TOWARDS UNDERSTANDING CONVERGENCE AND GENERALIZATION OF ADAMW

Anonymous authors
Paper under double-blind review

ABSTRACT

AdamW modifies vanilla Adam by decaying network weights per training iteration, and shows remarkable generalization superiority over Adam and its ℓ_2-regularized variant. In context of adaptive gradient algorithms (*e.g.* Adam), the decoupled weight decay in AdamW differs from the widely used ℓ_2-regularizer, since the former does not affect optimization steps, while the latter changes the first- and second-order gradient moments and thus the optimization steps. Despite its great success on both vision transformers and CNNs, for AdamW, its convergence behavior and its generalization improvement over (ℓ_2-regularized) Adam remain absent yet. To solve this issue, we prove the convergence of AdamW and justify its generalization advantages over Adam and its ℓ_2-regularized version. Specifically, AdamW can provably converge but minimizes a dynamically regularized loss that combines a vanilla loss and a dynamical regularization induced by the decoupled weight decay, thus leading to its different behaviors compared with Adam and its ℓ_2-regularized version. Moreover, on both general nonconvex problems and PŁ-conditioned problems, we establish the stochastic gradient complexity of AdamW to find a stationary point. Such complexity is also applicable to Adam and its ℓ_2-regularized variant, and indeed improves their previously known complexity, especially for modern over-parametrized networks. Besides, we theoretically show that AdamW often enjoys smaller generalization error bound than both Adam and its ℓ_2-regularized variant from the Bayesian posterior aspect. This result, for the first time, explicitly reveals the benefits of the unique decoupled weight decay in AdamW. We hope the theoretical results in this work could motivate researchers to propose novel optimizers with faster convergence and better generalization. Experimental results testify our theoretical implications.

1 INTRODUCTION

Adaptive gradient algorithms, *e.g.* Adam (Kingma & Ba, 2014), have become the most popular optimizers to train deep networks because of their faster convergence speed than SGD (Robbins & Monro, 1951), with many successful applications witnessed to computer vision (Dosovitskiy et al., 2020; Zhou et al., 2022) and natural language processing (Sainath et al., 2013; Abdel-Hamid et al., 2014), to name a few. Similar to the precondition spirit in the second-order algorithms (Süli & Mayers, 2003), adaptive gradient algorithms precondition the landscape curvature of loss objective, and accordingly adjust the learning rate for each gradient coordinate. This precondition often helps these adaptive algorithms achieve faster convergence speed than their non-adaptive counterparts across many applications, *e.g.* SGD which uses a single learning rate for all gradient coordinates. Unfortunately, this precondition also brings negative effect. That is, adaptive algorithms usually suffer from worse generalization performance than SGD (Keskar & Socher, 2017; Luo et al., 2019).

As a recent leading adaptive gradient approach, AdamW (Loshchilov & Hutter, 2018) greatly improves the generalization performance of adaptive algorithms, and has shown superior generalization performance over other adaptive and non-adaptive algorithms, *e.g.* Adam and SGD, on vision transformers, such as ViTs (Touvron et al., 2021) and Swin (Liu et al., 2021), and CNN, *e.g.* ResNet (He et al., 2016; Touvron et al., 2021) and ConvNext (Liu et al., 2022). The core of AdamW is a decoupled weight decay and also its integration with Adam. Specifically, AdamW first uses an exponential moving average to estimate the first-order moment $\boldsymbol{m}_k$ and second-order moment $\boldsymbol{n}_k$ of gradient like Adam, and then updates network weights $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta \boldsymbol{m}_k / \sqrt{\boldsymbol{n}_k + \delta} - \eta\lambda\boldsymbol{x}_k$ with a learning rate η, a weight decay parameter λ, and a small constant δ. One can observe that the update of AdamW

decouples the weight decay from the optimization steps taken w.r.t. the loss function, since the weight decay is always $-\eta\lambda\boldsymbol{x}_k$ no matter what the loss and optimization step are. This decoupled weight decay degenerates to ℓ_2-regularization for SGD, but differs from ℓ_2-regularization for adaptive algorithms. Because of its simplicity and strong compatibility, AdamW has been widely used in network training. But there remain many mysteries about AdamW yet. Firstly, for convergence, it is still not clear whether AdamW can theoretically converge or not, and if yes, what convergence rate it can achieve. Moreover, for the generalization superiority of AdamW over the widely used (ℓ_2-regularized) Adam, the theoretical reasons are rarely investigated though heavily desired.

Contributions: In this work, to resolve these issues, we provide a new viewpoint to understand the convergence and generalization behaviors of AdamW. Particularly, we theoretically prove the convergence of AdamW, and also further justify the generalization superiority of AdamW over Adam and its ℓ_2-regularized version. Our main contributions are highlighted below.

Firstly, we prove that AdamW can converge but minimizes a dynamically regularized loss that combines the vanilla loss and a dynamical regularization induced by the decoupled weight decay. Interestingly, this dynamical regularization distinguishes from the commonly used ℓ_2-regularization, and thus leads to the different behaviors between AdamW and (ℓ_2-regularized) Adam. For convergence speed, on general nonconvex problems, AdamW can find an ϵ-accurate first-order stationary point within stochastic gradient complexity $\mathcal{O}(c_\infty^{2.5}\epsilon^{-4})$ when using constant learning rate and $\mathcal{O}(c_\infty^{1.25}\epsilon^{-4}\log(\frac{1}{\epsilon}))$ with decayed learning rate, where c_∞ is the ℓ_∞-norm upper bound of stochastic gradient. When ignoring constant and logarithm terms, both complexities match the lower complexity bound $\mathcal{O}(\epsilon^{-4})$ in (Arjevani et al., 2019) under the same assumptions. These complexities are also applicable to Adam and its ℓ_2-regularized version, and improve their previously known complexities $\mathcal{O}(c_\infty\sqrt{d}\epsilon^{-4})$ and $\mathcal{O}(c_\infty\sqrt{d}\epsilon^{-4}\log(\frac{1}{\epsilon}))$ when respectively using constant and decayed learning rate (Zhou et al., 2018; Chen et al., 2021; Guo et al., 2021), since c_∞ is often much smaller than the network dimension d, especially for modern over-parametrized networks. On PŁ-conditioned nonconvex problems, our established complexity of AdamW also enjoys similar advantages.

Next, we theoretically show the benefits of the decoupled weight decay in AdamW to the generalization performance from the Bayesian posterior aspect. Specifically, we show that a proper decoupled weight decay $\lambda > 0$ helps AdamW achieve smaller generalization error, indicating the superiority of AdamW over vanilla Adam (without ℓ_2-regularization) which corresponds to $\lambda = 0$. Moreover, we further analyze ℓ_2-regularized Adam, and observe that AdamW often enjoys smaller generalization error bound than ℓ_2-regularized Adam. To our best knowledge, this work is the first one that explicitly shows the superiority of AdamW over Adam with or without ℓ_2-regularization.

2 Related Work

Convergence Analysis. Adaptive gradient algorithms, *e.g.* Adam (Kingma & Ba, 2014), have become the default optimizers in deep learning because of their fast convergence speed. Subsequently, many works investigate their convergence to deepen their understanding. On convex problems, Adam-type algorithms, *e.g.* Adam and AMSGrad (Reddi et al., 2019), are well studied and shown to enjoy the regret $\mathcal{O}(\sqrt{T})$ under the online learning setting with the time horizon T. For more practical nonconvex problems widely occurred in deep learning, under Lipschitz gradient condition, Guo et al. (2021) and Chen et al. (2018) established the stochastic gradient complexity $\mathcal{O}\big(c_\infty\sqrt{d}\epsilon^{-4}\big)$ of Adam-type algorithms to achieve an ϵ-accurate stationary point, where d is the problem dimension and c_∞ is the ℓ_∞-norm upper bound of stochastic gradient. RMSProp and Padam (Chen et al., 2021) are proved to have the complexity $\mathcal{O}\big(\sqrt{c_\infty}d\epsilon^{-4}\big)$ (Zhou et al., 2018), and Adabelief (Zhuang et al., 2020) has $\mathcal{O}\big(c_2^6\epsilon^{-4}\big)$ complexity, where c_2 is the ℓ_2-norm upper bound of stochastic gradient. The recently proposed Adan (Xie et al., 2022) enjoys the complexity $\mathcal{O}(c_\infty\epsilon^{-4})$ which is slightly better than the aforementioned complexity. Unfortunately, the convergence behaviors of AdamW remains unclear, even though it is the dominant optimizer with higher and more stable performance on vision transformers (Touvron et al., 2021; Liu et al., 2021) and CNNs (Touvron et al., 2021).

Generalization Analysis. Most works, *e.g.* (Mandt et al., 2016; Zhu et al., 2018), analyze the generalization of an algorithm through studying its stochastic differential equations (SDEs) because of the similar convergence behaviors of an algorithm and its SDE. For instance, Jastrzkebski et al. (2017) and Simsekli et al. (2019) respectively formulated SGD into Brownian- and Lévy-driven SDEs via assuming Gaussian or heavy-tailed gradient noise, and both proved that SGD tends to converge to flatter minima instead of sharp minima and thus enjoys good generalization. See more

results in (Pavlyukevich, 2011; Chaudhari & Soatto, 2018). But they all do not analyzed AdamW. Recently, some works also study the effects of weight decay. The works (Van Laarhoven, 2017; Zhang et al., 2018; Hoffer et al., 2018) intuitively claim that for layers followed by normalizations, *e.g.* BatchNormalization (Ioffe & Szegedy, 2015), weight decay increases the effective learning rate by reducing the scale of the network weights, and higher learning rates give larger gradient noise which often acts a stochastic regularizer. But Zhou et al. (2021b) argued the benefits of weight decay to the layers without normalization, *e.g.* fully-connected networks, and further empirically found the effects of weight decay to the last fully-connected layer of a network: it constrains the weight norm and controls the cross-boundary risk. Unfortunately, none of them explicitly show the generalization benefits of weight decay in AdamW. In this work, we borrow the aforementioned SDE tool and PAC Bayesian framework (McAllester, 1999) to explicitly and rigorously analyze the generalization effects of decoupled weight decay of AdamW and also its superiority over (ℓ_2-regularized) Adam.

3 Notation and Preliminarily

AdamW & ℓ_2-regularized Adam. Here we first briefly recall the steps of AdamW and (ℓ_2-regularized) Adam to solve the following stochastic nonconvex optimization problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} F(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{D}}[f(\boldsymbol{x},\boldsymbol{\xi})], \tag{1}$$

where loss f is differentiable and nonconvex, sample $\boldsymbol{\xi}$ is drawn from a distribution $\mathcal{D}$. To solve (1), at the k-th iteration, AdamW estimates the current gradient $\nabla F(\boldsymbol{x}_k)$ as the minibatch gradient $\boldsymbol{g}_k = \frac{1}{b}\sum_{i=1}^b \nabla f(\boldsymbol{x}_k;\boldsymbol{\xi}_i)$, and updates the variable $\boldsymbol{x}$ with $\boldsymbol{\beta}_1 \in [0,1)$, $\boldsymbol{\beta}_2 \in [0,1]$ and $\delta > 0$ as

$$\boldsymbol{m}_k = (1-\boldsymbol{\beta}_1)\boldsymbol{m}_{k-1} + \boldsymbol{\beta}_1\boldsymbol{g}_k,\ \ \boldsymbol{n}_k = (1-\boldsymbol{\beta}_2)\boldsymbol{n}_{k-1} + \boldsymbol{\beta}_2\boldsymbol{g}_k^2,\ \ \boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta\boldsymbol{m}_k/\sqrt{\boldsymbol{n}_k+\delta} - \eta\lambda\boldsymbol{x}_k, \tag{2}$$

where $\boldsymbol{m}_0 = \boldsymbol{g}_0$ and $\boldsymbol{n}_0 = \boldsymbol{g}_0^2$. See detailed AdamW algorithm in Algorithm 1 in Appendix C. The term $(-\eta\lambda\boldsymbol{x}_k)$ comes from the decoupled weight decay. AdamW only differs from vanilla Adam in the third step in Eqn. (2). Specifically, AdamW decouples the weight decay from the optimization steps, as the weight decay is always $-\eta\lambda\boldsymbol{x}_k$ no matter what the loss and optimization step are, while Adam often directly employs the ℓ_2-regularization. In this way, Adam adds a conventional weight decay $\lambda\boldsymbol{x}_k$ into the gradient estimation $\boldsymbol{g}_k = \frac{1}{b}\sum_{i=1}^b \nabla f(\boldsymbol{x}_k;\boldsymbol{\xi}_i) + \lambda\boldsymbol{x}_k$, then updates $\boldsymbol{m}_k$ and $\boldsymbol{n}_k$ as in (2), and finally updates $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta\boldsymbol{m}_k/\sqrt{\boldsymbol{n}_k+\delta}$. The decoupled weight decay in AdamW often achieves better generalization than the conventional one in Adam on many networks, *e.g.* ViTs (Touvron et al., 2021; Liu et al., 2021), and CNNs (Touvron et al., 2021; Liu et al., 2022).

Analysis Assumptions. Here we introduce necessary assumptions for theoretical analysis, which are commonly used in (Kingma & Ba, 2014; Reddi et al., 2019; Luo et al., 2019; Duchi et al., 2011).

Assumption 1 (L-smoothness)**.** *The function $f(\cdot,\cdot)$ is L-smooth w.r.t. the parameter, if $\exists L > 0$,*

$$\|\nabla f(\boldsymbol{x}_1,\boldsymbol{\xi}) - \nabla f(\boldsymbol{x}_2,\boldsymbol{\xi})\|_2 \le L\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2, \qquad \forall \boldsymbol{x}_1,\boldsymbol{x}_2 \text{ and } \boldsymbol{\xi}\sim\mathcal{D}.$$

Assumption 2 (Gradient estimation assumption)**.** *The gradient estimation $\boldsymbol{g}_k$ is unbiased, and its magnitude and variance are bounded as follows*

$$\mathbb{E}[\boldsymbol{g}_k] = \nabla F(\boldsymbol{x}_k), \qquad \|\boldsymbol{g}_k\|_\infty \le c_\infty, \qquad \mathbb{E}[\|\nabla F(\boldsymbol{x}_k) - \boldsymbol{g}_k\|_2] \le \sigma, \quad \forall k.$$

When a nonconvex problem satisfies both Assumptions 1 and 2, the lower bound of the stochastic gradient complexity (a.k.a. IFO complexity) to find an ϵ-accurate first-order stationary point is $\Omega(\epsilon^{-4})$ (Arjevani et al., 2019; 2020). Next, we introduce Polyak-Łojasiewicz (PŁ) condition which is widely used in deep network analysis, since as observed or proved in (Hardt & Ma, 2016; Xie et al., 2017; Li & Yuan, 2017; Charles & Papailiopoulos, 2018; Zhou & Liang, 2018; Zhou et al., 2021a), deep neural networks often satisfy PŁ condition at least around a local minimum.

Assumption 3 (PŁ Condition)**.** *Let $\boldsymbol{x}_* \in \operatorname{argmin}_{\boldsymbol{x}} F(\boldsymbol{x})$. We say a function $F(\boldsymbol{x})$ satisfies μ-PŁ condition if it satisfies $2\mu(F(\boldsymbol{x}) - F(\boldsymbol{x}_*)) \le \|\nabla F(\boldsymbol{x})\|_2^2$ $(\forall \boldsymbol{x})$ with a universal constant μ.*

4 Convergence Analysis

In this section, we first investigate the convergence performance of AdamW on general nonconvex problems and then show its performance improvement when the problems further satisfy the PŁ condition. Since AdamW is mostly used in the highly nonconvex deep networks, in this work we analyze it on the nonconvex problems to match its real application setting.

To begin with, we first define the following dynamic function $F_k(\boldsymbol{x})$ at the k-th iteration which is indeed the combination of the vanilla loss $F(\boldsymbol{x})$ in Eqn. (1) and a dynamic regularization $\frac{\lambda}{2}\|\boldsymbol{x}\|_{\boldsymbol{v}_k}^2$:

$$F_k(\boldsymbol{x}) = F(\boldsymbol{x}) + \frac{\lambda}{2}\|\boldsymbol{x}\|_{\boldsymbol{v}_k}^2 = \mathbb{E}_{\boldsymbol{\xi}}[f(\boldsymbol{x};\boldsymbol{\xi})] + \frac{\lambda}{2}\|\boldsymbol{x}\|_{\boldsymbol{v}_k}^2, \tag{3}$$

where $\boldsymbol{v}_k = \sqrt{\boldsymbol{n}_k + \delta}$ and $\|\boldsymbol{x}\|_{\boldsymbol{v}_k} = \sqrt{\langle \boldsymbol{x}, \boldsymbol{v}_k \otimes \boldsymbol{x}\rangle}$ in which $\otimes$ denotes element-wise product. To obtain (3), one can approximate vanilla loss $F(\boldsymbol{x})$ by its Taylor expansion, and compute $\boldsymbol{x}_{k+1}$:

$$\boldsymbol{x}_{k+1} \approx \operatorname{argmin}_{\boldsymbol{x}} F(\boldsymbol{x}_k) + \langle \nabla F(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k\rangle + \frac{1}{2\eta}\|\boldsymbol{x} - \boldsymbol{x}_k\|_{\boldsymbol{v}_k}^2 + \frac{\lambda}{2}\|\boldsymbol{x}\|_{\boldsymbol{v}_k}^2 = \frac{1}{1+\lambda\eta}\left[\boldsymbol{x}_k - \eta\frac{\nabla F(\boldsymbol{x}_k)}{\boldsymbol{v}_k}\right],$$

In this approximation, similar to adaptive gradient algorithms, *e.g.* Adam, $\boldsymbol{v}_k$ plays a role similar to the Hessian at the point $\boldsymbol{x}_k$. Then considering η is very small in practice, one can approximate $\frac{1}{1+\lambda\eta} \approx 1 - \lambda\eta$, and the factor $\lambda\eta^2$ for the term $F(\boldsymbol{x}_k)/\boldsymbol{v}_k$ is too small and can be ignored compared with η. Finally, in stochastic setting, one can use the gradient estimation $\boldsymbol{m}_k$ to estimate the full gradient $\nabla F(\boldsymbol{x}_k)$, and thus achieves $\boldsymbol{x}_{k+1} = (1 - \lambda\eta)\boldsymbol{x}_k - \eta\boldsymbol{m}_k/\boldsymbol{v}_k$ which accords with the update (2) of AdamW. From this process, one can also observe that the dynamic regularizer $\frac{\lambda}{2}\|\boldsymbol{x}\|_{\boldsymbol{v}_k}^2$ is induced by the decoupled weight decay $-\lambda\eta\boldsymbol{x}_k$ in AdamW. In the following, we will show that AdamW actually minimizes this dynamic function $F_k(\boldsymbol{x})$ instead of the vanilla loss $F(\boldsymbol{x})$.

4.1 Results on General Nonconvex Problems

Following many works which analyze adaptive gradient algorithms, *e.g.* (Tijmen & Geoffrey, 2012; Zhou et al., 2018; Zhuang et al., 2020; Guo et al., 2021; Xie et al., 2022), we first provide the convergence results of AdamW by using a constant learning rate η. Theorem 1 summarizes the main results on a general nonconvex problem with its proof in Appendix E.1.

Theorem 1. *Suppose Assumptions 1 and 2 hold, $\boldsymbol{x}_* \in \operatorname{argmin}_{\boldsymbol{x}} F(\boldsymbol{x})$. By setting $\eta \le \frac{\delta^{1.25} b\epsilon^2}{6(c_\infty^2+\delta)^{0.75}\sigma^2 L}$, $\boldsymbol{\beta}_1 \le \frac{\delta^{0.5} b\epsilon^2}{3(c_\infty^2+\delta)^{0.5}\sigma^2}$ and $\boldsymbol{\beta}_2 \in (0,1)$ for all iterations, after $T = \mathcal{O}\big(\max\big(\frac{c_\infty^{2.5} L\Delta\sigma^2}{\delta^{1.25} b\epsilon^4}, \frac{c_\infty^2\sigma^4}{\delta b^2\epsilon^4}\big)\big)$ iterations with $\Delta = F(\boldsymbol{x}_0) - F(\boldsymbol{x}_*)$, the sequence $\{\boldsymbol{x}_k\}_{k=0}^T$ generated by AdamW in Eqn. (2) obeys*

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\Big[\|\nabla \boldsymbol{F}_k(\boldsymbol{x}_k)\|_2^2\Big] \le \epsilon^2, \quad \frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\Big[\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|_{\boldsymbol{v}_k}^2\Big] \le \frac{\eta^2\epsilon^2}{4}, \quad \frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\Big[\|\boldsymbol{m}_k - \nabla F(\boldsymbol{x}_k)\|_2^2\Big] \le 8\epsilon^2. \tag{4}$$

Moreover, the total stochastic gradient complexity to achieve (4) is $\mathcal{O}\big(\max\big(\frac{c_\infty^{2.5} L\Delta\sigma^2}{\delta^{1.25}\epsilon^4}, \frac{c_\infty^2\sigma^4}{\delta b\epsilon^4}\big)\big)$.

Theorem 1 guarantees the convergence of AdamW on the general nonconvex problems. Specifically, within $T = \mathcal{O}\big(\max\big(\frac{c_\infty^{2.5} L\Delta\sigma^2}{\delta^{1.25} b\epsilon^4}, \frac{c_\infty^2\sigma^4}{\delta b^2\epsilon^4}\big)\big)$ iterations, the average gradient $\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\big[\|\nabla \boldsymbol{F}_k(\boldsymbol{x}_k)\|_2^2\big]$ is smaller than ϵ^2, indicating the convergence of AdamW. The second inequality in Eqn. (4) guarantees the small distance between two neighboring solutions $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k+1}$, also showing the good convergence behaviors of AdamW. The last inequality in Eqn. (4) reveals that the exponential moving average (EMA) $\boldsymbol{m}_k$ of all historical stochastic gradient is indeed very close to the full gradient $\nabla F(\boldsymbol{x}_k)$ and thus helps explain the success of EMA gradient estimation.

Besides, as shown in Theorem 1, to find an ϵ-accurate first-order stationary point (ϵ-ASP), the stochastic gradient complexity of AdamW is $\mathcal{O}\big(c_\infty^{2.5}\epsilon^{-4}\big)$ when ignoring some other constant factors like other algorithms (Zhuang et al., 2020; Guo et al., 2021; Xie et al., 2022). Such a complexity accords with the lower bound $\Omega(\epsilon^{-4})$ in (Arjevani et al., 2019; 2020) (up to constant factors). Compared with other optimizers, AdamW enjoys lower complexity than $\mathcal{O}\big(c_2^6\epsilon^{-4}\big)$ of Adabelief (Zhuang et al., 2020) and $\mathcal{O}\big(c_2\sqrt{d}\epsilon^{-4}\big)$ of LAMB (You et al., 2019) , especially on over-parameterized networks, where c_2 upper bounds the ℓ_2-norm of stochastic gradient. This is because for the d-dimensional gradient, compared with its ℓ_2-norm c_2, its ℓ_∞-norm c_∞ is usually much smaller, and can be $\sqrt{d}\times$ smaller for the best case. For Adam and its ℓ_2-regularized variant, since our Theorem 1 still holds for the cases where 1) $\lambda = 0$ or 2) the objective loss $F(\boldsymbol{x})$ is a combination of objective loss and an ℓ_2-regularization, they also enjoy the complexity $\mathcal{O}\big(c_\infty^{2.5}\epsilon^{-4}\big)$. Such complexity is superior than the previously known complexity $\mathcal{O}(c_\infty\sqrt{d}\epsilon^{-4})$ of Adam-type optimizers analyzed in (Zhou et al., 2018; Chen et al., 2021; Guo et al., 2021), *e.g.* Adam *with or without* ℓ_2-regularizer, AdaGrad (Duchi et al., 2011), AdaBound (Luo et al., 2018), *etc*. Though sharing the same complexity with (ℓ_2-regularized) Adam, AdamW separates the ℓ_2-regularizer with the loss objective via

the decoupled weight decay whose generalization benefits have been validated empirically in many works, *e.g.* (Touvron et al., 2021; Liu et al., 2021), and theoretically in our Sec. 5.

Now we investigate the convergence performance of AdamW when using a decayed learning rate η_k. Compared with the constant learning rate, this decay strategy is more widely used in practice, but is rarely investigated in other optimization analysis (e.g. (Zhou et al., 2018; You et al., 2019; Zhuang et al., 2020)) except for (Guo et al., 2021). Theorem 1 formally states our main results.

Theorem 2. *Suppose Assumptions 1 and 2 hold, and* $\boldsymbol{x}_* \in \operatorname{argmin}_{\boldsymbol{x}} F(\boldsymbol{x})$. *By setting* $\eta_k = \frac{\gamma\delta^{0.75}}{2(c_\infty^2+\delta)^{0.25}L\sqrt{k+1}}$, $\boldsymbol{\beta}_{1k} = \frac{\gamma}{\sqrt{k+1}}$ *and* $\boldsymbol{\beta}_{2k} = \boldsymbol{\beta}_2 \in (0,1)$ *with* $\gamma = \max\big(1, \frac{c_2^{0.25}L^{0.5}\Delta^{0.5}}{\delta^{0.125}\sigma}\big)$ *for the* k*-th training iteration, then to achieve the results in Eqn. (4) with* η *replaced by* η_1, *the stochastic gradient complexity of AdamW in Eqn. (2) is* $\mathcal{O}\big(\max\big(\frac{c_\infty^{1.25}L^{0.5}\Delta^{0.5}\sigma}{\delta^{0.625}\epsilon^4}\log\big(\frac{1}{\epsilon}\big), \frac{c_\infty\sigma^2}{\delta^{0.5}\epsilon^4}\log\big(\frac{1}{\epsilon}\big)\big)\big)$.

See its proof in appendix E.2. Theorem 2 shows that by using the decayed learning rate $\eta_k = \frac{\gamma\delta^{0.75}}{2(c_\infty^2+\delta)^{0.25}L\sqrt{k+1}}$ for the k-th iteration, AdamW can converge and share almost the same results (4) in Theorem 1 when it uses the constant learning rate. To achieve ϵ-ASP, the stochastic gradient complexity of AdamW with decayed learning rate is $\mathcal{O}\big(\max\big(\frac{c_\infty^{1.25}L^{0.5}\Delta^{0.5}\sigma}{\delta^{0.625}\epsilon^4}\log\big(\frac{1}{\epsilon}\big), \frac{c_\infty\sigma^2}{\delta^{0.5}\epsilon^4}\log\big(\frac{1}{\epsilon}\big)\big)\big)$ and slightly differs from the one $\mathcal{O}\big(\max\big(\frac{c_\infty^{2.5}L\Delta\sigma^2}{\delta^{1.25}\epsilon^4}, \frac{c_\infty^2\sigma^4}{\delta b\epsilon^4}\big)\big)$ of AdamW using constant learning rate. Indeed, by comparing each term in the complexity, decayed learning rate respectively improves the constant one by factors $\frac{c_\infty^{1.25}L^{0.5}\Delta^{0.5}\sigma}{\delta^{0.625}}\log^{-1}(\frac{1}{\epsilon})$ and $\frac{c_\infty^2\sigma^2}{\delta^{0.5}}\log^{-1}(\frac{1}{\epsilon})$. Consider the fact that $\frac{c_\infty^{1.25}L^{0.5}\Delta^{0.5}\sigma}{\delta^{0.625}}$ and $\frac{c_\infty\sigma^2}{\delta^{0.5}}$ are often large than $\log(\frac{1}{\epsilon})$ because the ℓ_1-norm upper bound c_∞ of stochastic gradient is often not small and δ is usually very small, *e.g.* 10^{-4} in default, decayed learning rate is superior than constant learning rate which accords with the practical observations. When 1) $\lambda = 0$ or 2) the loss $F(\boldsymbol{x})$ is a ℓ_2-regularized objective loss, Theorem 2 still holds. So the stochastic complexity in Theorem 2 is also applicable to Adam and its ℓ_2-regularized variant. Guo et al. (2021) proved the stochastic gradient complexity $\mathcal{O}\big(\max\big(\frac{c_\infty^{2.5}L^2\sigma^2}{\delta^{2.5}\epsilon^4}\log\big(\frac{1}{\epsilon}\big), \frac{c_\infty^2\sigma^4}{\delta^2\epsilon^4}\log\big(\frac{1}{\epsilon}\big)\big)\big)$ of Adam-type algorithms, *e.g.* Adam and its ℓ_2-regularized variant, with decayed learning rate, which, however, is inferior than the complexity in this work, since as aforementioned, δ is often very small.

4.2 Results on PŁ-conditioned Nonconvex Problems

In this work, we are also particularly interested in the nonconvex problems under PŁ condition, since as observed or proved in (Hardt & Ma, 2016; Xie et al., 2017; Li & Yuan, 2017; Zhou & Liang, 2018), deep learning models often satisfy PŁ condition at least around a local minimum. For this special nonconvex problem, we follow (Reddi et al., 2016; Guo et al., 2021), and divide the whole optimization into K stages. Specifically, for constant learning rate setting, AdamW uses learning rate η_k in the whole k-th stage; while for decayed learning rate setting, it uses a decayed η_{k_i} for the k-th stage which satisfies $\eta_{k_i} < \eta_{k_j}$ if $i > j$, where η_{k_i} denotes the learning rate of the i-th iteration of the k-th stage. Moreover, for both learning rate settings, at the k-th stage, AdamW is allowed to run T_k iterations for achieving $\mathbb{E}\left[\boldsymbol{F}_k(\boldsymbol{x}_k) - \boldsymbol{F}_k(\boldsymbol{x}_*)\right] \le \epsilon_k$, where $\boldsymbol{x}_* \in \operatorname{argmin}_{\boldsymbol{x}} F(\boldsymbol{x})$, $\boldsymbol{x}_k$ is the output of the k-stage and $\epsilon_k = \frac{1}{2^k}[\boldsymbol{F}_0(\boldsymbol{x}_0) - \boldsymbol{F}_0(\boldsymbol{x}_*)]$ denotes the optimization accuracy. See detailed Algorithm 2 in Appendix C. At below, we provide the convergence results of AdamW under both settings of constant or decayed learning rate in Theorem 3 with proof in appendix E.3.

Theorem 3. *Suppose Assumptions 1 and 2 hold, and* $\boldsymbol{x}_* \in \operatorname{argmin}_{\boldsymbol{x}} F(\boldsymbol{x})$. *Assume the loss* $\boldsymbol{F}_k(\boldsymbol{x}_k)$ *in (3) and* $\boldsymbol{F}_k(\boldsymbol{x}_*)$ *satisfy the PŁ condition in Assumption 3.*

1) *For constant learning rate setting, assume a constant learning rate* $\eta_k \le \frac{\delta^{1.25}\mu b\epsilon_k}{12(c_\infty^2+\delta)^{0.75}\sigma^2 L}$, *constant* $\boldsymbol{\beta}_{1k} \le \frac{\delta^{0.5}\mu b\epsilon_k}{6(c_\infty^2+\delta)^{0.5}\sigma^2}$ *and* $\boldsymbol{\beta}_{2k} \in (0,1)$ *at the* k*-th stage. We have the following two properties:*

- ***1.1)*** *For the* k*-th stage, AdamW runs at most* $T_k = \mathcal{O}\big(\max\big(\frac{c_\infty^{2.5}\Delta\sigma^2 L}{\delta^{1.25}\mu^2 b\epsilon_k^2}, \frac{c_\infty^2\sigma^4}{\delta\mu^2 b^2\epsilon_k^2}\big)\big)$ *iterations to achieve* $\mathbb{E}\left[\boldsymbol{F}_k(\boldsymbol{x}_k) - \boldsymbol{F}_k(\boldsymbol{x}_*)\right] \le \epsilon_k$, *where the output* $\boldsymbol{x}_k$ *is uniformly randomly selected from the sequence* $\{\boldsymbol{x}_{k_i}\}_{i=1}^{T_k}$ *at the* k*-th stage.*
- ***1.2)*** *For* K *stages, the total stochastic complexity is* $\mathcal{O}\big(\max\big(\frac{c_\infty^{2.5}L\Delta\sigma^2}{\delta^{1.25}\mu^2\epsilon^2}, \frac{c_\infty^2\sigma^4}{\delta\mu^2\epsilon^2}\big)\big)$ *to achieve*

$$\min_{1\le k\le K}\mathbb{E}\left[\boldsymbol{F}_k(\boldsymbol{x}_k) - \boldsymbol{F}_k(\boldsymbol{x}_*)\right] \le \epsilon. \tag{5}$$

2) *For decayed learning rate setting, we set* $\eta_{k_i} \leq \frac{\gamma\delta^{0.75}}{2(c_\infty^2+\delta)^{0.25}L\sqrt{i+1}}$, $\boldsymbol{\beta}_{1k_i} \leq \frac{\gamma}{\sqrt{i+1}}$, $\boldsymbol{\beta}_{2k_i} = \boldsymbol{\beta}_{2k} \in (0,1)$ *at the* i*-th iteration of the* k*-th stage where* $\gamma = \max\big(1, \frac{(c_\infty^2+\delta)^{0.125}L^{0.5}\Delta^{0.5}}{\delta^{0.125}\sigma}\big)$.

2.1) *For the* k*-th stage, AdamW runs at most* $T_k = \mathcal{O}\big(\max\big(\frac{c_\infty^{1.25}L^{0.5}\Delta_k^{0.5}\sigma}{\delta^{0.625}b\mu^2\epsilon_k^2}, \frac{c_\infty\sigma^2}{\delta^{0.5}b\mu^2\epsilon_k^2}\big)\log\big(\frac{1}{\epsilon_k}\big)\big)$ *iterations to achieve* $\mathbb{E}\left[\boldsymbol{F}_k(\boldsymbol{x}_k) - \boldsymbol{F}_k(\boldsymbol{x}_*)\right] \leq \epsilon_k$*, where the output* $\boldsymbol{x}_k$ *is randomly selected from the sequence* $\{\boldsymbol{x}_{k_i}\}_{i=1}^{T_k}$ *at the* k*-th stage according to the probability distribution* $\big\{\frac{\eta_{k_i}}{\sum_{j=1}^{T_k}\eta_{k_j}}\big\}_{i=1}^{T_k}$.

2.2) *For* K *stages, the total stochastic complexity is* $\mathcal{O}\big(\max\big(\frac{c_\infty^{1.25}L^{0.5}\sigma}{\delta^{0.625}\mu^2\epsilon^2}, \frac{c_\infty\sigma^2}{\delta^{0.5}\mu^2\epsilon^2}\big)\big)$ *to achieve* (5).

By inspecting Theorem 3, one can observe that AdamW can always converge under both constant and decayed learning rate settings. Moreover, by comparison, to achieve ϵ-ASP in Eqn. (5), decayed learning rate has the total stochastic complexity $\mathcal{O}\big(\max\big(\frac{c_\infty^{1.25}L^{0.5}\sigma}{\delta^{0.625}\mu^2\epsilon^2}, \frac{c_\infty\sigma^2}{\delta^{0.5}\mu^2\epsilon^2}\big)\big)$, and shows the superiority over the constant learning rate whose complexity is $\mathcal{O}\big(\max\big(\frac{c_\infty^{2.5}L\Delta\sigma^2}{\delta^{1.25}\mu^2\epsilon^2}, \frac{c_\infty^2\sigma^4}{\delta\mu^2\epsilon^2}\big)\big)$. This conclusion is consistent with the one on the general nonconvex problems. It should be also noted that the complexity of AdamW on this special nonconvex problems (i.e. with PŁ condition) enjoys lower complexity than the one on the general nonconvex problems, since PŁ condition ensures a convexity-alike landscape of the loss objective and thus can be optimized faster.

5 Generalization Analysis

Here we first investigate the generalization error of AdamW via analyzing its hypothesis posterior, and then compare AdamW with (ℓ_2-regularized) Adam in terms of generalization performance.

5.1 Generalization Results

Analysis on hypothesis posterior. As shown in the classical PAC-Bayesian framework (McAllester, 1999), there is strong relations between the generalization error bound and the hypothesis posterior learned by an algorithm. So we first analyze the hypothesis posterior learned by AdamW, and then accordingly investigate the generalization error bound of AdamW. Specifically, following the works (Xie et al., 2021; Mandt et al., 2016; Chaudhari & Soatto, 2018; Jastrzkebski et al., 2017; Zhu et al., 2018; Zhou et al., 2020), we also study the corresponding stochastic differential equations (SDEs) of an algorithm to investigate its posterior and generalization behaviors because of the similar convergence behaviors of an algorithm and its SDE. For analysis, here we follow (Staib et al., 2019), and consider the matrix form of the second-order moment $\boldsymbol{n}_{t+1} = (1-\boldsymbol{\beta}_2)\boldsymbol{n}_t + \boldsymbol{\beta}_2\boldsymbol{g}_t\boldsymbol{g}_t^\top$ which is actually the vanilla version of second-order moment in AdaGrad (Duchi et al., 2011) and can reveal very similar functionality of the diagonal approximation $\mathrm{diag}\left(\boldsymbol{n}_{t+1}\right)$. Meanwhile, because we analyze the local convergence around an optimum since Sec. 4 already guarantees the convergence of AdamW to a local minimum, the observed fisher information matrix $\frac{1}{n}\sum_{i=1}^n \nabla F(\boldsymbol{x}_t;\boldsymbol{\xi}_i)\nabla F(\boldsymbol{x}_t;\boldsymbol{\xi}_i)^\top$ can well approximate the Hessian matrix $\boldsymbol{H}_{\boldsymbol{x}_t}$ near a minimum (Pawitan, 2001; Jastrzkebski et al., 2017; Zhu et al., 2018). Then we follow these works and approximate $\boldsymbol{n}_t$ as the Hessian matrix:

$$\boldsymbol{n}_t \approx \boldsymbol{H}_{\boldsymbol{x}_t}. \tag{6}$$

The validity of this approximation is also proved in a recent work (Staib et al., 2019). Accordingly, the updating rule of AdamW can be formulated as follows:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta\boldsymbol{Q}_t\boldsymbol{m}_t - \eta\lambda\boldsymbol{x}_t = \boldsymbol{x}_t - \eta\boldsymbol{Q}_t\nabla F(\boldsymbol{x}_t) - \eta\lambda\boldsymbol{x}_t + \eta\boldsymbol{Q}_t\boldsymbol{u}_t, \tag{7}$$

where $\boldsymbol{u}_t = \nabla F(\boldsymbol{x}_t) - \boldsymbol{m}_t$ denotes gradient noise, $\boldsymbol{Q}_t = \boldsymbol{n}_t^{-\frac{1}{2}}$ is defined for brevity. In the above AdamW formulation, the small constant δ in (2) is ignored for convenience which but does not affect our following results. Then following (Mandt et al., 2017; Smith & Le, 2017; Chaudhari & Soatto, 2018; Zhu et al., 2018), we assume the gradient noise $\boldsymbol{u}_t$ satisfies Gaussian distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{x}_t})$ because of the Central Limit theory. Accordingly, one can write SDE of AdamW as follows:

$$\mathsf{d}\boldsymbol{x}_t = -\boldsymbol{Q}_t\nabla F(\boldsymbol{x}_t)\mathsf{d}t - \lambda\boldsymbol{x}_t\mathsf{d}t + \boldsymbol{Q}_t\left(2\boldsymbol{\Sigma}_t\right)^{\frac{1}{2}}\mathsf{d}\boldsymbol{\zeta}_t,$$

where $\mathsf{d}\boldsymbol{\zeta}_t \sim \mathcal{N}(0, \boldsymbol{I}\mathsf{d}t)$ and $\boldsymbol{\Sigma}_t = \frac{\eta}{2}\boldsymbol{C}_{\boldsymbol{x}_t}$ in which $\boldsymbol{C}_{\boldsymbol{x}_t}$ is defined in the above literatures as

$$\boldsymbol{C}_{\boldsymbol{x}_t} = \frac{1}{b}\Big[\frac{1}{n}\sum_{i=1}^n \nabla f(\boldsymbol{x}_t;\boldsymbol{\xi}_i)\nabla f(\boldsymbol{x}_t;\boldsymbol{\xi}_i)^\top - \nabla F(\boldsymbol{x}_t)\nabla F(\boldsymbol{x}_t)^\top\Big],$$

where n is the training sample number of a dataset, and b is minibatch size. Since we analyze the local convergence around an optimum, this means $\nabla F(\boldsymbol{x}_t) \approx 0$ and the variance of the gradient

noise would dominate (Mandt et al., 2017; Smith & Le, 2017; Chaudhari & Soatto, 2018; Zhu et al., 2018). So following these works, we approximate $\boldsymbol{C}_{\boldsymbol{x}_t}$ as

$$\boldsymbol{C}_{\boldsymbol{x}_t} \approx \frac{1}{b}\Big[\frac{1}{n}\sum\nolimits_{i=1}^{n} \nabla f(\boldsymbol{x}_t;\boldsymbol{\xi}_i)\nabla f(\boldsymbol{x}_t;\boldsymbol{\xi}_i)^\top\Big] \approx \frac{1}{b}\boldsymbol{H}_{\boldsymbol{x}_t}. \tag{8}$$

Based on these approximations, we can derive the distribution of the hypothesis posterior learnt by AdamW, and summarize the main results in Lemma 4. See its proof in Appendix F.1.

Lemma 4. *Assume the objective loss can be approximated by a second-order Taylor approximation, namely, $F(\boldsymbol{x}) = F(\boldsymbol{x}_*)+\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_*)^\top \boldsymbol{H}_*(\boldsymbol{x}-\boldsymbol{x}_*)$. Then the solution $\boldsymbol{x}_t$ of AdamW obeys a Gaussian distribution $\mathcal{N}(\boldsymbol{x}_*, \boldsymbol{M}_{AdamW})$ where the covariance matrix $\boldsymbol{M}_{AdamW} = \mathbb{E}\left[\boldsymbol{x}_t\boldsymbol{x}_t^\top\right]$ is defined*

$$\boldsymbol{M}_{AdamW} = \frac{\eta}{2B}\boldsymbol{U}(\boldsymbol{S}^{\frac{1}{2}} + \lambda\boldsymbol{I})^{-1}\boldsymbol{U}^\top,$$

where $\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top$ is the SVD of $\boldsymbol{H}_*$ in which $\boldsymbol{S} = \mathrm{diag}\,(\sigma_1, \sigma_2, \cdots, \sigma_d)$ with singular values $\{\sigma_i\}_{i=1}^d$.

Lemma 4 tells that AdamW can converge to a solution which concentrates around the minimum $\boldsymbol{x}_*$. This also guarantees the good convergence behaviors of AdamW but from a SDE aspect. Then we look at the effects of the decoupled weight decay parameter λ. By observing the covariance matrix $\boldsymbol{M}_{\text{AdamW}}$, one can see that all singular values of $\boldsymbol{M}_{\text{AdamW}}$ becomes smaller when λ increases. It indeed indicates that decoupled weight decay in AdamW can make the algorithm more stable, and also benefits its convergence to the minimizer $\boldsymbol{x}_*$ which often enjoys better test performance.

Generalization analysis. Based on the above posterior analysis, we employ PAC Bayesian framework (McAllester, 1999) to explicitly analyze the generalization performance of AdamW. Given an algorithm $\mathcal{A}$ and a training dataset $\mathcal{D}_{\text{tr}}$ whose samples $\boldsymbol{\xi}$ are drawn from an unknown distribution $\mathcal{D}$, one often trains a model to obtain a posterior hypothesis $\boldsymbol{x}$ drawn from a hypothesis distribution $\mathcal{P} \sim \mathcal{N}(\boldsymbol{x}_*, \boldsymbol{M}_{\text{AdamW}})$ in Lemma 4. Then we denote the expected risk w.r.t. the hypothesis distribution $\mathcal{P}$ as $\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{D},\boldsymbol{x}\sim\mathcal{P}}[f(\boldsymbol{x},\boldsymbol{\xi})]$ and the empirical risk w.r.t. the distribution $\mathcal{P}$ as $\mathbb{E}_{\boldsymbol{\xi}\in\mathcal{D}_{\text{tr}},\boldsymbol{x}\sim\mathcal{P}}[f(\boldsymbol{x},\boldsymbol{\xi})]$. In practice, one often assumes the prior hypothesis satisfies Gaussian distribution $\mathcal{P}_{\text{pre}} \sim \mathcal{N}(\boldsymbol{0}, \rho\boldsymbol{I})$ (Lin et al., 2013; Simonyan & Zisserman, 2014; Szegedy et al., 2015; He et al., 2016; Dosovitskiy et al., 2020), because we do not know any information on the posterior hypothesis. Based on Lemma 4, we can derive the generalization error bound of AdamW in Theorem 5.

Theorem 5. *Assume the prior hypothesis $\boldsymbol{x}_0$ satisfies $\mathcal{P}_{pre} \sim \mathcal{N}(\boldsymbol{0}, \rho\boldsymbol{I})$. Then the expected risk for the posterior hypothesis $\boldsymbol{x} \sim \mathcal{P}$ of AdamW learned on training dataset $\mathcal{D}_{tr} \sim \mathcal{D}$ with n samples holds*

$$\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{D},\boldsymbol{x}\sim\mathcal{P}}[f(\boldsymbol{x},\boldsymbol{\xi})] - \mathbb{E}_{\boldsymbol{\xi}\in\mathcal{D}_{tr},\boldsymbol{x}\sim\mathcal{P}}[f(\boldsymbol{x},\boldsymbol{\xi})] \leq \frac{\sqrt{8}}{\sqrt{n}}\bigg(\sum_{i=1}^{d}\log\frac{2\rho b(\sigma_i^{\frac{1}{2}}+\lambda)}{\eta} + \frac{\eta}{2\rho b}\sum_{i=1}^{d}\frac{1}{\sigma_i^{\frac{1}{2}}+\lambda} + c_0\bigg)^{\frac{1}{2}},$$

with at least probability $1-\tau$, where $\tau \in (0,1)$ and $c_0 = \frac{1}{2\rho}\|\boldsymbol{x}_*\|^2 - \frac{d}{2} + 2\ln\left(\frac{2n}{\tau}\right)$.

See its proof in Appendix F.2. Theorem 5 guarantees that the generalization error of AdamW on a general learning problem can be upper bounded by $\mathcal{O}\big(\frac{1}{\sqrt{n}}\big)$ (up to other factors) which matches the error bound in (Vapnik, 2006; Hardt et al., 2016; Zhou & Feng, 2018a;b; Shalev-Shwartz & Ben-David, 2014) which but are derived from PAC theory or stability or uniform convergence aspects.

Then we inspect the effect of the decoupled weight decay parameter λ to the upper bound of AdamW. When λ increases, the first term $\sum_{i=1}^{d}\log 2\rho b(\sigma_i^{\frac{1}{2}}+\lambda)\eta^{-1}$ in the bound becomes larger, while the second term $\frac{\eta}{2\rho b}\sum_{i=1}^{d}(\sigma_i^{\frac{1}{2}}+\lambda)^{-1}$ in the bound decreases. Though in practice, it is hard to precisely decide the value of λ, from the above discussion, at least we know that tuning λ can yield smaller generalization error. Since the empirical risk $\mathbb{E}_{\boldsymbol{\xi}\in\mathcal{D}_{\text{tr}},\boldsymbol{x}\sim\mathcal{P}}[f(\boldsymbol{x},\boldsymbol{\xi})]$ is often small especially for a modern over-parametrized network, a proper λ can benefit AdamW in terms of the expected risk error, explaining the better test performance of AdamW over vanilla Adam (without ℓ_2-regularization) which corresponds to $\lambda = 0$ in AdamW. Finally, our result is the first one that theoretically and explicitly show the benefits of the decoupled weight decay to the generalization of AdamW.

5.2 Comparison with ℓ_2-regularized Adam

Now we compare AdamW (i.e. decoupled weight decay) with ℓ_2-regularized Adam (i.e. conventional weigh decay). To diminish the effects of historical gradient to the current optimization and also analyze the effects of current gradient to the behaviors of adaptive algorithms, many works, *e.g.* (Lyu et al., 2022; Malladi et al., 2022), set $\boldsymbol{\beta}_1 = \boldsymbol{\beta}_2 = 1$ in (2) to focus on concurrent optimiza-

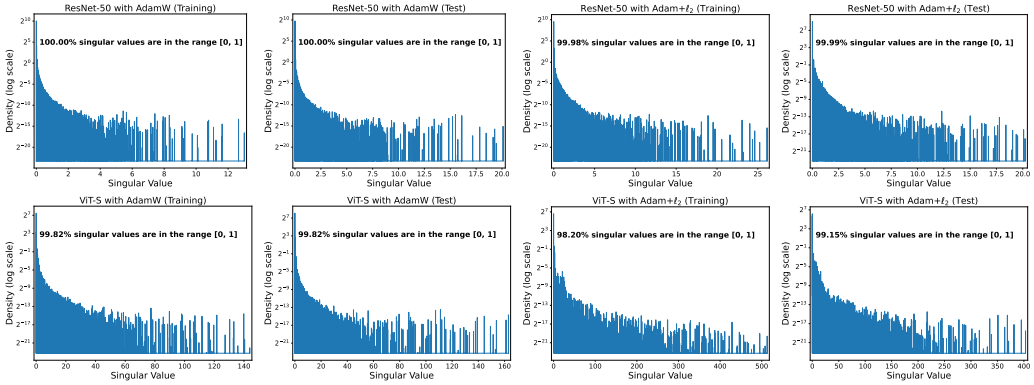


Figure 1: Visualization of singular values in ResNet50 and ViT-small trained by AdamW and ℓ_2-regularized Adam. See more visualization results, *e.g.* ResNet18, in Fig. 3 of Appendix B.

tion process of adaptive algorithms. Here we also follow this setting to investigate ℓ_2-regularized Adam whose updating rule can be formulated as follows:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \boldsymbol{Q}_t(\boldsymbol{m}_t + \lambda \boldsymbol{x}_t) = \boldsymbol{x}_t - \eta \boldsymbol{Q}_t(\nabla F(\boldsymbol{x}_t) + \lambda \boldsymbol{x}_t) + \eta \boldsymbol{Q}_t \boldsymbol{u}_t,$$

where both $\boldsymbol{u}_t = \nabla F(\boldsymbol{x}_t) - \boldsymbol{m}_t$ and $\boldsymbol{Q}_t = \boldsymbol{n}_t^{-\frac{1}{2}}$ have the same meanings in Eqn. (7). Then similarly, one can write the SDE of ℓ_2-regularized Adam as follows:

$$\mathsf{d}\boldsymbol{x}_t = -\boldsymbol{Q}_t(\nabla F(\boldsymbol{x}_t) + \lambda \boldsymbol{x}_t)\mathsf{d}t + \boldsymbol{Q}_t \left(2\boldsymbol{\Sigma}_t\right)^{\frac{1}{2}} \mathsf{d}\boldsymbol{\zeta}_t,$$

where $\mathsf{d}\boldsymbol{\zeta}_t \sim \mathcal{N}(0, \boldsymbol{I}\mathsf{d}t)$ and $\boldsymbol{\Sigma}_t = \frac{\eta}{2}\boldsymbol{C}_{\boldsymbol{x}_t}$ in which $\boldsymbol{C}_{\boldsymbol{x}_t}$ is given in (8).

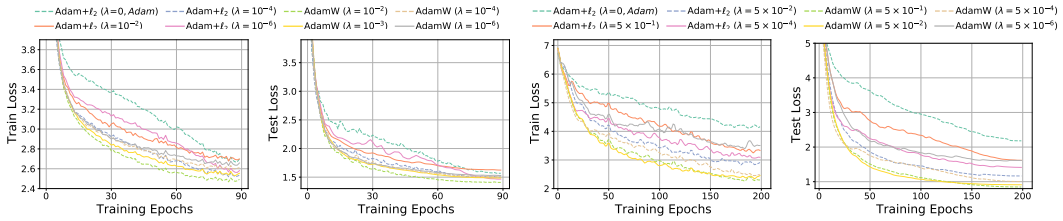
Theorem 6. *Assume the prior hypothesis $\boldsymbol{x}_0$ satisfies $\mathcal{P}_{pre} \sim \mathcal{N}(\boldsymbol{0}, \rho \boldsymbol{I})$. Then with at least probability $1-\tau$ and a constant c_0 in Theorem 5, the expected risk for the posterior hypothesis $\boldsymbol{x} \sim \mathcal{P}_{Adam+\ell_2}$ of ℓ_2-regularized Adam on a training dataset $\mathcal{D}_{tr} \sim \mathcal{D}$ with n samples can be upper bounded by*

$$\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{D}, \boldsymbol{x}\sim\mathcal{P}_{Adam+\ell_2}}[f(\boldsymbol{x},\boldsymbol{\xi})] - \mathbb{E}_{\boldsymbol{\xi}\in\mathcal{D}_{tr}, \boldsymbol{x}\sim\mathcal{P}_{Adam+\ell_2}}[f(\boldsymbol{x},\boldsymbol{\xi})] \leq \frac{\sqrt{8}}{\sqrt{n}}\left(\sum_{i=1}^{d} \log \frac{2\rho b(\sigma_i+\lambda)}{\eta \sigma_i^{\frac{1}{2}}} + \frac{\eta}{2\rho b}\sum_{i=1}^{d}\frac{\sigma_i^{\frac{1}{2}}}{\sigma_i+\lambda} + c_0\right)^{\frac{1}{2}}.$$

See its proof in Appendix F.3. Theorem 6 shows the generalization error bound $\mathcal{O}\big(\frac{1}{\sqrt{n}}\big)$ (up to other factors) of ℓ_2-regularized Adam. Moreover, when $\lambda = 0$, AdamW and ℓ_2-regularized Adam are exactly the same, and their error bounds are also the same as shown in Theorems 5 and 6.

Next, we compare the generalization error bounds of AdamW and ℓ_2-regularized Adam by comparing their different terms, i.e. $\mathsf{err}_{\text{adamw}} = \sum_{i=1}^{d} h(x_{\text{adamw}}^{(i)})$ with $x_{\text{adamw}}^{(i)} = 2\eta^{-1}\rho b(\sigma_i^{\frac{1}{2}} + \lambda)$ in AdamW and $\mathsf{err}_{\text{adam+}\ell_2} = \sum_{i=1}^{d} h(x_{\text{adam+}\ell_2}^{(i)})$ with $x_{\text{adam+}\ell_2}^{(i)} = 2\eta^{-1}\rho b(\sigma_i^{\frac{1}{2}} + \lambda\sigma_i^{-\frac{1}{2}})$ in ℓ_2-regularized Adam, where $h(x) = \log x + \frac{1}{x}$. For $h(x)$, we know $h'(x) = \frac{x-1}{x^2}$ and thus $h(x)$ will increase when $x \in (1, +\infty)$. Meanwhile, generally, we have $x_{\text{adam+}\ell_2}^{(i)} > x_{\text{adamw}}^{(i)} > 1$ for most $i \in [d]$ because of the following reasons. 1) Most of the singular values $\{\sigma_i\}_{i=1}^{d}$ of Hessian matrix in deep networks are much smaller than one which is well observed in many works, *e.g.* fully connected networks, AlexNet, VGG and ResNet (Sagun et al., 2016; 2017; Ghorbani et al., 2019; Sankar et al., 2021) and our experimental results on ResNet50 and ViT-small in Fig. 1. 2) The learning rate when reaching the minimum is often set to be very small in practice. 3) The minibatch size b is of order of thousand to train a modern network, and the variance ρ for the Gaussian initialization distribution $\mathcal{P}_{\text{pre}} \sim \mathcal{N}(\boldsymbol{0}, \rho \boldsymbol{I})$ is often at the order of $\mathcal{O}(1/\sqrt{d_i})$ (Glorot & Bengio, 2010; He et al., 2015), where d_i denotes the input dimension. So these factors together would indicate $x_{\text{adam+}\ell_2}^{(i)} > x_{\text{adamw}}^{(i)} > 1$. So the generalization error term $\mathsf{err}_{\text{adamw}}$ of AdamW is smaller than $\mathsf{err}_{\text{adam+}\ell_2}$ of ℓ_2-regularized Adam, which actually is also empirically testified by our experimental results on ResNet18, ResNet50 and ViT-small in Sec. 6. So AdamW often enjoys better generalization performance than ℓ_2-regularized Adam, also validated in Sec. 6. This actually shows the superiority of decoupled weight decay in AdamW over the conventional weight decay used in Adam.

The above analysis shows the generalization benefits of the dynamic regularization $\frac{\lambda}{2}\|\boldsymbol{x}\|_{\boldsymbol{v}_k}^2$ of AdamW in Eqn. (3). So we hope the theoretical results in this work could motivate more explo-



(a) ResNet18 (b) ViT-small

Figure 2: Training and test curves comparison on ImageNet. See more results in Appendix B.

Table 1: Generalization investigation of AdamW and ℓ-regularized Adam on ImageNet.

model	ResNet18		ResNet50		ViT-small					
train epoch	90		100		100		200		300	
optimizer	AdamW	Adam+ℓ_2	AdamW	Adam+ℓ_2	AdamW	Adam+ℓ_2	AdamW	Adam+ℓ_2	AdamW	Adam+ℓ_2
err in bound	3.43	3.85	3.42	3.78	3.62	3.75	3.58	3.72	3.47	3.70
test acc. (%)	67.9	67.2	77.0	76.5	76.1	75.3	79.2	77.6	79.8	78.5

ration on how to improve this dynamic regularization or propose a new dynamic regularizer for further generalization improvement in adaptive gradient algorithms.

6 Experiments

Investigation on singular values of Hessian. We first respectively use AdamW and ℓ_2-regularized Adam to train two popular network architectures on ImageNet (Deng et al., 2009), i.e. ResNet50 (He et al., 2016) and vision transformer small (ViT-small) (Dosovitskiy et al., 2020) for both 100 epochs. Then we adopt the method in (Yao et al., 2020) to estimate the singular values of these two trained networks. Fig. 1 plots the spectral density of these singular values on both training and test data of ImageNet, and shows that there are more than 99% singular values that are in the range $[0, 1]$ and indeed are much smaller than one. This also accords with the observations in (Sagun et al., 2016; 2017; Ghorbani et al., 2019; Sankar et al., 2021) that most of the singular values are much smaller than one in deep networks, *e.g.* AlexNet (Krizhevsky et al., 2017), VGG (Simonyan & Zisserman, 2014) and ResNet (He et al., 2016). All these observations support the results in Sec. 5.2.

Investigation on generalization. We compare the different terms, i.e. $\mathsf{err}_{\text{adamw}}$ and $\mathsf{err}_{\text{adam+}\ell_2}$ defined at end of Sec. 5.2, in the generalization error bounds of AdamW and ℓ_2-regularized Adam. To this end, we receptively train three models, including ResNet18, ResNet50 and ViT-small, on ImageNet by using AdamW and ℓ_2-regularized Adam, and well tune the hyper-parameters of these two optimizers, *e.g.* learning rate and weight decay (regularization) parameter λ. Note, here ℓ_2-regularized Adam includes Adam by setting $\lambda = 0$. Next, we compute $\mathsf{err}_{\text{adamw}}$ and $\mathsf{err}_{\text{adam+}\ell_2}$ on the test dataset of ImageNet, since test dataset can better reveal the generalization ability of an algorithm. Table 1 shows that on all test cases, $\mathsf{err}_{\text{adamw}}$ is often smaller than $\mathsf{err}_{\text{adam+}\ell_2}$ by a remarkable margin. These results empirically support the superior generalization error bound of AdamW over ℓ_2-regularized Adam. Moreover, Table 1 also reveals the higher test accuracy of AdamW over ℓ_2-regularized Adam. All these results accord with our theoretical results in Sec. 5.2.

Investigation on convergence. We also plot the training and test curves of AdamW and ℓ_2-regularized Adam on ImageNet in Fig. 2. See multiple experimental trials in Fig. 4 of Appendix B. One can find that on ResNet50 and ViT-small, 1) AdamW shows slightly faster convergence speed than Adam and ℓ_2-regularized Adam when their weight decay parameter are well-tuned, *e.g.* $\lambda = 5\times10^{-1}$ for AdamW and $\lambda = 5\times10^{-2}$ for ℓ_2-regularized Adam on ViT-small; 2) weight decay (regularization) parameter λ has big effects to the convergence speed of AdamW and ℓ_2-regularized Adam. So under the same computational cost, the faster convergence of AdamW could also partially explain the above better generalization performance of AdamW over (ℓ_2-regularized) Adam.

7 Conclusion

In this work, we first prove the convergence of AdamW using both constant and decayed learning rates on the general nonconvex problems and PŁ-conditioned problems. Moreover, we find that AdamW provably minimizes a dynamically regularized loss that combines a vanilla loss and a dynamical regularization, and thus its behaviors differ from those in (ℓ_2-regularized) Adam. Besides, for the first time, we explicitly justify the generalization superiority of AdamW over both Adam and its ℓ_2-regularized variant. Finally, experimental results validate the implications of our theory.

REFERENCES

O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10):1533–1545, 2014.

Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

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

Z. Charles and D. Papailiopoulos. Stability and generalization of learning algorithms that converge to global optima. In *Proc. Int'l Conf. Machine Learning*, pp. 745–754. PMLR, 2018.

Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–10. IEEE, 2018.

Jinghui Chen, Dongruo Zhou, Yiqi Tang, Ziyan Yang, Yuan Cao, and Quanquan Gu. Closing the generalization gap of adaptive gradient methods in training deep neural networks. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 3267–3275, 2021.

Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. *arXiv preprint arXiv:1808.02941*, 2018.

J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pp. 2232–2241. PMLR, 2019.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Zhishuai Guo, Yi Xu, Wotao Yin, Rong Jin, and Tianbao Yang. A novel convergence analysis for algorithms of the adam family and beyond. *arXiv e-prints*, pp. arXiv–2104, 2021.

M. Hardt and T. Ma. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pp. 1225–1234. PMLR, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

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

Elad Hoffer, Ron Banner, Itay Golan, and Daniel Soudry. Norm matters: efficient and accurate normalization schemes in deep networks. *Advances in Neural Information Processing Systems*, 31, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Stanislaw Jastrzkebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from adam to sgd. *arXiv preprint arXiv:1712.07628*, 2017.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. *Int'l Conf. Learning Representations*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Z. Li and Y. Yuan. Convergence analysis of two-layer neural networks with relu activation. In *Proc. Conf. Neural Information Processing Systems*, 2017.

Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. In *International Conference on Learning Representations*, 2018.

Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. *arXiv preprint arXiv:1902.09843*, 2019.

Kaifeng Lyu, Zhiyuan Li, and Sanjeev Arora. Understanding the generalization benefit of normalization layers: Sharpness reduction. *arXiv preprint arXiv:2206.07085*, 2022.

Sadhika Malladi, Kaifeng Lyu, Abhishek Panigrahi, and Sanjeev Arora. On the sdes and scaling rules for adaptive gradient algorithms. *arXiv preprint arXiv:2205.10287*, 2022.

S. Mandt, M. Hoffman, and D. Blei. A variational analysis of stochastic gradient algorithms. In *Proc. Int'l Conf. Machine Learning*, pp. 354–363, 2016.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *arXiv preprint arXiv:1704.04289*, 2017.

David A McAllester. Some pac-bayesian theorems. *Machine Learning*, 37(3):355–363, 1999.

Leandro Pardo. *Statistical inference based on divergence measures*. Chapman and Hall/CRC, 2018.

I. Pavlyukevich. First exit times of solutions of stochastic differential equations driven by multiplicative lévy noise with heavy tails. *Stochastics and Dynamics*, 11(02n03):495–519, 2011.

Yudi Pawitan. *In all likelihood: statistical modelling and inference using likelihood*. Oxford University Press, 2001.

Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pp. 314–323. PMLR, 2016.

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

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

Levent Sagun, Leon Bottou, and Yann LeCun. Eigenvalues of the hessian in deep learning: Singularity and beyond. *arXiv preprint arXiv:1611.07476*, 2016.

Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

T. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran. Deep convolutional neural networks for LVCSR. In *ICASSP*, pp. 8614–8618. IEEE, 2013.

Adepu Ravi Sankar, Yash Khasbage, Rahul Vigneswaran, and Vineeth N Balasubramanian. A deeper look at the hessian eigenspectrum of deep neural networks and its applications to regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9481–9488, 2021.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

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

U. Simsekli, L. Sagun, and M. Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In *Proc. Int'l Conf. Machine Learning*, 2019.

Samuel L Smith and Quoc V Le. A bayesian perspective on generalization and stochastic gradient descent. *arXiv preprint arXiv:1710.06451*, 2017.

Matthew Staib, Sashank Reddi, Satyen Kale, Sanjiv Kumar, and Suvrit Sra. Escaping saddle points with adaptive gradient methods. In *International Conference on Machine Learning*, pp. 5956–5965. PMLR, 2019.

Endre Süli and David F Mayers. *An introduction to numerical analysis*. Cambridge university press, 2003.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Tieleman Tijmen and Hinton Geoffrey. Lecture 6.5-rmsprop: Divide the gradient by a run- ning average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4, 2012.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Twan Van Laarhoven. L2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*, 2017.

Vladimir Vapnik. *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006.

Bo Xie, Yingyu Liang, and Le Song. Diverse neural network learns true target functions. In *Artificial Intelligence and Statistics*, pp. 1216–1224. PMLR, 2017.

Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, and Shuicheng Yan. Adan: Adaptive nesterov momentum algorithm for faster optimizing both cnns and vits. *Axriv*, 2022.

Zeke Xie, Li Yuan, Zhanxing Zhu, and Masashi Sugiyama. Positive-negative momentum: Manipulating stochastic gradient noise to improve generalization. In *International Conference on Machine Learning*, pp. 11448–11458. PMLR, 2021.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, pp. 581–590. IEEE, 2020.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations*, 2019.

Guodong Zhang, Chaoqi Wang, Bowen Xu, and Roger Grosse. Three mechanisms of weight decay regularization. *arXiv preprint arXiv:1810.12281*, 2018.

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.

Pan Zhou and Jiashi Feng. Empirical risk landscape analysis for understanding deep neural networks. In *International conference on learning representations*, 2018a.

Pan Zhou and Jiashi Feng. Understanding generalization and optimization performance of deep cnns. In *International Conference on Machine Learning*, pp. 5960–5969. PMLR, 2018b.

Pan Zhou, Jiashi Feng, Chao Ma, Caiming Xiong, Steven Chu Hong Hoi, et al. Towards theoretically understanding why sgd generalizes better than adam in deep learning. *Advances in Neural Information Processing Systems*, 33:21285–21296, 2020.

Pan Zhou, Hanshu Yan, Xiaotong Yuan, Jiashi Feng, and Shuicheng Yan. Towards understanding why lookahead generalizes better than sgd and beyond. In *Neural Information Processing Systems*, 2021a.

Pan Zhou, Yichen Zhou, Chenyang Si, Weihao Yu, Teck Khim Ng, and Shuicheng Yan. Mugs: A multi-granular self-supervised learning framework. In *arXiv preprint arXiv:2203.14415*, 2022.

Y. Zhou and Y. Liang. Characterization of gradient dominance and regularity conditions for neural networks. In *Int'l Conf. Learning Representations*, 2018.

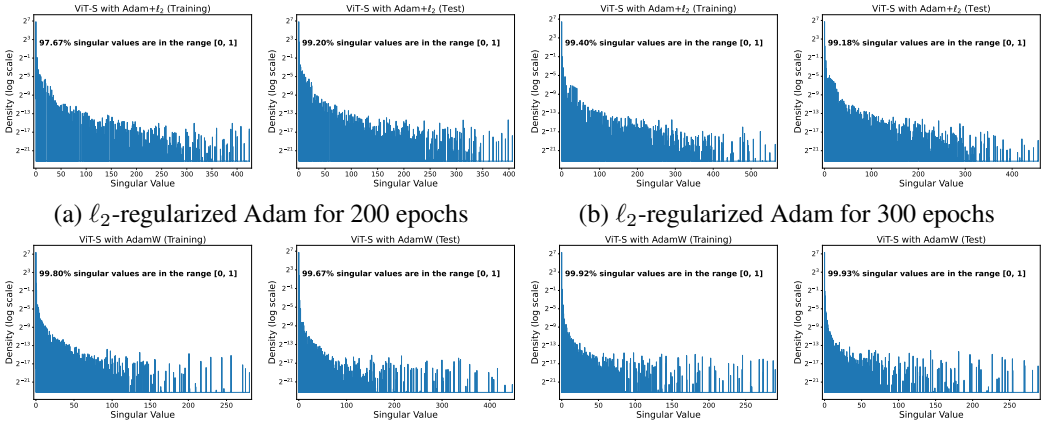
Yucong Zhou, Yunxiao Sun, and Zhao Zhong. Fixnorm: Dissecting weight decay for training deep neural networks. *arXiv preprint arXiv:2103.15345*, 2021b.

Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. *arXiv preprint arXiv:1803.00195*, 2018.

Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *Advances in Neural Information Processing Systems*, 33:18795–18806, 2020.

A APPENDIX

The appendix contains the technical proofs of convergence results and some additional experimental results of the paper entitled "Towards Understanding Convergence and Generalization of AdamW". It is structured as follows. In Appendix C, we first give the detailed algorithmic frameworks of AdamW and its stagewise variant in Algorithms 1 and 2. Then Appendix D provides some auxiliary lemmas throughout this document. Then Appendix E presents the proof of the convergence results in Sec. 4, i.e., the proof of Theorems 1 $\sim$ 3. Next, we introduce the proof of generalization results in Sec. 5, including Lemma 4 and Theorems 5 and 6. Finally, Appendix G provides the proofs of some auxiliary lemmas in Appendix D.



(a) ℓ_2-regularized Adam for 200 epochs (b) ℓ_2-regularized Adam for 300 epochs

(c) AdamW for 200 epochs (d) AdamW for 300 epochs

Figure 3: Visualization of singular values in ViT-small trained by ℓ_2-regularized Adam and AdamW for 200 and 300 epochs.

B MORE EXPERIMENTAL RESULTS

Here we give more experimental investigation on singular values of Hessian in deep networks. In the manuscript, we provide investigation by training ResNet50 (He et al., 2016) and vision transformer small (ViT-small) (Dosovitskiy et al., 2020) for both 100 epochs. Here we provide more visualization results of ResNet50 (He et al., 2016) and vision transformer small (ViT-small) (Dosovitskiy et al., 2020) trained by 200 and 300 epochs. Similarly, we adopt the singular value estimation method in (Yao et al., 2020) to estimate the singular values of these two trained networks. Fig. 3 plots the spectral density of these singular values, and shows that there are more than 99% singular values that are in the range $[0, 1]$ and indeed are much smaller than one. All these results also accords with the observations on ResNet50 and ViT-small trained by 100 epochs. All these observations support the results in Sec. 5.2.

Algorithm 1: AdamW (Loshchilov & Hutter, 2018)

Input: initialization $\boldsymbol{\theta}_0$, step size $\{\eta_k\}_{k=0}^T$, hyper-parameters $\{\boldsymbol{\beta}_{1k}\}_{k=0}^T$ and $\{\boldsymbol{\beta}_{2k}\}_{k=0}^T$ for first- and second-order moments $\{\boldsymbol{m}_k\}_{k=0}^T$ and $\{\boldsymbol{n}_k\}_{k=0}^T$.

Output: some average of $\{\boldsymbol{x}_k\}_{k=0}^T$.

while $k < T$ **do**

  estimate stochastic gradient $\boldsymbol{g}_k = \frac{1}{b}\sum_{i=1}^b \nabla f(\boldsymbol{x}_k; \boldsymbol{\xi}_i)$;

  estimate first-order moment $\boldsymbol{m}_k = (1-\boldsymbol{\beta}_{1k})\boldsymbol{m}_k + \boldsymbol{\beta}_{1k}\boldsymbol{g}_k$;

  estimate second-order moment $\boldsymbol{n}_k = (1-\boldsymbol{\beta}_{2k})\boldsymbol{n}_k + \boldsymbol{\beta}_{2k}\boldsymbol{g}_k^2$;

  update parameter $\boldsymbol{x}_{k+1} = (1-\lambda\eta_k)\boldsymbol{x}_k - \eta_k \boldsymbol{m}_k/\sqrt{\boldsymbol{n}_k + \delta}$;

end while

For multiple trials of the experiments, we independently test AdamW on ResNet18 by using three different seeds, and plot the average and variance in Fig. 4. Similarly, we evaluate ℓ_2-regularized

Algorithm 2: Stagewise AdamW

Input: initialization $\boldsymbol{\theta}_0$, optimization accuracy $\{\epsilon_k\}_{k=1}^K$.
Output: some average of $\{\boldsymbol{x}_k\}_{k=0}^T$.
while $k < K$ **do**
  optimize the loss objective by AdamW (algorithm 1) to accuracy ϵ_k, and output solution $\boldsymbol{x}_k$;
end while

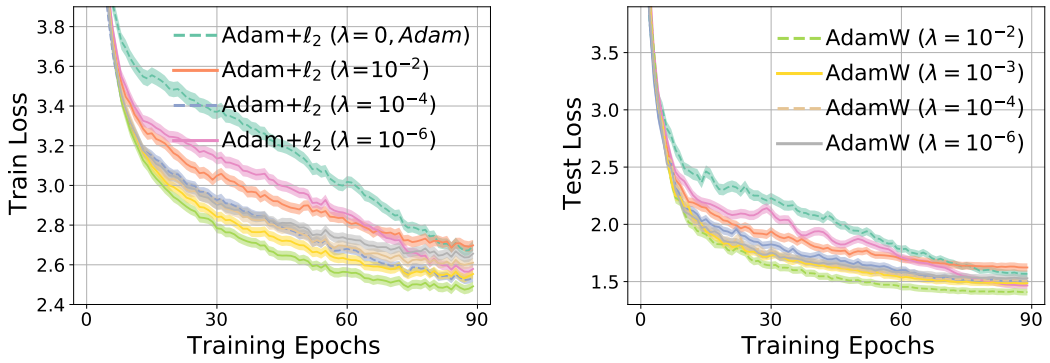


Figure 4: Training and test curves comparison on ImageNet. We independently test AdamW on ResNet18 by using three different seeds, and plot the average and variance. Similarly, we evaluate ℓ_2-regularized Adam with three different seeds.

Adam with three different seeds. From Fig. 4, one can observe that the performance of these algorithms are stable and consistent. But one can also observe there are too many curves especially with the variance boundary which weakens the readability. So we only put Fig. 4 in the appendix and mention readers to refer this Fig. 4 for multiple experiment trials.

C Details of AdamW and its Stagewise Variant

Due to space limitation, in the manuscript, we do not provide the detailed AdamW. Here we give algorithmic framework of AdamW in Algorithm 1 to help understand. Since in Sec. 4.2 we further propose the stagewise AdamW algorithm to solve PŁ-conditioned nonconvex problems, here we also provide the algorithmic framework of stagewise AdamW in Algorithm 2.

D Auxiliary Lemmas

Before giving our analysis, we first provide some important lemmas.

Lemma 7. *Assume $c_{s,\infty} \leq \|g_k\|_\infty \leq c_\infty$, then we have*

$$\|\boldsymbol{m}_k\|_\infty \leq c_\infty, \quad \|\boldsymbol{n}_i + \delta\|_\infty \leq c_\infty^2 + \delta, \quad \left\| \frac{(\boldsymbol{n}_k + \delta)^p}{(\boldsymbol{n}_{k+1} + \delta)^p} \right\|_\infty \in [1-\mu, 1+\mu] \ \ (\forall p \in [0,1]),$$

where $\mu = \frac{\boldsymbol{\beta}_2 c_\infty^2}{c_{s,\infty}^2 + \delta}$.

See its proof in Appendix G.1.

Lemma 8. *(Xie et al., 2022) The sequence $\{\boldsymbol{x}_k\}_{k=0}^T$ generated by AdamW in Eqn. (2) satisfies*

$$\frac{\lambda_{k+1}}{1-\mu} \|\boldsymbol{x}_{k+1}\|_{\boldsymbol{v}_k}^2 \leq \lambda_k \|\boldsymbol{x}_k\|_{\boldsymbol{v}_k}^2 + \lambda \langle \boldsymbol{x}_{k+1} - \boldsymbol{x}_k, \boldsymbol{x}_k \rangle_{\boldsymbol{v}_k} + \frac{\lambda}{2} \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_{\boldsymbol{v}_k}^2 .$$

Lemma 9. *(Xie et al., 2022) The sequence $\{\boldsymbol{x}_k\}_{k=0}^T$ generated by AdamW in Eqn. (2) satisfies*

$$\begin{aligned}&\mathbb{E}\left[\left\|\boldsymbol{m}_k-\nabla F(\boldsymbol{x}_k)\right\|^2\right]\\ \leq&(1-\boldsymbol{\beta}_1)\mathbb{E}\left[\left\|\boldsymbol{m}_{k-1}-\nabla F(\boldsymbol{x}_{k-1})\right\|^2\right]+\frac{(1-\boldsymbol{\beta}_1)^2L^2}{\boldsymbol{\beta}_1}\mathbb{E}\left[\left\|\boldsymbol{x}_k-\boldsymbol{x}_{k-1}\right\|^2\right]+\frac{\boldsymbol{\beta}_1^2\sigma^2}{b}.\end{aligned}$$

E Proof of Theorem 4

E.1 Proof of Theorem 1

Proof. For brevity, we let

$$\boldsymbol{v}_k=\sqrt{\boldsymbol{n}_k+\delta}.$$

When $\|\boldsymbol{g}_i\|_\infty\leq c_\infty$, we have $\|\boldsymbol{m}_k\|_\infty\leq c_\infty$ and $\delta\leq\|\boldsymbol{n}_i+\delta\|_\infty\leq c_\infty^2+\delta$ in Lemma 7. For brevity, let

$$c_1:=\delta^p\leq\|\boldsymbol{v}_k\|_\infty\leq c_2:=(c_\infty^2+\delta)^p.$$

Also we define

$$\boldsymbol{u}_k:=\boldsymbol{m}_k+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k,\qquad\boldsymbol{x}_{k+1}-\boldsymbol{x}_k=-\eta\frac{\boldsymbol{m}_k+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k}{\boldsymbol{v}_k}=-\eta\frac{\boldsymbol{u}_k}{\boldsymbol{v}_k}.$$

Moreover, we also define $\widetilde{F}_k(\boldsymbol{x}_k)$ as follows:

$$\widetilde{F}_k(\boldsymbol{x}_k)=F(\boldsymbol{x})+\lambda_k\left\|\boldsymbol{x}\right\|_{\boldsymbol{v}_k}^2=\mathbb{E}_{\boldsymbol{\xi}}[f(\boldsymbol{\theta};\boldsymbol{\xi})]+\lambda_k\left\|\boldsymbol{x}\right\|_{\boldsymbol{v}_k}^2,$$

where $\lambda_k=\frac{\lambda}{2}\sum_{i=1}^k\left(\frac{1-\mu}{2}\right)^i\ (k>0)$ and $\lambda_0=0$ in which $\mu=\frac{\boldsymbol{\beta}_2c_\infty^2}{\delta}$.

Then by using the smoothness of $f(\boldsymbol{x};\boldsymbol{\xi})$, we can obtain

$$\begin{aligned}&\widetilde{F}_{k+1}(\boldsymbol{x}_{k+1})\\ \leq&F(\boldsymbol{x}_k)+\langle\nabla F(\boldsymbol{x}_k),\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\rangle+\frac{L}{2}\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\right\|^2+\lambda_{k+1}\left\|\boldsymbol{x}_{k+1}\right\|_{\boldsymbol{v}_{k+1}}^2\\ \overset{①}{\leq}&F(\boldsymbol{x}_k)+\langle\nabla F(\boldsymbol{x}_k),\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\rangle+\frac{L}{2}\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\right\|^2+\frac{\lambda_{k+1}}{1-\mu}\left\|\boldsymbol{x}_{k+1}\right\|_{\boldsymbol{v}_k}^2\\ \overset{②}{\leq}&F(\boldsymbol{x}_k)+\lambda_k\left\|\boldsymbol{x}_k\right\|_{\boldsymbol{v}_k}^2+\langle\nabla F(\boldsymbol{x}_k)+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k,\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\rangle+\frac{L}{2}\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\right\|^2+\frac{\lambda}{2}\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\right\|_{\boldsymbol{v}_k}^2\\ =&\widetilde{F}_k(\boldsymbol{x}_k)-\eta\left\langle\nabla F(\boldsymbol{x}_k)+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k,\frac{\boldsymbol{u}_k}{\boldsymbol{v}_k}\right\rangle+\frac{L\eta^2}{2}\left\|\frac{\boldsymbol{u}_k}{\boldsymbol{v}_k}\right\|^2+\frac{\lambda\eta^2}{2}\left\|\frac{\boldsymbol{u}_k}{\boldsymbol{v}_k}\right\|_{\boldsymbol{v}_k}^2\\ =&\widetilde{F}_k(\boldsymbol{x}_k)+\frac{1}{2}\left\|\sqrt{\frac{\eta}{\boldsymbol{v}_k}}\left(\nabla F(\boldsymbol{x}_k)+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k-\boldsymbol{u}_k\right)\right\|^2-\frac{1}{2}\left\|\sqrt{\frac{\eta}{\boldsymbol{v}_k}}\left(\nabla F(\boldsymbol{x}_k)+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\right)\right\|^2\\ &-\frac{1}{2}\left\|\sqrt{\frac{\eta}{\boldsymbol{v}_k}}\boldsymbol{u}_k\right\|^2+\frac{L\eta^2}{2}\left\|\frac{\boldsymbol{u}_k}{\boldsymbol{v}_k}\right\|^2+\frac{\lambda\eta^2}{2}\left\|\frac{\boldsymbol{u}_k}{\boldsymbol{v}_k}\right\|_{\boldsymbol{v}_k}^2\\ \leq&\widetilde{F}_k(\boldsymbol{x}_k)+\frac{\eta}{2c_1}\left\|\nabla F(\boldsymbol{x}_k)-\boldsymbol{m}_k\right\|^2-\frac{\eta}{2c_2}\left\|\nabla F(\boldsymbol{x}_k)+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\right\|^2-\left[\frac{\eta}{2c_2}-\frac{L\eta^2}{2c_1^2}-\frac{\lambda\eta^2}{2c_1}\right]\left\|\boldsymbol{u}_k\right\|^2\\ \overset{③}{\leq}&\widetilde{F}_k(\boldsymbol{x}_k)+\frac{\eta}{2c_1}\left\|\nabla F(\boldsymbol{x}_k)-\boldsymbol{m}_k\right\|^2-\frac{\eta}{2c_2}\left\|\nabla F(\boldsymbol{x}_k)+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\right\|^2-\frac{\eta}{4c_2}\left\|\boldsymbol{u}_k\right\|^2\end{aligned}\tag{9}$$

where ① holds since Lemma 7 proves $\left\|\frac{(\boldsymbol{n}_k+\delta)^{\frac{1}{2}}}{(\boldsymbol{n}_{k+1}+\delta)^{\frac{1}{2}}}\right\|_\infty\in[1-\mu,1+\mu]\ (\forall p\in[0,1])$ in which $\mu=\frac{\boldsymbol{\beta}_2c_\infty^2}{\delta}$; ② holds because in Lemma 8, we have

$$\frac{\lambda_{k+1}}{1-\mu}\left\|\boldsymbol{x}_{k+1}\right\|_{\boldsymbol{v}_k}^2\leq\lambda_k\left\|\boldsymbol{x}_k\right\|_{\boldsymbol{v}_k}^2+\lambda\left\langle\boldsymbol{x}_{k+1}-\boldsymbol{x}_k,\boldsymbol{x}_k\right\rangle_{\boldsymbol{v}_k}+\frac{\lambda}{2}\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\right\|_{\boldsymbol{v}_k}^2;$$

③ holds, since we set $\eta \leq \frac{c_1^2}{2c_2(L+\lambda c_1)}$ such that $\frac{\eta}{4c_2} \geq \frac{L\eta^2}{2c_1^2} + \frac{\lambda\eta^2}{2c_1}$.

From Lemma 9, we have

$$\begin{aligned}
&\mathbb{E}\left[\|\boldsymbol{m}_k - \nabla F(\boldsymbol{x}_k)\|^2\right] \\
\leq&(1-\boldsymbol{\beta}_1)\mathbb{E}\left[\|\boldsymbol{m}_{k-1} - \nabla F(\boldsymbol{x}_{k-1})\|^2\right] + \frac{(1-\boldsymbol{\beta}_1)^2L^2}{\boldsymbol{\beta}_1}\mathbb{E}\left[\|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|^2\right] + \frac{\boldsymbol{\beta}_1^2\sigma^2}{b} \\
\leq&(1-\boldsymbol{\beta}_1)\mathbb{E}\left[\|\boldsymbol{m}_{k-1} - \nabla F(\boldsymbol{x}_{k-1})\|^2\right] + \frac{(1-\boldsymbol{\beta}_1)^2L^2\eta^2}{\boldsymbol{\beta}_1 c_1^2}\mathbb{E}\left[\|\boldsymbol{u}_k\|^2\right] + \frac{\boldsymbol{\beta}_1^2\sigma^2}{b}
\end{aligned} \tag{10}$$

where we use $\boldsymbol{x}_k - \boldsymbol{x}_{k-1} = \eta\frac{\boldsymbol{u}_k}{\boldsymbol{v}_k}$.

Then we add Eqn. (12) and $\alpha\times$ (13) as follows:

$$\begin{aligned}
&\widetilde{F}_{k+1}(\boldsymbol{x}_{k+1}) + \alpha\mathbb{E}\left[\|\boldsymbol{m}_{k+1} - \nabla F(\boldsymbol{x}_{k+1})\|^2\right] \\
\leq&\widetilde{F}_k(\boldsymbol{x}_k) - \frac{\eta}{2c_2}\|\nabla F(\boldsymbol{x}_k) + \lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\|^2 + \left[(1-\boldsymbol{\beta}_1)\alpha + \frac{\eta}{2c_1}\right]\mathbb{E}\left[\|\boldsymbol{m}_{k-1} - \nabla F(\boldsymbol{x}_{k-1})\|^2\right] \\
&- \left[\frac{\eta}{4c_2} - \frac{\alpha(1-\boldsymbol{\beta}_1)^2L^2\eta^2}{\boldsymbol{\beta}_1 c_1^2}\right]\mathbb{E}\left[\|\boldsymbol{u}_k\|^2\right] + \frac{\alpha\boldsymbol{\beta}_1^2\sigma^2}{b}.
\end{aligned}$$

Then by setting $\alpha = \frac{\eta}{2c_1\boldsymbol{\beta}_1}$ and $G(\boldsymbol{x}_{k+1}) = \widetilde{F}_{k+1}(\boldsymbol{x}_{k+1}) + \frac{\eta}{2c_1\boldsymbol{\beta}_1}\mathbb{E}\left[\|\boldsymbol{m}_{k+1} - \nabla F(\boldsymbol{x}_{k+1})\|^2\right]$, we can obtain

$$\begin{aligned}
G(\boldsymbol{x}_{k+1}) \leq&G(\boldsymbol{x}_k) - \frac{\eta}{2c_2}\mathbb{E}\|\nabla F(\boldsymbol{x}_k) + \lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\|^2 - \frac{\eta}{4c_2}\left[1 - \frac{2c_2(1-\boldsymbol{\beta}_1)^2L^2\eta^2}{\boldsymbol{\beta}_1^2c_1^3}\right]\mathbb{E}\left[\|\boldsymbol{u}_k\|^2\right] + \frac{\eta\boldsymbol{\beta}_1\sigma^2}{2c_1b} \\
\overset{①}{\leq}&G(\boldsymbol{x}_k) - \frac{\eta}{2c_2}\mathbb{E}\left[\|\nabla F(\boldsymbol{x}_k) + \lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\|^2\right] - \frac{\eta}{8c_2}\mathbb{E}\left[\|\boldsymbol{u}_k\|^2\right] + \frac{\eta\boldsymbol{\beta}_1\sigma^2}{2c_1b},
\end{aligned}$$

where ① holds since set $\eta \leq \frac{\boldsymbol{\beta}_1 c_1}{2(1-\boldsymbol{\beta}_1)L}\sqrt{\frac{c_1}{c_2}}$ such that $\frac{2c_2(1-\boldsymbol{\beta}_1)^2L^2\eta^2}{\boldsymbol{\beta}_1^2c_1^3} \leq \frac{1}{2}$.

Then summing the above inequality from $k=0$ to $k=T-1$ gives

$$\begin{aligned}
\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\nabla F(\boldsymbol{x}_k) + \lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\|^2 + \frac{1}{4}\|\boldsymbol{u}_k\|^2\right] \leq&\frac{2c_2}{\eta T}\left[G(\boldsymbol{x}_0) - G(\boldsymbol{x}_T)\right] + \frac{c_2\boldsymbol{\beta}_1\sigma^2}{c_1b} \\
\leq&\frac{2c_2\Delta}{\eta T} + \frac{c_2\sigma^2}{c_1\boldsymbol{\beta}_1bT} + \frac{c_2\boldsymbol{\beta}_1\sigma^2}{c_1b} \\
\leq&\epsilon^2,
\end{aligned} \tag{11}$$

where we set $T \geq \max\left(\frac{6c_2\Delta}{\eta\epsilon^2}, \frac{3c_2\sigma^2}{c_1\boldsymbol{\beta}_1b\epsilon^2}\right)$ and $\boldsymbol{\beta}_1 \leq \frac{c_1b\epsilon^2}{3c_2\boldsymbol{\beta}_1\sigma^2}$, in which

$$\begin{aligned}
&G(\boldsymbol{x}_0) - G(\boldsymbol{x}_T) \\
=&\widetilde{F}_0(\boldsymbol{x}_0) + \frac{\eta}{2c_1\boldsymbol{\beta}_1}\mathbb{E}\left[\|\boldsymbol{m}_0 - \nabla F(\boldsymbol{x}_0)\|^2\right] - \widetilde{F}_T(\boldsymbol{x}_T) - \frac{\eta}{2c_1\boldsymbol{\beta}_1}\mathbb{E}\left[\|\boldsymbol{m}_T - \nabla F(\boldsymbol{x}_T)\|^2\right] \\
=&F(\boldsymbol{x}_0) + \frac{\eta}{2c_1\boldsymbol{\beta}_1}\mathbb{E}\left[\|\boldsymbol{m}_0 - \nabla F(\boldsymbol{x}_0)\|^2\right] - F(\boldsymbol{x}_T) - \lambda_T\|\boldsymbol{x}_T\|_{\boldsymbol{v}_T} - \frac{\eta}{2c_1\boldsymbol{\beta}_1}\mathbb{E}\left[\|\boldsymbol{m}_T - \nabla F(\boldsymbol{x}_T)\|^2\right] \\
\leq&F(\boldsymbol{x}_0) + \frac{\eta}{2c_1\boldsymbol{\beta}_1}\mathbb{E}\left[\|\boldsymbol{m}_0 - \nabla F(\boldsymbol{x}_0)\|^2\right] - F(\boldsymbol{x}_T) \\
\leq&\Delta + \frac{\eta}{2c_1\boldsymbol{\beta}_1}\mathbb{E}\left[\|\boldsymbol{m}_0 - \nabla F(\boldsymbol{x}_0)\|^2\right] \\
\leq&\Delta + \frac{\eta\sigma^2}{2c_1\boldsymbol{\beta}_1b},
\end{aligned}$$

where $\Delta = F(\boldsymbol{x}_0) - F(\boldsymbol{x}_*)$. This result directly bounds

$$\frac{1}{T}\sum_{k=0}^{T-1}\|\boldsymbol{v}_k\otimes(\boldsymbol{x}_k - \boldsymbol{x}_{k+1})\|^2 = \frac{\eta^2}{T}\sum_{k=0}^{T-1}\|\boldsymbol{m}_k + \lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\|^2 \leq \frac{\eta^2}{T}\sum_{k=0}^{T-1}\|\boldsymbol{u}_k\|^2 \leq 4\eta^2\epsilon^2.$$

and

$$\frac{1}{T}\sum_{k=0}^{T-1}\|\boldsymbol{x}_k-\boldsymbol{x}_{k+1}\|^2\leq\frac{4\eta^2\epsilon^2}{c_1^2}.$$

Besides, we have

$$\begin{aligned}\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\boldsymbol{m}_k-\nabla F(\boldsymbol{x}_k)\|^2\right]\leq&\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\boldsymbol{m}_k+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k-\nabla F(\boldsymbol{x}_k)-\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\|^2\right]\\ \leq&\frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\boldsymbol{m}_k+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\|^2+\|\nabla F(\boldsymbol{x}_k)-\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\|^2\right]\\ =&\frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\boldsymbol{m}_k+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\|^2+\|\boldsymbol{u}_k\|^2\right]\\ \leq&2\left[\epsilon^2+\frac{3}{4}\times4\epsilon^2\right]\leq8\epsilon^2.\end{aligned}$$

For all hyper-parameters, we put their constrains together:

$$\boldsymbol{\beta}_1\leq\frac{c_1b\epsilon^2}{3c_2\sigma^2},$$

where $c_1=\delta^p\leq\|\boldsymbol{v}_k\|_\infty\leq\left(c_\infty^2+\delta\right)^p=c_2=\mathcal{O}\left(c_\infty^{2p}\right)$. For η, it should satisfy

$$\eta\leq\frac{\boldsymbol{\beta}_1c_1}{2(1-\boldsymbol{\beta}_1)L}\sqrt{\frac{c_1}{c_2}}\leq\frac{c_1b\epsilon^2}{3c_2\sigma^2}\frac{c_1}{2L}\sqrt{\frac{c_1}{c_2}}=\frac{c_1^2b\epsilon^2}{6c_2\sigma^2L}\sqrt{\frac{c_1}{c_2}}.$$

where δ is often much smaller than one, and $\boldsymbol{\beta}_1$ is very small. For T, we have

$$\begin{aligned}T\geq&\max\left(\frac{6c_2\Delta}{\eta\epsilon^2},\frac{3c_2\sigma^2}{c_1\boldsymbol{\beta}_1b\epsilon^2}\right)=\mathcal{O}\left(\max\left(\frac{6c_2\Delta}{\epsilon^2}\frac{6c_2\sigma^2L}{c_1^2b\epsilon^2}\sqrt{\frac{c_2}{c_1}},\frac{3c_2\sigma^2}{c_1b\epsilon^2}\frac{3c_2\sigma^2}{c_1b\epsilon^2}\right)\right)\\ =&\mathcal{O}\left(\max\left(\frac{36c_2^{2.5}\Delta\sigma^2L}{c_1^{2.5}b\epsilon^4},\frac{9c_2^2\sigma^4}{c_1^2b^2\epsilon^4}\right)\right)=\mathcal{O}\left(\max\left(\frac{36c_\infty^{2.5}\Delta\sigma^2L}{\delta^{1.25}b\epsilon^4},\frac{9c_\infty^2\sigma^4}{\delta b^2\epsilon^4}\right)\right).\end{aligned}$$

Now we compute the stochastic gradient complexity. For T iterations, the complexity is

$$\mathcal{O}\left(Tb\right)=\mathcal{O}\left(\max\left(\frac{36c_2^{2.5}\Delta\sigma^2L}{c_1^{2.5}\epsilon^4},\frac{9c_2^2\sigma^4}{c_1^2b\epsilon^4}\right)\right)=\mathcal{O}\left(\max\left(\frac{36c_\infty^{2.5}\Delta\sigma^2L}{\delta^{1.25}\epsilon^4},\frac{9c_\infty^2\sigma^4}{\delta b\epsilon^4}\right)\right).$$

The proof is completed. □

E.2 Proof of Theorem 2

Proof. For brevity, we let $\boldsymbol{v}_k=\sqrt{\boldsymbol{n}_k+\delta}$. Since we have $\|\boldsymbol{m}_k\|_\infty\leq c_\infty$ and $\delta\leq\|\boldsymbol{n}_i+\delta\|_\infty\leq c_\infty^2+\delta$ in Lemma 7, for brevity, let

$$c_1:=\delta^{0.5}\leq\|\boldsymbol{v}_k\|_\infty\leq c_2:=(c_\infty^2+\delta)^{0.5}.$$

Also we define

$$\boldsymbol{u}_k:=\boldsymbol{m}_k+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k,\qquad\boldsymbol{x}_{k+1}-\boldsymbol{x}_k=-\eta_k\frac{\boldsymbol{m}_k+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k}{\boldsymbol{v}_k}=-\eta_k\frac{\boldsymbol{u}_k}{\boldsymbol{v}_k}.$$

Then by using the smoothness of $f(\boldsymbol{x};\boldsymbol{\xi})$, we can obtain

$$
\begin{aligned}
&\widetilde{F}_{k+1}(\boldsymbol{x}_{k+1}) \\
\leq& F(\boldsymbol{x}_k) + \langle \nabla F(\boldsymbol{x}_k), \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \rangle + \frac{L}{2} \left\| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \right\|^2 + \lambda_{k+1} \left\| \boldsymbol{x}_{k+1} \right\|_{\boldsymbol{v}_{k+1}}^2 \\
\overset{①}{\leq}& F(\boldsymbol{x}_k) + \langle \nabla F(\boldsymbol{x}_k), \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \rangle + \frac{L}{2} \left\| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \right\|^2 + \frac{\lambda_{k+1}}{1-\mu} \left\| \boldsymbol{x}_{k+1} \right\|_{\boldsymbol{v}_k}^2 \\
\overset{②}{\leq}& F(\boldsymbol{x}_k) + \lambda_k \left\| \boldsymbol{x}_k \right\|_{\boldsymbol{v}_k}^2 + \langle \nabla F(\boldsymbol{x}_k) + \lambda \boldsymbol{x}_k \otimes \boldsymbol{v}_k, \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \rangle + \frac{L}{2} \left\| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \right\|^2 + \frac{\lambda}{2} \left\| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \right\|_{\boldsymbol{v}_k}^2 \\
=& \widetilde{F}_k(\boldsymbol{x}_k) - \eta_k \left\langle \nabla F(\boldsymbol{x}_k) + \lambda \boldsymbol{x}_k \otimes \boldsymbol{v}_k, \frac{\boldsymbol{u}_k}{\boldsymbol{v}_k} \right\rangle + \frac{L\eta_k^2}{2} \left\| \frac{\boldsymbol{u}_k}{\boldsymbol{v}_k} \right\|^2 + \frac{\lambda \eta_k^2}{2} \left\| \frac{\boldsymbol{u}_k}{\boldsymbol{v}_k} \right\|_{\boldsymbol{v}_k}^2 \\
=& \widetilde{F}_k(\boldsymbol{x}_k) + \frac{1}{2} \left\| \sqrt{\frac{\eta_k}{\boldsymbol{v}_k}} \left(\nabla F(\boldsymbol{x}_k) + \lambda \boldsymbol{x}_k \otimes \boldsymbol{v}_k - \boldsymbol{u}_k \right) \right\|^2 - \frac{1}{2} \left\| \sqrt{\frac{\eta_k}{\boldsymbol{v}_k}} \left(\nabla F(\boldsymbol{x}_k) + \lambda \boldsymbol{x}_k \otimes \boldsymbol{v}_k \right) \right\|^2 \\
&- \frac{1}{2} \left\| \sqrt{\frac{\eta_k}{\boldsymbol{v}_k}} \boldsymbol{u}_k \right\|^2 + \frac{L\eta_k^2}{2} \left\| \frac{\boldsymbol{u}_k}{\boldsymbol{v}_k} \right\|^2 + \frac{\lambda \eta_k^2}{2} \left\| \frac{\boldsymbol{u}_k}{\boldsymbol{v}_k} \right\|_{\boldsymbol{v}_k}^2 \\
\leq& \widetilde{F}_k(\boldsymbol{x}_k) + \frac{\eta_k}{2c_1} \left\| \nabla F(\boldsymbol{x}_k) - \boldsymbol{m}_k \right\|^2 - \frac{\eta_k}{2c_2} \left\| \nabla F(\boldsymbol{x}_k) + \lambda \boldsymbol{x}_k \otimes \boldsymbol{v}_k \right\|^2 - \left[\frac{\eta_k}{2c_2} - \frac{L\eta_k^2}{2c_1^2} - \frac{\lambda \eta_k^2}{2c_1} \right] \left\| \boldsymbol{u}_k \right\|^2 \\
\overset{③}{\leq}& \widetilde{F}_k(\boldsymbol{x}_k) + \frac{\eta_k}{2c_1} \left\| \nabla F(\boldsymbol{x}_k) - \boldsymbol{m}_k \right\|^2 - \frac{\eta_k}{2c_2} \left\| \nabla F(\boldsymbol{x}_k) + \lambda \boldsymbol{x}_k \otimes \boldsymbol{v}_k \right\|^2 - \frac{\eta_k}{4c_2} \left\| \boldsymbol{u}_k \right\|^2
\end{aligned}
\tag{12}
$$

where ① holds since Lemma 7 proves $\left\| \frac{(\boldsymbol{n}_k + \delta)^{0.5}}{(\boldsymbol{n}_{k+1} + \delta)^{0.5}} \right\|_\infty \in [1-\mu, 1+\mu]$ $(\forall p \in [0,1])$ in which $\mu = \frac{\boldsymbol{\beta}_2 c_\infty^2}{\delta}$; ② holds because in Lemma 8, we have

$$
\frac{\lambda_{k+1}}{1-\mu} \left\| \boldsymbol{x}_{k+1} \right\|_{\boldsymbol{v}_k}^2 \leq \lambda_k \left\| \boldsymbol{x}_k \right\|_{\boldsymbol{v}_k}^2 + \lambda \left\langle \boldsymbol{x}_{k+1} - \boldsymbol{x}_k, \boldsymbol{x}_k \right\rangle_{\boldsymbol{v}_k} + \frac{\lambda}{2} \left\| \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \right\|_{\boldsymbol{v}_k}^2 ;
$$

③ holds, since we set $\eta_k \leq \frac{c_1^2}{2c_2(L+\lambda c_1)}$ such that $\frac{\eta_k}{4c_2} \geq \frac{L\eta_k^2}{2c_1^2} + \frac{\lambda \eta_k^2}{2c_1}$.

From Lemma 9, we have

$$
\begin{aligned}
&\mathbb{E}\left[\left\| \boldsymbol{m}_k - \nabla F(\boldsymbol{x}_k) \right\|^2 \right] \\
\leq& (1-\boldsymbol{\beta}_{1,k}) \mathbb{E}\left[\left\| \boldsymbol{m}_{k-1} - \nabla F(\boldsymbol{x}_{k-1}) \right\|^2 \right] + \frac{(1-\boldsymbol{\beta}_{1,k})^2 L^2}{\boldsymbol{\beta}_{1,k}} \mathbb{E}\left[\left\| \boldsymbol{x}_k - \boldsymbol{x}_{k-1} \right\|^2 \right] + \frac{\boldsymbol{\beta}_{1,k}^2 \sigma^2}{b} \\
\leq& (1-\boldsymbol{\beta}_{1,k}) \mathbb{E}\left[\left\| \boldsymbol{m}_{k-1} - \nabla F(\boldsymbol{x}_{k-1}) \right\|^2 \right] + \frac{(1-\boldsymbol{\beta}_{1,k})^2 L^2 \eta_k^2}{\boldsymbol{\beta}_{1,k} c_1^2} \mathbb{E}\left[\left\| \boldsymbol{u}_k \right\|^2 \right] + \frac{\boldsymbol{\beta}_{1,k}^2 \sigma^2}{b}
\end{aligned}
\tag{13}
$$

where we use $\boldsymbol{x}_k - \boldsymbol{x}_{k-1} = \eta_k \frac{\boldsymbol{u}_k}{\boldsymbol{v}_k}$.

Then we add Eqn. (12) and $\alpha\times$ (13) as follows:

$$
\begin{aligned}
&\widetilde{F}_{k+1}(\boldsymbol{x}_{k+1}) + \alpha \mathbb{E}\left[\left\| \boldsymbol{m}_{k+1} - \nabla F(\boldsymbol{x}_{k+1}) \right\|^2 \right] \\
\leq& \widetilde{F}_k(\boldsymbol{x}_k) - \frac{\eta_k}{2c_2} \left\| \nabla F(\boldsymbol{x}_k) + \lambda \boldsymbol{x}_k \otimes \boldsymbol{v}_k \right\|^2 + \left[(1-\boldsymbol{\beta}_{1,k})\alpha + \frac{\eta_k}{2c_1} \right] \mathbb{E}\left[\left\| \boldsymbol{m}_{k-1} - \nabla F(\boldsymbol{x}_{k-1}) \right\|^2 \right] \\
&- \left[\frac{\eta_k}{4c_2} - \frac{\alpha(1-\boldsymbol{\beta}_{1,k})^2 L^2 \eta_k^2}{\boldsymbol{\beta}_{1,k} c_1^2} \right] \mathbb{E}\left[\left\| \boldsymbol{u}_k \right\|^2 \right] + \frac{\alpha \boldsymbol{\beta}_{1,k}^2 \sigma^2}{b}.
\end{aligned}
$$

Then by setting $\alpha = \frac{\eta_k}{2c_1\boldsymbol{\beta}_{1,k}}$ and $G(\boldsymbol{x}_{k+1}) = \widetilde{F}_{k+1}(\boldsymbol{x}_{k+1}) + \frac{\eta_k}{2c_1\boldsymbol{\beta}_{1,k}}\mathbb{E}\left[\|\boldsymbol{m}_{k+1} - \nabla F(\boldsymbol{x}_{k+1})\|^2\right]$, we can obtain

$$
\begin{aligned}
&G(\boldsymbol{x}_{k+1}) \\
\leq& G(\boldsymbol{x}_k) - \frac{\eta_k}{2c_2}\mathbb{E}\left\|\nabla F(\boldsymbol{x}_k) + \lambda \boldsymbol{x}_k \otimes \boldsymbol{v}_k\right\|^2 - \frac{\eta_k}{4c_2}\left[1 - \frac{2c_2(1-\boldsymbol{\beta}_{1,k})^2 L^2 \eta_k^2}{\boldsymbol{\beta}_{1,k}^2 c_1^3}\right]\mathbb{E}\left[\|\boldsymbol{u}_k\|^2\right] + \frac{\eta_k \boldsymbol{\beta}_{1,k}\sigma^2}{2c_1 b} \\
\overset{①}{\leq}& G(\boldsymbol{x}_k) - \frac{\eta_k}{2c_2}\mathbb{E}\left[\left\|\nabla F(\boldsymbol{x}_k) + \lambda \boldsymbol{x}_k \otimes \boldsymbol{v}_k\right\|^2\right] - \frac{\eta_k}{8c_2}\mathbb{E}\left[\|\boldsymbol{u}_k\|^2\right] + \frac{\eta_k \boldsymbol{\beta}_{1,k}\sigma^2}{2c_1 b},
\end{aligned}
$$

where ① holds since we set $\eta_k \leq \frac{\boldsymbol{\beta}_{1,k}c_1}{2(1-\boldsymbol{\beta}_{1,k})L}\sqrt{\frac{c_1}{c_2}}$ such that $\frac{2c_2(1-\boldsymbol{\beta}_{1,k})^2 L^2 \eta_k^2}{\boldsymbol{\beta}_{1,k}^2 c_1^3} \leq \frac{1}{2}$.

Then summing the above inequality from $k = 0$ to $k = T-1$ gives

$$
\begin{aligned}
&\sum_{k=0}^{T-1} \frac{\eta_k}{\sum_{k=0}^{T-1}\eta_k}\mathbb{E}\left[\left\|\nabla F(\boldsymbol{x}_k) + \lambda \boldsymbol{x}_k \otimes \boldsymbol{v}_k\right\|^2 + \frac{1}{4}\|\boldsymbol{u}_k\|^2\right] \\
\leq& \frac{2c_2}{\sum_{k=0}^{T-1}\eta_k}\left[G(\boldsymbol{x}_0) - G(\boldsymbol{x}_T)\right] + \frac{c_2\sum_{k=0}^{T-1}\eta_k \boldsymbol{\beta}_{1,k}\sigma^2}{c_1 b\sum_{k=0}^{T-1}\eta_k} \\
\leq& \frac{2c_2\Delta}{\sum_{k=0}^{T-1}\eta_k} + \frac{c_2\eta_0\sigma^2}{c_1\boldsymbol{\beta}_{1,0}b\sum_{k=0}^{T-1}\eta_k} + \frac{c_2\sigma^2\sum_{k=0}^{T-1}\eta_k\boldsymbol{\beta}_{1,k}}{c_1 b\sum_{k=0}^{T-1}\eta_k},
\end{aligned} \tag{14}
$$

where

$$
\begin{aligned}
&G(\boldsymbol{x}_0) - G(\boldsymbol{x}_T) \\
=& \widetilde{F}_0(\boldsymbol{x}_0) + \frac{\eta_0}{2c_1\boldsymbol{\beta}_{1,0}}\mathbb{E}\left[\|\boldsymbol{m}_0 - \nabla F(\boldsymbol{x}_0)\|^2\right] - \widetilde{F}_T(\boldsymbol{x}_T) - \frac{\eta_0}{2c_1\boldsymbol{\beta}_{1,0}}\mathbb{E}\left[\|\boldsymbol{m}_T - \nabla F(\boldsymbol{x}_T)\|^2\right] \\
=& F(\boldsymbol{x}_0) + \frac{\eta_0}{2c_1\boldsymbol{\beta}_{1,0}}\mathbb{E}\left[\|\boldsymbol{m}_0 - \nabla F(\boldsymbol{x}_0)\|^2\right] - F(\boldsymbol{x}_T) - \lambda_T\|\boldsymbol{x}_T\|_{\boldsymbol{v}_T} - \frac{\eta_0}{2c_1\boldsymbol{\beta}_{1,0}}\mathbb{E}\left[\|\boldsymbol{m}_T - \nabla F(\boldsymbol{x}_T)\|^2\right] \\
\leq& F(\boldsymbol{x}_0) + \frac{\eta_0}{2c_1\boldsymbol{\beta}_{1,0}}\mathbb{E}\left[\|\boldsymbol{m}_0 - \nabla F(\boldsymbol{x}_0)\|^2\right] - F(\boldsymbol{x}_T) \\
\leq& \Delta + \frac{\eta_0}{2c_1\boldsymbol{\beta}_{1,0}}\mathbb{E}\left[\|\boldsymbol{m}_0 - \nabla F(\boldsymbol{x}_0)\|^2\right] \\
\leq& \Delta + \frac{\eta_0\sigma^2}{2c_1\boldsymbol{\beta}_{1,0}b},
\end{aligned}
$$

where $\Delta = F(\boldsymbol{x}_0) - F(\boldsymbol{x}_*)$. Then by setting $\boldsymbol{\beta}_{1,k} = \frac{\gamma_1}{\sqrt{k+1}}$ and $\eta_k = \gamma_2\boldsymbol{\beta}_{1,k}$ where $\gamma_2 = \frac{c_1^{1.5}}{2c_2^{0.5}L}\gamma_3$ and $\gamma_3 = 1$ to satisfy $\eta_k \leq \frac{\boldsymbol{\beta}_{1,k}c_1}{2(1-\boldsymbol{\beta}_{1,k})L}\sqrt{\frac{c_1}{c_2}}$, we have

$$
\begin{aligned}
&\sum_{k=0}^{T-1} \frac{\eta_k}{\sum_{k=0}^{T-1}\eta_k}\mathbb{E}\left[\left\|\nabla F(\boldsymbol{x}_k) + \lambda \boldsymbol{x}_k \otimes \boldsymbol{v}_k\right\|^2 + \frac{1}{4}\|\boldsymbol{u}_k\|^2\right] \\
\leq& \frac{2c_2\Delta}{\sum_{k=0}^{T-1}\eta_k} + \frac{c_2\eta_0\sigma^2}{c_1\boldsymbol{\beta}_{1,0}b\sum_{k=0}^{T-1}\eta_k} + \frac{c_2\sigma^2\sum_{k=0}^{T-1}\eta_k\boldsymbol{\beta}_{1,k}}{c_1 b\sum_{k=0}^{T-1}\eta_k} \\
\overset{①}{\leq}& \frac{c_2\Delta}{\gamma_1\gamma_2(\sqrt{T+1}-2)} + \frac{c_2\sigma^2}{2c_1 b\gamma_1(\sqrt{T+1}-2)} + \frac{c_2\gamma_1\sigma^2\log(T)}{2c_1 b(\sqrt{T+1}-2)} \\
=& \frac{2c_2^{1.5}\Delta L}{c_1^{1.5}\gamma_1\gamma_3(\sqrt{T+1}-2)} + \frac{c_2\sigma^2}{2c_1 b\gamma_1(\sqrt{T+1}-2)} + \frac{c_2\gamma_1\sigma^2\log(T)}{2c_1 b(\sqrt{T+1}-2)} \\
\overset{②}{\leq}& \frac{2c_2}{c_1\gamma_1(\sqrt{T+1}-2)}\left(\frac{c_2^{0.5}L\Delta}{c_1^{0.5}} + \sigma^2\right) + \frac{c_2\gamma_1\sigma^2\log(T)}{2c_1 b(\sqrt{T+1}-2)} \\
\leq& \epsilon^2,
\end{aligned}
$$

where ① uses $\sum_{k=0}^{T-1}\boldsymbol{\beta}_{1,k} \geq \int_2^{T+1}\frac{\gamma_1}{\sqrt{x}}dx = 2\gamma_1(\sqrt{T+1}-2)$ and $\sum_{k=0}^{T-1}\eta_k\boldsymbol{\beta}_{1,k} \leq \gamma_1^2\gamma_2\int_1^T\frac{1}{x}dx = \gamma_1^2\gamma_2\log(T)$, and ② holds by setting

$$
\begin{aligned}
T =&\mathcal{O}\left(\max\left(\frac{4c_2}{c_1\gamma_1\epsilon^4}\left(\frac{c_2^{0.5}L\Delta}{c_1^{0.5}}+\sigma^2\right),\frac{c_2\gamma_1\sigma^2\log\left(\frac{1}{\epsilon}\right)}{2c_1b\epsilon^4}\right)\right)\\
=&\mathcal{O}\left(\max\left(\frac{c_2}{c_1\gamma_1\epsilon^4}\left(\frac{c_2^{0.5}L\Delta}{c_1^{0.5}}+\sigma^2\right),\frac{c_2\gamma_1\sigma^2\log\left(\frac{1}{\epsilon}\right)}{c_1b\epsilon^4}\right)\right)\\
=&\mathcal{O}\left(\max\left(\frac{c_2}{c_1\epsilon^4\max\left(1,\frac{c_2^{0.25}L^{0.5}\Delta^{0.5}}{c_1^{0.25}\sigma}\right)}\left(\frac{c_2^{0.5}L\Delta}{c_1^{0.5}}+\sigma^2\right),\frac{c_2\sigma^2\log\left(\frac{1}{\epsilon}\right)\max\left(1,\frac{c_2^{0.25}L^{0.5}\Delta^{0.5}}{c_1^{0.25}\sigma}\right)}{c_1b\epsilon^4}\right)\right)\\
=&\mathcal{O}\left(\max\left(\frac{c_2\sigma^2}{c_1b\epsilon^4}\log\left(\frac{1}{\epsilon}\right),\frac{c_2^{1.25}L^{0.5}\Delta^{0.5}\sigma}{c_1^{1.25}b\epsilon^4}\log\left(\frac{1}{\epsilon}\right)\right)\right)
\end{aligned}
$$

where we set $\gamma_1 = \max\left(1,\frac{c_2^{0.25}L^{0.5}\Delta^{0.5}}{c_1^{0.25}\sigma}\right)$.

For all hyper-parameters, we put their constrains together:

$$
\boldsymbol{\beta}_{1,k} = \frac{\gamma}{\sqrt{k+1}}, \quad \eta_k = \frac{c_1^{1.5}}{2c_2^{0.5}L}\boldsymbol{\beta}_{1,k} = \frac{\gamma c_1^{1.5}}{2c_2^{0.5}L\sqrt{k+1}} = \frac{\gamma\delta^{0.75}}{2(c_\infty^2+\delta)^{0.25}L\sqrt{k+1}},
$$

where $\gamma = \max\left(1,\frac{c_2^{0.25}L^{0.5}\Delta^{0.5}}{c_1^{0.25}\sigma}\right)$, $c_1 = \delta^{0.5} \leq \|\boldsymbol{v}_k\|_\infty \leq \left(c_\infty^2+\delta\right)^{0.5} = c_2$. Then by setting minibatch size as one, one can easily compute the stochastic gradient complexity

$$
\begin{aligned}
\mathcal{O}\left(Tb\right) =&\mathcal{O}\left(\max\left(\frac{c_2\sigma^2}{c_1\epsilon^4}\log\left(\frac{1}{\epsilon}\right),\frac{c_2^{1.25}L^{0.5}\Delta^{0.5}\sigma}{c_1^{1.25}\epsilon^4}\log\left(\frac{1}{\epsilon}\right)\right)\right)\\
=&\mathcal{O}\left(\max\left(\frac{c_\infty\sigma^2}{\delta^{0.5}\epsilon^4}\log\left(\frac{1}{\epsilon}\right),\frac{c_\infty^{1.25}L^{0.5}\Delta^{0.5}\sigma}{\delta^{0.625}\epsilon^4}\log\left(\frac{1}{\epsilon}\right)\right)\right).
\end{aligned}
$$

The above result directly bounds

$$
\begin{aligned}
\sum_{k=0}^{T-1}\frac{\eta_k}{\sum_{k=0}^{T-1}\eta_k}\left\|\boldsymbol{v}_k\otimes(\boldsymbol{x}_k-\boldsymbol{x}_{k+1})\right\|^2 =&\sum_{k=0}^{T-1}\frac{\eta_k^3}{\sum_{k=0}^{T-1}\eta_k}\left\|\boldsymbol{m}_k+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\right\|^2\\
=&\max_k\eta_k^2\left(\sum_{k=0}^{T-1}\frac{\eta_k}{\sum_{k=0}^{T-1}\eta_k}\left\|\boldsymbol{m}_k+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\right\|^2\right)\\
\leq&\eta_1^2\sum_{k=0}^{T-1}\frac{\eta_k}{\sum_{k=0}^{T-1}\eta_k}\left\|\boldsymbol{u}_k\right\|^2\\
\leq&4\eta_1^2\epsilon^2.
\end{aligned}
$$

Besides, we have

$$
\begin{aligned}
\sum_{k=0}^{T-1}\frac{\eta_k}{\sum_{k=0}^{T-1}\eta_k}\mathbb{E}\left[\left\|\boldsymbol{m}_k-\nabla F(\boldsymbol{x}_k)\right\|^2\right] \leq&\sum_{k=0}^{T-1}\frac{\eta_k}{\sum_{k=0}^{T-1}\eta_k}\mathbb{E}\left[\left\|\boldsymbol{m}_k+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k-\nabla F(\boldsymbol{x}_k)-\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\right\|^2\right]\\
\leq&2\sum_{k=0}^{T-1}\frac{\eta_k}{\sum_{k=0}^{T-1}\eta_k}\mathbb{E}\left[\left\|\boldsymbol{m}_k+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\right\|^2+\left\|\nabla F(\boldsymbol{x}_k)-\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\right\|^2\right]\\
=&2\sum_{k=0}^{T-1}\frac{\eta_k}{\sum_{k=0}^{T-1}\eta_k}\mathbb{E}\left[\left\|\boldsymbol{m}_k+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\right\|^2+\left\|\boldsymbol{u}_k\right\|^2\right]\\
\leq&2\left[\epsilon^2+\frac{3}{4}\times4\epsilon^2\right]\leq8\epsilon^2.
\end{aligned}
$$

The proof is completed. □

E.3 PROOF OF THEOREM 3

Proof. **Step 1. Results under constant learning rate.** Here we first consider the conventional one stage training. Firstly, we borrow the results in Eqn. (11) in Appendix E.1 (proofs for Theorem 1), we have

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\nabla F(\boldsymbol{x}_k)+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\right\|^2+\frac{1}{4}\left\|\boldsymbol{m}_k+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\right\|^2\right]\leq\frac{2c_2\Delta}{\eta T}+\frac{c_2\sigma^2}{c_1\boldsymbol{\beta}_1 bT}+\frac{c_2\boldsymbol{\beta}_1\sigma^2}{c_1 b},$$

where $\Delta=F(\boldsymbol{x}_0)-F(\boldsymbol{x}_*)$. In this way, by setting $T\geq\max\left(\frac{12c_2\Delta}{\eta\mu\epsilon^2},\frac{6c_2\sigma^2}{c_1\mu\boldsymbol{\beta}_1 b\epsilon^2}\right)$ and $\boldsymbol{\beta}_1\leq\frac{c_1\mu b\epsilon^2}{6c_2\boldsymbol{\beta}_1\sigma^2}$,

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\nabla F(\boldsymbol{x}_k)+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\right\|^2+\frac{1}{4}\left\|\boldsymbol{m}_k+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\right\|^2\right]\leq\frac{\mu\epsilon^2}{2}.$$

Then we consider all constrains of hyper-parameters in Appendix E.1 together:

$$\boldsymbol{\beta}_1\leq\frac{c_1\mu b\epsilon^2}{6c_2\sigma^2},\qquad\eta\leq\frac{\boldsymbol{\beta}_1 c_1}{2(1-\boldsymbol{\beta}_1)L}\sqrt{\frac{c_1}{c_2}}\leq\frac{c_1\mu b\epsilon^2}{6c_2\sigma^2}\frac{c_1}{2L}\sqrt{\frac{c_1}{c_2}}=\frac{c_1^2\mu b\epsilon^2}{12c_2\sigma^2 L}\sqrt{\frac{c_1}{c_2}},$$

where $c_1=\delta^{0.5}\leq\|\boldsymbol{v}_k\|_\infty\leq\left(c_\infty^2+\delta\right)^{0.5}=c_2$. For T, we have

$$\begin{aligned}T\geq&\max\left(\frac{12c_2\Delta}{\eta\mu\epsilon^2},\frac{6c_2\sigma^2}{c_1\mu\boldsymbol{\beta}_1 b\epsilon^2}\right)=\mathcal{O}\left(\max\left(\frac{12c_2\Delta}{\mu\epsilon^2}\frac{12c_2\sigma^2 L}{c_1^2\mu b\epsilon^2}\sqrt{\frac{c_2}{c_1}},\frac{6c_2\sigma^2}{c_1\mu b\epsilon^2}\frac{6c_2\sigma^2}{c_1\mu b\epsilon^2}\right)\right)\\=&\mathcal{O}\left(\max\left(\frac{144c_2^{2.5}\Delta\sigma^2 L}{c_1^{2.5}\mu^2 b\epsilon^4},\frac{36c_2^2\sigma^4}{c_1^2\mu^2 b^2\epsilon^4}\right)\right).\end{aligned}$$

Now we compute the stochastic gradient complexity. For T iterations, the complexity is

$$\mathcal{O}\left(Tb\right)=\mathcal{O}\left(\max\left(\frac{144c_2^{2.5}\Delta\sigma^2 L}{c_1^{2.5}\mu^2\epsilon^4},\frac{36c_2^2\sigma^4}{c_1^2\mu^2 b\epsilon^4}\right)\right).$$

Then we consider multiple stage training. For each stage, by setting $\epsilon_k=\frac{\epsilon_0}{2^k}$, we run T_k iterations and hope to achieve

$$\mathbb{E}\left[\boldsymbol{F}_k(\boldsymbol{x}_k)-\boldsymbol{F}_k(\boldsymbol{x}_*)\right]\leq\frac{1}{T_k}\sum_{i=0}^{T_k-1}\mathbb{E}\left[\boldsymbol{F}_i(\boldsymbol{x}_i)-\boldsymbol{F}_i(\boldsymbol{x}_*)\right]\leq\frac{1}{2\mu T_k}\sum_{i=0}^{T_k-1}\mathbb{E}\left[\left\|\nabla\boldsymbol{F}_i(\boldsymbol{x}_i)\right\|^2\right]\leq\epsilon_k,\tag{15}$$

where the last inequality uses the PŁ condition. This ϵ_k-accuracy solution can be achievable according to the above results. Specifically, by setting

$$\begin{aligned}&\boldsymbol{\beta}_{1k}\leq\frac{c_1\mu b\epsilon_k}{6c_2\sigma^2}=\frac{\delta^{0.5}\mu b\epsilon_k}{6(c_\infty^2+\delta)^{0.5}\sigma^2},\qquad\eta\leq\frac{c_1^{2.5}\mu b\epsilon_k}{12c_2^{1.5}\sigma^2 L}=\frac{\delta^{1.25}\mu b\epsilon_k}{12(c_\infty^2+\delta)^{0.75}\sigma^2 L},\\&T_k\geq\mathcal{O}\left(\max\left(\frac{144c_2^{2.5}\Delta\sigma^2 L}{c_1^{2.5}\mu^2 b\epsilon_k^2},\frac{36c_2^2\sigma^4}{c_1^2\mu^2 b^2\epsilon_k^2}\right)\right)=\mathcal{O}\big(\max\big(\frac{c_\infty^{2.5}\Delta\sigma^2 L}{\delta^{1.25}\mu^2 b\epsilon_k^2},\frac{c_\infty^2\sigma^4}{\delta\mu^2 b^2\epsilon_k^2}\big)\big).\end{aligned}$$

we can achieve the target in (15) with stochastic complexity as $\mathcal{O}\left(T_k b\right)=\mathcal{O}\left(\max\left(\frac{144c_2^{2.5}\Delta\sigma^2 L}{c_1^{2.5}\mu^2\epsilon_k^2},\frac{36c_2^2\sigma^4}{c_1^2\mu^2 b\epsilon_k^2}\right)\right)$.

Finally, to achieve ϵ-accuracy solution, we only need to run at most K stages which should satisfy

$$\epsilon_K=\frac{\epsilon_0}{2^K}\leq\epsilon.$$

So it means that K should obey

$$K\geq\log_2\left(\frac{1}{\epsilon}\right).$$

In this way, we can compute the total computational complexity as follows:

$$\begin{aligned}\sum_{k=1}^{K}\mathbb{E}\left[T_k b\right] =&\mathbb{E}\left[\sum_{k=1}^{K}\mathcal{O}\left(\max\left(\frac{144c_2^{2.5}\Delta\sigma^2 L}{c_1^{2.5}\mu^2\epsilon_k^2},\frac{36c_2^2\sigma^4}{c_1^2\mu^2 b\epsilon_k^2}\right)\right)\right]\\ =&\mathcal{O}\left(\max\left(\frac{144c_2^{2.5}\Delta\sigma^2 L}{c_1^{2.5}\mu^2\epsilon^2},\frac{36c_2^2\sigma^4}{c_1^2\mu^2 b\epsilon^2}\right)\right)=\mathcal{O}\big(\max\big(\frac{c_\infty^{2.5}L\Delta\sigma^2}{\delta^{1.25}\mu^2\epsilon^2},\frac{c_\infty^2\sigma^4}{\delta\mu^2\epsilon^2}\big)\big)\end{aligned}$$

where

$$\mathbb{E}\left[\sum_{k=1}^{K}\frac{1}{\epsilon_k^2}\right]=\mathbb{E}\left[\sum_{k=1}^{K}\frac{2^{2k}}{\epsilon_0^2}\right]=\frac{16(2^{2K}-1)}{15\epsilon_0^2}\leq\frac{16(\frac{\epsilon_0^2}{\epsilon^2}-1)}{15\epsilon_0^2}=\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$$

Step 2. Results under decaying learning rate. Firstly, we borrow the results in Eqn. (14) in Appendix E.2 (proofs for Theorem 2), we have

$$\begin{aligned}&\sum_{k=0}^{T-1}\frac{\eta_k}{\sum_{k=0}^{T-1}\eta_k}\mathbb{E}\left[\left\|\nabla F(\boldsymbol{x}_k)+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\right\|^2+\frac{1}{4}\left\|\boldsymbol{m}_k+\lambda\boldsymbol{x}_k\otimes\boldsymbol{v}_k\right\|^2\right]\\ \leq&\frac{2c_2\Delta}{\sum_{k=0}^{T-1}\eta_k}+\frac{c_2\eta_0\sigma^2}{c_1\boldsymbol{\beta}_{1,0}b\sum_{k=0}^{T-1}\eta_k}+\frac{c_2\sigma^2\sum_{k=0}^{T-1}\eta_k\boldsymbol{\beta}_{1,k}}{c_1 b\sum_{k=0}^{T-1}\eta_k},\end{aligned}$$

where $\Delta=F(\boldsymbol{x}_0)-F(\boldsymbol{x}_*)$. In this way, by setting

$$\begin{aligned}\boldsymbol{\beta}_{1,k}=&\frac{\gamma}{\sqrt{k+1}},\quad \eta_k=\frac{c_1^{1.5}}{2c_2^{0.5}L}\boldsymbol{\beta}_{1,k}=\frac{\gamma c_1^{1.5}}{2c_2^{0.5}L\sqrt{k+1}},\\ T=&\mathcal{O}\left(\max\left(\frac{c_2\sigma^2}{c_1 b\mu^2\epsilon^4}\log\left(\frac{1}{\epsilon}\right),\frac{c_2^{1.25}L^{0.5}\Delta^{0.5}\sigma}{c_1^{1.25}b\mu^2\epsilon^4}\log\left(\frac{1}{\epsilon}\right)\right)\right),\end{aligned}$$

where $\gamma=\max\left(1,\frac{c_2^{0.25}L^{0.5}\Delta^{0.5}}{c_1^{0.25}\sigma}\right)$, $c_1=\delta^p\leq\|\boldsymbol{v}_k\|_\infty\leq\left(c_\infty^2+\delta\right)^p=c_2$, we have

$$\begin{aligned}&\mathbb{E}\left[\left\|\nabla F(\boldsymbol{x}_{k'})+\lambda\boldsymbol{x}_{k'}\otimes\boldsymbol{v}_{k'}\right\|^2+\frac{1}{4}\left\|\boldsymbol{m}_{k'}+\lambda\boldsymbol{x}_{k'}\otimes\boldsymbol{v}_{k'}\right\|^2\right]\\ \leq&\frac{2c_2}{c_1\gamma_1(\sqrt{T+1}-2)}\left(\frac{c_2^{0.5}L\Delta}{c_1^{0.5}}+\sigma^2\right)+\frac{c_2\gamma_1\sigma^2\log(T)}{2c_1 b(\sqrt{T+1}-2)}\leq\frac{\mu\epsilon^2}{2}.\end{aligned}$$

where $\boldsymbol{x}_{k'}$ is the final output, and k' is selected from $\{1,\cdots,T\}$ according to the distribution $\{\frac{\eta_k}{\sum_{i=0}^{T-1}\eta_i}\}$. Then by setting minibatch size as one, one can easily compute the stochastic gradient complexity

$$\mathcal{O}\left(Tb\right)=\mathcal{O}\left(\max\left(\frac{c_2\sigma^2}{c_1\mu^2\epsilon^4}\log\left(\frac{1}{\epsilon}\right),\frac{c_2^{1.25}L^{0.5}\Delta^{0.5}\sigma}{c_1^{1.25}\mu^2\epsilon^4}\log\left(\frac{1}{\epsilon}\right)\right)\right).$$

Then we consider multiple stage training. For each stage, by setting $\epsilon_k=\frac{\epsilon_0}{2^k}$, we run T_k iterations and hope to achieve

$$\Delta_k=\mathbb{E}\left[\boldsymbol{F}_k(\boldsymbol{x}_k)-\boldsymbol{F}_k(\boldsymbol{x}_*)\right]\leq\frac{1}{T_k}\sum_{i=0}^{T_k-1}\mathbb{E}\left[\boldsymbol{F}_i(\boldsymbol{x}_i)-\boldsymbol{F}_i(\boldsymbol{x}_*)\right]\leq\frac{1}{2\mu T_k}\sum_{i=0}^{T_k-1}\mathbb{E}\left[\left\|\nabla\boldsymbol{F}_i(\boldsymbol{x}_i)\right\|^2\right]\leq\epsilon_k,\tag{16}$$

where the last inequality uses the PŁ condition. This ϵ_k-accuracy solution can be achievable according to the above results. Specifically, by setting

$$\begin{aligned}\boldsymbol{\beta}_{1,k_i}=&\frac{\gamma}{\sqrt{i+1}},\quad \eta_{k_i}=\frac{c_1^{1.5}}{2c_2^{0.5}L}\boldsymbol{\beta}_{1,k_i}=\frac{\gamma c_1^{1.5}}{2c_2^{0.5}L\sqrt{i+1}}=\frac{\gamma\delta^{1.5p}}{2(c_\infty^2+\delta)^{0.5p}L\sqrt{i+1}},\\ T_k=&\mathcal{O}\left(\max\left(\frac{c_2\sigma^2}{c_1 b\mu^2\epsilon_k^2}\log\left(\frac{1}{\epsilon_k}\right),\frac{c_2^{1.25}L^{0.5}\Delta_k^{0.5}\sigma}{c_1^{1.25}b\mu^2\epsilon_k^2}\log\left(\frac{1}{\epsilon_k}\right)\right)\right)\\ =&\mathcal{O}\big(\max\big(\frac{c_\infty\sigma^2}{\delta^{0.5}b\mu^2\epsilon_k^2}\log\big(\frac{1}{\epsilon_k}\big),\frac{c_\infty^{1.25}L^{0.5}\Delta_k^{0.5}\sigma}{\delta^{0.625}b\mu^2\epsilon_k^2}\log\big(\frac{1}{\epsilon_k}\big)\big)\big),\end{aligned}$$

where $\gamma = \max\left(1, \frac{c_2^{0.25} L^{0.5} \Delta^{0.5}}{c_1^{0.25} \sigma}\right)$, $c_1 = \delta^{0.5} \leq \|\boldsymbol{v}_k\|_\infty \leq \left(c_\infty^2 + \delta\right)^{0.5} = c_2$, we can achieve the target in (16) with stochastic complexity as $\mathcal{O}\left(T_k b\right) = \mathcal{O}\left(\max\left(\frac{c_2 \sigma^2}{c_1 \mu^2 \epsilon_k^2} \log\left(\frac{1}{\epsilon_k}\right), \frac{c_2^{1.25} L^{0.5} \Delta_k^{0.5} \sigma}{c_1^{1.25} \mu^2 \epsilon_k^2} \log\left(\frac{1}{\epsilon_k}\right)\right)\right)$

Finally, to achieve ϵ-accuracy solution, we only need to run at most K stages which should satisfy

$$\epsilon_K = \frac{\epsilon_0}{2^K} \leq \epsilon.$$

So it means that K should obey

$$K \geq \log_2\left(\frac{1}{\epsilon}\right).$$

In this way, we can compute the total computational complexity as follows:

$$\begin{aligned}
\sum_{k=1}^{K} \mathbb{E}\left[T_k b\right] =& \mathbb{E}\left[\sum_{k=1}^{K} \mathcal{O}\left(\max\left(\frac{c_2 \sigma^2}{c_1 \mu^2 \epsilon_k^2} \log\left(\frac{1}{\epsilon_k}\right), \frac{c_2^{1.25} L^{0.5} \Delta_k^{0.5} \sigma}{c_1^{1.25} \mu^2 \epsilon_k^2} \log\left(\frac{1}{\epsilon_k}\right)\right)\right)\right] \\
=& \mathcal{O}\left(\max\left(\frac{c_2 \sigma^2}{c_1 \mu^2} \mathbb{E}\left[\sum_{k=1}^{K} \frac{1}{\epsilon_k^2} \log\left(\frac{1}{\epsilon_k}\right)\right], \frac{c_2^{1.25} L^{0.5} \sigma}{c_1^{1.25} \mu^2} \mathbb{E}\left[\sum_{k=1}^{K} \frac{\Delta_k^{0.5}}{\epsilon_k^2} \log\left(\frac{1}{\epsilon_k}\right)\right]\right)\right) \\
=& \mathcal{O}\left(\max\left(\frac{c_2 \sigma^2}{c_1 \mu^2 \epsilon^2}, \frac{c_2^{1.25} L^{0.5} \sigma}{c_1^{1.25} \mu^2 \epsilon^2}\right)\right) = \mathcal{O}\left(\max\left(\frac{c_\infty \sigma^2}{\delta^{0.5} \mu^2 \epsilon^2}, \frac{c_\infty^{1.25} L^{0.5} \sigma}{\delta^{0.625} \mu^2 \epsilon^2}\right)\right)
\end{aligned}$$

where

$$\mathbb{E}\left[\sum_{k=1}^{K} \frac{1}{\epsilon_k^2} \log\left(\frac{1}{\epsilon_k}\right)\right] \overset{①}{=} \mathbb{E}\left[\sum_{k=1}^{K} \frac{2^{2k}}{\epsilon_0^2} \log\left(\frac{2^k}{\epsilon_0}\right)\right] = \mathcal{O}\left(\mathbb{E}\left[\sum_{k=1}^{K} \frac{k \cdot 2^{2k}}{\epsilon_0^2}\right]\right) = \mathcal{O}\left(\mathbb{E}\left[S_K\right]\right) \overset{②}{=} \mathcal{O}\left(\frac{1}{\epsilon^2}\right)$$

$$\mathbb{E}\left[\sum_{k=1}^{K} \frac{\Delta_k^{0.5}}{\epsilon_k^2} \log\left(\frac{1}{\epsilon_k}\right)\right] \overset{③}{\leq} \mathbb{E}\left[\sum_{k=1}^{K} \frac{\epsilon_k^{0.5}}{\epsilon_k^2} \log\left(\frac{1}{\epsilon_k}\right)\right] = \mathbb{E}\left[\sum_{k=1}^{K} \frac{k \cdot 2^{1.5k}}{\epsilon_0^{1.5}}\right] = \mathbb{E}\left[S_K'\right] = \mathcal{O}\left(\frac{1}{\epsilon^2}\right)$$

where we use $\epsilon_k = \frac{\epsilon_0}{2^k}$ in ①, and Eqn. (16) in ③. For ②, we can compute

$$S_K - 4S_{K-1} = \sum_{k=1}^{K} \frac{k \cdot 2^{2k}}{\epsilon_0^2} - 4 \sum_{k=1}^{K-1} \frac{k \cdot 2^{2k}}{\epsilon_0^2} = \frac{4}{\epsilon_0^2}.$$

Consider $S_1 = \frac{4}{\epsilon_0^2}$, then we have

$$S_K + \frac{4}{3\epsilon_0^2} = 4\left(S_{K-1} + \frac{4}{3\epsilon_0^2}\right) = 4^{K-1}\left(S_1 + \frac{4}{3\epsilon_0^2}\right) = \frac{4^{K+2}}{3\epsilon_0^2} = \frac{16}{3\epsilon^2},$$

where we use $\frac{\epsilon_0}{2^K} = \epsilon$. Similarly, we compute

$$\begin{aligned}
S_K' + \frac{2^{1.5}}{(2^{1.5}-1)\epsilon_0^{1.5}} =& 2^{1.5}\left(S_{K-1}' + \frac{2^{1.5}}{(2^{1.5}-1)\epsilon_0^{1.5}}\right) = 2^{1.5(K-1)}\left(S_1' + \frac{2^{1.5}}{(2^{1.5}-1)\epsilon_0^{1.5}}\right) \\
=& \frac{2 \cdot 2^{1.5(K+1)}}{\epsilon_0^{1.5}} \leq \frac{8}{\epsilon^2}.
\end{aligned}$$

The proof is completed. □

F PROOF OF RESULTS IN SEC. 5

To begin with, we first give one useful lemma to prove our generalization error bound.

Lemma 10. *(PAC-Bayesian generalization bound) (McAllester, 1999) For any $\tau \in (0,1)$, the expected risk for the posterior hypothesis of an algorithm over a training dataset $\mathcal{D}_{tr} \sim \mathcal{D}$ with n samples holds with at least probability $1-\tau$:*

$$\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{D},\boldsymbol{x}\sim\mathcal{P}}[f(\boldsymbol{x},\boldsymbol{\xi})] - \mathbb{E}_{\boldsymbol{\xi}\in\mathcal{D}_{tr},\boldsymbol{x}\sim\mathcal{P}}[f(\boldsymbol{x},\boldsymbol{\xi})] \leq 4\sqrt{\frac{1}{n}\left(KL(\mathcal{P}\|\mathcal{P}_{pre}) + \ln\left(\frac{2n}{\tau}\right)\right)},$$

where $KL(\mathcal{P}\|\mathcal{P}_{pre})$ denotes the Kullback-Leibler divergence from prior $\mathcal{P}_{pre}$ to posterior $\mathcal{P}$.

F.1 Proof of Lemma 4

Proof. Based on the assumptions in Lemma 4 and Eqn. (8), we can write the SDE equations as follows:

$$\begin{aligned}\mathsf{d}\boldsymbol{x}_t =& -\boldsymbol{Q}_t\nabla F(\boldsymbol{x}_t)\mathsf{d}t - \lambda\boldsymbol{x}_t\mathsf{d}t + \boldsymbol{Q}_t\left(2\boldsymbol{\Sigma}_t\right)^{\frac{1}{2}}\mathsf{d}\boldsymbol{\zeta}_t\\ =& -\boldsymbol{Q}_t\boldsymbol{H}_*\boldsymbol{x}_t\mathsf{d}t - \lambda\boldsymbol{x}_t\mathsf{d}t + \boldsymbol{Q}_t\left(2\boldsymbol{\Sigma}_t\right)^{\frac{1}{2}}\mathsf{d}\boldsymbol{\zeta}_t\\ =& -\left(\boldsymbol{H}_*^{\frac{1}{2}} + \lambda\boldsymbol{I}\right)\boldsymbol{x}_t\mathsf{d}t + \sqrt{\frac{\eta}{b}}\mathsf{d}\boldsymbol{\zeta}_t,\end{aligned}$$

where $\mathsf{d}\boldsymbol{\zeta}_t \sim \mathcal{N}(0, \boldsymbol{I}\mathsf{d}t)$, $\boldsymbol{\Sigma}_t \approx \frac{\eta}{2B}\boldsymbol{H}_*$; $\boldsymbol{Q}_t = \boldsymbol{H}_*^{-\frac{1}{2}}$. Then for this Ornstein–Uhlenbeck process, we can compute its closed form solution as follows:

$$\boldsymbol{x}_t = \exp\left(-(\boldsymbol{H}_*^{\frac{1}{2}} + \lambda\boldsymbol{I})t\right)\boldsymbol{x}_0 + \sqrt{\frac{\eta}{b}}\int_0^t \exp\left(-(\boldsymbol{H}_*^{\frac{1}{2}} + \lambda\boldsymbol{I})(t-t')\right)\mathsf{d}\boldsymbol{\zeta}_{t'}.$$

Let $\boldsymbol{M} = \mathbb{E}\left[\boldsymbol{x}_t\boldsymbol{x}_t^\top\right]$. In this way, considering $(\boldsymbol{H}_*^{\frac{1}{2}} + \lambda\boldsymbol{I}) = (\boldsymbol{H}_*^{\frac{1}{2}} + \lambda\boldsymbol{I})^\top$, we follow (Mandt et al., 2017) (see their Appendix b) and can further compute the algebraic relation for the stationary covariance of the multivariate Ornstein–Uhlenbeck process as follows:

$$\begin{aligned}&(\boldsymbol{H}_*^{\frac{1}{2}} + \lambda\boldsymbol{I})\boldsymbol{M} + \boldsymbol{M}(\boldsymbol{H}_*^{\frac{1}{2}} + \lambda\boldsymbol{I})\\ =&\frac{\eta}{b}\int_{-\infty}^t (\boldsymbol{H}_*^{\frac{1}{2}} + \lambda\boldsymbol{I})\exp\left(-(\boldsymbol{H}_*^{\frac{1}{2}} + \lambda\boldsymbol{I})(t-t')\right)\exp\left(-(\boldsymbol{H}_*^{\frac{1}{2}} + \lambda\boldsymbol{I})(t-t')\right)\mathsf{d}t'\\ &+\frac{\eta}{b}\int_{-\infty}^t \exp\left(-(\boldsymbol{H}_*^{\frac{1}{2}} + \lambda\boldsymbol{I})(t-t')\right)\exp\left(-(\boldsymbol{H}_*^{\frac{1}{2}} + \lambda\boldsymbol{I})(t-t')\right)\mathsf{d}t'(\boldsymbol{H}_*^{\frac{1}{2}} + \lambda\boldsymbol{I})\\ =&\frac{\eta}{b}\int_{-\infty}^t \frac{\mathsf{d}}{\mathsf{d}t'}\left(\exp\left(-(\boldsymbol{H}_*^{\frac{1}{2}} + \lambda\boldsymbol{I})(t-t')\right)\exp\left(-(\boldsymbol{H}_*^{\frac{1}{2}} + \lambda\boldsymbol{I})(t-t')\right)\right)\\ =&\frac{\eta}{b}\boldsymbol{I},\end{aligned}$$

where we use the lower limits of the integral vanishes by the positivity of the eigenvalues of $\boldsymbol{H}_*^{\frac{1}{2}} + \lambda\boldsymbol{I}$. Next, let $\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top$ is the SVD of $\boldsymbol{H}_*$, where $\boldsymbol{S} = \mathrm{diag}\left(\sigma_1, \sigma_2, \cdots, \sigma_d\right)$. Then we have

$$\boldsymbol{U}(\boldsymbol{S}^{\frac{1}{2}} + \lambda\boldsymbol{I})\boldsymbol{U}^\top\boldsymbol{M} + \boldsymbol{M}\boldsymbol{U}(\boldsymbol{S}^{\frac{1}{2}} + \lambda\boldsymbol{I})\boldsymbol{U}^\top = \frac{\eta}{b}\boldsymbol{I}.$$

Then multiplying $\boldsymbol{U}^\top$ from left side and also $\boldsymbol{U}$ from right side gives

$$(\boldsymbol{S}^{\frac{1}{2}} + \lambda\boldsymbol{I})\boldsymbol{U}^\top\boldsymbol{M}\boldsymbol{U} + \boldsymbol{U}^\top\boldsymbol{M}\boldsymbol{U}(\boldsymbol{S}^{\frac{1}{2}} + \lambda\boldsymbol{I}) = \frac{\eta}{b}\boldsymbol{I}.$$

Therefore, we know

$$\boldsymbol{M}_{\text{AdamW}} = \frac{\eta}{2B}\boldsymbol{U}(\boldsymbol{S}^{\frac{1}{2}} + \lambda\boldsymbol{I})^{-1}\boldsymbol{U}^\top.$$

The proof is completed. □

F.2 Proof of Theorem 5

Proof. According to the assumption in Theorem 5 and Lemma 4, we know that for AdamW, its prior and posterior distributions are both Gaussian distribution, namely $\mathcal{P}_{\text{pre}} \sim \mathcal{N}(0, \rho\boldsymbol{I})$ and $\mathcal{P} \sim \mathcal{N}(\boldsymbol{x}_*, \boldsymbol{M}_{\text{AdamW}})$ where

$$\boldsymbol{M}_{\text{AdamW}} = \frac{\eta}{2B}\boldsymbol{U}(\boldsymbol{S}^{\frac{1}{2}} + \lambda\boldsymbol{I})^{-1}\boldsymbol{U}^\top,$$

where $\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top$ is the SVD of $\boldsymbol{H}_*$ in which $\boldsymbol{S} = \mathrm{diag}\left(\sigma_1, \sigma_2, \cdots, \sigma_d\right)$.

On the other hand, for KL between two Gaussian distributions $\boldsymbol{W}_1 \sim (\boldsymbol{u}_1, \boldsymbol{\Sigma}_1)$ and $\boldsymbol{W}_2 \sim (\boldsymbol{u}_2, \boldsymbol{\Sigma}_2)$, we can follow (Pardo, 2018) and compute it as follows:

$$\mathrm{KL}(\boldsymbol{W}_2\|\boldsymbol{W}_1) = \frac{1}{2}\left[\log\frac{\det(\boldsymbol{\Sigma}_1)}{\det(\boldsymbol{\Sigma}_2)} + \mathrm{Tr}\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2\right)\right] + \frac{1}{2}(\boldsymbol{u}_1 - \boldsymbol{u}_2)^\top\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{u}_1 - \boldsymbol{u}_2) - \frac{d}{2}.$$

Accordingly, for AdamW, we can compute

$$\begin{aligned}\mathrm{KL}(\mathcal{P}\|\mathcal{P}_{\mathrm{pre}}) =& \frac{1}{2}\left[\log\frac{\rho^d}{\left(\frac{\eta}{2b}\right)^d\prod_{i=1}^d\frac{1}{\sigma_i^{\frac{1}{2}}+\lambda}}+\frac{\eta}{2\rho b}\sum_{i=1}^d\frac{1}{\sigma_i^{\frac{1}{2}}+\lambda}+\frac{1}{2\rho}\|\boldsymbol{x}_*\|^2-\frac{d}{2}\right]\\ =& \frac{1}{2}\left[\sum_{i=1}^d\log\frac{2\rho b(\sigma_i^{\frac{1}{2}}+\lambda)}{\eta}+\frac{\eta}{2\rho b}\sum_{i=1}^d\frac{1}{\sigma_i^{\frac{1}{2}}+\lambda}+\frac{1}{2\rho}\|\boldsymbol{x}_*\|^2-\frac{d}{2}\right].\end{aligned}$$

Then by using Lemma 10, it further yields the generalization bound of AdamW as follows:

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{D},\boldsymbol{x}\sim\mathcal{P}}[f(\boldsymbol{x},\boldsymbol{\xi})]-\mathbb{E}_{\boldsymbol{\xi}\in\mathcal{D}_{\mathrm{tr}},\boldsymbol{x}\sim\mathcal{P}}[f(\boldsymbol{x},\boldsymbol{\xi})]\\ \le& 4\sqrt{\frac{1}{2n}\left(\sum_{i=1}^d\log\frac{2\rho b(\sigma_i^{\frac{1}{2}}+\lambda)}{\eta}+\frac{\eta}{2\rho b}\sum_{i=1}^d\frac{1}{\sigma_i^{\frac{1}{2}}+\lambda}+c_0\right)},\end{aligned}$$

where $c_0=\frac{1}{2\rho}\|\boldsymbol{x}_*\|^2-\frac{d}{2}+2\ln\left(\frac{2n}{\tau}\right)$. The proof is completed. □

F.3 Proof of Theorem 6

Proof. **Step 1. Posterior Analysis on Adam+ℓ_2-Regularization.** Here we borrow the same idea in Lemma 4 and Theorem 5 to analyze the covariance matrix $\boldsymbol{M}=\mathbb{E}\left[\boldsymbol{x}_t\boldsymbol{x}_t^\top\right]$. To begin with, we simplify the SDE of Adam+ℓ_2-Regularization. Based on the assumptions in Theorem 6 and Eqn. (8), we can write the SDE equations as follows:

$$\begin{aligned}\mathsf{d}\boldsymbol{x}_t =& -\boldsymbol{Q}_t\nabla F(\boldsymbol{x}_t)\mathsf{d}t-\lambda\boldsymbol{Q}_t\boldsymbol{x}_t\mathsf{d}t+\boldsymbol{Q}_t\left(2\boldsymbol{\Sigma}_t\right)^{\frac{1}{2}}\mathsf{d}\boldsymbol{\zeta}_t\\ =& -\boldsymbol{Q}_t\boldsymbol{H}_*\boldsymbol{x}_t\mathsf{d}t-\lambda\boldsymbol{Q}_t\boldsymbol{x}_t\mathsf{d}t+\boldsymbol{Q}_t\left(2\boldsymbol{\Sigma}_t\right)^{\frac{1}{2}}\mathsf{d}\\ =& -(\boldsymbol{H}_*^{\frac{1}{2}}+\lambda\boldsymbol{H}_*^{-\frac{1}{2}})\boldsymbol{x}_t\mathsf{d}t+\sqrt{\frac{\eta}{b}}\mathsf{d}\boldsymbol{\zeta}_t,\end{aligned}$$

where $\mathsf{d}\boldsymbol{\zeta}_t\sim\mathcal{N}(0,\boldsymbol{I}\mathsf{d}t)$, $\boldsymbol{\Sigma}_t\approx\frac{\eta}{2B}\boldsymbol{H}_*$; $\boldsymbol{Q}_t=\boldsymbol{H}_*^{-\frac{1}{2}}$. Then for this Ornstein–Uhlenbeck process, we can compute its closed form solution as follows:

$$\boldsymbol{x}_t=\exp\left(-(\boldsymbol{H}_*^{\frac{1}{2}}+\lambda\boldsymbol{H}_*^{-\frac{1}{2}})t\right)\boldsymbol{x}_0+\sqrt{\frac{\eta}{b}}\int_0^t\exp\left(-(\boldsymbol{H}_*^{\frac{1}{2}}+\lambda\boldsymbol{H}_*^{-\frac{1}{2}})(t-t')\right)\mathsf{d}\boldsymbol{\zeta}_{t'}.$$

Let $\boldsymbol{M}=\mathbb{E}\left[\boldsymbol{x}_t\boldsymbol{x}_t^\top\right]$. In this way, considering $(\boldsymbol{H}_*^{\frac{1}{2}}+\lambda\boldsymbol{H}_*^{-\frac{1}{2}}))=(\boldsymbol{H}_*^{\frac{1}{2}}+\lambda\boldsymbol{H}_*^{-\frac{1}{2}}))^\top$, we follow (Mandt et al., 2017) (see their Appendix b) and can further compute the algebraic relation for the stationary covariance of the multivariate Ornstein–Uhlenbeck process as follows:

$$\begin{aligned}&(\boldsymbol{H}_*^{\frac{1}{2}}+\lambda\boldsymbol{H}_*^{-\frac{1}{2}})\boldsymbol{M}+\boldsymbol{M}(\boldsymbol{H}_*^{\frac{1}{2}}+\lambda\boldsymbol{H}_*^{-\frac{1}{2}})\\ =&\frac{\eta}{b}\int_{-\infty}^t(\boldsymbol{H}_*^{\frac{1}{2}}+\lambda\boldsymbol{H}_*^{-\frac{1}{2}})\exp\left(-(\boldsymbol{H}_*^{\frac{1}{2}}+\lambda\boldsymbol{H}_*^{-\frac{1}{2}})(t-t')\right)\exp\left(-(\boldsymbol{H}_*^{\frac{1}{2}}+\lambda\boldsymbol{H}_*^{-\frac{1}{2}})(t-t')\right)\mathsf{d}t'\\ &+\frac{\eta}{b}\int_{-\infty}^t\exp\left(-(\boldsymbol{H}_*^{\frac{1}{2}}+\lambda\boldsymbol{H}_*^{-\frac{1}{2}})(t-t')\right)\exp\left(-(\boldsymbol{H}_*^{\frac{1}{2}}+\lambda\boldsymbol{H}_*^{-\frac{1}{2}})(t-t')\right)\mathsf{d}t'(\boldsymbol{H}_*^{\frac{1}{2}}+\lambda\boldsymbol{H}_*^{-\frac{1}{2}})\\ =&\frac{\eta}{b}\int_{-\infty}^t\frac{\mathsf{d}}{\mathsf{d}t'}\left(\exp\left(-(\boldsymbol{H}_*^{\frac{1}{2}}+\lambda\boldsymbol{H}_*^{-\frac{1}{2}})(t-t')\right)\exp\left(-(\boldsymbol{H}_*^{\frac{1}{2}}+\lambda\boldsymbol{H}_*^{-\frac{1}{2}})(t-t')\right)\right)\\ =&\frac{\eta}{b}\boldsymbol{I},\end{aligned}$$

where we use the lower limits of the integral vanishes by the positivity of the eigenvalues of $\boldsymbol{H}_*^{\frac{1}{2}}+\lambda\boldsymbol{I}$. Next, let $\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top$ is the SVD of $\boldsymbol{H}_*$, where $\boldsymbol{S}=\mathrm{diag}\left(\sigma_1,\sigma_2,\cdots,\sigma_d\right)$. Then we have

$$\boldsymbol{U}(\boldsymbol{S}^{\frac{1}{2}}+\lambda\boldsymbol{S}^{-\frac{1}{2}})\boldsymbol{U}^\top\boldsymbol{M}+\boldsymbol{M}\boldsymbol{U}(\boldsymbol{S}^{\frac{1}{2}}+\lambda\boldsymbol{S}^{-\frac{1}{2}})\boldsymbol{U}^\top=\frac{\eta}{b}\boldsymbol{I}.$$

Then multiplying $\boldsymbol{U}^\top$ from left side and also $\boldsymbol{U}$ from right side gives

$$(\boldsymbol{S}^{\frac{1}{2}} + \lambda \boldsymbol{S}^{-\frac{1}{2}})\boldsymbol{U}^\top \boldsymbol{M}\boldsymbol{U} + \boldsymbol{U}^\top \boldsymbol{M}\boldsymbol{U}(\boldsymbol{S}^{\frac{1}{2}} + \lambda \boldsymbol{S}^{-\frac{1}{2}}) = \frac{\eta}{b}\boldsymbol{I}.$$

Therefore, we know

$$\boldsymbol{M}_{\text{Adam+}\ell_2\text{-Reg.}} = \frac{\eta}{2B}\boldsymbol{U}\boldsymbol{S}^{\frac{1}{2}}(\boldsymbol{S} + \lambda \boldsymbol{I})^{-1}\boldsymbol{U}^\top.$$

Step 2. Generalization Analysis. According to the assumption in Theorem 6 and Lemma 4, we know that for Adam + ℓ_2 regularization, its prior and posterior distributions are both Gaussian distribution, namely $\mathcal{P}_{\text{pre}} \sim \mathcal{N}(0, \rho \boldsymbol{I})$ and $\mathcal{P} \sim \mathcal{N}(\boldsymbol{x}_*, \boldsymbol{M}_{\text{Adam+}\ell_2\text{-Reg.}})$ where

$$\boldsymbol{M}_{\text{Adam+}\ell_2\text{-Reg.}} = \frac{\eta}{2B}\boldsymbol{U}\boldsymbol{S}^{\frac{1}{2}}(\boldsymbol{S} + \lambda \boldsymbol{I})^{-1}\boldsymbol{U}^\top.$$

where $\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top$ is the SVD of $\boldsymbol{H}_*$ in which $\boldsymbol{S} = \operatorname{diag}(\sigma_1, \sigma_2, \cdots, \sigma_d)$.

On the other hand, for KL between two Gaussian distributions $\boldsymbol{W}_1 \sim (\boldsymbol{u}_1, \boldsymbol{\Sigma}_1)$ and $\boldsymbol{W}_2 \sim (\boldsymbol{u}_2, \boldsymbol{\Sigma}_2)$, we can follow (Pardo, 2018) and can compute it as follows:

$$\mathrm{KL}(\boldsymbol{W}_2 \| \boldsymbol{W}_1) = \frac{1}{2}\left[\log \frac{\det(\boldsymbol{\Sigma}_1)}{\det(\boldsymbol{\Sigma}_2)} + \operatorname{Tr}\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2\right)\right] + \frac{1}{2}(\boldsymbol{u}_1 - \boldsymbol{u}_2)^\top \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{u}_1 - \boldsymbol{u}_2) - \frac{d}{2}.$$

Accordingly, for Adam + ℓ_2 regularization, we can compute

$$\begin{aligned}
\mathrm{KL}(\mathcal{P} \| \mathcal{P}_{\text{pre}}) =& \frac{1}{2}\left[\log \frac{\rho^d}{\left(\frac{\eta}{2b}\right)^d \prod_{i=1}^d \frac{\sigma_i^{\frac{1}{2}}}{\sigma_i + \lambda}} + \frac{\eta}{2\rho b}\sum_{i=1}^d \frac{\sigma_i^{\frac{1}{2}}}{\sigma_i + \lambda} + \frac{1}{2\rho}\|\boldsymbol{x}_*\|^2 - \frac{d}{2}\right] \\
=& \frac{1}{2}\left[\sum_{i=1}^d \log \frac{2\rho b(\sigma_i + \lambda)}{\eta \sigma_i^{\frac{1}{2}}} + \frac{\eta}{2\rho b}\sum_{i=1}^d \frac{\sigma_i^{\frac{1}{2}}}{\sigma_i + \lambda} + \frac{1}{2\rho}\|\boldsymbol{x}_*\|^2 - \frac{d}{2}\right]
\end{aligned}$$

This further yields the generalization bound of Adam+ℓ_2-Reg. as follows:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi} \sim \mathcal{D}, \boldsymbol{x} \sim \mathcal{P}}[f(\boldsymbol{x}, \boldsymbol{\xi})] - \mathbb{E}_{\boldsymbol{\xi} \in \mathcal{D}_{\text{tr}}, \boldsymbol{x} \sim \mathcal{P}}[f(\boldsymbol{x}, \boldsymbol{\xi})] \\
\leq& 4\sqrt{\frac{1}{2n}\left(\sum_{i=1}^d \log \frac{2\rho b(\sigma_i + \lambda)}{\eta \sigma_i^{\frac{1}{2}}} + \frac{\eta}{2\rho b}\sum_{i=1}^d \frac{\sigma_i^{\frac{1}{2}}}{\sigma_i + \lambda} + c_0\right)},
\end{aligned}$$

where $c_0 = \frac{1}{2\rho}\|\boldsymbol{x}_*\|^2 - \frac{d}{2} + 2\ln\left(\frac{2n}{\tau}\right)$. The proof is completed. □

G Proofs of Auxiliary Lemmas

G.1 Proof of Lemma 7

Proof. Here we use mathematical induction to prove the first two results. Assume for $t \leq k$, we have $\|\boldsymbol{m}_t\|_\infty \leq c_\infty$ and $\|\boldsymbol{n}_t + \delta\|_\infty \leq c_\infty + \delta$. Then for $k+1$, we have

$$\begin{aligned}
\|\boldsymbol{m}_{k+1}\|_\infty &= \|(1 - \boldsymbol{\beta}_1)\boldsymbol{m}_k + \boldsymbol{\beta}_1 \boldsymbol{g}_k\|_\infty \leq (1 - \boldsymbol{\beta}_1)\|\boldsymbol{m}_k\|_\infty + \boldsymbol{\beta}_1 \|\boldsymbol{g}_k\|_\infty \leq c_\infty, \\
\|\boldsymbol{n}_{k+1}\|_\infty &= \left\|(1 - \boldsymbol{\beta}_2)\boldsymbol{n}_k + \boldsymbol{\beta}_2 \boldsymbol{g}_k^2\right\|_\infty \leq (1 - \boldsymbol{\beta}_2)\|\boldsymbol{n}_k\|_\infty + \boldsymbol{\beta}_2 \left\|\boldsymbol{g}_k^2\right\|_\infty \leq c_\infty^2,
\end{aligned}$$

where $\boldsymbol{g}_k = \frac{1}{b}\sum_{i=1}^b \nabla f(\boldsymbol{x}_k; \boldsymbol{\xi}_i)$. On the other hand, we have

$$\left\|\frac{\boldsymbol{n}_k + \delta}{\boldsymbol{n}_{k+1} + \delta}\right\|_\infty = \left\|1 + \frac{\boldsymbol{n}_k - \boldsymbol{n}_{k+1}}{\boldsymbol{n}_{k+1} + \delta}\right\|_\infty = \left\|1 + \frac{\boldsymbol{\beta}_2(\boldsymbol{n}_k - \boldsymbol{g}_k^2)}{\boldsymbol{n}_{k+1} + \delta}\right\|_\infty \in \left[1 - \frac{\boldsymbol{\beta}_2 c_\infty^2}{c_{s,\infty}^2 + \delta}, 1 + \frac{\boldsymbol{\beta}_2 c_\infty^2}{c_{s,\infty}^2 + \delta}\right]$$

where $\boldsymbol{n}_{k+1} = (1 - \boldsymbol{\beta}_2)\boldsymbol{n}_k + \boldsymbol{\beta}_2 \boldsymbol{g}_k^2$. Therefore, for any $1 \geq p \geq 0$, we can easily obtain

$$\left\|\frac{(\boldsymbol{n}_k + \delta)^p}{(\boldsymbol{n}_{k+1} + \delta)^p}\right\|_\infty \in \left[\left(1 - \frac{\boldsymbol{\beta}_2 c_\infty^2}{c_{s,\infty}^2 + \delta}\right)^p, \left(1 + \frac{\boldsymbol{\beta}_2 c_\infty^2}{c_{s,\infty}^2 + \delta}\right)^p\right] \in \left[1 - \frac{\boldsymbol{\beta}_2 c_\infty^2}{c_{s,\infty}^2 + \delta}, 1 + \frac{\boldsymbol{\beta}_2 c_\infty^2}{c_{s,\infty}^2 + \delta}\right].$$

where $\boldsymbol{n}_{k+1} = (1 - \boldsymbol{\beta}_2)\boldsymbol{n}_k + \boldsymbol{\beta}_2 \boldsymbol{g}_k^2$. The proof is completed. □