2014), AdaFactor (Shazeer & Stern, 2018), AMSGrad (Reddi et al., 2018), AdaBound (Luo et al., 2018), Lion (Chen et al., 2024), *etc*. Recently there have been also adaptive methods attempting to accelerate by leveraging the second-order information, *e.g.*, AdaHessian (Yao et al., 2021) and Sophia (Liu et al., 2023). However, most algorithms that are able to train large language models adopt coordinate-wise adaptivity. In contrast, stochastic gradient descent, even equipped with global gradient norm clipping, cannot match the performance of coordinate-wise adaptive algorithms on language tasks (Li et al., 2022a). Previous work has given convergence rate for RMSProp and Adam under different assumptions (Chen et al., 2018; Zou et al., 2019; Shi & Li, 2021; Guo et al., 2021; Défossez et al., 2022; Zhang et al., 2022).

Our work shows that `AdamW` and `SignGD` with weight decay converge to the same point assuming convergence. Balles & Hennig (2018); Kunstner et al. (2022) point out that the similarity with `SignGD` largely accounts for the advantage of `Adam` over `SGD`. Moreover, when `SignGD` is equipped with momentum which is one key component of `Adam`, it can achieve comparable empirical results with `Adam` for

various tasks (Balles & Hennig, 2018; Kunstner et al., 2022; Bernstein et al., 2018; Crawshaw et al., 2022).

Role of Weight Decay: The usage of weight decay, which refers to shrinking the parameter by a small constant fraction, can be dated back to the 1980s (Rumelhart et al., 1986; Hinton, 1987). It has been recognized as a standard trick to improve the generalization performance of neural networks (Krogh & Hertz, 1991; Bos & Chug, 1996) for a long time. Krizhevsky et al. (2012) first noticed that weight decay can sometimes accelerate optimization in deep learning. For modern architectures equipped with normalization layers, *e.g.*, BatchNorm (Ioffe & Szegedy, 2015) and LayerNorm (Ba et al., 2016), only the direction of the parameters before normalization layers matters, rather than their norms. Turning on weight decay in such settings changes the effective learning rate of the parameters (Hoffer et al., 2018; Arora et al., 2018; Zhang et al., 2018; Li & Arora, 2019; Li et al., 2020).

Though weight decay is equivalent to ℓ_2 regularization for SGD, for steepest descent methods with general norms and adaptive methods like `Adam`, they lead to different optimization trajectories (Loshchilov & Hutter, 2018; Zhang et al., 2018; Zhuang et al., 2022). The empirical benefit of weight decay over ℓ_2 regularization when they are different is not well-understood in theory.

Implicit Regularization: Our main result Theorem 1.1 shows that `AdamW` regularizes the ℓ_∞ norm of the learned solution implicitly through modifying the optimization dynamics, rather than directly modifying the objective, like `Adam` with ℓ_2 regularization. This kind of behavior is termed *implicit regularization* or *implicit bias* of optimization algorithms. Though there has been a large volume of works studying the implicit bias of (Stochastic) GD and its non-adaptive variants, including settings related to max margin (Soudry et al., 2018; Gunasekar et al., 2018; Nacson et al., 2019a;b; Ji & Telgarsky, 2018; Lyu & Li, 2019), initialization with norm (Gunasekar et al., 2017; Arora et al., 2019a; Li et al., 2019; Lyu et al., 2021), kernel regime (Jacot et al., 2018; Arora et al., 2019b;c), and flatness (Blanc et al., 2020; Damian et al., 2021; Li et al., 2021; Arora et al., 2022; Li et al., 2022b; Wen et al., 2022; Damian et al., 2022), very few results are known about the implicit bias of adaptive methods like `Adam`. Wilson et al. (2017) shows that for linear regression problem, adaptive methods can converge to a solution whose elements have the same magnitude under certain conditions. Their converged solution thus has small ℓ_∞ norm while the converged solution of non-adaptive methods is known to have the smallest ℓ_2 norm among all the global minimizers. Wang et al. (2021) shows that `Adam` behaves similarly to non-adaptive methods like GD when the cross-entropy loss converges to 0 due to the positive numerical stability hyperparameter ϵ in the denominator of `Adam`'s update rule. The theoretical derivation by Cattaneo et al. (2023) argues that `Adam` tends to find interpolating solutions with small ℓ_1 norm.

The concurrent work by Chen et al. (2023) is arguably the most related work to us, where the recently discovered optimization algorithm by auto-search, Lion (Chen et al., 2024), is elegantly generalized to a family of algorithms, Lion-$\mathcal{K}$, where $\mathcal{K}$ is some convex function. When $\mathcal{K}$ is chosen to be the dual norm and momentum in Lion-$\mathcal{K}$ is turned off, Lion-$\mathcal{K}$ becomes the normalized steepest descent. Their analysis shows that even with momentum, the steepest normalized descent with weight decay can be viewed as optimization under the original norm constraint. However, in any Lion-$\mathcal{K}$ algorithm, the update at one step t only depends on past iterates through first-order momentum $\boldsymbol{m}_t$. Their analysis cannot be applied to `AdamW` because `AdamW` cannot be written in the form of Lion-$\mathcal{K}$ for any convex function $\mathcal{K}$. To see this, simply note that the update of Lion-$\mathcal{K}$ for a fixed $\mathcal{K}$ is completely determined by $\boldsymbol{g}_t, \boldsymbol{m}_t$ and $\boldsymbol{x}_t$ while the update of `AdamW` can still be different if the second order momentum $\boldsymbol{v}_t$ is different. In terms of proof technique, Chen et al. (2023) constructs the Lyapunov function while we directly characterize the KKT point and connect the converged point to KKT point through the weighted average update.

7. Discussion and Future Works

This work focuses on the implicit bias of `AdamW` in the deterministic (or full-batch) case. Though our upper bound on the average update size of `Adam` holds unconditionally on the input gradients, regardless of stochasticity or not, it is unlikely that the $\frac{1}{\lambda}$ upper bound can be reached when there is large gradient noise, especially when β_2 is very close to 1. In that case, the denominator of the update of `AdamW` is roughly the square root of the square of the expected gradient plus some additional gradient variance term, which strictly dominates the expected gradient in the numerator. Malladi et al. (2022) uses Stochastic Differential Equation (SDE) approximation to model the trajectories of `Adam` in such regime and empirically tests the implication of SDE approximation, namely the square root scaling rule.

The most interesting future direction is to understand in what sense the optimization advantage of coordinate-wise adaptive methods like `Adam` over standard SGD for language modeling tasks can be explained by the conjecture implied by the findings of Kunstner et al. (2022), that *the loss function for language modeling tasks has better properties under ℓ_∞ geometry*. It would be interesting to understand if the loss landscape in real-world applications shares common properties with our toy quadratic example and induces similar results in Figure 2.

Another important future direction is to provide non-asymptotic convergence rates for AdamW in both convex and non-convex settings.

8. Conclusions

We make the first step towards understanding the benefit of AdamW over Adam with ℓ_2 regularization by characterizing the implicit bias of AdamW, *i.e.*, it can only converge to KKT points of the ℓ_∞ norm constrained optimization problem. There are two main insights behind this result: (1)Adam is a smoothed version of SignGD, which is the normalized steepest descent w.r.t. ℓ_∞ norm; (2). for any norm, the corresponding normalized steepest descent with weight decay is essentially Frank-Wolfe over the corresponding norm ball, which is known to perform constrained optimization. Our main technical contribution is a tight upper bound of the average update size of Adam updates. We test its prediction on the relationship between the ℓ_∞ norm of the parameters and the AdamW hyperparameters η, λ, β_1, β_2 on a language modeling task.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgement

We thank Xinran Gu for pointing us to important related work and anonymous reviewers for their valuable feedback.

References

Arora, S., Li, Z., and Lyu, K. Theoretical Analysis of Auto Rate-Tuning by Batch Normalization. In *International Conference on Learning Representations*, 2018. 8

Arora, S., Cohen, N., Hu, W., and Luo, Y. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems*, 2019a. 8

Arora, S., Du, S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*. PMLR, 2019b. 8

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019c. 8

Arora, S., Li, Z., and Panigrahi, A. Understanding gradient descent on the edge of stability in deep learning. In *International Conference on Machine Learning*. PMLR, 2022. 8

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 8

Balles, L. and Hennig, P. Dissecting adam: The sign, magnitude and variance of stochastic gradients. In *International Conference on Machine Learning*, 2018. 7, 8

Bernstein, J., Wang, Y.-X., Azizzadenesheli, K., and Anandkumar, A. signSGD: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, 2018. 8

Blanc, G., Gupta, N., Valiant, G., and Valiant, P. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. In *Conference on learning theory*. PMLR, 2020. 8

Bolte, J., Combettes, C. W., and Pauwels, E. The iterates of the Frank–Wolfe algorithm may not converge. *Mathematics of Operations Research*, 2023. 4

Bos, S. and Chug, E. Using weight decay to optimize the generalization ability of a perceptron. In *Proceedings of International Conference on Neural Networks (ICNN'96)*. IEEE, 1996. 8

Cattaneo, M. D., Klusowski, J. M., and Shigida, B. On the implicit bias of adam. *arXiv preprint arXiv:2309.00079*, 2023. 8

Chen, L., Liu, B., Liang, K., et al. Lion Secretly Solves a Constrained Optimization: As Lyapunov Predicts. In *The Twelfth International Conference on Learning Representations*, 2023. 3, 8

Chen, X., Liu, S., Sun, R., and Hong, M. On the Convergence of A Class of Adam-Type Algorithms for Non-Convex Optimization. In *International Conference on Learning Representations*, 2018. 7

Chen, X., Liang, C., Huang, D., Real, E., Wang, K., Pham, H., Dong, X., Luong, T., Hsieh, C.-J., Lu, Y., et al. Symbolic discovery of optimization algorithms. *Advances in Neural Information Processing Systems*, 2024. 7, 8

Crawshaw, M., Liu, M., Orabona, F., Zhang, W., and Zhuang, Z. Robustness to Unbounded Smoothness of Generalized SignSGD. In *Advances in Neural Information Processing Systems*, 2022. 8

Damian, A., Ma, T., and Lee, J. D. Label Noise SGD Provably Prefers Flat Global Minimizers. In *Advances in Neural Information Processing Systems*, 2021. 8

Damian, A., Nichani, E., and Lee, J. D. Self-Stabilization: The Implicit Bias of Gradient Descent at the Edge of Stability. In *The Eleventh International Conference on Learning Representations*, 2022. 8

Défossez, A., Bottou, L., Bach, F., and Usunier, N. A Simple Convergence Proof of Adam and Adagrad. *Transactions on Machine Learning Research*, 2022. 7

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 2011. 7

Frank, M., Wolfe, P., et al. An algorithm for quadratic programming. *Naval research logistics quarterly*, 1956. 3

Gunasekar, S., Woodworth, B. E., Bhojanapalli, S., Neyshabur, B., and Srebro, N. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*, 2017. 8

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*. PMLR, 2018. 8

Guo, Z., Xu, Y., Yin, W., Jin, R., and Yang, T. A novel convergence analysis for algorithms of the adam family. *arXiv preprint arXiv:2112.03459*, 2021. 7

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 7

Hinton, G. E. Learning translation invariant recognition in a massively parallel networks. In *International conference on parallel architectures and languages Europe*, 1987. 8

Hoffer, E., Banner, R., Golan, I., and Soudry, D. Norm matters: efficient and accurate normalization schemes in deep networks. In *Advances in Neural Information Processing Systems*, 2018. 8

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 2015. 8

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, 2018. 8

Jaggi, M. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*, 2013. 3, 4

Ji, Z. and Telgarsky, M. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2018. 8

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1, 7

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012. 8

Krogh, A. and Hertz, J. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 1991. 8

Kunstner, F., Chen, J., Lavington, J. W., and Schmidt, M. 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*, 2022. 1, 2, 6, 7, 8, 21

Li, Y., Wei, C., and Ma, T. Towards explaining the regularization effect of initial large learning rate in training neural networks. In *Advances in Neural Information Processing Systems*, 2019. 8

Li, Z. and Arora, S. An Exponential Learning Rate Schedule for Deep Learning. In *International Conference on Learning Representations*, 2019. 8

Li, Z., Lyu, K., and Arora, S. Reconciling modern deep learning with traditional optimization analyses: The intrinsic learning rate. *Advances in Neural Information Processing Systems*, 2020. 8

Li, Z., Wang, T., and Arora, S. What Happens after SGD Reaches Zero Loss?–A Mathematical Framework. In *International Conference on Learning Representations*, 2021. 8

Li, Z., Bhojanapalli, S., Zaheer, M., Reddi, S., and Kumar, S. Robust training of neural networks using scale invariant architectures. In *International Conference on Machine Learning*, 2022a. 7

Li, Z., Wang, T., and Yu, D. Fast Mixing of Stochastic Gradient Descent with Normalization and Weight Decay. In *Advances in Neural Information Processing Systems*, 2022b. 8

Liu, H., Li, Z., Hall, D. L. W., Liang, P., and Ma, T. Sophia: A Scalable Stochastic Second-order Optimizer for Language Model Pre-training. In *The Twelfth International Conference on Learning Representations*, 2023. 7

Loshchilov, I. and Hutter, F. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2018. 1, 8

Luo, L., Xiong, Y., Liu, Y., and Sun, X. Adaptive Gradient Methods with Dynamic Bound of Learning Rate. In *International Conference on Learning Representations*, 2018. 7

Lyu, K. and Li, J. Gradient Descent Maximizes the Margin of Homogeneous Neural Networks. In *International Conference on Learning Representations*, 2019. 8

Lyu, K., Li, Z., Wang, R., and Arora, S. Gradient Descent on Two-layer Nets: Margin Maximization and Simplicity Bias. In *Advances in Neural Information Processing Systems*, 2021. 8

Malladi, S., Lyu, K., Panigrahi, A., and Arora, S. On the SDEs and Scaling Rules for Adaptive Gradient Algorithms. In *Advances in Neural Information Processing Systems*, 2022. 8

Marcus, M., Santorini, B., and Marcinkiewicz, M. A. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 1993. 6

Nacson, M. S., Gunasekar, S., Lee, J., Srebro, N., and Soudry, D. Lexicographic and depth-sensitive margins in homogeneous and non-homogeneous deep models. In *International Conference on Machine Learning*, 2019a. 8

Nacson, M. S., Lee, J., Gunasekar, S., Savarese, P. H. P., Srebro, N., and Soudry, D. Convergence of gradient descent on separable data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019b. 8

Reddi, S. J., Kale, S., and Kumar, S. On the Convergence of Adam and Beyond. In *International Conference on Learning Representations*, 2018. 7

Robbins, H. and Monro, S. A stochastic approximation method. *The annals of mathematical statistics*, 1951. 7

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *nature*, 1986. 8

Shazeer, N. and Stern, M. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, 2018. 7

Shi, N. and Li, D. Rmsprop converges with proper hyperparameter. In *International conference on learning representation*, 2021. 7

Soudry, D., Hoffer, E., Nacson, M. S., and Srebro, N. The Implicit Bias of Gradient Descent on Separable Data. In *International Conference on Learning Representations*, 2018. 8

Tieleman, T. and Hinton, G. Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. *University of Toronto, Technical Report*, 2012. 7

Wang, B., Meng, Q., Chen, W., and Liu, T.-Y. The implicit bias for adaptive optimization algorithms on homogeneous neural networks. In *International Conference on Machine Learning*, 2021. 8

Wen, K., Ma, T., and Li, Z. How Sharpness-Aware Minimization Minimizes Sharpness? In *The Eleventh International Conference on Learning Representations*, 2022. 8

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, 2017. 8

Yao, Z., Gholami, A., Shen, S., Mustafa, M., Keutzer, K., and Mahoney, M. Adahessian: An adaptive second order optimizer for machine learning. In *proceedings of the AAAI conference on artificial intelligence*, 2021. 7

Zeiler, M. D. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 7

Zhang, G., Wang, C., Xu, B., and Grosse, R. Three Mechanisms of Weight Decay Regularization. In *International Conference on Learning Representations*, 2018. 8

Zhang, J., Karimireddy, S. P., Veit, A., Kim, S., Reddi, S., Kumar, S., and Sra, S. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 2020. 1, 7

Zhang, Y., Chen, C., Shi, N., Sun, R., and Luo, Z.-Q. Adam Can Converge Without Any Modification On Update Rules. In *Advances in Neural Information Processing Systems*, 2022. 7

Zhuang, Z., Liu, M., Cutkosky, A., and Orabona, F. Understanding adamw through proximal methods and scale-freeness. *Transactions on Machine Learning Research*, 2022. 1, 8

Zou, F., Shen, L., Jie, Z., Zhang, W., and Liu, W. A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2019. 7

Contents

1 Introduction **1**

2 Preliminaries and Notations **2**

3 Warm Up: Implicit Bias of Normalized Steepest Descent w. Weight Decay **3**

3.1 Convex Setting: Constrained Optimization 3

3.2 Non-convex Setting: Convergence to KKT Points 4

4 Implicit Bias of AdamW **5**

4.1 Upper Bound for Average Update Size of `Adam` 5

5 Experiments **6**

5.1 Language Modeling Task on PTB 6

5.2 Synthetic Problem 7

6 Related Work **7**

7 Discussion and Future Works **8**

8 Conclusions **9**

A Omitted Proofs in Section 3 **12**

A.1 Omitted proofs for convergence into norm ball with bounded update 13

A.2 Omitted proofs for convergence to constrained minimizer with proper learning rates 13

A.3 Omitted Proofs for Lemma 3.9 15

B Omitted Proofs in Section 4 **16**

B.1 Omitted proofs for upper bound for average update size of `Adam` 16

B.2 Proof for Lemma 4.1 17

B.3 A counter example when $\beta_1 > \beta_2$ 19

B.4 Proof for upper bound for norm of iterates in `AdamW` 20

C Experimental Details and More Results **21**

A. Omitted Proofs in Section 3

In this section, we provide the omitted proofs in Section 3, which shows the iterates and the converged solution by normalized steepest descent with decoupled weight decay before diving into the analysis on `AdamW`. In Appendix A.1, we prove that the iterates will enter or stay in the norm ball with radius $\frac{1}{\lambda}$ for any normalized update. In Appendix A.2, we prove that the iterates of normalized steepest descent with weight decay will converge to the constrained minimizer of $L(\boldsymbol{x})$ in the same ball with proper learning rates.

A.1. Omitted proofs for convergence into norm ball with bounded update

Proof of Lemma 3.1. We prove by induction that $\|\boldsymbol{x}_t\| \leq \frac{1}{\lambda} + \prod_{i=1}^{t}(1-\lambda\eta_i)\left(\|\boldsymbol{x}_0\| - \frac{1}{\lambda}\right)$.

$$\begin{aligned}\|\boldsymbol{x}_t\| - \frac{1}{\lambda} &= \|(1-\lambda\eta_t)\boldsymbol{x}_{t-1} - \eta_t\boldsymbol{\Delta}_t\| - \frac{1}{\lambda} \leq (1-\lambda\eta_t)\|\boldsymbol{x}_{t-1}\| + \eta_t\|\boldsymbol{\Delta}_t\| - \frac{1}{\lambda} \leq (1-\lambda\eta_t)\|\boldsymbol{x}_{t-1}\| + \eta_t - \frac{1}{\lambda} \\ &= (1-\lambda\eta_t)\left(\|\boldsymbol{x}_{t-1}\| - \frac{1}{\lambda}\right) \leq \prod_{i=1}^{t}(1-\lambda\eta_i)\left(\|\boldsymbol{x}_0\| - \frac{1}{\lambda}\right).\end{aligned}$$

When $\|\boldsymbol{x}_0\| > \frac{1}{\lambda}$, we have that

$$\|\boldsymbol{x}_t\| - \frac{1}{\lambda} \leq \prod_{i=1}^{t}(1-\lambda\eta_i)\left(\|\boldsymbol{x}_0\| - \frac{1}{\lambda}\right) \leq \prod_{i=1}^{t}\exp(-\lambda\eta_i)\left(\|\boldsymbol{x}_0\| - \frac{1}{\lambda}\right) = \exp(-\lambda\sum_{i=1}^{t}\eta_i)\left(\|\boldsymbol{x}_0\| - \frac{1}{\lambda}\right).$$

When $\|\boldsymbol{x}_0\| \leq \frac{1}{\lambda}$, $\|\boldsymbol{x}_t\| - \frac{1}{\lambda} \leq 0$. This completes the proof. □

A.2. Omitted proofs for convergence to constrained minimizer with proper learning rates

Proof of Lemma 3.3. For normalized steepest descent update $\boldsymbol{\Delta}_t$ from Equation 1,

$$\begin{aligned}\nabla L(\boldsymbol{x}_{t-1})^\top(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}) &= -\eta_t\nabla L(\boldsymbol{x}_{t-1})^\top\boldsymbol{\Delta}_t - \lambda\eta_t\nabla L(\boldsymbol{x}_{t-1})^\top\boldsymbol{x}_{t-1} \\ &= -\eta_t\|\nabla L(\boldsymbol{x}_{t-1})\|_* - \lambda\eta_t\nabla L(\boldsymbol{x}_{t-1})^\top(\boldsymbol{x}_{t-1} - \boldsymbol{x}^*) - \lambda\eta_t\nabla L(\boldsymbol{x}_{t-1})^\top\boldsymbol{x}^* \\ &\leq -\eta_t\|\nabla L(\boldsymbol{x}_{t-1})\|_* - \lambda\eta_t\left(L(\boldsymbol{x}_{t-1}) - L(\boldsymbol{x}^*)\right) + \lambda\eta_t\|\nabla L(\boldsymbol{x}_{t-1})\|_*\|\boldsymbol{x}^*\| \\ &\leq -\lambda\eta_t\left(L(\boldsymbol{x}_{t-1}) - L(\boldsymbol{x}^*)\right),\end{aligned}$$

where the first inequality we use convexity of L and the second inequality uses $\|\boldsymbol{x}^*\| \leq 1$.

Since the gradient of L is H-lipschitz, by Taylor expansion, we have that

$$\begin{aligned}L(\boldsymbol{x}_t) - L(\boldsymbol{x}_{t-1}) &= \int_0^1 \nabla L\left(\boldsymbol{x}_{t-1} + \alpha(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})\right)^\top(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})\,d\alpha \\ &= \nabla L(\boldsymbol{x}_{t-1})^\top(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}) + \int_0^1 \left(\nabla L\left(\boldsymbol{x}_{t-1} + \alpha(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})\right) - \nabla L(\boldsymbol{x}_{t-1})\right)^\top(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})\,d\alpha \\ &\leq -\lambda\eta_t\left(L(\boldsymbol{x}_{t-1}) - L(\boldsymbol{x}^*)\right) + \int_0^1 \|\nabla L\left(\boldsymbol{x}_{t-1} + \alpha(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})\right) - \nabla L(\boldsymbol{x}_{t-1})\|_*\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|\,d\alpha \\ &\leq -\lambda\eta_t\left(L(\boldsymbol{x}_{t-1}) - L(\boldsymbol{x}^*)\right) + \int_0^1 H\alpha\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|^2\,d\alpha \\ &= -\lambda\eta_t\left(L(\boldsymbol{x}_{t-1}) - L(\boldsymbol{x}^*)\right) + \frac{H}{2}\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|^2\end{aligned}$$

Because the update $\boldsymbol{\Delta}_t$ is normalized and thus have unit norm by definition, it holds that

$$\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|^2 = \|-\eta_t\boldsymbol{\Delta}_t - \lambda\eta_t\boldsymbol{x}_{t-1}\|^2 \leq \eta_t^2\left(\|\boldsymbol{\Delta}_t\| + \lambda\|\boldsymbol{x}_{t-1}\|\right)^2 \leq \eta_t^2\left(1 + \lambda\|\boldsymbol{x}_{t-1}\|\right)^2$$

Finally, we conclude that

$$L(\boldsymbol{x}_t) - L(\boldsymbol{x}^*) \leq (1-\lambda\eta_t)(L(\boldsymbol{x}_{t-1}) - L(\boldsymbol{x}^*)) + \frac{H}{2}\eta_t^2\left(1 + \lambda\|\boldsymbol{x}_{t-1}\|\right)^2.$$

□

Proof of Theorem 3.4. The proof of Theorem 3.4 is a direct application of Lemma A.1 on the one-step descent lemma Lemma 3.3. □

Lemma A.1. *Assume that* $\eta_t \geq 0$, $\lim_{t\to\infty} \eta_t = 0$ *and* $\sum_{t=1}^{\infty} \eta_t = \infty$. C *is any positive number and* $a_0 \geq 0$. *If the sequence* $\{a_t\}_{t=0}^{\infty}$ *satisfies that* $a_t \leq (1-\eta_t)a_{t-1} + C\eta_t^2$, *then* $\lim_{t\to\infty} a_t = 0$.

Proof of Lemma A.1. First we show by induction that $a_t \leq a_0 \exp\left(-\sum_{i=1}^{t} \eta_i\right) + C\sum_{i=1}^{t} \eta_i^2 \exp\left(-\sum_{j=i+1}^{t} \eta_j\right)$. Suppose the inequality holds for $t-1$, then we have that

$$
\begin{aligned}
a_t &\leq (1-\eta_t)a_{t-1} + C\eta_t^2 \leq \exp(-\eta_t)\, a_{t-1} + C\eta_t^2 \\
&\leq \exp(-\eta_t)\left[a_0 \exp\left(-\sum_{i=1}^{t-1} \eta_i\right) + C\sum_{i=1}^{t-1} \eta_i^2 \exp\left(-\sum_{j=i+1}^{t-1} \eta_j\right)\right] + C\eta_t^2 \\
&= a_0 \exp\left(-\sum_{i=1}^{t} \eta_i\right) + C\sum_{i=1}^{t-1} \eta_i^2 \exp\left(-\sum_{j=i+1}^{t} \eta_j\right) + C\eta_t^2 \\
&= a_0 \exp\left(-\sum_{i=1}^{t} \eta_i\right) + C\sum_{i=1}^{t} \eta_i^2 \exp\left(-\sum_{j=i+1}^{t} \eta_j\right).
\end{aligned}
$$

Because $\sum_{t=1}^{\infty} \eta_t = \infty$, $\lim_{t\to\infty} a_0 \exp\left(-\sum_{i=1}^{t} \eta_i\right) = 0$. In order to show $\lim_{t\to\infty} a_t = 0$, it's sufficient to show $\lim_{t\to\infty} \sum_{i=1}^{t} \eta_i^2 \exp\left(-\sum_{j=i+1}^{t} \eta_j\right) = 0$.

For any $\epsilon > 0$, η^* is chosen such that $\eta^* e^{\eta^*} = \frac{\epsilon}{2}$. There exists $\tau \in \mathbb{N}^+$ such that $\eta_i \leq \eta^*$ for $i \geq \tau$. We choose T such that $\exp(-\sum_{j=\tau}^{T} \eta_j) \leq \frac{\epsilon}{2\sum_{i=1}^{\tau-1} \eta_i^2}$. Then for any $t \geq T$, we have that

$$
\sum_{i=1}^{\tau-1} \eta_i^2 \exp\left(-\sum_{j=i+1}^{t} \eta_j\right) \leq \sum_{i=0}^{\tau-1} \eta_i^2 \exp\left(-\sum_{j=\tau}^{t} \eta_j\right) \leq \left(\sum_{i=1}^{\tau-1} \eta_i^2\right) \exp\left(-\sum_{j=\tau}^{T} \eta_j\right) \leq \frac{\epsilon}{2}
$$

and

$$
\begin{aligned}
\sum_{i=\tau}^{t} \eta_i^2 \exp\left(-\sum_{j=i+1}^{t} \eta_j\right) &\leq \eta^* \sum_{i=\tau}^{t} \eta_i \exp\left(-\sum_{j=i+1}^{t} \eta_j\right) \\
&\leq \eta^* \sum_{i=\tau}^{t} (\exp(\eta_i) - 1) \exp\left(-\sum_{j=i+1}^{t} \eta_j\right) \\
&= \eta^* \sum_{i=\tau}^{t} \exp(\eta_i)\,(1 - \exp(-\eta_i)) \exp\left(-\sum_{j=i+1}^{t} \eta_j\right) \\
&\leq \eta^* \sum_{i=\tau}^{t-1} \exp(\eta^*)\,(1 - \exp(-\eta_i)) \exp\left(-\sum_{j=i+1}^{t} \eta_j\right) \\
&= \eta^* \exp(\eta^*) \sum_{i=\tau}^{t} \left[\exp\left(-\sum_{j=i+1}^{t} \eta_j\right) - \exp\left(-\sum_{j=i}^{t} \eta_j\right)\right] \\
&= \eta^* \exp(\eta^*) \left[1 - \exp\left(-\sum_{j=\tau}^{t} \eta_j\right)\right] \leq \eta^* \exp(\eta^*) = \frac{\epsilon}{2}.
\end{aligned}
$$

When summing them up, we have that

$$
\sum_{i=1}^{t} \eta_i^2 \exp\left(-\sum_{j=i+1}^{t} \eta_j\right) = \sum_{i=1}^{\tau-1} \eta_i^2 \exp\left(-\sum_{j=i+1}^{t} \eta_j\right) + \sum_{i=\tau}^{t} \eta_i^2 \exp\left(-\sum_{j=i+1}^{t} \eta_j\right) \leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.
$$

□

Proof of Theorem 3.5. From Lemma 3.1, $\|\boldsymbol{x}_t\| \le \max\{\|\boldsymbol{x}_0\|, \frac{1}{\lambda}\} = B$ for $t \ge 0$. Define $C \triangleq \frac{H(1+\lambda B)^2}{2\lambda^2}\frac{4}{(t+1)^2}$. We have that for $t = 1$,

$$L(\boldsymbol{x}_1) - L(\boldsymbol{x}^*) \le (1-1)(L(\boldsymbol{x}_0) - L(\boldsymbol{x}^*)) + \frac{H(1+\lambda B)^2}{2\lambda^2} = C \le \frac{4C}{3}.$$

Suppose $L(\boldsymbol{x}_{t-1}) - L(\boldsymbol{x}^*) \le \frac{4C}{t+1}$. We have that

$$\begin{aligned} L(\boldsymbol{x}_t) - L(\boldsymbol{x}^*) &\le (1 - \frac{2}{t+1})(L(\boldsymbol{x}_{t-1}) - L(\boldsymbol{x}^*)) + \frac{H(1+\lambda B)^2}{2\lambda^2}\frac{4}{(t+1)^2} \\ &\le \frac{t-1}{t+1}\frac{4C}{t+1} + \frac{4C}{(t+1)^2} = \frac{4Ct}{(t+1)^2} \le \frac{4C}{t+2}. \end{aligned}$$

□

A.3. Omitted Proofs for Lemma 3.9

Proof of Lemma 3.9.

1. For any $\epsilon > 0$, there exists t' such that $\|\boldsymbol{x}_t - \boldsymbol{x}_\infty\| \le \frac{\epsilon}{2\lambda}$ for any $t > t'$. Because $\eta_t \boldsymbol{\Delta}_t = \boldsymbol{x}_{t-1} - \boldsymbol{x}_t - \lambda\eta_t \boldsymbol{x}_{t-1}$, we have that

$$\begin{aligned} \frac{\sum_{t=1}^T \eta_t \boldsymbol{\Delta}_t}{\sum_{t=1}^T \eta_t} &= \frac{\boldsymbol{x}_0 - \boldsymbol{x}_T - \lambda \sum_{t=1}^T \eta_t \boldsymbol{x}_{t-1}}{\sum_{t=1}^T \eta_t} \\ &= \frac{\boldsymbol{x}_0 - \boldsymbol{x}_T - \lambda(\sum_{t=1}^{t'} \eta_t \boldsymbol{x}_{t-1} - \sum_{t=1}^{t'} \eta_t \boldsymbol{x}_\infty)}{\sum_{t=1}^T \eta_t} - \lambda \frac{\sum_{t=1}^{t'} \eta_t \boldsymbol{x}_\infty + \sum_{t=t'+1}^T \eta_t \boldsymbol{x}_{t-1}}{\sum_{t=1}^T \eta_t} \end{aligned}$$

Then we choose $T' \ge t'$ such that $\sum_{t=1}^T \eta_t \ge \frac{2}{\epsilon}\left(\left\|\boldsymbol{x}_0 - \boldsymbol{x}_\infty - \lambda\left(\sum_{t=1}^{t'} \eta_t \boldsymbol{x}_{t-1} - \sum_{t=1}^{t'} \eta_t \boldsymbol{x}_\infty\right)\right\| + \frac{\epsilon}{2}\right)$ for $T \ge T'$ and have that

$$\begin{aligned} \left\|\frac{\sum_{t=1}^T \eta_t \boldsymbol{\Delta}_t}{\sum_{t=1}^T \eta_t} + \lambda \boldsymbol{x}_\infty\right\| &\le \frac{\left\|\boldsymbol{x}_0 - \boldsymbol{x}_T - \lambda\left(\sum_{t=1}^{t'} \eta_t \boldsymbol{x}_{t-1} - \sum_{t=1}^{t'} \eta_t \boldsymbol{x}_\infty\right)\right\|}{\sum_{t=1}^T \eta_t} + \lambda \frac{\sum_{t=t'+1}^T \eta_t \|\boldsymbol{x}_{t-1} - \boldsymbol{x}_\infty\|}{\sum_{t=1}^T \eta_t} \\ &\le \frac{\left\|\boldsymbol{x}_0 - \boldsymbol{x}_\infty - \lambda\left(\sum_{t=1}^{t'} \eta_t \boldsymbol{x}_{t-1} - \sum_{t=1}^{t'} \eta_t \boldsymbol{x}_\infty\right)\right\| + \|\boldsymbol{x}_T - \boldsymbol{x}_\infty\|}{\sum_{t=1}^T \eta_t} + \lambda \frac{\epsilon}{2\lambda} \\ &\le \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon \end{aligned}$$

So $\boldsymbol{\Delta}_\infty := \frac{\sum_{t=1}^T \eta_t \boldsymbol{\Delta}_t}{\sum_{t=1}^T \eta_t}$ exists and $\boldsymbol{\Delta}_\infty = -\lambda \boldsymbol{x}_\infty$.

2. Because $\nabla L(\boldsymbol{x})$ is a continuous function and $\lim_{t\to\infty} \boldsymbol{x}_t = \boldsymbol{x}_\infty$, $\lim_{t\to\infty} \nabla L(\boldsymbol{x}_t) = \nabla L(\boldsymbol{x}_\infty)$. For any $\epsilon > 0$, there exists T_1 such that $|\|\nabla L(\boldsymbol{x}_t)\|_* - \|\nabla L(\boldsymbol{x}_\infty)\|_*| \le \|\nabla L(\boldsymbol{x}_t) - \nabla L(\boldsymbol{x}_\infty)\|_* \le \frac{\epsilon}{3}$ for any $t \ge T_1$. It also holds that $|\langle \nabla L(\boldsymbol{x}_t) - \nabla L(\boldsymbol{x}_\infty), \boldsymbol{\Delta}_t\rangle| \le \|\nabla L(\boldsymbol{x}_t) - \nabla L(\boldsymbol{x}_\infty)\|_* \|\boldsymbol{\Delta}_t\| \le \frac{\epsilon}{3}$ because $\|\boldsymbol{\Delta}_t\| \le 1$. Because $\sum_{t=1}^\infty \eta_t = \infty$, there exists $T_2 \ge T_1$ such that $\sum_{t=1}^{T_2} \eta_t \ge \frac{3}{\epsilon}\left|\sum_{t=1}^{T_1} \eta_t \left(\|\nabla L(\boldsymbol{x}_t)\|_* - \|\nabla L(\boldsymbol{x}_\infty)\|_* + \langle \nabla L(\boldsymbol{x}_\infty) - \nabla L(\boldsymbol{x}_t), \boldsymbol{\Delta}_t\rangle\right)\right|$. Then for any $T \ge T_2$, we have that

$$
\begin{aligned}
&\left|\frac{\sum_{t=1}^{T}\eta_t\left\langle\nabla L(\boldsymbol{x}_\infty),\boldsymbol{\Delta}_t\right\rangle}{\sum_{t=1}^{T}\eta_t}-\left\|\nabla L(\boldsymbol{x}_\infty)\right\|_*\right| \\
=&\left|\frac{\sum_{t=1}^{T}\eta_t\left\langle\nabla L(\boldsymbol{x}_t),\boldsymbol{\Delta}_t\right\rangle}{\sum_{t=1}^{T}\eta_t}+\frac{\sum_{t=1}^{T}\eta_t\left\langle\nabla L(\boldsymbol{x}_\infty)-\nabla L(\boldsymbol{x}_t),\boldsymbol{\Delta}_t\right\rangle}{\sum_{t=1}^{T}\eta_t}-\left\|\nabla L(\boldsymbol{x}_\infty)\right\|_*\right| \\
=&\left|\frac{\sum_{t=1}^{T}\eta_t\left\|\nabla L(\boldsymbol{x}_t)\right\|_*}{\sum_{t=1}^{T}\eta_t}+\frac{\sum_{t=1}^{T}\eta_t\left\langle\nabla L(\boldsymbol{x}_\infty)-\nabla L(\boldsymbol{x}_t),\boldsymbol{\Delta}_t\right\rangle}{\sum_{t=1}^{T}\eta_t}-\left\|\nabla L(\boldsymbol{x}_\infty)\right\|_*\right| \\
=&\left|\frac{\sum_{t=1}^{T}\eta_t(\left\|\nabla L(\boldsymbol{x}_t)\right\|_*-\left\|\nabla L(\boldsymbol{x}_\infty)\right\|_*)}{\sum_{t=1}^{T}\eta_t}+\frac{\sum_{t=1}^{T}\eta_t\left\langle\nabla L(\boldsymbol{x}_\infty)-\nabla L(\boldsymbol{x}_t),\boldsymbol{\Delta}_t\right\rangle}{\sum_{t=1}^{T}\eta_t}\right| \\
\leq&\frac{\left|\sum_{t=1}^{T_1}\eta_t\left(\left\|\nabla L(\boldsymbol{x}_t)\right\|_*-\left\|\nabla L(\boldsymbol{x}_\infty)\right\|_*+\left\langle\nabla L(\boldsymbol{x}_\infty)-\nabla L(\boldsymbol{x}_t),\boldsymbol{\Delta}_t\right\rangle\right)\right|}{\sum_{t=1}^{T}\eta_t} \\
+&\frac{\sum_{t=T_1+1}^{T}\eta_t\left|\left\|\nabla L(\boldsymbol{x}_t)\right\|_*-\left\|\nabla L(\boldsymbol{x}_\infty)\right\|_*\right|}{\sum_{t=1}^{T}\eta_t}+\frac{\sum_{t=T_1+1}^{T}\eta_t\left|\left\langle\nabla L(\boldsymbol{x}_\infty)-\nabla L(\boldsymbol{x}_t),\boldsymbol{\Delta}_t\right\rangle\right|}{\sum_{t=1}^{T}\eta_t} \\
\leq&\frac{\epsilon}{3}+\frac{\epsilon}{3}\frac{\sum_{t=T_1+1}^{T}\eta_t}{\sum_{t=1}^{T}\eta_t}+\frac{\epsilon}{3}\frac{\sum_{t=T_1+1}^{T}\eta_t}{\sum_{t=1}^{T}\eta_t}\leq\epsilon
\end{aligned}
$$

Therefore, we prove that $\left\|\nabla L(\boldsymbol{x}_\infty)\right\|_*=\lim_{T\to\infty}\frac{\sum_{t=1}^{T}\eta_t\langle\nabla L(\boldsymbol{x}_\infty),\boldsymbol{\Delta}_t\rangle}{\sum_{t=1}^{T}\eta_t}$. On the other hand, we have that

$$
\begin{aligned}
\left\langle\nabla L(\boldsymbol{x}_\infty),\boldsymbol{\Delta}_\infty\right\rangle&=\left\langle\nabla L(\boldsymbol{x}_\infty),\lim_{T\to\infty}\frac{\sum_{t=1}^{T}\eta_t\boldsymbol{\Delta}_t}{\sum_{t=1}^{T}\eta_t}\right\rangle=\lim_{T\to\infty}\left\langle\nabla L(\boldsymbol{x}_\infty),\frac{\sum_{t=1}^{T}\eta_t\boldsymbol{\Delta}_t}{\sum_{t=1}^{T}\eta_t}\right\rangle \\
&=\lim_{T\to\infty}\frac{\sum_{t=1}^{T}\eta_t\left\langle\nabla L(\boldsymbol{x}_\infty),\boldsymbol{\Delta}_t\right\rangle}{\sum_{t=1}^{T}\eta_t},
\end{aligned}
$$

which finishes the proof.

3. For any T, $\left\|\frac{\sum_{t=1}^{T}\eta_t\boldsymbol{\Delta}_t}{\sum_{t=1}^{T}\eta_t}\right\|\leq\frac{\sum_{t=1}^{T}\eta_t\|\boldsymbol{\Delta}_t\|}{\sum_{t=1}^{T}\eta_t}\leq\frac{\sum_{t=1}^{T}\eta_t}{\sum_{t=1}^{T}\eta_t}=1$. By continuity of $\|\cdot\|$, $\|\Delta_\infty\|=\lim_{T\to\infty}\left\|\frac{\sum_{t=1}^{T}\eta_t\boldsymbol{\Delta}_t}{\sum_{t=1}^{T}\eta_t}\right\|\leq 1$.

□

B. Omitted Proofs in Section 4

B.1. Omitted proofs for upper bound for average update size of `Adam`

Proof of Lemma 4.2. We first represent m_t and v_t as a weighted sum of g_t and g_t^2. $m_t=\beta_1 m_{t-1}+(1-\beta_1)g_t=\beta_1^t m_0+(1-\beta_1)\sum_{i=0}^{t-1}\beta_1^i g_{t-i}$. $v_t\geq\beta_2 v_{t-1}+(1-\beta_2)g_t^2\geq\beta_2^t v_0+(1-\beta_2)\sum_{i=0}^{t-1}\beta_2^i g_{t-i}^2$. By Cauchy–Schwarz inequality, we have that

$$
\begin{aligned}
\left|\sum_{t=1}^{T}\eta_t\Delta_t\right|&=\left|\sum_{t=1}^{T}\eta_t\frac{\beta_1^t m_0+(1-\beta_1)\sum_{i=0}^{t-1}\beta_1^i g_{t-i}}{\sqrt{v_t}}\right| \\
&\leq\left[\sum_{t=1}^{T}\eta_t\left(\frac{\beta_1^t m_0^2}{v_t}+\sum_{i=0}^{t-1}(1-\beta_1)\beta_1^i\frac{g_{t-i}^2}{v_t}\right)\right]^{\frac{1}{2}}\left[\sum_{t=1}^{T}\eta_t\left(\beta_1^t+\sum_{i=0}^{t-1}(1-\beta_1)\beta_1^i\right)\right]^{\frac{1}{2}} \\
&\leq\left(\sum_{t=1}^{T}\eta_t\frac{\beta_1^t v_0}{v_t}+\sum_{t=1}^{T}\eta_t\sum_{i=0}^{t-1}(1-\beta_1)\beta_1^i\frac{v_{t-i}-\beta_2 v_{t-i-1}}{(1-\beta_2)v_t}\right)^{\frac{1}{2}}\left(\sum_{t=1}^{T}\eta_t\right)^{\frac{1}{2}}.
\end{aligned}
$$

We only calculate the first term in the following way

$$
\begin{aligned}
&\sum_{t=1}^{T}\eta_t\frac{\beta_1^t v_0}{v_t}+\sum_{t=1}^{T}\eta_t\sum_{i=0}^{t-1}(1-\beta_1)\beta_1^i\frac{v_{t-i}-\beta_2 v_{t-i-1}}{(1-\beta_2)v_t}\\
=&\sum_{t=1}^{T}\eta_t\frac{\beta_1^t v_0}{v_t}+\sum_{t=1}^{T}\eta_t\frac{1-\beta_1}{(1-\beta_2)v_t}\left(v_t-\beta_1^{t-1}\beta_2 v_0+\sum_{i=1}^{t-1}(\beta_1^i-\beta_2\beta_1^{i-1})v_{t-i}\right)\\
=&\sum_{t=1}^{T}\eta_t+\frac{\beta_2-\beta_1}{1-\beta_2}\sum_{t=1}^{T}\eta_t\left(1-\beta_1^{t-1}\frac{v_0}{v_t}-(1-\beta_1)\sum_{i=1}^{t-1}\beta_1^{i-1}\frac{v_{t-i}}{v_t}\right)\\
\le&\sum_{t=1}^{T}\eta_t+\frac{\beta_2-\beta_1}{1-\beta_2}\sum_{t=1}^{T}\eta_t\left(\beta_1^{t-1}+(1-\beta_1)\sum_{i=1}^{t-1}\beta_1^{i-1}\left(1-\frac{v_{t-i}}{v_t}\right)\right)\\
\le&\sum_{t=1}^{T}\eta_t+\frac{\beta_2-\beta_1}{1-\beta_2}\sum_{t=1}^{T}\eta_t\left(\beta_1^{t-1}+(1-\beta_1)\sum_{i=1}^{t-1}\beta_1^{i-1}\ln\left(\frac{v_t}{v_{t-i}}\right)\right)\\
=&\sum_{t=1}^{T}\eta_t+\frac{\beta_2-\beta_1}{1-\beta_2}\sum_{t=1}^{T}\eta_t\beta_1^{t-1}+\frac{(\beta_2-\beta_1)(1-\beta_1)}{1-\beta_2}\sum_{t=1}^{T}\left(\eta_t\frac{1-\beta_1^{t-1}}{1-\beta_1}-\sum_{i=1}^{T-t}\eta_{t+i}\beta_1^{i-1}\right)\ln v_t\\
=&\sum_{t=1}^{T}\eta_t+\frac{\beta_2-\beta_1}{1-\beta_2}\sum_{t=1}^{T}\eta_t\beta_1^{t-1}+\frac{(\beta_2-\beta_1)(1-\beta_1)}{1-\beta_2}\sum_{t=2}^{T}\left(\eta_t\frac{1-\beta_1^{t-1}}{1-\beta_1}-\sum_{i=1}^{T-t}\eta_{t+i}\beta_1^{i-1}\right)\ln\frac{v_t}{v_1}.
\end{aligned}
$$

When $\beta_1=\beta_2$, we have that

$$
\begin{aligned}
|\Delta_t|=&\left|\frac{\beta_1^t m_0+(1-\beta_1)\sum_{i=0}^{t-1}\beta_1^i g_{t-i}}{\sqrt{v_t}}\right|\le\left(\frac{\beta_1^t m_0^2+(1-\beta_1)\sum_{i=0}^{t-1}\beta_1^i g_{t-i}^2}{v_t}\right)^{\frac{1}{2}}\left(\beta_1^t+\sum_{i=0}^{t-1}(1-\beta_1)\beta_1^i\right)^{\frac{1}{2}}\\
\le&\left(\frac{\beta_1^t v_0+(1-\beta_1)\sum_{i=0}^{t-1}\beta_1^i g_{t-i}^2}{v_t}\right)^{\frac{1}{2}}=1
\end{aligned}
$$

□

B.2. Proof for Lemma 4.1

Proof of Lemma 4.1.

1. The proof for this part is the same as the proof of Lemma 3.9 in Appendix A.3.
2. If $\nabla L(\boldsymbol{x}_\infty)=\mathbf{0}$, $\langle\nabla L(\boldsymbol{x}_\infty),\boldsymbol{\Delta}_\infty\rangle=0=\|\nabla L(\boldsymbol{x}_\infty)\|_1$.
   If $\nabla L(\boldsymbol{x}_\infty)\neq\mathbf{0}$, we consider each coordinate j such that $\nabla L(\boldsymbol{x}_\infty)_j\neq 0$. $\lim_{t\to\infty}\nabla L(\boldsymbol{x}_t)_j=\nabla L(\boldsymbol{x}_\infty)_j$.
   $\boldsymbol{m}_{t,j}=(1-\beta_1)\sum_{i=0}^{t}\beta_1^i\boldsymbol{g}_{t-i,j}\to\nabla L(\boldsymbol{x}_\infty)_j$ and $\boldsymbol{v}_{t,j}=(1-\beta_2)\sum_{i=0}^{t}\beta_2^i\boldsymbol{g}_{t-i,j}^2\to\nabla L(\boldsymbol{x}_\infty)_j^2$. $\lim_{t\to\infty}\boldsymbol{\Delta}_{t,j}=\lim_{t\to\infty}\frac{\boldsymbol{m}_{t,j}}{\sqrt{\boldsymbol{v}_{t,j}}}=\text{sign}(\nabla L(\boldsymbol{x}_\infty)_j)$. For any $\epsilon>0$, there exists t' such that $\|\boldsymbol{\Delta}_{t,j}-\text{sign}\left(\nabla L(\boldsymbol{x}_\infty)_j\right)\|\le\frac{\epsilon}{2}$ for $t\ge t'$.
   And there exists $T'\ge t'$ such that $\sum_{t=1}^{T}\eta_t\ge\frac{2}{\epsilon}\sum_{t=1}^{t'}\eta_t\left(\boldsymbol{\Delta}_{t,j}-\text{sign}\left(\nabla L(\boldsymbol{x}_\infty)_j\right)\right)$ for any $T\ge T'$. Then for any $T\ge T'$, we have that

$$
\begin{aligned}
&\left\|\frac{\sum_{t=1}^{T}\eta_t\boldsymbol{\Delta}_{t,j}}{\sum_{t=1}^{T}\eta_t}-\text{sign}(\nabla L(\boldsymbol{x}_\infty)_j)\right\|\\
\le&\left\|\frac{\sum_{t=1}^{t'}\eta_t\left(\boldsymbol{\Delta}_{t,j}-\text{sign}(\nabla L(\boldsymbol{x}_\infty)_j)\right)}{\sum_{t=1}^{T}\eta_t}\right\|+\frac{\sum_{t=t'+1}^{T}\eta_t\left\|\boldsymbol{\Delta}_{t,j}-\text{sign}(\nabla L(\boldsymbol{x}_\infty)_j)\right\|}{\sum_{t=1}^{T}\eta_t}\\
\le&\frac{\epsilon}{2}+\frac{\epsilon}{2}=\epsilon.
\end{aligned}
$$

So $\boldsymbol{\Delta}_{\infty,j} = \lim_{T\to\infty} \frac{\sum_{t=1}^T \eta_t \boldsymbol{\Delta}_{t,j}}{\sum_{t=1}^T \eta_t} = \text{sign}(\nabla L(\boldsymbol{x}_\infty)_j)$ for $\nabla L(\boldsymbol{x}_\infty)_j \neq 0$. Then we have that

$$\langle \nabla L(\boldsymbol{x}_\infty), \boldsymbol{\Delta}_\infty \rangle = \sum_{\nabla L(\boldsymbol{x}_\infty)_j \neq 0} \nabla L(\boldsymbol{x}_\infty)_j \boldsymbol{\Delta}_{\infty,j} = \sum_{\nabla L(\boldsymbol{x}_\infty)_j \neq 0} |\nabla L(\boldsymbol{x}_\infty)_j| = \|\nabla L(\boldsymbol{x}_\infty)\|_1 .$$

3. For any nonzero coordinate j of $\nabla L(\boldsymbol{x}_\infty)$, from above we have $|\boldsymbol{\Delta}_{\infty,j}| = |\text{sign}(\nabla L(\boldsymbol{x}_\infty)_j)| = 1$.

   For j such that $\nabla L(\boldsymbol{x}_\infty)_j = 0$, we know $\lim_{t\to\infty} \boldsymbol{g}_{t,j} = \lim_{t\to\infty} \boldsymbol{m}_{t,j} = \lim_{t\to\infty} \boldsymbol{v}_{t,j} = 0$. We employ the upper bound for average update in Lemma 4.2 since $\{\boldsymbol{g}_{t,j}\}_{t=1}^\infty$ and $\{\boldsymbol{v}_{t,j}\}_{t=0}^\infty$ in Algorithm 1 satisfy the condition that $\boldsymbol{v}_{t,j} - \beta_2 \boldsymbol{v}_{t-1,j} \geq (1-\beta_2)\boldsymbol{g}_{t,j}^2$ and $\boldsymbol{m}_{0,j} = 0 \leq \sqrt{\boldsymbol{v}_{0,j}}$. By Lemma 4.2 we have

$$\begin{aligned}
&\left| \frac{\sum_{t=1}^T \eta_t \boldsymbol{\Delta}_{t,j}}{\sum_{t=1}^T \eta_t} \right| \\
&\leq \left(\frac{\sum_{t=1}^T \eta_t + \frac{\beta_2-\beta_1}{1-\beta_2} \sum_{t=1}^T \eta_t \beta_1^{t-1} + \frac{(\beta_2-\beta_1)(1-\beta_1)}{1-\beta_2} \sum_{t=2}^T \left(\eta_t \frac{1-\beta_1^{t-1}}{1-\beta_1} - \sum_{i=1}^{T-t} \eta_{t+i} \beta_1^{i-1} \right) \ln \frac{\boldsymbol{v}_{t,j}}{\boldsymbol{v}_{1,j}}}{\sum_{t=1}^T \eta_t} \right)^{\frac{1}{2}} .
\end{aligned}$$

   The denominator goes to ∞ when $T \to \infty$. So it suffices to bound the last two terms in the numerator by constants in order to show $\|\boldsymbol{\Delta}_\infty\| \leq 1$. Because η_t is non-increasing in t, it holds that

$$\sum_{t=1}^T \eta_t \beta_1^t \leq \sum_{t=1}^T \eta_1 \beta_1^t \leq \frac{\eta_1 \beta_1}{1-\beta_1}.$$

   For the last term, we first analyze the coefficient between each $\ln \boldsymbol{v}_{t,j}$. Define $\alpha_t = \eta_t \frac{1-\beta_1^{t-1}}{1-\beta_1} - \sum_{i=1}^{T-t} \eta_{t+i} \beta_1^{i-1}$. We claim that $|\alpha_t| \leq \max\{\frac{\beta_1^{t-1}}{1-\beta_1}\eta_{t+1}, \frac{\eta_t}{1-\beta_1}\} = \frac{\eta_t}{1-\beta_1}$. This is because

$$\alpha_t \leq \eta_t \frac{1-\beta_1^{t-1}}{1-\beta_1} \leq \frac{\eta_t}{1-\beta_1},$$

   and again by monotonicity of learning rates η_t, we have that

$$\alpha_t \geq \eta_{t+1} \frac{1-\beta_1^{t-1}}{1-\beta_1} - \sum_{i=1}^{T-t} \eta_{t+1} \beta_1^{i-1} \geq \eta_{t+1} \left(\frac{1-\beta_1^{t-1}}{1-\beta_1} - \sum_{i=1}^{\infty} \beta_1^{i-1} \right) = -\frac{\beta_1^{t-1}}{1-\beta_1} \eta_{t+1}.$$

   We can also have $\ln \frac{\boldsymbol{v}_{t,j}}{\boldsymbol{v}_{1,j}} \geq (t-1)\ln\beta_2$ because $\ln \boldsymbol{v}_{t,j} - \ln \boldsymbol{v}_{t-1,j} = \ln \frac{\boldsymbol{v}_{t,j}}{\boldsymbol{v}_{t-1,j}} = \ln \frac{\beta_2 \boldsymbol{v}_{t-1,j} + (1-\beta_2)\boldsymbol{g}_{t,j}^2}{\boldsymbol{v}_{t-1,j}} \geq \ln \beta_2$. And there exists t' such that $\ln \frac{\boldsymbol{v}_{t,j}}{\boldsymbol{v}_{1,j}} \leq 0$ for any $t \geq t'$ because $\lim_{t\to\infty} \boldsymbol{v}_{t,j} = 0$. Then it holds that

$$\begin{aligned}
\sum_{t=2}^T \alpha_t \ln \frac{\boldsymbol{v}_{t,j}}{\boldsymbol{v}_{1,j}} &\leq \sum_{t=2}^{t'} |\alpha_t| \left| \ln \frac{\boldsymbol{v}_{t,j}}{\boldsymbol{v}_{1,j}} \right| + \sum_{t=t'+1}^T \alpha_t \ln \frac{\boldsymbol{v}_{t,j}}{\boldsymbol{v}_{1,j}} \\
&\leq \sum_{t=2}^{t'} \frac{\eta_t}{1-\beta_1} \left| \ln \frac{\boldsymbol{v}_{t,j}}{\boldsymbol{v}_{1,j}} \right| + \sum_{\alpha_t \leq 0,\ t'<t\leq T} \alpha_t \ln \frac{\boldsymbol{v}_{t,j}}{\boldsymbol{v}_{1,j}} \\
&\leq \frac{\sum_{t=2}^{t'} \eta_t \left| \ln \frac{\boldsymbol{v}_{t,j}}{\boldsymbol{v}_{1,j}} \right|}{1-\beta_1} + \sum_{\alpha_t \leq 0,\ t'<t\leq T} \left(-\frac{\beta_1^t}{1-\beta_1} \eta_{t+1} \right) (t-1)\ln\beta_2 \\
&\leq \frac{\sum_{t=2}^{t'} \eta_t \left| \ln \frac{\boldsymbol{v}_{t,j}}{\boldsymbol{v}_{1,j}} \right|}{1-\beta_1} + (-\ln\beta_2) \sum_{t=1}^T \frac{(t-1)\beta_1^t}{1-\beta_1} \eta_{t+1} \\
&\leq \frac{\sum_{t=2}^{t'} \eta_t \left| \ln \frac{\boldsymbol{v}_{t,j}}{\boldsymbol{v}_{1,j}} \right|}{1-\beta_1} + (-\ln\beta_2)\eta_1 \sum_{t=1}^T \frac{(t-1)\beta_1^t}{1-\beta_1} \\
&\leq \frac{\sum_{t=2}^{t'} \eta_t \left| \ln \frac{\boldsymbol{v}_{t,j}}{\boldsymbol{v}_{1,j}} \right|}{1-\beta_1} - \frac{\eta_1 \beta_1^2 \ln\beta_2}{(1-\beta_1)^3} .
\end{aligned}$$

Define $C := \frac{(\beta_2-\beta_1)\eta_1\beta_1}{(1-\beta_2)(1-\beta_1)} + \frac{\beta_2-\beta_1}{1-\beta_2}\left(\sum_{t=2}^{t'} \eta_t \left|\ln \boldsymbol{v}_{t,j}\right| - \frac{\eta_1\beta_1^2 \ln \beta_2}{(1-\beta_1)^2}\right)$, we now have

$$\left|\frac{\sum_{t=1}^{T} \eta_t \boldsymbol{\Delta}_{t,j}}{\sum_{t=1}^{T} \eta_t}\right| \leq \left(\frac{\sum_{t=1}^{T} \eta_t + C}{\sum_{t=1}^{T} \eta_t}\right)^{\frac{1}{2}}.$$

Therefore $|\boldsymbol{\Delta}_{\infty,j}| = \left|\lim_{T\to\infty} \frac{\sum_{t=1}^{T} \eta_t \boldsymbol{\Delta}_{t,j}}{\sum_{t=1}^{T} \eta_t}\right| \leq \lim_{T\to\infty} \left|\frac{\sum_{t=1}^{T} \eta_t \boldsymbol{\Delta}_{t,j}}{\sum_{t=1}^{T} \eta_t}\right| \leq 1$, since $\sum_{t=1}^{\infty} \eta_T = \infty$. This completes the proof.

□

B.3. A counter example when $\beta_1 > \beta_2$

For any λ and $\beta_1 > \beta_2$, we provide the following example for which the iterates of `AdamW` will converge and the ℓ_∞ norm of the converged solution is larger than $\frac{1}{\lambda}$.

For some sufficiently small η, denote $\tilde{x} = -\frac{1}{\lambda}\frac{\frac{1-\beta_1}{1-\lambda\eta-\beta_1}}{\sqrt{\frac{1-\beta_2}{(1-\lambda\eta)^2-\beta_2}}}$. $L(x)$ is defined as $\frac{1}{2}(x-\tilde{x})^2$. For any starting point $x_0 > \tilde{x}$, m_0 is set as $\frac{1-\beta_1}{1-\lambda\eta-\beta_1} g_1$ and v_0 is set as $\frac{1-\beta_2}{(1-\lambda\eta)^2-\beta_2} g_1^2$ with $g_1 = \nabla L(x_0) = x_0 - \tilde{x}$. We show by induction that $\frac{m_t}{\sqrt{v_t}} = -\lambda\tilde{x}$ and $x_t - \tilde{x} = (1-\lambda\eta)(x_{t-1}-\tilde{x})$ for any $t \geq 1$.

When $t = 1$, we have that $m_1 = \beta_1 m_0 + (1-\beta_1) g_1 = \frac{(1-\beta_1)(1-\lambda\eta)}{1-\lambda\eta-\beta_1} g_1$ and $v_1 = \beta_2 v_0 + (1-\beta_2) g_1^2 = \frac{(1-\beta_2)(1-\lambda\eta)^2}{(1-\lambda\eta)^2-\beta_2} g_1^2$. Then $\frac{m_1}{\sqrt{v_1}} = \text{sign}(g_1)\frac{\frac{1-\beta_1}{1-\lambda\eta-\beta_1}}{\sqrt{\frac{1-\beta_2}{(1-\lambda\eta)^2-\beta_2}}} = \frac{\frac{1-\beta_1}{1-\lambda\eta-\beta_1}}{\sqrt{\frac{1-\beta_2}{(1-\lambda\eta)^2-\beta_2}}} = -\lambda\tilde{x}$, which proves the first claim. For the second claim, we have that

$$x_1 - \tilde{x} = x_0 - \eta\frac{m_1}{\sqrt{v_1}} - \lambda\eta x_0 - \tilde{x} = x_0 + \lambda\eta\tilde{x} - \lambda\eta x_0 - \tilde{x} = (1-\lambda\eta)(x_0-\tilde{x}).$$

Suppose the claims hold for any $0 \leq i < t$. Then $g_{i+1} = \nabla L(x_i) = x_i - \tilde{x} = (1-\lambda\eta)^i g_1$. We have that

$$\begin{aligned} m_t &= \beta_1^t m_0 + (1-\beta_1)\sum_{i=1}^{t} \beta_1^{t-i} g_i = \beta_1^t \frac{1-\beta_1}{1-\lambda\eta-\beta_1} g_1 + (1-\beta_1)\sum_{i=1}^{t} \beta_1^{t-i}(1-\lambda\eta)^{i-1} g_1 \\ &= \left(\beta_1^t \frac{1-\beta_1}{1-\lambda\eta-\beta_1} + (1-\beta_1)\frac{(1-\lambda\eta)^t - \beta_1^t}{1-\lambda\eta-\beta_1}\right) g_1 = \frac{1-\beta_1}{1-\lambda\eta-\beta_1}(1-\lambda\eta)^t g_1, \end{aligned}$$

and

$$\begin{aligned} v_t &= \beta_2^t v_0 + (1-\beta_2)\sum_{i=1}^{t} \beta_2^{t-i} g_i^2 = \beta_2^t \frac{1-\beta_2}{(1-\lambda\eta)^2-\beta_2} g_1^2 + (1-\beta_2)\sum_{i=1}^{t} \beta_2^{t-i}(1-\lambda\eta)^{2(i-1)} g_1^2 \\ &= \left(\beta_2^t \frac{1-\beta_2}{(1-\lambda\eta)^2-\beta_2} + (1-\beta_2)\frac{(1-\lambda\eta)^{2t} - \beta_2^t}{(1-\lambda\eta)^2-\beta_2}\right) g_1^2 = \frac{1-\beta_2}{(1-\lambda\eta)^2-\beta_2}(1-\lambda\eta)^{2t} g_1^2. \end{aligned}$$

Then $\frac{m_t}{\sqrt{v_t}} = \frac{\frac{1-\beta_1}{1-\lambda\eta-\beta_1}}{\sqrt{\frac{1-\beta_2}{(1-\lambda\eta)^2-\beta_2}}} = -\lambda\tilde{x}$. We also have that

$$x_t - \tilde{x} = x_{t-1} - \eta\frac{m_t}{\sqrt{v_t}} - \lambda\eta x_{t-1} - \tilde{x} = x_0 + \lambda\eta\tilde{x} - \lambda\eta x_0 - \tilde{x} = (1-\lambda\eta)(x_0-\tilde{x}).$$

.

In this regime, x_t will converge to $\tilde{x}$ because $|x_t - \tilde{x}| = O((1-\lambda\eta)^t)$. However, when $\beta_1 > \beta_2$ and $\lambda\eta$ is very small, $|\tilde{x}|$ can be larger than $\frac{1}{\lambda}$. For example, when $\beta_1 = 0.99$, $\beta_2 = 0.9$, $\lambda = 0.1$ and $\eta = 0.01$, $|\tilde{x}| = 10.999 > \frac{1}{\lambda}$.

B.4. Proof for upper bound for norm of iterates in AdamW

Proof of Lemma 4.3. For AdamW with constant learning rate η and each coordinate j, $\boldsymbol{x}_{T,j}$ can be written as weighted average of past update

$$\boldsymbol{x}_{T,j} = (1-\lambda\eta)^T \boldsymbol{x}_{0,j} + \sum_{t=0}^{T-1} \eta(1-\lambda\eta)^t \frac{\boldsymbol{m}_{T-t,j}}{\sqrt{\boldsymbol{v}_{T-t,j}}} = (1-\lambda\eta)^T \boldsymbol{x}_{0,j} + \sum_{t=1}^{T} \eta(1-\lambda\eta)^{T-t} \frac{\boldsymbol{m}_{t,j}}{\sqrt{\boldsymbol{v}_{t,j}}}.$$

Define $\eta_t = \eta(1-\lambda\eta)^{T-t}$ for $1 \le t \le T$. We apply Lemma 4.2 on $\{\boldsymbol{v}_{t,j}\}_{t=1}^T$ and $\{\boldsymbol{g}_{t,j}\}_{t=1}^T$ to bound $\left|\sum_{t=1}^T \eta(1-\lambda\eta)^{T-t} \frac{\boldsymbol{m}_{t,j}}{\sqrt{\boldsymbol{v}_{t,j}}}\right|$.

We first compute $\sum_{t=1}^T \eta_t = \frac{1-(1-\lambda\eta)^T}{\lambda} \le \frac{1}{\lambda}$. For the second term in Equation 4, we have that

$$\begin{aligned}
\frac{\beta_2-\beta_1}{1-\beta_2}\sum_{t=1}^T \eta_t \beta_1^{t-1} &= \frac{\beta_2-\beta_1}{1-\beta_2}\sum_{t=1}^T \eta(1-\lambda\eta)^{T-t}\beta_1^{t-1} = \frac{1}{\lambda}\frac{(\beta_2-\beta_1)\lambda\eta[(1-\lambda\eta)^T-\beta_1^T]}{(1-\beta_2)(1-\lambda\eta-\beta_1)} \\
&\le \frac{(\beta_2-\beta_1)\eta}{(1-\beta_2)\left|1-\lambda\eta-\beta_1\right|}\left[\beta_1^T + (1-\lambda\eta)^T\right].
\end{aligned}$$

For the last term, we define $\alpha_t = \eta_t \frac{1-\beta_1^{t-1}}{1-\beta_1} - \sum_{i=1}^{T-t} \eta_{t+i}\beta_1^{i-1}$ and we can compute the exact form of α_t as following

$$\begin{aligned}
\alpha_t &= \eta(1-\lambda\eta)^{T-t}\frac{1-\beta_1^{t-1}}{1-\beta_1} - \sum_{i=1}^{T-t}\eta(1-\lambda\eta)^{T-t-i}\beta_1^{i-1} = \eta(1-\lambda\eta)^{T-t}\frac{1-\beta_1^{t-1}}{1-\beta_1} - \frac{\eta[(1-\lambda\eta)^{T-t}-\beta_1^{T-t}]}{1-\lambda\eta-\beta_1} \\
&= \frac{\eta(1-\lambda\eta)^{T-t}(-\lambda\eta)}{(1-\beta_1)(1-\lambda\eta-\beta_1)} - \frac{\eta(1-\lambda\eta)^{T-t}\beta_1^{t-1}}{1-\beta_1} + \frac{\eta\beta_1^{T-t}}{1-\lambda\eta-\beta_1}.
\end{aligned}$$

Then we can bound the last term by showing that

$$\begin{aligned}
&\frac{(\beta_2-\beta_1)(1-\beta_1)}{1-\beta_2}\sum_{t=2}^T \alpha_t \ln\frac{\boldsymbol{v}_{t,j}}{\boldsymbol{v}_{1,j}} \\
\le& \frac{(\beta_2-\beta_1)(1-\beta_1)}{1-\beta_2}\sum_{t=2}^T |\alpha_t|\left|\ln\frac{\boldsymbol{v}_{t,j}}{\boldsymbol{v}_{1,j}}\right| \\
\le& C\frac{(\beta_2-\beta_1)(1-\beta_1)}{1-\beta_2}\left[\sum_{t=2}^T \frac{\lambda\eta^2(1-\lambda\eta)^{T-t}}{(1-\beta_1)|1-\lambda\eta-\beta_1|} + \sum_{t=2}^T \frac{\eta(1-\lambda\eta)^{T-t}\beta_1^{t-1}}{1-\beta_1} + \sum_{t=2}^T \frac{\eta\beta_1^{T-t}}{|1-\lambda\eta-\beta_1|}\right] \\
=& C\frac{(\beta_2-\beta_1)(1-\beta_1)}{1-\beta_2}\left[\frac{\eta[1-(1-\lambda\eta)^{T-1}]}{(1-\beta_1)|1-\lambda\eta-\beta_1|} + \frac{\beta_1\eta|(1-\lambda\eta)^{T-1}-\beta_1^{T-1}|}{(1-\beta_1)|1-\lambda\eta-\beta_1|} + \frac{\eta(1-\beta_1^{T-1})}{|1-\lambda\eta-\beta_1|(1-\beta_1)}\right] \\
\le& C\frac{(\beta_2-\beta_1)(1-\beta_1)}{1-\beta_2}\left[\frac{\eta[1-(1-\lambda\eta)^{T-1}]}{(1-\beta_1)|1-\lambda\eta-\beta_1|} + \frac{\beta_1\eta\left((1-\lambda\eta)^{T-1}+\beta_1^{T-1}\right)}{(1-\beta_1)|1-\lambda\eta-\beta_1|} + \frac{\eta(1-\beta_1^{T-1})}{|1-\lambda\eta-\beta_1|(1-\beta_1)}\right] \\
=& \frac{2C\eta(\beta_2-\beta_1)}{(1-\beta_2)|1-\lambda\eta-\beta_1|} + \frac{C\eta(\beta_2-\beta_1)(\beta_1-1)}{(1-\beta_2)|1-\lambda\eta-\beta_1|}\left[\beta_1^{T-1}+(1-\lambda\eta)^{T-1}\right] \le \frac{2C\eta(\beta_2-\beta_1)}{(1-\beta_2)|1-\lambda\eta-\beta_1|}.
\end{aligned}$$

Therefore, we have the following bound

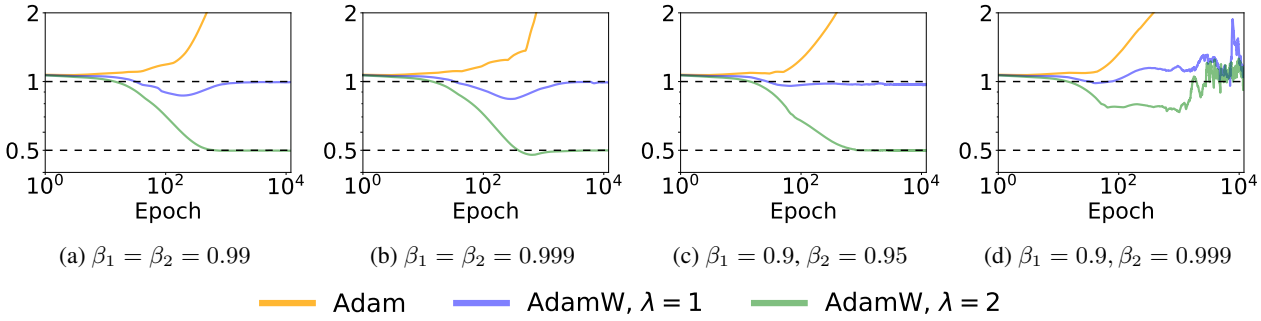
$$
\begin{aligned}
\lambda|\boldsymbol{x}_{T,j}| &\leq \lambda(1-\lambda\eta)^T|\boldsymbol{x}_{0,j}| + \lambda\left|\sum_{t=1}^{T}\eta_t\frac{\boldsymbol{m}_{t,j}}{\sqrt{\boldsymbol{v}_{t,j}}}\right| \\
&\leq \lambda(1-\lambda\eta)^T|\boldsymbol{x}_{0,j}| + \lambda\sum_{t=1}^{T}\eta_t\left[1+\frac{(\beta_2-\beta_1)\eta\left[\beta_1^T+(1-\lambda\eta)^T\right]}{\sum_{t=1}^{T}\eta_t(1-\beta_2)\left|1-\lambda\eta-\beta_1\right|}+\frac{2C\eta(\beta_2-\beta_1)}{\sum_{t=1}^{T}\eta_t(1-\beta_2)|1-\lambda\eta-\beta_1|}\right]^{\frac{1}{2}} \\
&\leq \lambda(1-\lambda\eta)^T|\boldsymbol{x}_{0,j}| + \left[(\lambda\sum_{t=1}^{T}\eta_t)^2+\lambda^2\sum_{t=1}^{T}\eta_t\left[\frac{(\beta_2-\beta_1)\eta\left[\beta_1^T+(1-\lambda\eta)^T\right]+2C\eta(\beta_2-\beta_1)}{(1-\beta_2)\left|1-\lambda\eta-\beta_1\right|}\right]\right]^{\frac{1}{2}} \\
&\leq \lambda(1-\lambda\eta)^T|\boldsymbol{x}_{0,j}| + \left[1+\lambda\frac{(\beta_2-\beta_1)\eta\left[\beta_1^T+(1-\lambda\eta)^T\right]+2C\eta(\beta_2-\beta_1)}{(1-\beta_2)\left|1-\lambda\eta-\beta_1\right|}\right]^{\frac{1}{2}} \\
&\leq \lambda(1-\lambda\eta)^T|\boldsymbol{x}_{0,j}| + \left[1+\frac{\lambda\eta(\beta_2-\beta_1)\left[\beta_1^T+(1-\lambda\eta)^T\right]}{2(1-\beta_2)|1-\lambda\eta-\beta_1|}+C\frac{\lambda\eta(\beta_2-\beta_1)}{(1-\beta_2)|1-\lambda\eta-\beta_1|}\right] \\
&\leq 1+\lambda(1-\lambda\eta)^T\left\|\boldsymbol{x}_0\right\|_\infty + \frac{\lambda\eta(\beta_2-\beta_1)\left[\beta_1^T+(1-\lambda\eta)^T\right]}{2(1-\beta_2)|1-\lambda\eta-\beta_1|}+C\frac{\lambda\eta(\beta_2-\beta_1)}{(1-\beta_2)|1-\lambda\eta-\beta_1|}.
\end{aligned}
$$

This completes the proof. □

C. Experimental Details and More Results

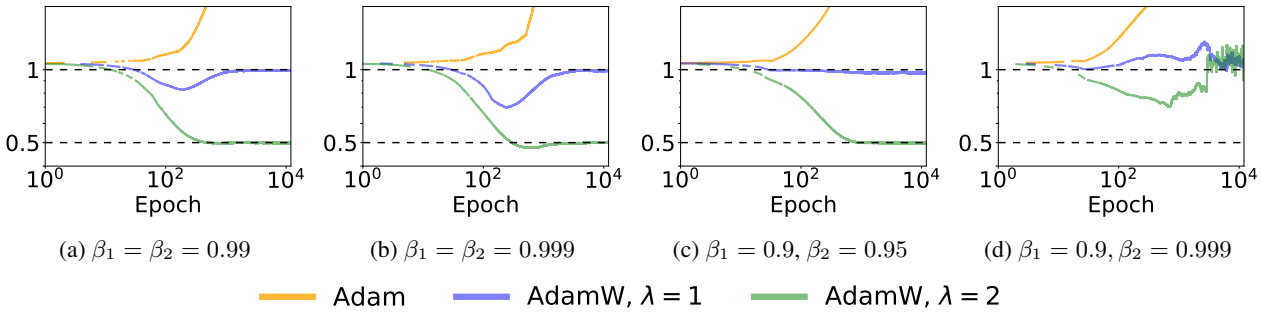
The architecture of the two-layer transformer is the same as in Kunstner et al. (2022), which is also used as a tutorial example in PyTorch. It consists of a 200-dimensional embedding layer, 2 transformer layers and a linear layer. Each transformer layer consists of a 2-head self-attention and an MLP with a hidden dimension 200. The experiments are run on a single A4000 or a single A6000.

As mentioned in Section 5, we present the results for another three random seeds in Figures 3 to 5. We also plot the results in full range for all the four random seeds in Figures 6 to 9 to show that the ℓ_∞ norm of parameters for `Adam` keeps increasing.



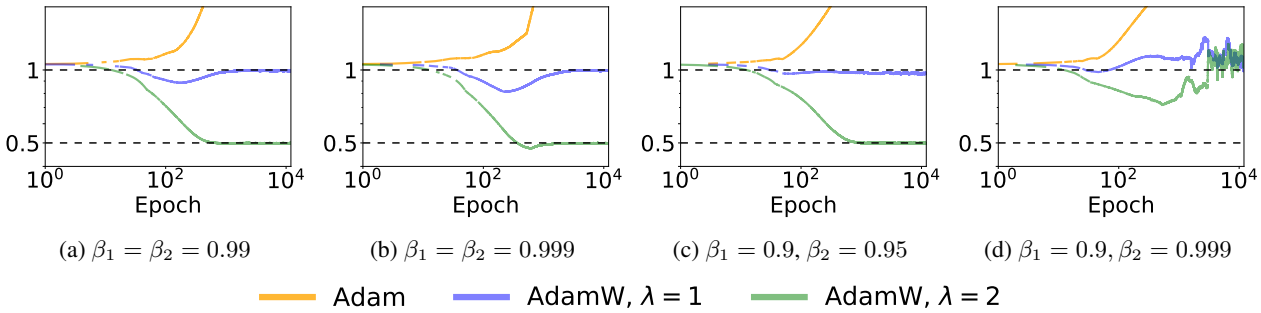
(a) $\beta_1 = \beta_2 = 0.99$ (b) $\beta_1 = \beta_2 = 0.999$ (c) $\beta_1 = 0.9, \beta_2 = 0.95$ (d) $\beta_1 = 0.9, \beta_2 = 0.999$

Figure 3: ℓ_∞ norm of parameters for `Adam` and `AdamW` with different β_1, β_2 for seed 1



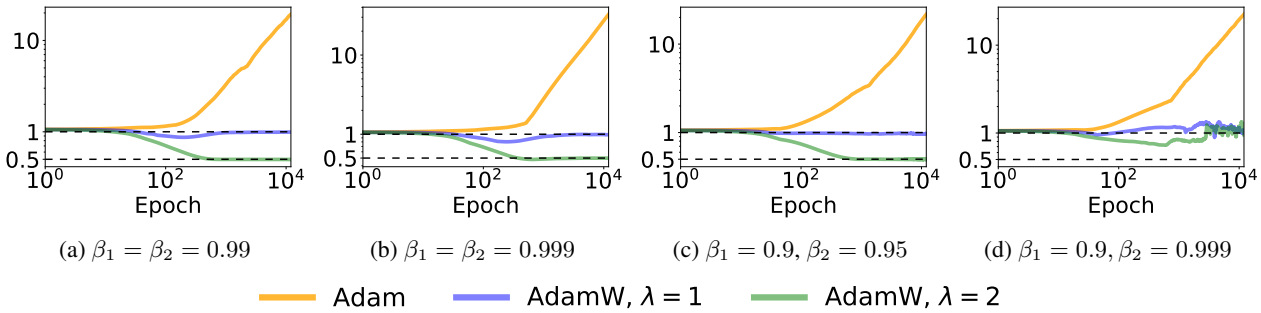
(a) $\beta_1 = \beta_2 = 0.99$ (b) $\beta_1 = \beta_2 = 0.999$ (c) $\beta_1 = 0.9, \beta_2 = 0.95$ (d) $\beta_1 = 0.9, \beta_2 = 0.999$

Figure 4: ℓ_∞ norm of parameters for `Adam` and `AdamW` with different β_1, β_2 for seed 2



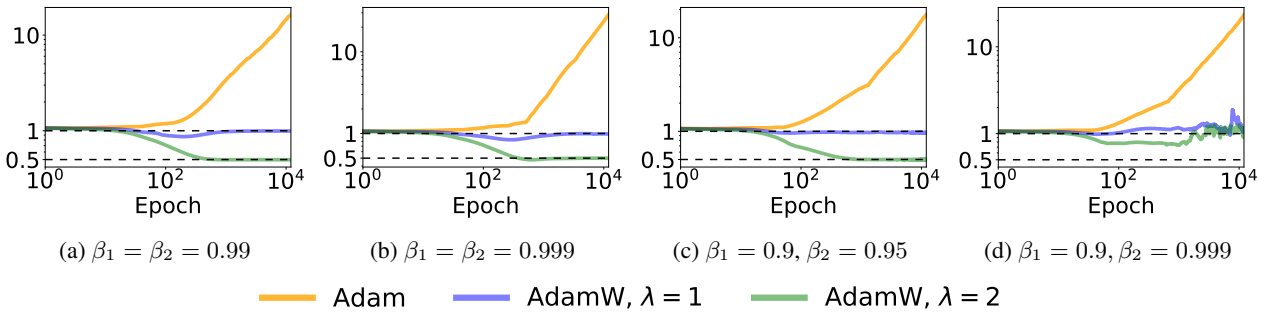
(a) $\beta_1 = \beta_2 = 0.99$ (b) $\beta_1 = \beta_2 = 0.999$ (c) $\beta_1 = 0.9, \beta_2 = 0.95$ (d) $\beta_1 = 0.9, \beta_2 = 0.999$

Figure 5: ℓ_∞ norm of parameters for `Adam` and `AdamW` with different β_1, β_2 for seed 3



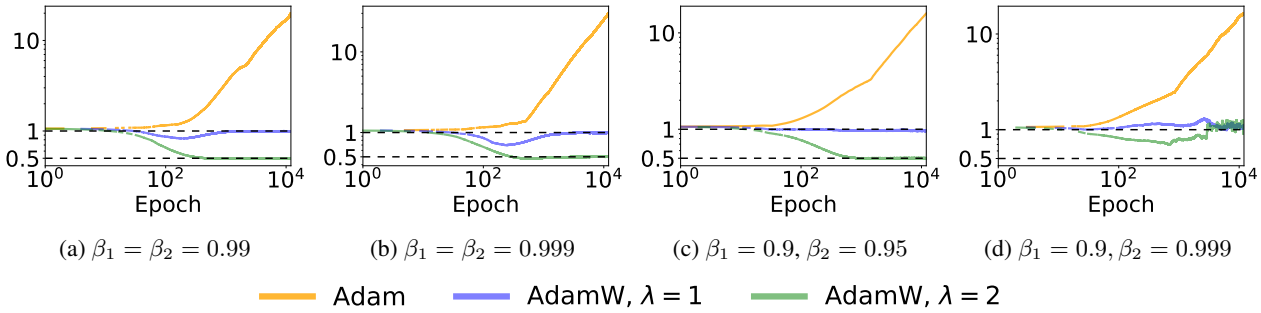
(a) $\beta_1 = \beta_2 = 0.99$ (b) $\beta_1 = \beta_2 = 0.999$ (c) $\beta_1 = 0.9, \beta_2 = 0.95$ (d) $\beta_1 = 0.9, \beta_2 = 0.999$

Figure 6: ℓ_∞ norm of parameters for `Adam` and `AdamW` with different β_1, β_2 for seed 0



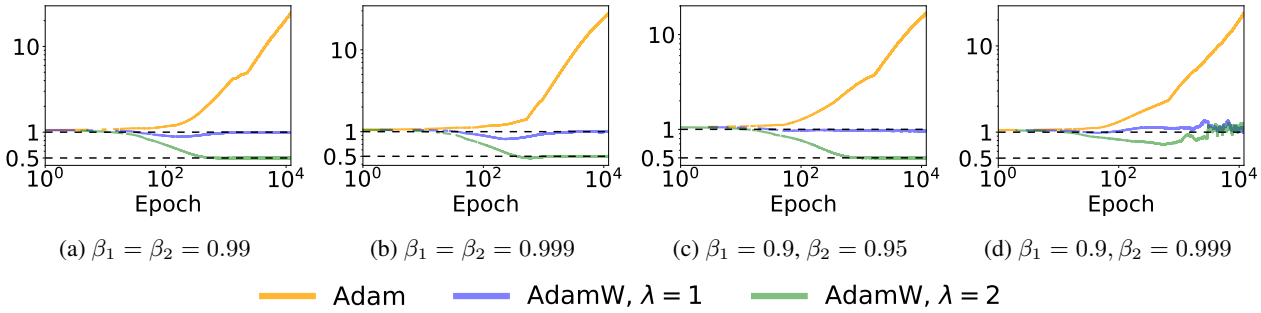
(a) $\beta_1 = \beta_2 = 0.99$ (b) $\beta_1 = \beta_2 = 0.999$ (c) $\beta_1 = 0.9, \beta_2 = 0.95$ (d) $\beta_1 = 0.9, \beta_2 = 0.999$

Figure 7: ℓ_∞ norm of parameters for `Adam` and `AdamW` with different β_1, β_2 for seed 1



(a) $\beta_1 = \beta_2 = 0.99$ (b) $\beta_1 = \beta_2 = 0.999$ (c) $\beta_1 = 0.9, \beta_2 = 0.95$ (d) $\beta_1 = 0.9, \beta_2 = 0.999$

Figure 8: ℓ_∞ norm of parameters for `Adam` and `AdamW` with different β_1, β_2 for seed 1



(a) $\beta_1 = \beta_2 = 0.99$ (b) $\beta_1 = \beta_2 = 0.999$ (c) $\beta_1 = 0.9, \beta_2 = 0.95$ (d) $\beta_1 = 0.9, \beta_2 = 0.999$

Figure 9: ℓ_∞ norm of parameters for `Adam` and `AdamW` with different β_1, β_2 for seed 3