We correct some typos and some updates in section 3.3 are highlighted in red to draw attention.

ADAM EXPLOITS ℓ_∞-GEOMETRY OF LOSS LANDSCAPE VIA COORDINATE-WISE ADAPTIVITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Adam outperforms SGD when training language models. Yet such benefits are not well-understood theoretically – previous convergence analysis for Adam and SGD mainly focuses on the number of steps T and is already minimax-optimal in non-convex cases, which are both $O(T^{-1/4})$. In this work, we argue that the better dependence on the loss smoothness is the key advantage of Adam over SGD. More specifically, we give a new convergence analysis for Adam under novel assumptions that loss is smooth under ℓ_∞ geometry rather than the more common ℓ_2 geometry, which yields a much better empirical smoothness constant for GPT-2 and ResNet models. Moreover, we show that if we rotate the training loss randomly, Adam can be outperformed by some variants of SGD which is invariant to rotations. This implies that any practically relevant explanation of Adam's optimization benefit must involve non-rotational invariant properties of loss, such as ℓ_∞ smoothness as used in our analysis. We also extend the convergence analysis to blockwise Adam, which is a generalization of standard Adam.

1 INTRODUCTION

Large language models (LLMs) have gained phenomenal capabilities as their scale grows (Radford et al., 2019; Kaplan et al., 2020; Brown et al., 2020; Zhang et al., 2022a; Touvron et al., 2023; OpenAI, 2023; Reid et al., 2024). However, pre-training LLMs are incredibly time-consuming. Adaptive Momentum Estimation (`Adam`)(Kingma & Ba, 2014) is the current to-go optimization algorithm for LLMs due to its fast convergence. In contrast, `SGD`, a popular and arguably the simplest optimizer, optimizes language model loss much more slowly than `Adam`.

However, the optimization benefit of `Adam` over `SGD` cannot be explained by existing theory. Existing convergence analyses for `Adam` and `SGD` focus on the dependence on the number of steps under assumptions on the smoothness and gradient bounds of the loss (Défossez et al., 2022), and it has been shown that both `Adam` and `SGD` achieve the minimax convergence rate $O(T^{-1/4})$ in the non-convex settings (Arjevani et al., 2023). Thus according to the theory, in the worst case, `SGD` would be more desirable compared to `Adam` because they have the same convergence rate, and yet `Adam` is less memory-efficient due to its coordinate-wise adaptivity, which needs to store the empirical moving average of second-order moments of past stochastic gradients. Therefore, we hypothesize that the coordinate-wise adaptivity in `Adam` is exploiting some unknown properties of LLMs which `SGD` cannot make use of.

Towards this end, we identified a significant difference between `Adam` and `SGD` in this paper. This difference, often ignored in previous works, is that `SGD` is rotation-invariant, while `Adam` is only permutation-invariant (see definitions in Section 2). Intuitively, this means if we rotate the loss landscape, the optimization trajectory of `SGD` would be the same (up to some rotation), while the trajectory of `Adam` could be completely different. If `Adam` optimizes much more slowly after rotation, then it suggests `Adam` is exploiting some non-rotational-invariant properties of the loss function, which is not captured by standard smoothness assumptions in the convergence analysis.

Figure 1 summarizes our findings by comparing `Adam` on the original and rotated loss. The performance of `Adam` on the rotated loss does become much worse than `Adam` on the original loss. We also test a memory-efficient and rotational-invariant variant of `SGD`, `AdaSGD` (Wang & Wiens, 2020)

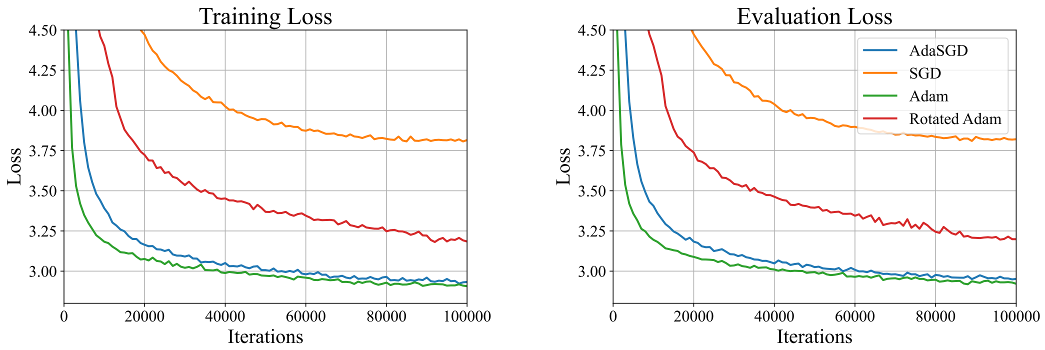

Figure 1: Training and validation losses of `Adam`, `AdaSGD` and `SGD` on GPT-2. `rotated Adam` means running `Adam` on a rotated loss. `Adam` on the original loss converges the fastest as expected. But convergence of `Adam` on a rotated loss is much slower, notably even worse than `AdaSGD`.

(defined in Algorithm 2)[1]. Surprisingly, the rotated `Adam` performs even much worse than the `SGD` variant. The results suggest it is impossible to explain the superior optimization performance of Adam over SGD just using rotationally invariant assumptions on the loss function, which raises the natural question,

> What are the non-rotation-invariant properties of a loss function that enable faster convergence of `Adam` than `SGD`?

We hypothesize that the ℓ_2 lipschitzness of loss gradient does not provide a tight-enough characterization of loss landscape of deep learning models in practice, such that we can separate `Adam` and other rotational-invariant algorithms. Inspired by the similarity between `Adam` and `SignGD` and the fact that `SignGD` is the normalized steepest descent with respect to ℓ_∞ norm, we propose to use ℓ_∞-norm related smoothness as a better tool to analyze `Adam`. In particular, our main results use the $(1,1)$-norm of Hessian of loss divided by variable dimension d in replacement of the spectral norm of Hessian as the smoothness measure, and prove a convergence rate of $O(\sqrt{\frac{1}{T}})$ for `Adam` without noise, or $O((\frac{\log T}{T})^{1/4})$ with noise. Our results have the same dependence on T as previous results, but much smaller smoothness constant when we measure it empirically. We empirically verify that $(1,1)$-norm of Hessian positively correlates with final training loss of `Adam` on both synthetic tasks like quadratic loss and real tasks like training GPT2 on OpenWebText and ResNet on CIFAR10.

We summarize our contributions below:

1. We show by experiments that the empirical optimization advantage of `Adam` over `SGD` can not be explained solely under rotation-invariant assumptions. (Figure 1)
2. We propose a new complexity metric for the optimization problem, which is the $(1,1)$-norm of the Hessian matrix of loss, $\left\|\nabla^2 L(x)\right\|_{1,1}$. We present a novel convergence result for `Adam` depending on this metric in the case of $\beta_1 = 0$. (Theorem 3.5)
3. We further generalize the theoretical analysis for `Adam` to blockwise `Adam` (Algorithm 3) whose convergence rate can be characterized by a novel smoothness measure (Theorem 3.12). `Adam` and `AdaSGD` are two notable examples of blockwise `Adam`. In `Adam`, all blocks are of size 1. In `AdaSGD`, there is only one block.
4. We empirically verify that when `Adam` converges more slowly on the rotated loss, the $(1,1)$-norm of Hessian also increases, which suggests that our new complexity metric for `Adam`'s convergence is practically relevant. (Section 4)

2 PRELIMINARIES

Notations. For $\boldsymbol{x} \in \mathbb{R}^d$, we define the vector p-norm $\|\boldsymbol{x}\|_p$ as $(\sum_{i=1}^d x_i^p)^{1/p}$ for $p \in [1,\infty]$. For a matrix $\boldsymbol{A} \in \mathbb{R}^{d_1 \times d_2}$, its $(1,1)$-norm $\|\boldsymbol{A}\|_{1,1}$ is defined as $\sum_{i=1}^{d_1}\sum_{j=1}^{d_2} |A_{i,j}|$ and its operator norm induced by vector p-norm $\|\cdot\|_p$ as $\sup_{\boldsymbol{x}\in\mathbb{R}^d} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_q}{\|\boldsymbol{x}\|_p}$, denoted by $\|\boldsymbol{A}\|_p$, where $\frac{1}{q}+\frac{1}{p}=1$ and $\|\cdot\|_q$

---

[1] There is one small difference. We use an exponential average of the gradient for $\boldsymbol{m}_t$ instead of momentum. Our definition makes `AdaSGD` the same as `Adam` in a one-dimensional problem.

Algorithm 1 Adam

```
Hyperparam: $\beta_1, \beta_2, \epsilon \geq 0$, total steps $T$, learning rate $\{\eta_t\}_{t=1}^{T}$, $\epsilon$, initial $\boldsymbol{m}_0, v_0$
Input: initialization $\boldsymbol{x}_0$, stochastic loss functions $\{L_t\}_{t=1}^{T}$
  $v_{0,i} \leftarrow v_0$
  for $t = 1, 2, \cdots, T$ :
    $g_{t,i} \leftarrow \nabla_i L_t(\boldsymbol{x}_{t-1})$
    $m_{t,i} \leftarrow \beta_1 m_{t-1,i} + (1-\beta_1) g_{t,i}$
    $v_{t,i} \leftarrow \beta_2 v_{t-1,i} + (1-\beta_2) g_{t,i}^2$
    $x_{t,i} \leftarrow x_{t-1,i} - \eta_t \frac{m_{t,i}}{\sqrt{v_{t,i}+\epsilon}}$
  return $\boldsymbol{x}_T$
```

Algorithm 2 AdaSGD

```
Hyperparam: $\beta_1, \beta_2, \epsilon \geq 0$, total steps $T$, learning rate $\{\eta_t\}_{t=1}^{T}$, initial $\boldsymbol{m}_0, v_0$
Input: initialization $\boldsymbol{x}_0$, stochastic loss functions $\{L_t\}_{t=1}^{T}$
  for $t = 1, 2, \cdots, T$ :
    $g_{t,i} \leftarrow \nabla_i L_t(\boldsymbol{x}_{t-1})$
    $m_{t,i} \leftarrow \beta_1 m_{t-1,i} + (1-\beta_1) g_{t,i}$
    $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2)(\|\boldsymbol{g}_t\|_2^2 / d)$
    $x_{t,i} \leftarrow x_{t-1,i} - \eta_t \frac{m_{t,i}}{\sqrt{v_t+\epsilon}}$
  return $\boldsymbol{x}_T$
```

is the dual norm of $\|\cdot\|_p$. For a deterministic loss function $L(\boldsymbol{x})$, we consider optimization over L with access only to independent stochastic functions $\{L_t(\boldsymbol{x})\}_{t=1}^{T}$ such that $\mathbb{E} L_t(\boldsymbol{x}) = L(\boldsymbol{x})$ for any input $\boldsymbol{x} \in \mathbb{R}^d$.

Rotation. For an invertible function $\mathcal{T} : \mathbb{R}^d \to \mathbb{R}^d$, $\mathcal{T}$ is a rotating transformation if there exists an orthogonal matrix $\boldsymbol{T} \in \mathbb{R}^{d \times d}$ such that $\mathcal{T}(\boldsymbol{x}) = \boldsymbol{T}\boldsymbol{x}$. $\mathcal{T}$ is a permutating transformation if there exists a permutation $\pi : [d] \to [d]$ such that $\mathcal{T}(\boldsymbol{x}) = [x_{\pi(1)}, \ldots, x_{\pi(d)}]^\top$. A permutating transformation is always a rotating transformation. We will use $\mathcal{R}$ to denote a rotating transformation.

Definition 2.1. *For initialization $\boldsymbol{x}_0$ and stochastic losses $\{L_t\}_{t=1}^{T}$, we can get $\boldsymbol{x}_t$ when running algorithm A on $(\boldsymbol{x}_0, \{L_t\}_{t=1}^{T})$. For a transformation $\mathcal{T}$, we can also get $\tilde{\boldsymbol{x}}_t$ when running A with the same hyperparameters on $(\tilde{\boldsymbol{x}}_0, \{\tilde{L}_t\}_{t=1}^{T})$ with $\tilde{\boldsymbol{x}}_0 = \mathcal{T}^{-1}(\boldsymbol{x}_0)$ and $\tilde{L}_t = L_t \circ \mathcal{T}$.*

An algorithm A is invariant w.r.t. $\mathcal{T}$ if it always holds that $\tilde{\boldsymbol{x}}_t = \mathcal{T}^{-1}(\boldsymbol{x}_t)$ for any hyperparameters, initialization and stochastic losses. An algorithm A is rotation invariant if it is invariant w.r.t. any rotating transformation $\mathcal{R}$. And A is permutation invariant if it is invariant w.r.t. any permutating transformation.

The following Theorem 2.2 shows the difference between `Adam` and `AdaSGD`, whose proof is in Appendix B.

Theorem 2.2. `SGD` *and* `AdaSGD` *are rotation-invariant.* `Adam` *and* `SignGD` *are permutation-invariant.*

3 MAIN RESULTS: CONVERGENCE RATES OF `Adam`

In this section, we present our main theoretical results, starting with a convergence analysis of `Adam` for stochastic smooth loss with coordinate-wise gradient noise (Theorem 3.5). We allow non-convex losses and thus the convergence is measured by the ℓ_1 norm of the gradient. For a deterministic loss, our best convergence rate (Theorem 3.2) is achieved by `SignGD` (`Adam` with $\beta_1 = \beta_2 = 0$). For a stochastic loss with bounded gradient noise variance, our best rate (Corollary 3.6) is achieved by `RMSProp` (`Adam` with $\beta_1 = 0$ and $\beta_2 \in [0, 1]$).

Then we extend our analysis of `Adam` to more general blockwise `Adam` (Theorem 3.12), which contains both `Adam` and `AdaSGD` as special cases. We also come up with novel smoothness measures (Definition 3.10) corresponding to the set of blocks used in blockwise `Adam`.

Similar to previous work (Défossez et al., 2022), our analysis could be extended to the most general case of `Adam`, where both β_1, β_2 are non-zero, but the rate becomes strictly worse than the `RMSProp` (the case of $\beta_1 = 0$), as there will be some extra polynomials of $\frac{1}{1-\beta_1}$. We decide not to include the result for the most general case, on one hand for ease of presentation, and on the other hand, because such result can not explain the optimization benefit of momentum ($\beta_1 > 0$) in practice and does not add any insight on the benefit of `Adam`. We hypothesize that we are missing some important features of loss landscape of transformers in the theoretical assumptions and we leave this for future work.

3.1 WARMUP: `SignGD` ($\beta_1 = \beta_2 = 0$)

In this section, we use the convergence analysis for `SignGD` (`Adam` with $\beta_1 = \beta_2 = 0$) as a warm-up and illustrate how `Adam` could benefit from a non-rotational invariant property of the loss landscape, which in particular is the ℓ_∞ smoothness. The key observation here is that `SignGD` is the normalized steepest descent with respect to ℓ_∞ norm (see (Xie & Li, 2024)), and thus it is more natural to analyze its convergence using ℓ_∞-norm-related geometry of the loss.

Definition 3.1. *Given a norm $\|\cdot\|$ on $\mathbb{R}^d$ and $\|\cdot\|_*$ as its dual norm, we say a function L is H-smooth w.r.t. $\|\cdot\|$ if for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, we have that $\|\nabla L(\boldsymbol{x}) - \nabla L(\boldsymbol{y})\|_* \leq H \|\boldsymbol{x} - \boldsymbol{y}\|$.*

Theorem 3.2. *Let L be a H-smooth with respect to $\|\cdot\|_\infty$ and $\{\boldsymbol{x}_t\}_{t=1}^T$ be the iterates of `SignGD` (`Adam` with $\beta_1 = \beta_2 = 0$) on L with initialization $\boldsymbol{x}_0$ and learning rate η, it holds that*

$$\min_{1\leq t\leq T} \|\nabla L(\boldsymbol{x}_t)\|_1 \leq \frac{L(\boldsymbol{x}_0) - \min L}{T\eta} + \frac{H\eta}{2}$$

if we choose $\eta = \sqrt{\frac{2(L(\boldsymbol{x}_0) - \min L)}{TH}}$, then $\min_{1\leq t\leq T} \|\nabla L(\boldsymbol{x}_t)\|_1 \leq \sqrt{\frac{2H(L(\boldsymbol{x}_0) - \min L)}{T}}$.

3.2 MAIN RESULT: `RMSProp` ($\beta_1 = 0, \beta_2 \in [0, 1]$)

It is well-known that `SignGD` might not converge in the stochastic case as the expectation of descent direction for mini-batch loss may not be a descent direction, and `RMSProp` is proposed to address this issue by using a moving average of the squared gradient per coordinate to reduce the coorleation between the denominator and the numerator, thus making the expected update direction less biased (Hinton et al., 2012). In this subsection we formalize the above intuition and show indeed a positive β_2 in `Adam` helps convergence in the stochastic case. The main challenges here are from both lower bounding the first-order term and upper bounding the second-order term in the modified descent lemma (the counterpart of Equation 1 for `RMSProp`).

$$L(\boldsymbol{x}_t) - L(\boldsymbol{x}_{t-1}) \leq -\eta_t \nabla L(\boldsymbol{x}_t)^\top \frac{\boldsymbol{g}_t}{\sqrt{\boldsymbol{v}_t + \epsilon}} + \frac{H}{2}\eta_t^2 \left\| \frac{\boldsymbol{g}_t}{\sqrt{\boldsymbol{v}_t + \epsilon}} \right\|_\infty^2$$

We can only upper bound $\left\| \frac{\boldsymbol{g}_t}{\sqrt{\boldsymbol{v}_t + \epsilon}} \right\|_\infty^2$ by $\frac{1}{1-\beta_2}$ without more fine-grained analysis on the relationship between gradients in each step, which will greatly hurt the dependence of convergence rate on $1-\beta_2$. However, even though the update at step t for one specific coordinate i can be as large as $\frac{1}{\sqrt{1-\beta_2}}$ with some very large $g_{t,i}$, the average moving speed for each coordinate should be close to 1. Therefore, we introduce a slightly stronger definition in Definition 3.3, which allows us to decompose the second order term into each coordinate according to Lemma 3.13. It also facilitates the analysis for the coordinate-wise first order term. We note this definition also appears in Assumption 2.3 of the concurrent work Maladkar et al. (2024).

Definition 3.3. *For any $\mathbf{H} = (H_1, \ldots, H_d) \in \mathbb{R}^d$, we say a function L is* $\mathbf{H}$-smooth coordinate-wisely w.r.t. ℓ_∞ norm , *iff for any $i \in [d]$, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, $|\nabla_i L(\boldsymbol{x}) - \nabla_i L(\boldsymbol{y})| \leq H_i \|\boldsymbol{x} - \boldsymbol{y}\|_\infty$.*

By definition, $\mathbf{H}$-smoothness coordinate-wisely w.r.t. ℓ_∞ norm implies $\sum_{i=1}^d H_i$ smoothness w.r.t. ℓ_∞ norm. We also need Assumption 3.4 to measure the influence of noise in stochastic setting.

Assumption 3.4 (Coordinate-wise noise)**.** *There exist constants σ_i such that*

$$\mathbb{E}\left(\nabla_i L_t(\boldsymbol{x}) - \nabla_i L(\boldsymbol{x})\right)^2 \leq \sigma_i^2$$

for any $i \in [d]$, $t \in \mathbb{N}$ and $\boldsymbol{x} \in \mathbb{R}^d$.

Due to the limitation of space, we only present the main result here. The sketch of the proof is presented in Section 3.4. We present the complete proof for the generalized blockwise Adam algorithm in Appendix C. The proof incorporates some key steps from Li & Lin (2024), extending them to accommodate the generalized algorithm and different smoothness assumptions.

Theorem 3.5 (Main, `Adam`)**.** *Let $\{L_t\}_{t=1}^T$ be independent stochastic losses satisfying Assumption 3.4 and that their expectation L is $\boldsymbol{H}$-coordinate-wisely smooth w.r.t. ℓ_∞ norm. For `Adam` with $\beta_1 = 0$, we have that*

$$\min_{\frac{T}{2} < t \leq T} \mathbb{E}\|\nabla L(\boldsymbol{x}_t)\|_1 \leq O\left(E + \sqrt{E}\sqrt{\frac{\beta_2^{\frac{T}{4}}}{T(1-\beta_2)} d v_0 + \sum_{i=1}^d \sigma_i + d\sqrt{\epsilon}} \right)$$

with

$$E = \frac{2}{\eta T}\mathbb{E}\left[L(\boldsymbol{x}_0) - L(\boldsymbol{x}_T)\right] + \left(1 + \frac{\beta_2 F}{T(1-\beta_2)}\right)\left(\eta\sum_{i=1}^{d} H_i + \sqrt{1-\beta_2}\sum_{i=1}^{d} \sigma_i\right)$$

and

$$F = \ln\left(1 + \frac{\sum_{i=1}^{d} \sigma_i^2 + \|\nabla L(\boldsymbol{x}_0)\|_\infty^2 + \max_{i\in[d]} H_i^2\eta^2 T(T+\frac{1}{1-\beta_2})}{v_0+\epsilon}\right)$$

We can determine the convergence rate of RMSprop by applying appropriate hyperparameters on Theorem 3.5. The optimal hyperparameters η and β_2 can be selected by minimizing E. We would assume that $v_0 + \epsilon > (\sum_{i=1}^{d}\sigma_i^2 + \|\nabla L(\boldsymbol{x}_0)\|_\infty^2 + \max_i H_i^2\eta^2)/\texttt{poly}(T)$ and $\frac{1}{1-\beta_2} = \texttt{poly}(T)$. Then we can simplify the term by considering $F = O(\log T)$.

The two terms involving $\sum_{i=1}^{d}\sigma_i$ have a lower bound $\Theta\left(\sum_{i=1}^{d}\sigma_i\left(\frac{\log T}{T}\right)^{\frac{1}{2}}\right)$, which can reached by $1-\beta_2 = \Theta\left(\frac{\log T}{T}\right)$. With this choice of $1-\beta_2$, the three terms involving η has a lower bound $\Theta\left(\sqrt{\frac{(L(\boldsymbol{x}_0)-\min_{\boldsymbol{x}} L(\boldsymbol{x}))\sum_{i=1}^{d} H_i}{T}}\right)$ reached by $\eta = \Theta\left(\sqrt{\frac{L(\boldsymbol{x}_0)-\min_{\boldsymbol{x}} L(\boldsymbol{x})}{T\sum_{i=1}^{d} H_i}}\right)$. Such choices of hyperparameters can give the optimal convergence rate for stochastic case in Corollary 3.6. For convenience, we define $R \triangleq (L(\boldsymbol{x}_0) - \min_{\boldsymbol{x}} L(\boldsymbol{x}))\sum_{i=1}^{d} H_i$, which will be the core term in Corollaries 3.6 to 3.8.

Corollary 3.6 (Stochastic Case, general σ_i)**.** *Let* $\{L_t\}_{t=1}^{T}$ *be independent stochastic losses satisfying Assumption 3.4 and that their expectation* L *is* $\boldsymbol{H}$*-coordinate-wisely smooth w.r.t.* ℓ_∞ *norm. For* $\beta_1 = 0$, $1-\beta_2 = \Theta(\frac{\log T}{T})$, $\epsilon = 0$, $\eta = \Theta\left(\sqrt{\frac{L(\boldsymbol{x}_0)-\min_{\boldsymbol{x}} L(\boldsymbol{x})}{T\sum_{i=1}^{d} H_i}}\right)$ *and* $v_0 > (\sum_{i=1}^{d}\sigma_i^2 + \|\nabla L(\boldsymbol{x}_0)\|_\infty^2 + \max_i H_i^2\eta^2)/\texttt{poly}(T)$*, we have that*

$$\min_{\frac{T}{2}<t\le T}\mathbb{E}\|\boldsymbol{g}_t\|_1 = O\left(\sqrt{\frac{R}{T}} + \sqrt{\frac{dv_0}{\texttt{poly}(T)} + \sum_{i=1}^{d}\sigma_i}\left[\left(\frac{R}{T}\right)^{\frac{1}{4}} + \sqrt{\sum_{i=1}^{d}\sigma_i}\left(\frac{\log T}{T}\right)^{\frac{1}{4}}\right]\right).$$

Even though the leading term w.r.t. T in the rate is $\left(\frac{\log T}{T}\right)^{\frac{1}{4}}$, its coefficient is $\sum_{i=1}^{d}\sigma_i$. It suggests that the rate can be much improved when noise is small. Below we get the convergence rate with the same hyperparameters in deterministic case in Corollary 3.7.

Corollary 3.7 (Deterministic Case, $\sigma_i = 0$)**.** *Let* $\{L_t\}_{t=1}^{T}$ *be deterministic losses satisfying Assumption 3.4 and that their expectation* L *is* $\boldsymbol{H}$*-coordinate-wisely smooth w.r.t.* ℓ_∞ *norm. For* $\beta_1 = 0$, $1-\beta_2 = \Theta(\frac{\log T}{T})$, $\epsilon = 0$, $\eta = \Theta\left(\sqrt{\frac{L(\boldsymbol{x}_0)-\min_{\boldsymbol{x}} L(\boldsymbol{x})}{T\sum_{i=1}^{d} H_i}}\right)$ *and* $v_0 > (\sum_{i=1}^{d}\sigma_i^2 + \|\nabla L(\boldsymbol{x}_0)\|_\infty^2 + \max_i H_i^2\eta^2)/\texttt{poly}(T)$ *for any polynomial* $\texttt{poly}(T)$*, we have that*

$$\min_{\frac{T}{2}<t\le T}\|\boldsymbol{g}_t\|_1 = O\left(\sqrt{\frac{R}{T}} + \sqrt{\frac{dv_0}{\texttt{poly}(T)}}\left(\frac{R}{T}\right)^{\frac{1}{4}}\right).$$

However, when $\sum_{i=1}^{d}\sigma_i = 0$, we have that $E = \frac{2}{\eta T}\mathbb{E}[L(\boldsymbol{x}_0) - L(\boldsymbol{x}_T)] + \eta\sum_{i=1}^{d} H_i\left(1 + \frac{\beta_2\log T}{(1-\beta_2)T}\right)$. Both E and the rate is a increasing function of β_2. So we should choose $\beta_2 = 0$ and $\eta = \Theta\left(\sqrt{\frac{L(\boldsymbol{x}_0)-\min_{\boldsymbol{x}} L(\boldsymbol{x})}{T\sum_{i=1}^{d} H_i}}\right)$. This will give the optimal convergence rate of deterministic case in Corollary 3.8. If we compare it with Corollary 3.7, the rate obtained by $1-\beta_2 = \Theta\left(\frac{\log T}{T}\right)$ is only slightly worse than the optimal rate.

Corollary 3.8 (Optimal Deterministic Case)**.** *Let* $\{L_t\}_{t=1}^{T}$ *be deterministic losses satisfying Assumption 3.4 and that their expectation* L *is* $\boldsymbol{H}$*-coordinate-wisely smooth w.r.t.* ℓ_∞ *norm. For* $\beta_1 = 0$, $\beta_2 = 0$, $\epsilon = 0$ *and* $\eta = \Theta\left(\sqrt{\frac{L(\boldsymbol{x}_0)-\min_{\boldsymbol{x}} L(\boldsymbol{x})}{T\sum_{i=1}^{d} H_i}}\right)$*, we have that* $\min_{\frac{T}{2}<t\le T}\|\boldsymbol{g}_t\|_1 = O\left(\sqrt{\frac{R}{T}}\right)$.

Algorithm 3 Blockwise `Adam`

```
Hyperparam: $\beta_1, \beta_2, \epsilon \geq 0$, block partition $\Phi : [d] \to [B]$, total steps $T$, learning rate schedule
    $\{\eta_t\}_{t=1}^{T}$, $\epsilon$, initial $\boldsymbol{m}_0$, $v_0$.
Input: initialization $\boldsymbol{x}_0$, stochastic loss functions $\{L_t\}_{t=1}^{T}$
    $v_{0,b} \leftarrow v_0$
    for $t = 1, 2, \cdots, T$ :
        $g_{t,i} \leftarrow \nabla_i L_t(\boldsymbol{x}_{t-1})$
        $m_{t,i} \leftarrow \beta_1 m_{t-1,i} + (1-\beta_1) g_{t,i}$
        $v_{t,b} \leftarrow \beta_2 v_{t-1,b} + (1-\beta_2)\left(\sum_{B(i)=b} g_{t,i}^2\right)/d_b$
        $x_{t,i} \leftarrow x_{t-1,i} - \eta_t \frac{m_{t,i}}{\sqrt{v_{t,B(i)}+\epsilon}}$
    return $\boldsymbol{x}_T$
```

Corollary 3.8 almost recovers Theorem 3.2, except the smoothness constant here $\sup_{\boldsymbol{x}} \|\nabla^2 L(\boldsymbol{x})\|_{(1,1)}$ is worse than that in Theorem 3.2, which is $\sup_{\boldsymbol{x}} \|\nabla^2 L(\boldsymbol{x})\|_{\infty}$, because it always holds that $\|\cdot\|_{1,1} \geq \|\cdot\|_{\infty}$. This gap is due to a technical difficulty of analyzing `Adam` or `RMSProp`, as mentioned in the beginning of Section 3.2.

Dependence on ϵ, v_0 and β_2. While many previous works rely on the relatively large magnitude of ϵ compared to $\boldsymbol{v}_t$ and give a bound in the regime of `SGD` when the adaptive effect is dominated by the constant ϵ (Zaheer et al., 2018; De et al., 2018), our result actually prefers ϵ to be 0 while maintaining the value of $v_0 + \epsilon$. We also note the dependence of our bound in Theorem 3.5 on v_0 is very mild and logarithmic. Theorem 3.5 has similar convergence rates for all v_0 of magnitude at most `poly`(T), while most previous result only addresses the case where $v_{0,i}$ is at the scale of noise (Li & Lin, 2024) or 0. The main reason for this adaptivity to a wide range of v_0 is our specific choice of $\beta_2 = 1 - \Theta(\frac{\log T}{T})$, which allows the initial large v_0 to decay fast and resume normal training. Other existing results using $\beta_2 = 1 - \Theta(1/T)$ (Défossez et al., 2022; Li & Lin, 2024) cannot allow large initial value v_0 because v_0 only decays a constant fraction throughout the training and the effective learning rate will be too small.

3.3 A UNIFIED ANALYSIS FOR BLOCKWISE `Adam`

In this subsection, we present convergence analysis for a broader class of adaptive algorithms defined in Algorithm 3, which could be thought as a coarser version of `Adam`. It does pre-conditioning blockwisely (specified by a partition function $\Phi : [d] \to [B]$ where B is the number of blocks) instead of coordinate-wisely. Since `Adam` and `AdaSGD` can be viewed as special cases of blockwise `Adam` (Algorithm 3) with $\Phi_{\texttt{Adam}} : i \mapsto i$ and $\Phi_{\texttt{AdaSGD}} : i \mapsto 1$ respectively, any convergence results for Algorithm 3 would imply convergence of `Adam` and `AdaSGD`. Finally we also note that such blockwise `Adam` has been recently studied empirically by some concurrent work, where the algorithm is named by Adam-mini (Zhang et al., 2024b) and Adalayer (Zhao et al., 2024).

We first introduce more notations. d_b denotes $|\{i|\Phi(i) = b\}|$, the number of parameters in block b. We define the vector $\boldsymbol{x}_{(b)}$ as $[x_i]_{\Phi(i)=b}$ and the submatrix $\mathbf{A}_{(b),(b')}$ as $[A_{i,j}]_{\Phi(i)=b,\Phi(j)=b'}$.

Definition 3.9 (Φ-norm). *We define the $(\infty, 2)$-norm w.r.t. partition Φ of vector $\boldsymbol{x}$ as the ℓ_∞ norm of the vector $\left(\frac{\|\boldsymbol{x}_{(b)}\|_2}{\sqrt{d_b}}\right)_{b=1}^{B}$, which is $\max_{b\in[B]} \frac{\|\boldsymbol{x}_{(b)}\|_2}{\sqrt{d_b}}$. For convenience, we will denote it by $\|\boldsymbol{x}\|_\Phi$ or just call it Φ-norm. We denote its dual norm by $\|\boldsymbol{x}\|_{\Phi,*}$, which is equal to $\sum_{b=1}^{B} \sqrt{d_b}\left\|\boldsymbol{x}_{(b)}\right\|_2$.*

Definition 3.10 (Generalized version of Definition 3.3). *For any partition function $\Phi : [d] \to [B]$ and $\mathbf{H} = (H_1, \ldots, H_B) \in \mathbb{R}^B$, we say a function L is* $\mathbf{H}$-smooth blockwisely w.r.t. Φ-norm, *iff for any $b \in [B]$, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$,*

$$\sqrt{d_b}\left\|\nabla_{(b)} L(\boldsymbol{x}) - \nabla_{(b)} L(\boldsymbol{y})\right\|_2 \leq H_b \|\boldsymbol{x} - \boldsymbol{y}\|_\Phi$$

We further define the Φ-smoothness of loss L by $H(L, \Phi) = \sum_{b=1}^{B} H_b$, where $\{H_b\}_{b=1}^{B}$ are the smallest numbers making L $\mathbf{H}$-smooth blockwisely in the above sense.

We note that the above defined blockwise Φ-smoothness is both a generalization of the coordinate-wise smoothness defined in Definition 3.1 (corresponding to the case of each block only containing

1 coordinate) and the standard ℓ_2 smoothness (corresponding to the case of only having one block). In the former case, we have $B = d$, $\Phi_{\texttt{Adam}}$ is the identity mapping $i \mapsto i$ and it holds that $H(L, i \mapsto i) \geq \sup_{\boldsymbol{x}\in\mathbb{R}^d} \left\|\nabla^2 L(\boldsymbol{x})\right\|_{1,1} \geq \sup_{\boldsymbol{x}\in\mathbb{R}^d} \left\|\nabla^2 L(\boldsymbol{x})\right\|_{\infty}$. In the latter case, we have $B = 1$, $\Phi_{\texttt{AdaSGD}}$ is the mapping $i \mapsto 1$ and $H(L, i \mapsto 1) = d \sup_{\boldsymbol{x}\in\mathbb{R}^d} \left\|\nabla^2 L(\boldsymbol{x})\right\|_2$.

Similar to the coordinate-wise case, one can show that $\sum_{b=1}^{B} H_b$ w.r.t. partition Φ is an upper bound for the smoothness of loss L w.r.t. Φ-norm.

Assumption 3.11 (Generalized version of Assumption 3.4)**.** *There exists constant σ_b such that $\mathbb{E}\left\|\nabla_{(b)} L_t(\boldsymbol{x}) - \nabla_{(b)} L(\boldsymbol{x})\right\|_2^2 \leq d_b \sigma_b^2$ for any block $b \in [B], t \in \mathbb{N}$ and $\boldsymbol{x} \in \mathbb{R}^d$.*

Theorem 3.12 (Main, Blockwise `Adam`)**.** *Under Assumption 3.11, suppose L is $\mathbf{H}$-smooth blockwisely w.r.t. Φ-norm, where $\mathbf{H} = (H_1, \dots, H_B) \in \mathbb{R}^B$, for Algorithm 3, we have that*

$$\min_{\frac{T}{2}<t\leq T} \mathbb{E}\sum_{b=1}^{B} \sqrt{d_b}\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2 \leq E + \sqrt{E}\sqrt{\frac{\beta_2^{\frac{T}{4}}}{T(1-\beta_2)} d\sqrt{v_0} + \sum_{b=1}^{B} d_b\sigma_b + d\sqrt{\epsilon}}$$

with

$$E = \frac{2}{\eta T}\mathbb{E}\left[L(\boldsymbol{x}_0) - L(\boldsymbol{x}_T)\right] + \left(1 + \frac{\beta_2 F}{T(1-\beta_2)}\right)\left(\eta\sum_{b=1}^{B} H_b + \sqrt{1-\beta_2}\sum_{b=1}^{B} d_b\sigma_b\right),$$

and

$$F = O\left(\ln\left(1 + \frac{\sum_{b=1}^{B}\sigma_b^2 + \|\nabla L(\boldsymbol{x}_0)\|_\Phi^2 + \max_{b\in[B]} H_b^2\eta^2 T(T + \frac{1}{1-\beta_2})}{v_0 + \epsilon}\right)\right).$$

(1,1)-norm as a surrogate complexity measure for $H(L, \Phi_{\texttt{Adam}})$. H_i in Definition 3.3 is determined by $\sup_{\boldsymbol{x}} \sum_{j=1}^{d}\left|\nabla_{i,j}^2 L(\boldsymbol{x})\right|$, which is difficult to compute because it requires taking supreme over the entire domain. A computationally-tractable alternative is to approximate $\sum_{i=1}^{d} H_i$ locally by the $(1,1)$-norm of Hessian of loss along the training trajectory. We provide an efficient approximation algorithm with guarantees by using hessian-vector product against random Cauchy vectors in Appendix D.2.

Different norms for smoothness. As an implication of Theorem 3.12, we immediately get analogs of Corollaries 3.6 to 3.8 for `AdaSGD`, with the corresponding new noise assumption and smoothness assumption. When the optimization is not noise-dominated, *i.e.*, the main term in the deterministic case $\sqrt{\frac{R}{T}} = \sqrt{\frac{(L(\boldsymbol{x}_0) - \min_{\boldsymbol{x}} L(\boldsymbol{x}))H(L,\Phi)}{T}}$ becomes the leading term, the choice of Φ now matters a lot. Here the biggest change from `AdaSGD` to `Adam` is the difference between $H(L, \Phi_{\texttt{AdaSGD}})$ and $H(L, \Phi_{\texttt{Adam}})$, which roughly correspond to $d \sup_{\boldsymbol{x}} \|\nabla^2 L(\boldsymbol{x})\|_2$ and $\sup_{\boldsymbol{x}} \|\nabla^2 L(\boldsymbol{x})\|_{1,1}$, using the above mentioned approximation.

Previous analyses of `Adam`'s convergence (Shi & Li, 2021; Défossez et al., 2022; Li & Lin, 2024) usually assume smoothness under the ℓ_2 norm. When using this assumption, the resulting convergence rate for `Adam` ends up being identical to the rate for `AdaSGD`, which fails to capture why `Adam` often performs better than `AdaSGD` in practice. By shifting to an ℓ_∞ norm smoothness assumption, we can observe the key difference: the coefficient for `Adam`'s convergence rate changes from $d \sup_{\boldsymbol{x}} \|\nabla^2 L(\boldsymbol{x})\|_2$ to $\sup_{\boldsymbol{x}} \|\nabla^2 L(\boldsymbol{x})\|_{1,1}$, where the latter is typically much smaller when `Adam` optimizes faster. This change leads to a divergence in the rates of the two algorithms, and comparing these new coefficients can provide insight into which algorithm may be more effective under different conditions.

Finally, we note that $\Phi_{\texttt{Adam}}$-smoothness $H(L, \Phi_{\texttt{Adam}})$ is not rotation-invariant in the sense that $H(L, \Phi_{\texttt{Adam}}) \neq H(L \circ \mathcal{R}, \Phi_{\texttt{Adam}})$ for a typical rotation $\mathcal{R}$. In practice, the $(1,1)$-norm of Hessian matrix can vary a lot when a rotation is performed on the loss as shown in Section 4.1. In contrast, $\Phi_{\texttt{AdaSGD}}$-smoothness $H(L, \Phi_{\texttt{AdaSGD}})$ is invariant under loss rotations.

	$(1,1)$-norm$/d$	Loss ($\beta_1 = \beta_2 = 0$)	Loss ($\beta_1 = 0.9, \beta_2 = 0.99$)
AdaSGD	0.00582	0.00881	0.00172
Adam	0.00582	0.00030	0.00001
Adam ($\mathcal{R}_1$)	0.04162	0.00317	0.00062
Adam ($\mathcal{R}_2$)	0.25364	0.00588	0.00122
Adam ($\mathcal{R}_3$)	0.61866	0.00747	0.00179
Adam ($\mathcal{R}_4$)	1.29959	0.00920	0.00239

Table 1: The final loss values obtained by different optimizers and the $(1, 1)$-norm of Hessian matrix for the corresponding unrotated objective and rotated objectives. The spectral norm of the Hessian matrix is always 1. `Adam` optimizes worse when the $(1, 1)$-norm of Hessian matrix increases, as suggested by our Corollary 3.7. Moreover, when $(1, 1)$-norm is smaller than spectral norm times space dimension, `Adam` tends to optimize faster than `AdaSGD` and vice versa, which justifies the effectiveness of Φ-smoothness as a tool to predict the optimization speed of blockwise `Adam`.

3.4 PROOF SKETCH OF THEOREM 3.12

We start by considering the decrease of $L(\boldsymbol{x}_t)$ in a single step t. By applying a Taylor expansion, we can upper bound the second order term with Lemma 3.13 using the smoothness assumption.

Lemma 3.13. *For any twice differentiable loss which is $\mathbf{H}$-smooth block-wisely w.r.t. Φ-norm (Definition 3.10), we have for any $\boldsymbol{x}$ and $\boldsymbol{\Delta} \in \mathbb{R}^d$, $\boldsymbol{\Delta}^\top \nabla^2 L(\boldsymbol{x}) \boldsymbol{\Delta} \le \sum_{b=1}^{B} \frac{H_b}{d_b} \left\| \boldsymbol{\Delta}_{(b)} \right\|_2^2$.*

Then we can get the decrease in a single step

$$L(\boldsymbol{x}_t) - L(\boldsymbol{x}_{t-1}) = -\eta \sum_{i=1}^{d} \frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{v_{t,\Phi(i)} + \epsilon}} + \frac{1}{2}\eta^2 \sum_{b=1}^{B} \frac{H_b}{d_b} \frac{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2}{v_{t,b} + \epsilon} = -\eta \sum_{b=1}^{B} \frac{\boldsymbol{g}_{t,(b)}^\top \bar{\boldsymbol{g}}_{t,(b)}}{\sqrt{v_{t,b} + \epsilon}} + \frac{1}{2}\eta^2 \sum_{b=1}^{B} \frac{H_b}{d_b} \frac{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2}{v_{t,b} + \epsilon}$$

At each step, the second order term $\frac{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2/d_b}{v_{t,b}+\epsilon}$ can be as large as $\frac{1}{1-\beta_2}$. But if we sum over all the steps, we can employ Lemma C.1 to bound the sum by $T + \frac{\beta_2}{1-\beta_2} \ln \frac{v_{T,b}+\epsilon}{v_{0,b}+\epsilon}$ rather than $\frac{T}{1-\beta_2}$.

The first order term has correlated denominator and nominator, making it hard to analyze its expectation. So we employ Lemma C.2 to replace it by $\sum_{b=1}^{B} \frac{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2}{\sqrt{\tilde{v}_{t,b}+\epsilon}}$ in which $\tilde{v}_{t,b} = \beta_2 v_{t-1,b} + (1-\beta_2)\left(\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b + \sigma_b^2\right)$ while sacrificing a constant factor and error terms related to noise magnitude σ_i.

Finally, we employ Cauchy inequality to upper bound $\left\|\bar{\boldsymbol{g}}_t\right\|_{\Phi,*}$

$$\left\|\bar{\boldsymbol{g}}_t\right\|_{\Phi,*} \le \sqrt{\sum_{b=1}^{B} \frac{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2}{\sqrt{\tilde{v}_{t,b}+\epsilon}}} \sqrt{\sum_{b=1}^{B} d_b \sqrt{\tilde{v}_{t,b}+\epsilon}}.$$

We use Lemma C.3 to upper bound $d_b\sqrt{\tilde{v}_{t,b}+\epsilon}$ by $\frac{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2}{\sqrt{\tilde{v}_{t,b}+\epsilon}}$. $\sum_{b=1}^{B} \frac{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2}{\sqrt{\tilde{v}_{t,b}+\epsilon}}$ can be bounded from analysis above, which finishes the proof.

4 EXPERIMENTS

In order to empirically investigate and confirm the implications of our propsosed theory, we compare the training performance of `Adam` with `AdaSGD`, `SGD` and `rotated Adam` on multiple different tasks. The details of performing rotation and computing matrix norms can be found in Appendix D.

4.1 QUADRATIC LOSS

We perform controlled experiments on quadratic loss to study the relationship between optimization speed of `Adam` and the shape of Hessian, in terms of $\Phi_{\texttt{Adam}}$-smoothness. More specifically, we consider $\Sigma = \text{diag}(\underbrace{1, \cdots, 1}_{10}, 1, \frac{1}{2^2}, \frac{1}{3^2}, \cdots, \frac{1}{990^2}) \in \mathbb{R}^{1000\times1000}$ and optimize the corresponding quadratic loss $\frac{1}{2}\boldsymbol{x}^\top \Sigma \boldsymbol{x}$ by `Adam`, with different levels of rotations. We manually generate orthogonal matrices $\mathcal{R}_i$ in the following way. We first sample $\mathcal{M} \in \mathbb{R}^{d\times d}$ where $\mathcal{M}_{i,j}$ is i.i.d. sampled from

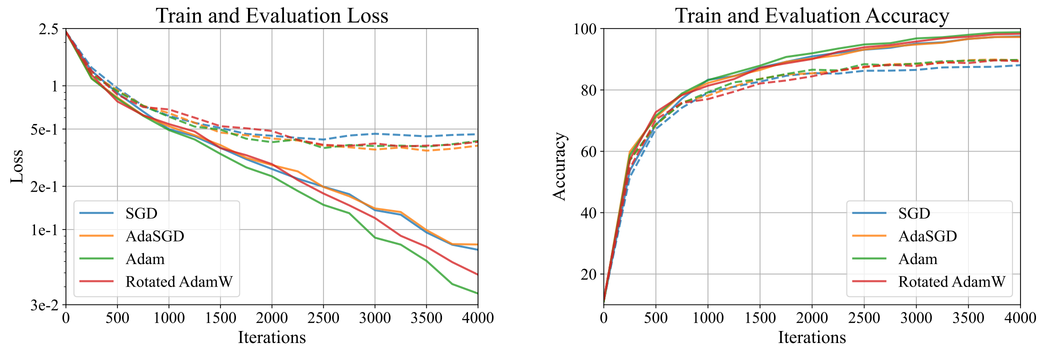

Figure 2: Training and test losses (left) and accuracies (right) of ResNet18 on CIFAR-10 with `Adam`, `AdaSGD`, `rotated Adam`, and `SGD`. We use batch size 256 and the optimal learning rate in terms of training loss from grid search. Solid and dashed lines correspond to the training and evaluation set metrics respectively. `Adam` converges faster than other algorithms.

$N(0,1)$. Then $\mathcal{A} = \mathcal{M} - \mathcal{M}^\top$ is a skew-symmetric matrix and $\exp(t\mathcal{A})$ represents a continuous family of matrices. We define $\mathcal{R}_i = \exp(t_i\mathcal{A})$ for different t_i. When $t_i = 0$, we know $\mathcal{R}_i = I$. When $t_i \to \infty$, $\mathcal{R}_i$ converges to a random orthogonal matrix in distribution.

Then we optimize $L_0(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top\Sigma\boldsymbol{x}$ with `AdaSGD` and `Adam` and optimize $L_i(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top\mathcal{R}_i^\top\Sigma\mathcal{R}_i\boldsymbol{x}$ with `Adam` for 100 steps. Because `AdaSGD` is rotational invariant, the optimization performance of `AdaSGD` is the same on all L_i. We tune learning rates for each setting and present their best results in Table 1.

We find a clear pattern that `Adam` optimizes worse when the $(1,1)$ norm of Hessian matrix increases, as suggested by our Corollary 3.7. Moreover, when $(1,1)$ norm divided by dimension is smaller than spectral norm, `Adam` tends to optimize faster than `AdaSGD` and vice versa, as suggested by our Theorem 3.12.

4.2 GPT-2 ON LANGUAGE MODELING TASK

We train GPT-2 on the OpenWebText corpus containing more than 9B tokens for 100k iterations. The training losses and evaluation losses of different optimizers are plotted in Figure 1. As mentioned in Section 1, `Adam` converges faster than `AdaSGD` while they both converge faster than `rotated Adam`. Since we propose the $(1,1)$-norm of Hessian as a non-rotation-invariant metric that can affect the convergence rate of `Adam`, we also measure it for original loss function L and rotated loss function $\tilde{L}$ on checkpoints trained with different losses. The results are presented in Table 2. The same correlation between norms and convergence rates holds here. The smaller the norm is, the faster the optimizer works.

	AdaSGD	Adam	Rotated Adam
Φ-smoothness Expression	$H(L, \Phi_{\texttt{AdaSGD}})$	$H(L, \Phi_{\texttt{Adam}})$	$H(L \circ \mathcal{R}, \Phi_{\texttt{Adam}})$
Φ-smoothness Estimation$/d$	24.86	3.2	36.16

Table 2: Hessian norms for the last GPT-2 checkpoints trained with different optimizers.

4.3 RESNET18 ON CIFAR-10

To further test whether the correlation between Φ-smoothness and the optimization performance holds for architectures other than transformers, we conduct a similar experiment on ResNet18 trained on CIFAR-10 (Krizhevsky, 2009). To do so, we first tuned each optimizer through searching over a grid of learning rates with weight decay being fixed to zero. More details can be found in Appendix D.3. Figure 2 depicts the loss and accuracy curves for the best performing hyperparameters chosen over the training set's final loss for batch size 256.[2] We also provide the results for other choices of batch size in Table 4.

When it comes to optimization speed, even for ResNet18, `Adam` is always better than `rotated Adam` and they are always better than `AdaSGD` and `SGD` across different batch sizes. Note that this does not contradict with common practice of training ResNet with `SGD`, where the main goal is to get

---

[2] We have intentionally limited the number of training iterations to emphasize the difference of optimizers in terms of training speed over generalization.

	AdaSGD	Adam	Rotated Adam
Φ-smoothness Expression	$H(L, \Phi_{\texttt{AdaSGD}})$	$H(L, \Phi_{\texttt{Adam}})$	$H(L \circ \mathcal{R}, \Phi_{\texttt{Adam}})$
Φ-smoothness Estimation$/d$	1.5355	0.0036	0.9868

Table 3: Hessian norms for optimal ResNet checkpoints trained with different optimizers and batch size 256.

better generalization and the training budget is large so all optimizers can easily achieve full training accuracy. In our experiment, we study optimization speed and intentionally limit the number of steps. We also measure the Hessian for checkpoints obtained at batch size 256 and the results are in Table 3. The correlation between norms and convergence rates still holds here. When the $(1, 1)$-norm divided by d is smaller than spectral norm, `Adam` optimizes faster than `AdaSGD`.

5 RELATED WORKS

Comparison between Adam and SGD Previous work tries to analyze the difference between `Adam` and `SGD` from different perspectives. Zhou et al. (2018) proves a faster convergence rate of `Adam` than `SGD` when the stochastic gradients are sparse. Zhang et al. (2020) suggests that `SGD` suffers more from heavy-tailed noise than `Adam`. Pan & Li (2023) claims that `Adam` has lower directional sharpness because of the effect of coordinate-wise clipping. Other works also consider the coordinate-wise normalization of `Adam` (Balles & Hennig, 2018; Kunstner et al., 2022). Kunstner et al. (2024) shows that the heavy-tailed class imbalance in language modeling tasks will cause `SGD` to converge slower when it can only optimize majority class well. Zhang et al. (2024a) finds that `Adam` is better at handling the block heterogeneity of Hessian matrix, which is a specific phenomenon in transformers. When viewing `Adam` as an adaptive method, there are works showing that adaptive methods have an advantage of achieving optimal convergence rate without relying on problem-dependent constant (Ward et al., 2020; Levy et al., 2021).

Convergence Rate of Adam There are many works showing convergence rate for `Adam` (Zhou et al., 2018; Chen et al., 2018; Zou et al., 2019; Shi & Li, 2021; Guo et al., 2021; Défossez et al., 2022; Zhang et al., 2022b). Most of them rely on the smoothness of the loss function, which is measured w.r.t. ℓ_2 norm. Zhang et al. (2019) proposes the (L_0, L_1) smoothness condition should be more reasonable than globally bounded smoothness. Li et al. (2024) further generalizes the (L_0, L_1) smoothness condition. However, they still focus on the default ℓ_2 norm which is rotation-invariant. To the best of our knowledge, we are the first to assume gradient Lipschitzness under ℓ_∞ norm for the analysis on `Adam`.

Comparison with Li & Lin (2024) Li & Lin (2024) employs the same ℓ_1 norm for gradient and improves the dependence on dimension d compared to previous results for ℓ_2 norm. But they still assume the common ℓ_2 norm smoothness while we adapt their results under ℓ_∞ norm smoothness to potentially further improve dependence on d. Another drawback of Li & Lin (2024) is setting $\boldsymbol{v}_0$ based on noise magnitude σ. which is impractical in real experiments because σ is unknown. Overestimation for σ will result in slow convergence because large $\boldsymbol{v}_0$ causes `Adam` to behave similarly with `SGD` without adjusting the coordinate-wise learning rate adaptively. In contrast, we allow for general initialization for $\boldsymbol{v}_0$ and our convergence rate can work well in both noisy setting and deterministic setting. We also use $1 - \beta_2 = \Theta\left(\log T/T\right)$ to obtain our convergence rate while Li & Lin (2024) requires $1 - \beta_2 = \Theta\left(1/T\right)$.

6 CONCLUSION

We give a new convergence analysis (Theorem 3.5) for `Adam` in the stochastic non-convex setting using a novel smoothness assumption. We show the convergence rate for the 1-norm of the gradient is $O(\frac{1}{\sqrt{T}})$ in the deterministic case (Corollaries 3.7 and 3.8) and $O(\frac{\log T}{T^{1/4}})$ in the stochastic case (Corollary 3.6). We also extend our analysis to blockwise `Adam` on loss L with respect to an arbitrary partition of the parameters Φ (Theorem 3.12) using the corresponding smoothness $H(L, \Phi)$ (Definition 3.10). Our bound for `Adam` involves $(1, 1)$-norm of Hessian, rather than the operator 2-norm of Hessian, which is relevant to the convergence speed of `AdaSGD`. This leads to significantly better smoothness conditions for deep learning models including ResNet-18 and GPT2 empirically. We also empirically verify that our smoothness measure $H(L, \Phi)$ positively correlates with the optimization speed of blockwise `Adam` with respect to the partition Φ.

REFERENCES

Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, 2023. URL `https://link.springer.com/article/10.1007/s10107-022-01822-7`. 1

Lukas Balles and Philipp Hennig. Dissecting adam: The sign, magnitude and variance of stochastic gradients. In *International Conference on Machine Learning*, 2018. URL `https://arxiv.org/pdf/1705.07774.pdf`. 10

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`. 1

Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. In *International Conference on Learning Representations*, 2018. URL `https://arxiv.org/pdf/1808.02941.pdf`. 10

Soham De, Anirbit Mukherjee, and Enayat Ullah. Convergence guarantees for rmsprop and adam in non-convex optimization and an empirical comparison to nesterov acceleration. *arXiv preprint arXiv:1807.06766*, 2018. URL `https://arxiv.org/pdf/1807.06766`. 6

Alexandre Défossez, Leon Bottou, Francis Bach, and Nicolas Usunier. A simple convergence proof of adam and adagrad. *Transactions on Machine Learning Research*, 2022. URL `https://arxiv.org/pdf/2003.02395.pdf`. 1, 3, 6, 7, 10

Zhishuai Guo, Yi Xu, Wotao Yin, Rong Jin, and Tianbao Yang. A novel convergence analysis for algorithms of the adam family. *arXiv preprint arXiv:2112.03459*, 2021. URL `https://arxiv.org/pdf/2112.03459.pdf`. 10

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 2012. URL `https://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf`. 4

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. URL `https://arxiv.org/pdf/2001.08361`. 1

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. URL `https://arxiv.org/pdf/1412.6980.pdf`. 1

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 9

Frederik Kunstner, Jacques Chen, Jonathan Wilder Lavington, and Mark Schmidt. Noise is not the main factor behind the gap between sgd and adam on transformers, but sign descent might be. In *The Eleventh International Conference on Learning Representations*, 2022. URL `https://arxiv.org/pdf/2304.13960.pdf`. 10

Frederik Kunstner, Robin Yadav, Alan Milligan, Mark Schmidt, and Alberto Bietti. Heavy-tailed class imbalance and why adam outperforms gradient descent on language models. *CoRR*, 2024. URL `https://arxiv.org/pdf/2402.19449`. 10

Kfir Levy, Ali Kavis, and Volkan Cevher. Storm+: Fully adaptive sgd with recursive momentum for nonconvex optimization. *Advances in Neural Information Processing Systems*, 2021. URL `https://proceedings.neurips.cc/paper/2021/file/ac10ff1941c540cd87c107330996f4f6-Paper.pdf`. 10

Haochuan Li, Alexander Rakhlin, and Ali Jadbabaie. Convergence of adam under relaxed assumptions. *Advances in Neural Information Processing Systems*, 2024. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/a3cc50126338b175e56bb3cad134db0b-Paper-Conference.pdf`. 10

Huan Li and Zhouchen Lin. On the $o(\frac{\sqrt{d}}{T^{1/4}})$ convergence rate of rmsprop and its momentum extension measured by ell_1 norm: Better dependence on the dimension. *arXiv preprint arXiv:2402.00389*, 2024. URL `https://arxiv.org/pdf/2402.00389`. 4, 6, 7, 10, 17

Devyani Maladkar, Ruichen Jiang, and Aryan Mokhtari. Convergence analysis of adaptive gradient methods under refined smoothness and noise assumptions. *arXiv preprint arXiv:2406.04592*, 2024. URL `https://arxiv.org/pdf/2406.04592`. 4

OpenAI. Gpt-4 technical report. *arXiv*, 2023. URL `https://arxiv.org/pdf/2303.08774`. 1

Yan Pan and Yuanzhi Li. Toward understanding why adam converges faster than sgd for transformers. *arXiv preprint arXiv:2306.00204*, 2023. URL `https://arxiv.org/pdf/2306.00204`. 10

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019. URL `https://insightcivic.s3.us-east-1.amazonaws.com/language-models.pdf`. 1

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. URL `https://arxiv.org/pdf/2403.05530`. 1

Naichen Shi and Dawei Li. Rmsprop converges with proper hyperparameter. In *International conference on learning representation*, 2021. URL `https://openreview.net/pdf?id=3UDSdyIcBDA`. 7, 10

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. URL `https://arxiv.org/pdf/2302.13971`. 1

Jiaxuan Wang and Jenna Wiens. Adasgd: Bridging the gap between sgd and adam. *arXiv preprint arXiv:2006.16541*, 2020. URL `https://arxiv.org/pdf/2006.16541`. 1

Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. *Journal of Machine Learning Research*, 2020. URL `https://www.jmlr.org/papers/volume21/18-352/18-352.pdf`. 10

Shuo Xie and Zhiyuan Li. Implicit bias of adamw: ℓ_∞ norm constrained optimization. *arXiv preprint arXiv:2404.04454*, 2024. URL `https://arxiv.org/pdf/2404.04454`. 4

Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. *Advances in neural information processing systems*, 2018. URL `https://proceedings.neurips.cc/paper/2018/file/90365351ccc7437a1309dc64e4db32a3-Paper.pdf`. 6

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2019. URL `https://arxiv.org/pdf/1905.11881`. 10

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 2020. URL `https://arxiv.org/pdf/1912.03194.pdf`. 10

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022a. URL `https://arxiv.org/pdf/2205.01068`. 1

Yushun Zhang, Congliang Chen, Naichen Shi, Ruoyu Sun, and Zhi-Quan Luo. Adam can converge without any modification on update rules. In *Advances in Neural Information Processing Systems*, 2022b. URL `https://arxiv.org/pdf/2208.09632.pdf`. 10

Yushun Zhang, Congliang Chen, Tian Ding, Ziniu Li, Ruoyu Sun, and Zhi-Quan Luo. Why transformers need adam: A hessian perspective. *arXiv preprint arXiv:2402.16788*, 2024a. URL `https://arxiv.org/pdf/2402.16788v1`. 10

Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenwei Wu, Yinyu Ye, Zhi-Quan Luo, and Ruoyu Sun. Adam-mini: Use fewer learning rates to gain more. *arXiv preprint arXiv:2406.16793*, 2024b. 6

Rosie Zhao, Depen Morwani, David Brandfonbrener, Nikhil Vyas, and Sham Kakade. Deconstructing what makes a good optimizer for language models. *arXiv preprint arXiv:2407.07972*, 2024. 6

Dongruo Zhou, Jinghui Chen, Yuan Cao, Yiqi Tang, Ziyan Yang, and Quanquan Gu. On the convergence of adaptive gradient methods for nonconvex optimization. *arXiv preprint arXiv:1808.05671*, 2018. URL `https://arxiv.org/pdf/1808.05671`. 10

Fangyu Zou, Li Shen, Zequn Jie, Weizhong Zhang, and Wei Liu. A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2019. URL `https://arxiv.org/pdf/1811.09358.pdf`. 10

A CONVERGENCE RATE OF `SignGD` FOR DETERMINISTIC LOSS

Proof of Theorem 3.2. We will directly prove a more general verion of Theorem 3.2. Because L is H-smooth with respect to $\|\cdot\|_\infty$, we have that

$$\begin{aligned} L(\boldsymbol{x}_{t+1}) - L(\boldsymbol{x}_t) &\leq -\nabla L(\boldsymbol{x}_t)^\top (\boldsymbol{x}_t - \boldsymbol{x}_{t+1}) + \frac{H}{2}\left\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\right\|^2 \\ &\leq -\eta \left\|\nabla L(\boldsymbol{x}_t)\right\|_* + \frac{\eta^2 H}{2}\eta^2 \end{aligned} \tag{1}$$

This implies that

$$\min_{1\leq t\leq T} \left\|\nabla L(\boldsymbol{x}_t)\right\|_* \leq \frac{1}{T}\sum_{t=1}^{T} \left\|\nabla L(\boldsymbol{x}_t)\right\|_* \leq \frac{L(\boldsymbol{x}_0) - L(\boldsymbol{x}_T)}{T\eta} + \frac{H\eta}{2},$$

which completes the proof. □

B INVARIANCE PROPERTY OF ADAM AND SGD

Theorem 2.2. `SGD` *and* `AdaSGD` *are rotation-invariant.* `Adam` *and* `SignGD` *are permutation-invariant.*

Proof of Theorem 2.2. For `SGD` and `AdaSGD`, we will show they are rotation-invariant by induction. For any rotating transformation $\mathcal{R}(\boldsymbol{x}) = \boldsymbol{R}\boldsymbol{x}$, suppose $\tilde{\boldsymbol{x}}_s = \mathcal{R}^{-1}(\boldsymbol{x}_s) = \boldsymbol{R}^\top \boldsymbol{x}_s$ holds for $s \leq t-1$. Then we have that $\tilde{\boldsymbol{g}}_t = \nabla_{\tilde{\boldsymbol{x}}} \tilde{L}_t(\tilde{\boldsymbol{x}}_t) = \boldsymbol{R}^\top \nabla_{\boldsymbol{x}} L(\boldsymbol{R}^{-1}\tilde{\boldsymbol{x}}_{t-1}) = \boldsymbol{R}^\top \nabla_{\boldsymbol{x}} L(\boldsymbol{x}_{t-1}) = \boldsymbol{R}^\top \boldsymbol{g}_t$ and $\tilde{\boldsymbol{m}}_t = \boldsymbol{R}^\top \boldsymbol{m}_t$. From the update rule of `SGD`, we have that $\tilde{\boldsymbol{x}}_t = \tilde{\boldsymbol{x}}_{t-1} - \eta_t \tilde{\boldsymbol{m}}_t = \boldsymbol{R}^\top \boldsymbol{x}_{t-1} - \eta_t \boldsymbol{R}^\top \boldsymbol{m}_t = \boldsymbol{R}^\top (\boldsymbol{x}_{t-1} - \eta_t \boldsymbol{m}_t) = \boldsymbol{R}^\top \boldsymbol{x}_t$. For the update rule of `AdaSGD`, we further have that $\|\tilde{\boldsymbol{g}}_t\|_2^2 = \|\boldsymbol{g}_t\|_2^2$ because $\boldsymbol{R}$ is an orthogonal matrix. Then $\tilde{v}_t = v_t$ and the derivation is similar.

For `Adam` and `SignGD`, it is easy to show by induction they are invariant w.r.t. any permutating transformation because the operation on gradient is performed on each coordinate separately. We only need to show they are not invariant w.r.t. a rotating transformation. We choose $\boldsymbol{R} = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}; \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]$, $L_t(\boldsymbol{x}) = L(\boldsymbol{x}) = 2x_1^2 + x_2^2$. Due to the update rule of `SignGD`, it can only update $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ in the direction of $[1,1]$ and $[1,-1]$. But when rotating the update direction on $\tilde{\boldsymbol{x}}$ back to the space of $\boldsymbol{x}$. The update direction can only be $[1,0]$ or $[0,1]$ that are different from the update direction in the original space. Because the first step in `Adam` takes the same direction in `SignGD`, we simultaneously show that both `SignGD` and `Adam` are not rotation-invariant. □

C PROOF DETAILS

C.1 PROOF FOR CONVERGENCE RATE OF BLOCKWISE ADAM

We also need new notations and different assumptions for specific partition $\Phi(\cdot)$. For specific block b, (b) is defined as $\Phi^{-1}(b) = \{i | \Phi(i) = b\}$ and $d_b = \#(b)$, the number of parameters in block b.

As mentioned in Section 3.4, we will use Lemma 3.13 to bound the second order term under smoothness Definition 3.10 and use Lemma C.1 to better control the growth of the sum of second order term.

Lemma 3.13. *For any twice differentiable loss which is* $\mathbf{H}$*-smooth block-wisely w.r.t.* Φ*-norm (Definition 3.10), we have for any* $\boldsymbol{x}$ *and* $\boldsymbol{\Delta} \in \mathbb{R}^d$, $\boldsymbol{\Delta}^\top \nabla^2 L(\boldsymbol{x}) \boldsymbol{\Delta} \le \sum_{b=1}^{B} \frac{H_b}{d_b} \left\| \boldsymbol{\Delta}_{(b)} \right\|_2^2$.

Proof of Lemma 3.13. From Definition 3.10, we know that

$$
\begin{aligned}
H_b &\geq \sup_{\boldsymbol{x}, \boldsymbol{\Delta}} \frac{\sqrt{d_b} \left\| \nabla_{(b)} L(\boldsymbol{x} + \boldsymbol{\Delta}) - \nabla_{(b)} L(\boldsymbol{x}) \right\|_2}{\max_{b' \in [B]} \frac{\left\| \boldsymbol{\Delta}_{(b')} \right\|_2}{\sqrt{d_{b'}}}} \\
&= \sup_{\boldsymbol{x}, \boldsymbol{\Delta}} \frac{\sqrt{d_b} \left\| \nabla^2_{(b),:} L(\boldsymbol{x}) \boldsymbol{\Delta} \right\|_2}{\max_{b' \in [B]} \frac{\left\| \boldsymbol{\Delta}_{(b')} \right\|_2}{\sqrt{d_{b'}}}} \\
&= \sup_{\boldsymbol{x}, \boldsymbol{\Delta}} \frac{\sqrt{d_b} \left\| \sum_{b'=1}^{B} \nabla^2_{(b),(b')} L(\boldsymbol{x}) \boldsymbol{\Delta}_{(b')} \right\|_2}{\max_{b' \in [B]} \frac{\left\| \boldsymbol{\Delta}_{(b')} \right\|_2}{\sqrt{d_{b'}}}} \\
&= \sup_{\boldsymbol{x}, \boldsymbol{\Delta}, \left\| \boldsymbol{\Delta}'_{(b)} \right\|_2 \le 1} \frac{\sqrt{d_b} \left\langle \boldsymbol{\Delta}'_{(b)}, \sum_{b'=1}^{B} \nabla^2_{(b),(b')} L(\boldsymbol{x}) \boldsymbol{\Delta}_{(b')} \right\rangle}{\max_{b' \in [B]} \frac{\left\| \boldsymbol{\Delta}_{(b')} \right\|_2}{\sqrt{d_{b'}}}} \\
&= \sup_{\boldsymbol{x}, \left\| \boldsymbol{\Delta}_{(b')} \right\|_2 \le \sqrt{d_{b'}}, \left\| \boldsymbol{\Delta}'_{(b)} \right\|_2 \le 1} \sqrt{d_b} \left\langle \boldsymbol{\Delta}'_{(b)}, \sum_{b'=1}^{B} \nabla^2_{(b),(b')} L(\boldsymbol{x}) \boldsymbol{\Delta}_{(b')} \right\rangle \\
&= \sup_{\boldsymbol{x}, \boldsymbol{\Delta}, \boldsymbol{\Delta}'} \sum_{b'=1}^{B} \frac{\sqrt{d_b} \sqrt{d_{b'}}}{\left\| \boldsymbol{\Delta}'_{(b)} \right\|_2 \left\| \boldsymbol{\Delta}_{(b')} \right\|_2} \left\langle \boldsymbol{\Delta}'_{(b)}, \nabla^2_{(b),(b')} L(\boldsymbol{x}) \boldsymbol{\Delta}_{(b')} \right\rangle
\end{aligned}
$$

Then for any $\boldsymbol{x}$ and $\boldsymbol{\Delta}$, we know that

$$
\begin{aligned}
\frac{H_b}{d_b} \left\| \boldsymbol{\Delta}_{(b)} \right\|_2^2 &\geq \frac{\left\| \boldsymbol{\Delta}_{(b)} \right\|_2^2}{d_b} \sum_{b'=1}^{B} \frac{\sqrt{d_b} \sqrt{d_{b'}}}{\left\| \boldsymbol{\Delta}_{(b)} \right\|_2 \left\| \boldsymbol{\Delta}_{(b')} \right\|_2} \left\langle \boldsymbol{\Delta}_{(b)}, \nabla^2_{(b),(b')} L(\boldsymbol{x}) \boldsymbol{\Delta}_{(b')} \right\rangle \\
&= \sum_{b'=1}^{B} \frac{\sqrt{d_{b'}} \left\| \boldsymbol{\Delta}_{(b)} \right\|_2}{\sqrt{d_b} \left\| \boldsymbol{\Delta}_{(b')} \right\|_2} \left\langle \boldsymbol{\Delta}_{(b)}, \nabla^2_{(b),(b')} L(\boldsymbol{x}) \boldsymbol{\Delta}_{(b')} \right\rangle
\end{aligned}
$$

and

$$
\begin{aligned}
&2 \sum_{b=1}^{B} \frac{H_b}{d_b} \left\| \boldsymbol{\Delta}_{(b)} \right\|_2^2 \\
=& \sum_{b=1}^{B} \frac{H_b}{d_b} \left\| \boldsymbol{\Delta}_{(b)} \right\|_2^2 + \sum_{b'=1}^{B} \frac{H_{b'}}{d_{b'}} \left\| \boldsymbol{\Delta}_{(b')} \right\|_2^2 \\
\geq& \sum_{b=1}^{B} \sum_{b'=1}^{B} \frac{\sqrt{d_{b'}} \left\| \boldsymbol{\Delta}_{(b)} \right\|_2}{\sqrt{d_b} \left\| \boldsymbol{\Delta}_{(b')} \right\|_2} \left\langle \boldsymbol{\Delta}_{(b)}, \nabla^2_{(b),(b')} L(\boldsymbol{x}) \boldsymbol{\Delta}_{(b')} \right\rangle + \sum_{b'=1}^{B} \sum_{b=1}^{B} \frac{\sqrt{d_b} \left\| \boldsymbol{\Delta}_{(b')} \right\|_2}{\sqrt{d_{b'}} \left\| \boldsymbol{\Delta}_{(b)} \right\|_2} \left\langle \boldsymbol{\Delta}_{(b')}, \nabla^2_{(b'),(b)} L(\boldsymbol{x}) \boldsymbol{\Delta}_{(b)} \right\rangle \\
\geq& 2 \sum_{b=1}^{B} \sum_{b'=1}^{B} \boldsymbol{\Delta}_{(b)}^\top \nabla^2_{(b),(b')} L(\boldsymbol{x}) \boldsymbol{\Delta}_{(b')} = 2 \boldsymbol{\Delta}^\top \nabla^2 L(\boldsymbol{x}) \boldsymbol{\Delta}.
\end{aligned}
$$

The last inequality comes from mean inequality. □

Lemma C.1. *Given any $0 < \beta_2 < 1$, suppose scalar sequences $\{v_t\}_{t=0}^T$ and $\{g_t\}_{t=1}^T$ satisfy that $v_0 \geq 0, v_1 > 0$ and $v_t - \beta_2 v_{t-1} \geq (1-\beta_2) g_t^2$ for $t \geq 1$. It holds that*

$$\sum_{t=1}^{T} \frac{g_t^2}{v_t} \leq T + \frac{\beta_2}{1-\beta_2} \ln \frac{v_T}{v_0}.$$

Proof of Lemma C.1. Notice that $1 - x \leq \ln \frac{1}{x}$ for any positive x. We can have that

$$\begin{aligned} \sum_{t=1}^{T} \frac{g_t^2}{v_t} &\leq \sum_{t=1}^{T} \frac{v_t - \beta_2 v_{t-1}}{(1-\beta_2) v_t} \\ &= \sum_{t=1}^{T} \left[1 + \frac{\beta_2}{1-\beta_2} \left(1 - \frac{v_{t-1}}{v_t} \right) \right] \\ &\leq T + \frac{\beta_2}{1-\beta_2} \sum_{t=1}^{T} \ln \frac{v_t}{v_{t-1}} \\ &= T + \frac{\beta_2}{1-\beta_2} \ln \frac{v_T}{v_0}. \end{aligned} \tag{2}$$

when $v_0 \neq 0$. When $v_0 = 0$, we can still have that

$$\begin{aligned} \sum_{t=1}^{T} \frac{g_t^2}{v_t} &\leq \frac{1}{1-\beta_2} + \sum_{t=2}^{T} \frac{g_t^2}{v_t} \\ &\leq \frac{1}{1-\beta_2} + (T-1) + \frac{\beta_2}{1-\beta_2} \ln \frac{v_T}{v_1} \\ &= T + \frac{\beta_2}{1-\beta_2} \ln \frac{v_T}{v_1/e}. \end{aligned}$$

□

Next we deal with the first order term by approximating it with a deterministic term. Here we need some new notations. Recall that $\boldsymbol{g}_t$ denotes the gradient of mini-batch $L_t(\boldsymbol{x}_{t-1})$ at step t. And $\mathbb{E}\left[\boldsymbol{g}_t | \boldsymbol{x}_{t-1}\right] = \nabla L(\boldsymbol{x}_{t-1})$ because $\mathbb{E} L_t = L$. The full-batch gradient is $\bar{\boldsymbol{g}}_t = \nabla L(\boldsymbol{x}_{t-1})$. Different kinds of second-order momentum are defined in the following way.

$$\begin{aligned} v_{t,b} &= \beta_2^t \left\| \boldsymbol{g}_{1,(b)} \right\|_2^2 / d_b + (1-\beta_2) \sum_{j=0}^{t-1} \beta_2^j \left(\left\| \boldsymbol{g}_{t-j,(b)} \right\|_2^2 \right) / d_b \\ \tilde{v}_{t,b} &= (1-\beta_2) \left(\left\| \bar{\boldsymbol{g}}_{t,(b)} \right\|_2^2 / d_b + \sigma_b^2 \right) + \beta_2 v_{t-1,b} \end{aligned}$$

Lemma C.2 (first-order approximation, no momentum)**.**

$$\mathbb{E} \sum_{t=1}^{T} \sum_{\Phi(i)=b} \frac{g_{t,i} \bar{g}_{t,i}}{\sqrt{v_{t,b} + \epsilon}} \geq \frac{1}{2} \mathbb{E} \sum_{t=1}^{T} \sum_{\Phi(i)=b} \frac{\bar{g}_{t,i}^2}{\sqrt{\tilde{v}_{t,b} + \epsilon}} - \sqrt{1-\beta_2} T d_b \sigma_b - \frac{d_b \sigma_b \beta_2}{\sqrt{1-\beta_2}} \mathbb{E}\left[\ln \frac{v_{T,b} + \epsilon}{v_{0,b} + \epsilon} \right].$$

Proof of Lemma C.2. The first order change can decomposed into two terms.

$$
\begin{aligned}
\mathbb{E}\sum_{t=1}^{T}\sum_{\Phi(i)=b}\frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{v_{t,b}+\epsilon}} &= \mathbb{E}\sum_{t=1}^{T}\sum_{\Phi(i)=b}\frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{\tilde{v}_{t,b}+\epsilon}} + \mathbb{E}\left[\sum_{t=1}^{T}\sum_{\Phi(i)=b}\frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{v_{t,b}+\epsilon}} - \frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{\tilde{v}_{t,b}+\epsilon}}\right] \\
&= \mathbb{E}\sum_{t=1}^{T}\sum_{\Phi(i)=b}\mathbb{E}\left[\frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{\tilde{v}_{t,b}+\epsilon}}\middle|\boldsymbol{x}_{t-1}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\sum_{\Phi(i)=b}\frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{v_{t,b}+\epsilon}} - \frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{\tilde{v}_{t,b}+\epsilon}}\right] \\
&= \mathbb{E}\sum_{t=1}^{T}\sum_{\Phi(i)=b}\frac{\bar{g}_{t,i}^2}{\sqrt{\tilde{v}_{t,b}+\epsilon}} + \mathbb{E}\left[\sum_{t=1}^{T}\sum_{\Phi(i)=b}\frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{v_{t,b}+\epsilon}} - \frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{\tilde{v}_{t,b}+\epsilon}}\right]
\end{aligned} \tag{3}
$$

For the second term, we have that

$$
\begin{aligned}
&\sum_{\Phi(i)=b}\left|g_{t,i}\bar{g}_{t,i}\left(\frac{1}{\sqrt{v_{t,b}+\epsilon}} - \frac{1}{\sqrt{\tilde{v}_{t,b}+\epsilon}}\right)\right| \\
=& \sum_{\Phi(i)=b}\frac{\left|g_{t,i}\bar{g}_{t,i}\left(\tilde{v}_{t,b}-v_{t,b}\right)\right|}{\sqrt{v_{t,b}+\epsilon}\sqrt{\tilde{v}_{t,b}+\epsilon}\left(\sqrt{v_{t,b}+\epsilon}+\sqrt{\tilde{v}_{t,b}+\epsilon}\right)} \\
=& \sum_{\Phi(i)=b}\frac{\left|g_{t,i}\bar{g}_{t,i}(1-\beta_2)\left(\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2-\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2/d_b\right)\right|}{\sqrt{v_{t,b}+\epsilon}\sqrt{\tilde{v}_{t,b}+\epsilon}\left(\sqrt{v_{t,b}+\epsilon}+\sqrt{\tilde{v}_{t,b}+\epsilon}\right)} \\
=& \sum_{\Phi(i)=b}\frac{\left|g_{t,i}\bar{g}_{t,i}(1-\beta_2)\left(\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2}+\sqrt{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2/d_b}\right)\left(\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2}-\sqrt{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2/d_b}\right)\right|}{\sqrt{v_{t,b}+\epsilon}\sqrt{\tilde{v}_{t,b}+\epsilon}\left(\sqrt{v_{t,b}+\epsilon}+\sqrt{\tilde{v}_{t,b}+\epsilon}\right)} \\
\leq& \sum_{\Phi(i)=b}\frac{\left|g_{t,i}\bar{g}_{t,i}\sqrt{1-\beta_2}\left(\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2}-\sqrt{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2/d_b}\right)\right|}{\sqrt{v_{t,b}+\epsilon}\sqrt{\tilde{v}_{t,b}+\epsilon}} \\
\leq& \frac{1}{2}\sum_{\Phi(i)=b}\frac{\bar{g}_{t,i}^2}{\sqrt{\tilde{v}_{t,b}+\epsilon}}\frac{\left(\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2}-\sqrt{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2/d_b}\right)^2}{\mathbb{E}[\left(\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2}-\sqrt{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2/d_b}\right)^2|\boldsymbol{x}_{t-1}]} \\
+& \frac{1}{2}\sum_{\Phi(i)=b}\frac{(1-\beta_2)g_{t,i}^2\mathbb{E}[\left(\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2}-\sqrt{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2/d_b}\right)^2|\boldsymbol{x}_{t-1}]}{(v_{t,b}+\epsilon)\sqrt{\tilde{v}_{t,b}+\epsilon}}
\end{aligned}
$$

The first inequality is because $v_{t,b}+\epsilon \geq (1-\beta_2)\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2/d_b$ and $\tilde{v}_{t,b}+\epsilon \geq (1-\beta_2)\left(\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2\right)$. For the first term, it will be exactly $\frac{1}{2}\frac{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2}{\sqrt{\tilde{v}_{t,b}+\epsilon}}$ after taking expectation conditional on $\boldsymbol{x}_{t-1}$. For the second term, we have the following inequality

$$
\begin{aligned}
&\mathbb{E}\left[\left(\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2}-\sqrt{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2/d_b}\right)^2\middle|\boldsymbol{x}_{t-1}\right]\\
=&\mathbb{E}\left[\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2+\sum_{\Phi(j)=b}g_{t,j}^2/d_b-2\sqrt{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2/d_b}\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_i^2}\middle|\boldsymbol{x}_{t-1}\right]\\
\le&2\left(\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2\right)-2\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2}\,\mathbb{E}\left[\sqrt{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2/d_b}\middle|\boldsymbol{x}_{t-1}\right]\\
\le&2\left(\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2\right)-2\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2}\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b}\\
=&2\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2}\left(\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2}-\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b}\right)\\
\le&2\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2}\,\sigma_b.
\end{aligned}
$$

The first inequality comes from Assumption 3.4. The second inequality is because ℓ_2 norm is a convex function. Then we know that

$$
\begin{aligned}
&\sum_{\Phi(i)=b}\frac{(1-\beta_2)g_{t,i}^2\mathbb{E}\left[\left(\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2}-\sqrt{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2/d_b}\right)^2|\boldsymbol{x}_{t-1}\right]}{(v_{t,b}+\epsilon)\sqrt{\tilde{v}_{t,b}+\epsilon}}\\
\le&\sum_{\Phi(i)=b}\frac{(1-\beta_2)g_{t,i}^2 2\sqrt{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b+\sigma_b^2}\,\sigma_b}{(v_{t,b}+\epsilon)\sqrt{\tilde{v}_{t,b}+\epsilon}}\\
\le&2\sqrt{1-\beta_2}\sigma_b\sum_{\Phi(i)=b}\frac{g_{t,i}^2}{v_{t,b}+\epsilon}.
\end{aligned}
$$

Then back to Equation 3, we have that

$$
\begin{aligned}
\mathbb{E}\sum_{t=1}^T\sum_{\Phi(i)=b}\frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{v_{t,b}+\epsilon}}&=\mathbb{E}\sum_{t=1}^T\sum_{\Phi(i)=b}\frac{\bar{g}_{t,i}^2}{\sqrt{\tilde{v}_{t,b}+\epsilon}}+\mathbb{E}\left[\sum_{t=1}^T\sum_{\Phi(i)=b}\frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{v_{t,b}+\epsilon}}-\frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{\tilde{v}_{t,b}+\epsilon}}\right]\\
&\ge\mathbb{E}\sum_{t=1}^T\sum_{\Phi(i)=b}\frac{\bar{g}_{t,i}^2}{\sqrt{\tilde{v}_{t,b}+\epsilon}}-\frac{1}{2}\mathbb{E}\sum_{t=1}^T\sum_{\Phi(i)=b}\frac{\bar{g}_{t,i}^2}{\sqrt{\tilde{v}_{t,b}+\epsilon}}-\frac{1}{2}2\sqrt{1-\beta_2}\sigma_b\mathbb{E}\sum_{t=1}^T\frac{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2}{v_{t,b}+\epsilon}\\
&=\frac{1}{2}\mathbb{E}\sum_{t=1}^T\sum_{\Phi(i)=b}\frac{\bar{g}_{t,i}^2}{\sqrt{\tilde{v}_{t,b}+\epsilon}}-\sqrt{1-\beta_2}\sigma_b\mathbb{E}\sum_{t=1}^T\frac{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2}{v_{t,b}+\epsilon}
\end{aligned}
$$

For the second term, we can apply Lemma C.1 and get that

$$
\sum_{t=1}^T\frac{\sum_{\Phi(i)=b}g_{t,i}^2/d_b}{v_{t,b}+\epsilon}\le T+\frac{\beta_2}{1-\beta_2}\ln\frac{v_{T,b}+\epsilon}{v_{0,b}+\epsilon}.
$$

Combining these two terms, we can get that

$$
\mathbb{E}\sum_{t=1}^T\sum_{\Phi(i)=b}\frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{v_{t,b}+\epsilon}}\ge\frac{1}{2}\mathbb{E}\sum_{t=1}^T\sum_{\Phi(i)=b}\frac{\bar{g}_{t,i}^2}{\sqrt{\tilde{v}_{t,b}+\epsilon}}-\sqrt{1-\beta_2}Td_b\sigma_b-\frac{d_b\sigma_b\beta_2}{\sqrt{1-\beta_2}}\mathbb{E}\left[\ln\frac{v_{T,b}+\epsilon}{v_{0,b}+\epsilon}\right].
$$

□

Next we need Lemma C.3 to deal with the denominator in the approximated first order term. The lemma is largely inspired by Lemma 6 in Li & Lin (2024), where we further generalize it to the case of block-wise `Adam`.

Lemma C.3. *For any $b \in [B]$, it holds that*

$$\sum_{t=\frac{T}{2}+1}^{T} \mathbb{E}\left[\sqrt{\tilde{v}_{t,b}+\epsilon}\right] \leq \frac{2\beta_2^{\frac{T}{4}}}{1-\beta_2}\sqrt{v_{0,b}} + \frac{T}{2}\sigma_b + \frac{T}{2}\sqrt{\epsilon} + 2\sum_{t=1}^{T}\mathbb{E}\left[\frac{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b}{\sqrt{\tilde{v}_{t,b}+\epsilon}}\right].$$

Proof of Lemma C.3. For each $t \leq T$, we have that

$$\begin{aligned}
&\mathbb{E}\left[\sqrt{\tilde{v}_{t,b}+\epsilon}\right]\\
=&\mathbb{E}\left[\sqrt{\beta_2 v_{t-1,b} + (1-\beta_2)(\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b + \sigma_b^2) + \epsilon}\right]\\
=&\mathbb{E}\left[\frac{\beta_2 v_{t-1,b} + (1-\beta_2)\sigma_b^2 + \epsilon}{\sqrt{\beta_2 v_{t-1,b} + (1-\beta_2)(\sum_{\Phi(i)=b}\bar{g}_{t,i}^2/d_b + \sigma_b^2) + \epsilon}}\right] + (1-\beta_2)\mathbb{E}\left[\frac{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b}{\sqrt{\tilde{v}_{t,b}+\epsilon}}\right]\\
\leq&\mathbb{E}\left[\sqrt{\beta_2 v_{t-1,b} + (1-\beta_2)\sigma_b^2 + \epsilon}\right] + (1-\beta_2)\mathbb{E}\left[\frac{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b}{\sqrt{\tilde{v}_{t,b}+\epsilon}}\right].
\end{aligned}$$

And for each $s \leq t-1$, we have that

$$\begin{aligned}
&\mathbb{E}\left[\sqrt{\beta_2^s v_{t-s,b} + (1-\beta_2^s)\sigma_b^2 + \epsilon}\right]\\
=&\mathbb{E}\left[\sqrt{\beta_2^{s+1} v_{t-s-1,b} + \beta_2^s(1-\beta_2)\sum_{\Phi(i)=b} g_{t-s,i}^2/d_b + (1-\beta_2^s)\sigma_b^2 + \epsilon}\right]\\
=&\mathbb{E}\left[\mathbb{E}\left[\sqrt{\beta_2^{s+1} v_{t-s-1,b} + \beta_2^s(1-\beta_2)\sum_{\Phi(i)=b} g_{t-s,i}^2/d_b + (1-\beta_2^s)\sigma_b^2 + \epsilon}\,\middle|\,\boldsymbol{x}_{t-s-1}\right]\right]\\
\leq&\mathbb{E}\left[\sqrt{\beta_2^{s+1} v_{t-s-1,b} + \beta_2^s(1-\beta_2)\mathbb{E}\left[\sum_{\Phi(i)=b} g_{t-s,i}^2/d_b\,\middle|\,\boldsymbol{x}_{t-s-1}\right] + (1-\beta_2^s)\sigma_b^2 + \epsilon}\right]\\
\leq&\mathbb{E}\left[\sqrt{\beta_2^{s+1} v_{t-s-1,b} + \beta_2^s(1-\beta_2)\sum_{\Phi(i)=b} \bar{g}_{t-s,i}^2/d_b + (1-\beta_2^{s+1})\sigma_b^2 + \epsilon}\right]\\
=&\mathbb{E}\left[\frac{\beta_2^{s+1} v_{t-s-1,b} + (1-\beta_2^{s+1})\sigma_b^2 + \epsilon}{\sqrt{\beta_2^{s+1} v_{t-s-1,b} + \beta_2^s(1-\beta_2)\sum_{\Phi(i)=b} \bar{g}_{t-s,i}^2/d_b + (1-\beta_2^{s+1})\sigma_b^2 + \epsilon}}\right]\\
+&\mathbb{E}\left[\frac{\beta_2^s(1-\beta_2)\sum_{\Phi(i)=b} \bar{g}_{t-s,i}^2/d_b}{\sqrt{\beta_2^{s+1} v_{t-s-1,b} + \beta_2^s(1-\beta_2)\sum_{\Phi(i)=b} \bar{g}_{t-s,i}^2/d_b + (1-\beta_2^{s+1})\sigma_b^2 + \epsilon}}\right]\\
\leq&\mathbb{E}\left[\sqrt{\beta_2^{s+1} v_{t-s-1,b} + (1-\beta_2^{s+1})\sigma_b^2 + \epsilon}\right] + \sqrt{\beta_2^s}(1-\beta_2)\mathbb{E}\left[\frac{\sum_{\Phi(i)=b}\bar{g}_{t-s,i}^2/d_b}{\sqrt{\tilde{v}_{t-s,b}+\epsilon}}\right].
\end{aligned}$$

By summing the above inequality over $s = 1, \cdots, t-1$, we have that

$$\begin{aligned}
&\mathbb{E}\left[\sqrt{\beta_2 v_{t-1,b} + (1-\beta_2)\sigma_b^2 + \epsilon}\right]\\
\leq&\mathbb{E}\left[\sqrt{\beta_2^t v_{0,b} + (1-\beta_2^t)\sigma_b^2 + \epsilon}\right] + \sum_{s=1}^{t-1}\sqrt{\beta_2^s}(1-\beta_2)\mathbb{E}\left[\frac{\sum_{\Phi(i)=b}\bar{g}_{t-s,i}^2/d_b}{\sqrt{\tilde{v}_{t-s,b}+\epsilon}}\right]\\
\leq&\sqrt{\beta_2^t v_{0,b}} + \sqrt{\sigma_b^2+\epsilon} + \sum_{s=1}^{t-1}\sqrt{\beta_2^s}(1-\beta_2)\mathbb{E}\left[\frac{\sum_{\Phi(i)=b}\bar{g}_{t-s,i}^2/d_b}{\sqrt{\tilde{v}_{t-s,b}+\epsilon}}\right].
\end{aligned}$$

and

$$\mathbb{E}\left[\sqrt{\tilde{v}_{t,b}+\epsilon}\right] \le \sqrt{\beta_2^t v_{0,b}} + \sqrt{\sigma_b^2+\epsilon} + \sum_{s=0}^{t-1} \sqrt{\beta_2^s}(1-\beta_2)\mathbb{E}\left[\frac{\sum_{\Phi(i)=b} \bar{g}_{t-s,i}^2/d_b}{\sqrt{\tilde{v}_{t-s,b}+\epsilon}}\right].$$

By summing the above inequality over $t = \frac{T}{2}+1, \cdots, T$, we have that

$$\begin{aligned}
\sum_{t=\frac{T}{2}+1}^{T}\left[\sqrt{\tilde{v}_{t,b}+\epsilon}\right] &\le \sum_{t=\frac{T}{2}+1}^{T} \sqrt{\beta_2^t v_{0,b}} + \frac{T}{2}\sqrt{\sigma_b^2+\epsilon} + \sum_{t=\frac{T}{2}+1}^{T}\sum_{s=0}^{t-1} \sqrt{\beta_2^s}(1-\beta_2)\mathbb{E}\left[\frac{\sum_{\Phi(i)=b} \bar{g}_{t-s,i}^2/d_b}{\sqrt{\tilde{v}_{t-s,b}+\epsilon}}\right] \\
&\le \frac{\beta_2^{\frac{T}{4}}}{1-\sqrt{\beta_2}}\sqrt{v_{0,b}} + \frac{T}{2}\sqrt{\sigma_b^2+\epsilon} + \frac{1-\beta_2}{1-\sqrt{\beta_2}}\sum_{t=1}^{T}\mathbb{E}\left[\frac{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b}{\sqrt{\tilde{v}_{t,b}+\epsilon}}\right] \\
&= \frac{\beta_2^{\frac{T}{4}}}{1-\sqrt{\beta_2}}\sqrt{v_{0,b}} + \frac{T}{2}\sqrt{\sigma_b^2+\epsilon} + (1+\sqrt{\beta_2})\sum_{t=1}^{T}\mathbb{E}\left[\frac{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b}{\sqrt{\tilde{v}_{t,b}+\epsilon}}\right] \\
&\le \frac{2\beta_2^{\frac{T}{4}}}{1-\beta_2}\sqrt{v_{0,b}} + \frac{T}{2}\sigma_b + \frac{T}{2}\sqrt{\epsilon} + 2\sum_{t=1}^{T}\mathbb{E}\left[\frac{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2/d_b}{\sqrt{\tilde{v}_{t,b}+\epsilon}}\right].
\end{aligned}$$

□

This following Lemma C.4 is to control the growth of $v_{T,b}$ so that the right hand side in Lemma C.1 is indeed $\Theta\left(T+\frac{\log T}{1-\beta_2}\right)$ instead of $\Theta(T)$ when all the constants are poly(T).

Lemma C.4. *For any T, it holds that*

$$\begin{aligned}
\mathbb{E}\max_{b\in[B]} v_{T,b}+\epsilon \le{}& 2\epsilon + 2v_0 + 4\sum_{b=1}^{B}\sigma_b^2 + 8\max_{b\in[B]}\left\|\nabla_{(b)}L(\boldsymbol{x}_0)\right\|_2^2/d_b + 8\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T^2 \\
&+ 8\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T\frac{\beta_2}{1-\beta_2}\ln 4\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T\frac{\beta_2}{(1-\beta_2)(v_0+\epsilon)}.
\end{aligned}$$

Proof of Lemma C.4. From the definition of $v_{t,i}$ and Assumption 3.4, we have that

$$
\begin{aligned}
&\mathbb{E}\max_{b\in[B]} v_{t,b}\\
=&\mathbb{E}\max_{b\in[B]}\left[\beta_2^t v_{0,b}+(1-\beta_2)\sum_{s=1}^{t}\beta_2^{t-s}\left\|\boldsymbol{g}_{s,(b)}\right\|_2^2/d_b\right]\\
\le&\beta_2^t\left\|\boldsymbol{v}_0\right\|_\infty+(1-\beta_2)\mathbb{E}\max_{b\in[B]}\sum_{s=1}^{t}\beta_2^{t-s}\left\|\boldsymbol{g}_{s,(b)}\right\|_2^2/d_b\\
=&\beta_2^t\left\|\boldsymbol{v}_0\right\|_\infty+(1-\beta_2)\mathbb{E}\max_{b\in[B]}\sum_{s=1}^{t}\beta_2^{t-s}\left\|\mathbb{E}[\boldsymbol{g}_{s,(b)}|\boldsymbol{x}_{s-1}]+\boldsymbol{g}_{s,(b)}-\mathbb{E}[\boldsymbol{g}_{s,(b)}|\boldsymbol{x}_{s-1}]\right\|_2^2/d_b\\
\le&\beta_2^t\left\|\boldsymbol{v}_0\right\|_\infty+(1-\beta_2)\mathbb{E}\max_{b\in[B]}\sum_{s=1}^{t}\beta_2^{t-s}\left[2\left\|\mathbb{E}[\boldsymbol{g}_{s,(b)}|\boldsymbol{x}_{s-1}]\right\|_2^2+2\left\|\boldsymbol{g}_{s,(b)}-\mathbb{E}[\boldsymbol{g}_{s,(b)}|\boldsymbol{x}_{s-1}]\right\|_2^2\right]/d_b\\
\le&\beta_2^t\left\|\boldsymbol{v}_0\right\|_\infty+2(1-\beta_2)\mathbb{E}\sum_{b=1}^{B}\sum_{s=1}^{t}\beta_2^{t-s}\left\|\boldsymbol{g}_{s,(b)}-\mathbb{E}[\boldsymbol{g}_{s,(b)}|\boldsymbol{x}_{s-1}]\right\|_2^2/d_b+2(1-\beta_2)\mathbb{E}\max_{b\in[B]}\sum_{s=1}^{t}\beta_2^{t-s}\left\|\nabla_{(b)}L(\boldsymbol{x}_{s-1})\right\|_2^2/d_b\\
\le&\beta_2^t\left\|\boldsymbol{v}_0\right\|_\infty+2(1-\beta_2^t)\sum_{b=1}^{B}\sigma_b^2+2(1-\beta_2)\mathbb{E}\max_{b\in[B]}\sum_{s=1}^{t}\beta_2^{t-s}\left[2\left\|\nabla_{(b)}L(\boldsymbol{x}_0)\right\|_2^2+2\left\|\nabla_{(b)}L(\boldsymbol{x}_{s-1})-\nabla_{(b)}L(\boldsymbol{x}_0)\right\|_2^2\right]/d_b\\
\le&\beta_2^t\left\|\boldsymbol{v}_0\right\|_\infty+2(1-\beta_2^t)\sum_{b=1}^{B}\sigma_b^2+4(1-\beta_2^t)\max_{b\in[B]}\left\|\nabla_{(b)}L(\boldsymbol{x}_0)\right\|_2^2/d_b\\
&+4(1-\beta_2)\mathbb{E}\max_{b\in[B]}\sum_{s=1}^{t}\beta_2^{t-s}\left\|\nabla_{(b)}L(\boldsymbol{x}_{s-1})-\nabla_{(b)}L(\boldsymbol{x}_0)\right\|_2^2/d_b\\
\le&\beta_2^t\left\|\boldsymbol{v}_0\right\|_\infty+2(1-\beta_2^t)\sum_{b=1}^{B}\sigma_b^2+4(1-\beta_2^t)\max_{b\in[B]}\left\|\nabla_{(b)}L(\boldsymbol{x}_0)\right\|_2^2/d_b\\
&+4(1-\beta_2)\mathbb{E}\max_{b\in[B]}\sum_{s=1}^{t}\beta_2^{t-s}\frac{H_b^2}{d_b^2}\max_{b'\in[B]}\frac{\left\|\boldsymbol{x}_{s-1,(b')}-\boldsymbol{x}_{0,(b')}\right\|_2^2}{d_{b'}}\\
\le&\beta_2^t\left\|\boldsymbol{v}_0\right\|_\infty+2\sum_{b=1}^{B}\sigma_b^2+4\max_{b\in[B]}\left\|\nabla_{(b)}L(\boldsymbol{x}_0)\right\|_2^2/d_b+4(1-\beta_2)(\max_{b\in[B]}\frac{H_b^2}{d_b^2})\mathbb{E}\sum_{s=1}^{t}\beta_2^{t-s}\max_{b'\in[B]}\frac{\left\|\boldsymbol{x}_{s-1,(b')}-\boldsymbol{x}_{0,(b')}\right\|_2^2}{d_{b'}}.
\end{aligned}
$$

We define $C=v_0+2\sum_{b=1}^{B}\sigma_b^2+4\max_{b\in[B]}\left\|\nabla_{(b)}L(\boldsymbol{x}_0)\right\|_2^2/d_b$ for simplicity. From Lemma C.1 and Cauchy inequality, we know that

$$
\begin{aligned}
\frac{1}{d_{b'}}\left\|\boldsymbol{x}_{t,(b')}-\boldsymbol{x}_{0,(b')}\right\|_2^2=&\frac{\eta^2}{d_{b'}}\sum_{\Phi(j)=b'}\left|\sum_{s=1}^{t}\frac{g_{s,j}}{\sqrt{v_{s,b'}+\epsilon}}\right|^2\\
\le&\frac{\eta^2}{d_{b'}}\sum_{\Phi(j)=b'}t\sum_{s=1}^{t}\frac{g_{s,j}^2}{v_{s,b'}+\epsilon}\\
=&\eta^2 t\sum_{s=1}^{t}\frac{\sum_{\Phi(j)=b'}g_{s,j}^2/d_{b'}}{v_{s,b'}+\epsilon}\\
\le&\eta^2 t\left(t+\frac{\beta_2}{1-\beta_2}\ln\frac{v_{t,b'}+\epsilon}{v_{0,b'}+\epsilon}\right)\\
\le&\eta^2t^2+\eta^2t\frac{\beta_2}{1-\beta_2}\ln\frac{\max_{b\in[B]}v_{t,b}+\epsilon}{v_0+\epsilon}
\end{aligned}
$$

The right hand side is independent of specific block b. So we can get that

$$
\max_{b'\in[B]}\frac{\left\|\boldsymbol{x}_{t,(b')}-\boldsymbol{x}_{0,(b')}\right\|_2^2}{d_{b'}}\le\eta^2t^2+\eta^2t\frac{\beta_2}{1-\beta_2}\ln\frac{\max_{b\in[B]}v_{t,b}+\epsilon}{v_0+\epsilon}
$$

and

$$\begin{aligned}
&\mathbb{E}\max_{b'\in[B]}\frac{\left\|\boldsymbol{x}_{t,(b')}-\boldsymbol{x}_{0,(b')}\right\|_2^2}{d_{b'}}\\
\leq&\eta^2t^2+\eta^2t\frac{\beta_2}{1-\beta_2}\mathbb{E}\ln\frac{\max_{b\in[B]}v_{t,b}+\epsilon}{v_0+\epsilon}\\
\leq&\eta^2t^2+\eta^2t\frac{\beta_2}{1-\beta_2}\ln\frac{\mathbb{E}\max_{b\in[B]}v_{t,b}+\epsilon}{v_0+\epsilon}\\
\leq&\eta^2t^2+\eta^2t\frac{\beta_2}{1-\beta_2}\ln\frac{\epsilon+C+4(1-\beta_2)\max_{b\in[B]}\frac{H_b^2}{d_b^2}\mathbb{E}\sum_{s=1}^t\beta_2^{t-s}\max_{b'\in[B]}\frac{\left\|\boldsymbol{x}_{s-1,(b')}-\boldsymbol{x}_{0,(b')}\right\|_2^2}{d_{b'}}}{v_0+\epsilon}.
\end{aligned}$$

Define $G=\max_{1\leq t\leq T}\mathbb{E}\max_{b'\in[B]}\frac{\left\|\boldsymbol{x}_{t,(b')}-\boldsymbol{x}_{0,(b')}\right\|_2^2}{d_{b'}}$. There exists $t\leq T$ such that

$$\begin{aligned}
G=&\mathbb{E}\max_{b'\in[B]}\frac{\left\|\boldsymbol{x}_{t,(b')}-\boldsymbol{x}_{0,(b')}\right\|_2^2}{d_{b'}}\\
\leq&\eta^2t^2+\eta^2t\frac{\beta_2}{1-\beta_2}\ln\frac{\epsilon+C+4(1-\beta_2)\max_{b\in[B]}\frac{H_b^2}{d_b^2}\mathbb{E}\sum_{s=1}^t\beta_2^{t-s}\max_{b'\in[B]}\frac{\left\|\boldsymbol{x}_{s-1,(b')}-\boldsymbol{x}_{0,(b')}\right\|_2^2}{d_{b'}}}{v_0+\epsilon}\\
\leq&\eta^2t^2+\eta^2t\frac{\beta_2}{1-\beta_2}\ln\frac{\epsilon+C+4(1-\beta_2)\max_{b\in[B]}\frac{H_b^2}{d_b^2}\sum_{s=1}^t\beta_2^{t-s}G}{v_0+\epsilon}\\
\leq&\eta^2T^2+\eta^2T\frac{\beta_2}{1-\beta_2}\ln\frac{\epsilon+C+4\max_{b\in[B]}\frac{H_b^2}{d_b^2}G}{v_0+\epsilon}.
\end{aligned}$$

We further define $G'=\epsilon+C+4\max_{b\in[B]}\frac{H_b^2}{d_b^2}G$ and get that

$$\begin{aligned}
G'\leq&\epsilon+C+4\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T^2+4\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T\frac{\beta_2}{1-\beta_2}\ln\frac{G'}{v_0+\epsilon}\\
\leq&\epsilon+C+4\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T^2\\
&+4\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T\frac{\beta_2}{1-\beta_2}\left(\ln\frac{G'(1-\beta_2)}{4\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T\beta_2}+\ln4\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T\frac{\beta_2}{(1-\beta_2)(v_0+\epsilon)}\right)\\
\leq&\epsilon+C+4\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T^2+\frac{G'}{2}+4\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T\frac{\beta_2}{1-\beta_2}\ln4\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T\frac{\beta_2}{(1-\beta_2)(v_0+\epsilon)}.
\end{aligned}$$

The last inequality comes from $\ln x\leq\frac{x}{2}$. Then we can get that

$$\begin{aligned}
\mathbb{E}\max_{b\in[B]}v_{T,b}+\epsilon\leq&\epsilon+C+4(1-\beta_2)\max_{b\in[B]}\frac{H_b^2}{d_b^2}\sum_{s=1}^t\beta_2^{t-s}G\\
\leq&\epsilon+C+4\max_{b\in[B]}\frac{H_b^2}{d_b^2}G=G'\\
\leq&2\epsilon+2C+8\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T^2+8\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T\frac{\beta_2}{1-\beta_2}\ln4\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T\frac{\beta_2}{(1-\beta_2)(v_0+\epsilon)}.
\end{aligned}$$

□

Finally, we give the proof for Theorem 3.12. When $\Phi(i)=i$, i.e., each parameter forms a single block, it becomes the proof for Theorem 3.5.

Theorem 3.12 (Main, Blockwise `Adam`)**.** *Under Assumption 3.11, suppose L is $\mathbf{H}$-smooth blockwisely w.r.t. Φ-norm, where $\mathbf{H} = (H_1, \dots, H_B) \in \mathbb{R}^B$, for Algorithm 3, we have that*

$$\min_{\frac{T}{2}<t\leq T} \mathbb{E} \sum_{b=1}^{B} \sqrt{d_b} \left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2 \leq E + \sqrt{E} \sqrt{\frac{\beta_2^{\frac{T}{4}}}{T(1-\beta_2)} d\sqrt{v_0} + \sum_{b=1}^{B} d_b\sigma_b + d\sqrt{\epsilon}}$$

with

$$E = \frac{2}{\eta T}\mathbb{E}\left[L(\boldsymbol{x}_0) - L(\boldsymbol{x}_T)\right] + \left(1 + \frac{\beta_2 F}{T(1-\beta_2)}\right) \left(\eta \sum_{b=1}^{B} H_b + \sqrt{1-\beta_2} \sum_{b=1}^{B} d_b\sigma_b\right),$$

and

$$F = O\left(\ln\left(1 + \frac{\sum_{b=1}^{B}\sigma_b^2 + \|\nabla L(\boldsymbol{x}_0)\|_\Phi^2 + \max_{b\in[B]} H_b^2\eta^2 T(T + \frac{1}{1-\beta_2})}{v_0 + \epsilon}\right)\right).$$

Proof of Theorem 3.12. In a single step, we can apply Lemma 3.13 and have that

$$\begin{aligned} L(\boldsymbol{x}_t) - L(\boldsymbol{x}_{t-1}) &\leq \nabla L(\boldsymbol{x}_{t-1})^\top(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}) + \frac{1}{2}\sum_{b=1}^{B}\frac{H_b}{d_b}\sum_{\Phi(i)=b}(x_{t,i} - x_{t-1,i})^2 \\ &= -\eta\sum_{i=1}^{d}\frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{v_{t,\Phi(i)} + \epsilon}} + \frac{1}{2}\eta^2\sum_{b=1}^{B}\frac{H_b}{d_b}\frac{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2}{v_{t,b}+\epsilon}. \end{aligned}$$

If we sum over t from 1 to T and take expectation, we can get

$$\begin{aligned} \mathbb{E}\left[L(\boldsymbol{x}_T) - L(\boldsymbol{x}_0)\right] &\leq -\mathbb{E}\left[\eta\sum_{i=1}^{d}\sum_{t=1}^{T}\frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{v_{t,\Phi(i)}+\epsilon}}\right] + \frac{1}{2}\eta^2\mathbb{E}\left[\sum_{b=1}^{B}\frac{H_b}{d_b}\sum_{t=1}^{T}\frac{\left\|\boldsymbol{g}_{t,(b)}\right\|_2^2}{v_{t,b}+\epsilon}\right] \\ &\leq -\mathbb{E}\left[\eta\sum_{i=1}^{d}\sum_{t=1}^{T}\frac{g_{t,i}\bar{g}_{t,i}}{\sqrt{v_{t,\Phi(i)}+\epsilon}}\right] + \frac{1}{2}\eta^2\mathbb{E}\left[\sum_{b=1}^{B}H_b\left(T + \frac{\beta_2}{1-\beta_2}\ln\frac{v_{T,b}+\epsilon}{v_{0,b}+\epsilon}\right)\right]. \end{aligned}$$

The second inequality comes from applying Lemma C.1. By Lemma C.2, we have that

$$\begin{aligned} \frac{1}{T}\mathbb{E}\left[\sum_{i=1}^{d}\sum_{t=1}^{T}\frac{\bar{g}_{t,i}^2}{\sqrt{\tilde{v}_{t,\Phi(i)}+\epsilon}}\right] &\leq \frac{2}{\eta T}\mathbb{E}\left[L(\boldsymbol{x}_0) - L(\boldsymbol{x}_T)\right] + \frac{\eta}{T}\mathbb{E}\left[\sum_{b=1}^{B}H_b\left(T + \frac{\beta_2}{1-\beta_2}\ln\frac{v_{T,b}+\epsilon}{v_{0,b}+\epsilon}\right)\right] \\ &\quad + \frac{1}{T}\sum_{b=1}^{B}d_b\sigma_b\sqrt{1-\beta_2}\left(T + \frac{\beta_2}{1-\beta_2}\mathbb{E}\ln\frac{v_{T,b}+\epsilon}{v_{0,b}+\epsilon}\right) \\ &\leq \frac{2}{\eta T}\mathbb{E}\left[L(\boldsymbol{x}_0) - L(\boldsymbol{x}_T)\right] + \eta\sum_{b=1}^{B}H_b + \sqrt{1-\beta_2}\sum_{b=1}^{B}d_b\sigma_b \\ &\quad + \frac{\beta_2}{T(1-\beta_2)}\left(\eta\sum_{b=1}^{B}H_b + \sqrt{1-\beta_2}\sum_{b=1}^{B}\sigma_b\right)\max_{b\in[B]}\mathbb{E}\ln\frac{v_{T,b}+\epsilon}{v_{0,b}+\epsilon} \\ &\leq \frac{2}{\eta T}\mathbb{E}\left[L(\boldsymbol{x}_0) - L(\boldsymbol{x}_T)\right] + \eta\sum_{b=1}^{B}H_b + \sqrt{1-\beta_2}\sum_{b=1}^{B}d_b\sigma_b \\ &\quad + \frac{\beta_2}{T(1-\beta_2)}\left(\eta\sum_{b=1}^{B}H_b + \sqrt{1-\beta_2}\sum_{b=1}^{B}d_b\sigma_b\right)\ln\frac{\mathbb{E}\max_{b\in[B]}v_{T,b}+\epsilon}{v_0+\epsilon} \end{aligned}$$

From Lemma C.4, we can define

$$\begin{aligned} E = {} & \frac{2}{\eta T}\mathbb{E}\left[L(\boldsymbol{x}_0) - L(\boldsymbol{x}_T)\right] + \eta\sum_{b=1}^{B}H_b + \sqrt{1-\beta_2}\sum_{b=1}^{B}d_b\sigma_b + \frac{\beta_2}{T(1-\beta_2)}\left(\eta\sum_{b=1}^{B}H_b + \sqrt{1-\beta_2}\sum_{b=1}^{B}d_b\sigma_b\right) \\ & \cdot\ln\frac{2\epsilon + 2C + 8\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T^2 + 8\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T\frac{\beta_2}{1-\beta_2}\ln 4\max_{b\in[B]}\frac{H_b^2}{d_b^2}\eta^2T\frac{\beta_2}{(1-\beta_2)(v_0+\epsilon)}}{v_0+\epsilon} \end{aligned}$$

and $C = v_0 + 2\sum_{b=1}^{B} \sigma_b^2 + 4\max_{b\in[B]} \left\|\nabla_{(b)} L(\boldsymbol{x}_0)\right\|_2^2 / d_b$. One can show that

$$E = \frac{2}{\eta T}\mathbb{E}\left[L(\boldsymbol{x}_0) - L(\boldsymbol{x}_T)\right] + \eta\sum_{b=1}^{B} H_b + \sqrt{1-\beta_2}\sum_{b=1}^{B} d_b\sigma_b + \frac{\beta_2}{T(1-\beta_2)}\left(\eta\sum_{b=1}^{B} H_b + \sqrt{1-\beta_2}\sum_{b=1}^{B} d_b\sigma_b\right)F,$$

with

$$F = \ln\left(1 + \frac{\sum_{b=1}^{B}\sigma_b^2 + \|\nabla L(\boldsymbol{x}_0)\|_\Phi^2 + \max_{b\in[B]} H_b^2\eta^2 T(T+\frac{1}{1-\beta_2})}{v_0+\epsilon}\right).$$

$$\frac{1}{T}\mathbb{E}\left[\sum_{i=1}^{d}\sum_{t=1}^{T}\frac{\bar{g}_{t,i}^2}{\sqrt{\tilde{v}_{t,\Phi(i)}+\epsilon}}\right] \le E.$$

By Lemma C.3 and Cauchy inequality, we have that

$$\begin{aligned}
\frac{2}{T}\mathbb{E}\sum_{t=\frac{T}{2}+1}^{T}\sum_{b=1}^{B}\sqrt{d_b}\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2 &\le \left(\frac{2}{T}\mathbb{E}\sum_{t=\frac{T}{2}+1}^{T}\sum_{b=1}^{B}\frac{\left\|\bar{\boldsymbol{g}}_{t,(b)}\right\|_2^2}{\sqrt{\tilde{v}_{t,b}+\epsilon}}\right)^{\frac{1}{2}}\left(\frac{2}{T}\mathbb{E}\sum_{t=\frac{T}{2}+1}^{T}\sum_{b=1}^{B} d_b\sqrt{\tilde{v}_{t,b}+\epsilon}\right)^{\frac{1}{2}}\\
&\le \left(\frac{2}{T}\mathbb{E}\sum_{t=\frac{T}{2}+1}^{T}\sum_{b=1}^{B}\sum_{\Phi(i)=b}\frac{\bar{g}_{t,i}^2}{\sqrt{\tilde{v}_{t,b}+\epsilon}}\right)^{\frac{1}{2}}\left(\frac{2}{T}\mathbb{E}\sum_{t=\frac{T}{2}+1}^{T}\sum_{b=1}^{B} d_b\sqrt{\tilde{v}_{t,b}+\epsilon}\right)^{\frac{1}{2}}\\
&\le \sqrt{2E}\left(4E + \frac{4\beta_2^{\frac{T}{4}}}{T(1-\beta_2)}d\sqrt{v_0} + \sum_{b=1}^{B} d_b\sigma_b + d\sqrt{\epsilon}\right)^{\frac{1}{2}}\\
&\le 2\sqrt{2}E + \sqrt{2}\sqrt{E}\sqrt{\frac{4\beta_2^{\frac{T}{4}}}{T(1-\beta_2)}d\sqrt{v_0} + \sum_{b=1}^{B} d_b\sigma_b + d\sqrt{\epsilon}}.
\end{aligned}$$

This completes the proof. ☐

D EXPERIMENT DETAILS

D.1 TRAINING ADAM ON A ROTATED LOSS

A key difficulty in implementing `rotated Adam` arises from applying an orthogonal rotation on the parameters before calculating the loss. It is computationally infeasible to apply a 125M × 125M orthogonal matrix on the 125M-sized parameter vector. To avoid such computation, we design a new orthogonal transformer to rotate the parameters of the network. In what follows, we elaborate on this rotation.

`RandPerm`. Given a vector v of size d, we can orthogonally rotate it by repeatedly applying these consecutive operations: 1. Permute the entries of the vector according to a randomly chosen permutation $\pi \in \mathbb{S}_d$. 2. Reshape the permuted vector into a 3D tensor of size $[s_1, s_2, s_3]$, apply a fixed orthogonal rotation of size $s \times s$ on each side of the tensor and then reshape it back to a vector of size d.

This operation performs an orthogonal transformation $\mathcal{R}$ on the input vector v. We can chain multiple operations of this kind and construct $\texttt{RandPerm}^k$, where k is a positive number indicating the number of consecutive `RandPerm` s applied. Building upon this rotation, we train GPT-2 125M with `Adam` on $L \circ \texttt{RandPerm}^2$ to analyze our hypothesis regarding the ℓ_∞ geometry of the loss landscape and to verify that `Adam` will indeed suffer from the induced orthogonal equivariance. Figure 1 confirms our findings, as the performance of `rotated Adam` with $\texttt{RandPerm}^2$ is significantly worse than `Adam`. This suggests that `Adam` is highly sensitive to the rotation and adaptivity alone can't explain its advantage.

D.2 COMPUTATION OF MATRIX NORMS

It is impossible to get the full Hessian matrix and directly compute norms of it. We can only leverage Hessian vector product function in pytorch to probe the Hessian matrix. The estimation of spectral norm is done by power iteration. The estimation of $(1,1)$-norm relies on the properties of Cauchy distribution. Given $a = [a_1, \cdots, a_n]$ and i.i.d. standard Cauchy variables $X_1, \cdots, X_n$, $\sum_{i=1}^n a_i X_i$ is Cauchy distributed with location 0 and scale $\sum_{i=1}^n |a_i|$. For a single value Cauchy distribution with location 0 and scaling γ, an estimator for γ is the median of the absolute values of all the samples.

Therefore, we propose Algorithm 4 to estimate the sum of absolute values for each row and sum over all the rows to get $(1,1)$-norm of Hessian matrix. We choose $n = 50$ for all the measurement experiments. We also prove a concentration inequality in Theorem D.1.

Algorithm 4 Estimation of $(1,1)$-Norm of Hessian, $\nabla^2 L(\theta)$

```
Input: Number of Cauchy vectors $n$, parameter $\theta \in \mathbb{R}^d$, loss $L$
for $i = 1 \to n$ :
    Sample a independent Cauchy vector $v^{(i)} \in \mathbb{R}^d$ where $v_j^{(i)} \overset{\text{i.i.d.}}{\sim}$ Cauchy$(0,1)$ for $j = 1, \ldots, d$.
    $\mathbf{H}_{:,i} \leftarrow \nabla^2 L(\theta) \cdot v^{(i)}$        (Using hessian-vector product)
return $\sum_{j=1}^d \mathrm{median}(\mathrm{abs}(\mathbf{H}_{j,:}))$
```

Theorem D.1. *For the estimate in Algorithm 4, it holds that*

$$P\left(\left|\sum_{j=1}^d \mathit{median}(|\mathbf{H}_{j,:}|) - \left\|\nabla^2 L(\theta)\right\|_{1,1}\right| \geq \delta \left\|\nabla^2 L(\theta)\right\|_{1,1}\right) \leq d \exp\left(-\frac{8n}{25\pi^2}\delta^2\right)$$

for $\delta \in (0,1)$.

Proof. We first prove a concentration inequality for $M_n = \mathrm{median}(|X_1|, \cdots, |X_n|)$ in which $X_1, \cdots, X_n \overset{\text{i.i.d.}}{\sim} \mathrm{Cauchy}(0,1)$.

Given $X \sim \mathrm{Cauchy}(0,1)$, the cumulative distribution function of $|X|$ is $F(x) = \frac{2}{\pi}\mathrm{arctan}(x)$ and the median of $|X|$ is 1 because $F(1) = 0.5$. For a fixed $\delta \in (0,1)$, define $p_1 = \frac{2}{\pi}\mathrm{arctan}(1-\delta)$ to be the probability that $|X|$ is smaller than $1-\delta$ and $S = \sum_{i=1}^n \mathbf{1}_{|X_i| \leq 1-\delta}$. Since $\mathbf{1}_{|X_i| \leq 1-\delta}$ follows i.i.d. Bernoulli distribution with p_1, $S \sim \mathrm{Bin}(n, p_1)$.

$M_n \leq 1-\delta$ if and only if at least $\frac{n+1}{2}$ X_i's are smaller than $1-\delta$. And we can apply Hoeffding's inequality on S and get that

$$\begin{aligned}
P(M_n \leq 1-\delta) &= P(S \geq \frac{n+1}{2}) \leq P(S \geq \frac{n}{2}) \\
&= P(S - np_1 \geq \frac{n}{2} - np_1) \\
&\leq \exp\left(-\frac{2(\frac{n}{2} - np_1)^2}{n}\right) \\
&\leq \exp\left(-2n\left(\frac{1}{2} - \frac{2}{\pi}\mathrm{arctan}(1-\delta)\right)^2\right).
\end{aligned}$$

With similar derivation, we can also get that

$$P(M_n \geq 1+\delta) \leq \exp\left(-2n\left(\frac{1}{2} - \frac{2}{\pi}\mathrm{arctan}(1+\delta)\right)^2\right).$$

When $\delta \in (-1,1)$, we have that

$$\left(\frac{1}{2} - \frac{2}{\pi}\mathrm{arctan}(1+\delta)\right)^2 = \frac{4}{\pi^2}\left(\mathrm{arctan}(1) - \mathrm{arctan}(1+\delta)\right)^2 \geq \frac{4}{25\pi^2}\delta^2.$$

	SGD	AdaSGD	Adam	Rotated Adam
Batch size 16	0.0777	0.114	0.064	0.0905
Batch size 64	0.0698	0.0854	0.0472	0.0574
Batch size 256	0.0723	0.0787	0.0359	0.0485
Batch size 1024	0.1115	0.0915	0.0735	0.0817

Table 4: Training losses of ResNet for different optimizers and different batch size. For each setting, we choose the optimal performance over all the learning rates.

because the derivative of $\arctan(x)$ is always greater than $\frac{1}{5}$ for $x \in (0, 2)$. Then we can have that

$$P(|M_n - 1| \geq \delta) \leq 2 \exp\left(-\frac{8n}{25\pi^2}\delta^2\right).$$

When $v_j^{(i)} \overset{\text{i.i.d.}}{\sim} \text{Cauchy}(0, 1)$, it holds that $H_{j,i} = \nabla^2 L(\theta)_{j,:} \cdot v^{(i)}$ follows $\text{Cauchy}(0, \sum_{k=1}^{d} \left|\nabla^2 L(\theta)_{j,k}\right|)$. We can have that

$$P\left(\left|\text{median}(|H_{j,1}|, \cdots, |H_{j,n}|) - \sum_{k=1}^{d} \left|\nabla^2 L(\theta)_{j,k}\right|\right| \geq \delta \sum_{k=1}^{d} \left|\nabla^2 L(\theta)_{j,k}\right|\right) \leq \exp\left(-\frac{8n}{25\pi^2}\delta^2\right).$$

And we can sum the tail probability over all the rows j and get that

$$P\left(\left|\sum_{j=1}^{d} \text{median}(|\mathbf{H}_{j,:}|) - \left\|\nabla^2 L(\theta)\right\|_{1,1}\right| \geq \delta \left\|\nabla^2 L(\theta)\right\|_{1,1}\right) \leq d \exp\left(-\frac{8n}{25\pi^2}\delta^2\right).$$

□

D.3 TRAINING DETAILS

In `Adam` and its variants (including AdaSGD) the momentums were set to $\beta_1, \beta_2 = (0.9, 0.99)$. Momentum in SGD was also set to 0.9. Weight decay is always deactivated.

For the ResNet experiment, we applied random crop and random horizontal flip augmentations over the training data to promote better generalization. We tuned each optimizer through searching over the same grid of learning rates[3] The number of iterations is adjusted per batch size to result in 20 epochs for each training run (for instance, 4000 iterations were used for a batch size of 256, and 1000 iterations were used for a batch size of 1024).

We didn't tune learning rate for GPT-2 experiment and just used the same peak learning rate of 6×10^{-4} for all optimization methods except `SGD`, for which we did a grid search to find the maximum possible peak learning rate.

---

[3] We used the following values: 6.25×10^{-4}, 1.25×10^{-3}, 2.5×10^{-3}, 5.0×10^{-3}, 1.0×10^{-2}, 2.0×10^{-2}, 4.0×10^{-2}, 8.0×10^{-2}, 1.6×10^{-1}, 3.2×10^{-1}, 6.4×10^{-1}, 1.28×10^{0}.