GENERAL INSTRUCTION

- **Authors: When you submit your corrections, please either annotate the IEEE Proof PDF or send a list of corrections. Do not send new source files as we do not reconvert them at this production stage.**
- **Authors: Carefully check the page proofs (and coordinate with all authors); additional changes or updates WILL NOT be accepted after the article is published online/print in its final form. Please check author names and affiliations, funding, as well as the overall article for any errors prior to sending in your author proof corrections. Your article has been peer reviewed, accepted as final, and sent in to IEEE. No text changes have been made to the main part of the article as dictated by the editorial level of service for your publication.**
- **Authors: Per IEEE policy, one complimentary proof will be sent to only the Corresponding Author. The Corresponding Author is responsible for uploading one set of corrected proofs to the IEEE Author Gateway.**

QUERIES

Q1. Author: Please confirm or add details for any funding or financial support for the research of this article.
Q2. Author: Please check and confirm whether the author affiliations in the first footnote are correct as set.
Q3. Author: Please provide page range for Refs. [1], [3], [7], [8], [10], [11], [16], [18], [19], [20], [23], [24], [26], [27], [37], [40], [41], [43], [44], and [55].

Short Papers

Towards Understanding Convergence and Generalization of AdamW

Pan Zhou, Xingyu Xie, Zhouchen Lin, *Fellow, IEEE*, and Shuicheng Yan, *Fellow, IEEE*

***Abstract*—AdamW modifies Adam by adding a decoupled weight decay to decay network weights per training iteration. For adaptive algorithms, this decoupled weight decay does not affect specific optimization steps, and differs from the widely used ℓ_2-regularizer which changes optimization steps via changing the first- and second-order gradient moments. Despite its great practical success, for AdamW, its convergence behavior and generalization improvement over Adam and ℓ_2-regularized Adam (ℓ_2-Adam) remain absent yet. To solve this issue, we prove the convergence of AdamW and justify its generalization advantages over Adam and ℓ_2-Adam. Specifically, AdamW provably converges but minimizes a dynamically regularized loss that combines vanilla loss and a dynamical regularization induced by decoupled weight decay, thus yielding different behaviors with Adam and ℓ_2-Adam. Moreover, on both general nonconvex problems and PŁ-conditioned problems, we establish stochastic gradient complexity of AdamW to find a stationary point. Such complexity is also applicable to Adam and ℓ_2-Adam, and improves their previously known complexity, especially for over-parametrized networks. Besides, we prove that AdamW enjoys smaller generalization errors than Adam and ℓ_2-Adam from the Bayesian posterior aspect. This result, for the first time, explicitly reveals the benefits of decoupled weight decay in AdamW. Experimental results validate our theory.**

***Index Terms*—Adaptive gradient algorithms, analysis of AdamW, convergence of AdamW, generalization of AdamW.**

I. Introduction

Adaptive gradient algorithms, e.g., Adam [1], have become the most popular optimizers to train deep networks because of their faster convergence speed than SGD [2], with many successful applications in computer vision [3], [4] and natural language processing [5], *etc.* Similar to the precondition in the second-order algorithms [6], adaptive algorithms precondition the landscape curvature of loss objective to adjust the learning rate for each gradient coordinate. This precondition often helps these adaptive algorithms achieve faster convergence speed than their non-adaptive counterparts, 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 [7], [8], [9], [10].

As a leading adaptive gradient approach, AdamW [11] greatly improves the generalization performance of adaptive algorithms on vision transformers (ViTs) [12] and CNNs [13], [14]. The core of AdamW is a decoupled weight decay. Specifically, AdamW uses an exponential moving average to estimate the first-order moment $\boldsymbol{m}_k$ and second-order moment $\boldsymbol{n}_k$ like Adam, and then updates network weights $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta \boldsymbol{m}_k / \sqrt{\boldsymbol{n}_k + \delta} - \eta\lambda_k \boldsymbol{x}_k$ with a learning rate η, a weight decay parameter λ_k, and a small constant δ. One can observe that AdamW decouples the weight decay from the optimization steps w.r.t. the loss function, since the weight decay is always $-\eta\lambda_k\boldsymbol{x}_k$ no matter what the loss and optimization step are. This decoupled weight decay becomes ℓ_2-regularization for SGD, but differs from ℓ_2-regularization for adaptive algorithms. Thanks to its effectiveness, AdamW has been widely used in network training. But there remain many mysteries about AdamW yet. Firstly, it is 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 Adam and ℓ_2-regularized Adam (ℓ_2-Adam), the theoretical reasons are rarely investigated though heavily desired.

Contributions: 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 justify its superior generalization to (ℓ_2)-Adam. 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 differs from the commonly used ℓ_2-regularization, and thus yields the different behaviors between AdamW and ℓ_2-Adam. For convergence speed, on general nonconvex problems, AdamW finds 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 decaying learning rate, where c_∞ is the ℓ_∞-norm upper bound of stochastic gradient. When ignoring logarithm terms, both complexities match the lower complexity bound $\mathcal{O}(\epsilon^{-4})$ in [15]. These complexities are applicable to Adam and ℓ_2-Adam, 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 decaying learning rate [16], [17], [18], as c_∞ is often much smaller than the network parameter dimension d. 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

Manuscript received 2 February 2023; revised 23 February 2024; accepted 15 March 2024. The work of Pan Zhou was supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant. The work of Zhouchen Lin was supported by the NSF China under Grant 62276004, and in part by the major key project of PCL, China under Grant PCL2021A12.
Q1 Recommended for acceptance by Y. Tai. *(Corresponding author: Shuicheng Yan.)*

Pan Zhou is with the School of Computing and Information Systems,
Q2 Singapore Management University, Singapore 188065 (e-mail: panzhou3@gmail.com).

Xingyu Xie is with the National Key Lab of General AI, School of Intelligence Science and Technology, Peking University, Beijing 100871, China (e-mail: xyxie@pku.edu.cn).

Zhouchen Lin is with the National Key Lab of General AI, School of Intelligence Science and Technology, Peking University, Beijing 100871, China, and with the Institute for Artificial Intelligence, Peking University, Beijing 100871, China, and also with Peng Cheng Laboratory, Shenzhen 518066, China (e-mail: zlin@pku.edu.cn).

Shuicheng Yan is with Skywork AI, Irvine CA 92617 USA (e-mail: shuicheng.yan@gmail.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3382294, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3382294

posterior aspect. Specifically, we show that a proper decoupled weight decay $\lambda_k > 0$ helps AdamW achieve smaller generalization error, indicating the superiority of AdamW over vanilla Adam that corresponds to $\lambda_k = 0$. 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 and its ℓ_2-regularized version.

II. Related Work

Convergence Analysis: Adaptive gradient algorithms, e.g., Adam, have become the default optimizers in deep learning because of their fast convergence speed. Accordingly, many works investigate their convergence to deepen their understanding. On convex problems, Adam-type algorithms, e.g., Adam and AMSGrad [19], enjoy the regret $\mathcal{O}(\sqrt{T})$ under the online learning setting with training iteration number T. For nonconvex problems, Adam-type algorithms have the stochastic gradient complexity $\mathcal{O}(c_\infty\sqrt{d}\epsilon^{-4})$ to find an ϵ-accurate stationary point [18], [20]. RMSProp and Padam [17] are proved to have the complexity $\mathcal{O}(\sqrt{c_\infty}d\epsilon^{-4})$ [16], and Adabelief [21] has $\mathcal{O}(c_2^6\epsilon^{-4})$ complexity, where c_2 is the ℓ_2-norm upper bound of stochastic gradient. But the convergence behaviors of AdamW remains unclear, though it is the dominant optimizer for vision transformers [12] and CNNs [13].

Generalization Analysis: Most works, e.g., [22], [23], [24], 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, by formulating SGD into Brownian- or Lévy-driven SDEs, SGD always provably tends to converge to flat minima and thus enjoys good generalization [9], [24]. Recently, for weight decay, the works [25], [26], [27] intuitively claim that for layers followed by normalizations, e.g., BatchNormalization [28], 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. [29] argued the benefits of weight decay to the layers without normalization, e.g., fully-connected networks, and further empirically found the regularization effects of weight decay to the last fully-connected layer of a network. Unfortunately, none of them explicitly show the generalization benefits of weight decay in AdamW. Here we borrow the aforementioned SDE tool and PAC Bayesian framework [30] to explicitly analyze the generalization effects of decoupled weight decay of AdamW and also its superiority over ℓ_2-Adam.

III. Notation and Preliminarily

AdamW & ℓ_2-Adam: We first briefly recall the steps of AdamW, Adam and ℓ_2-Adam to solve the stochastic nonconvex 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 problem (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 three constants $\beta_1 \in [0,1]$, $\beta_2 \in [0,1]$ and $\delta > 0$:

$$\boldsymbol{m}_k = (1-\beta_1)\boldsymbol{m}_k + \beta_1\boldsymbol{g}_k, \;\; \boldsymbol{n}_k = (1-\beta_2)\boldsymbol{n}_k + \beta_2\boldsymbol{g}_k^2,$$
$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta\boldsymbol{m}_k/\sqrt{\boldsymbol{n}_k+\delta} - \eta\lambda_k\boldsymbol{x}_k, \tag{2}$$

where $\boldsymbol{m}_0=\boldsymbol{g}_0$, $\boldsymbol{n}_0=\boldsymbol{g}_0^2$, and all operations (e.g., product, division) involved vectors are element-wise. Here we allow λ_k to evolve along iteration number k, as in practice, an evolving λ_k often shows better performance than a fixed one [4], [31], [32], [33]. See detailed AdamW in Algorithm 1 of Appendix B, available online. AdamW differs from vanilla Adam in the third step of (2). Specifically, AdamW decouples weight decay from the optimization steps, as weight decay is always $-\eta\lambda_k\boldsymbol{x}_k$ no matter what the loss and optimization step are. But ℓ_2-Adam adds a conventional weight decay $\lambda_k\boldsymbol{x}_k$ into the gradient estimation $\boldsymbol{g}_k = \frac{1}{b}\sum_{i=1}^{b}\nabla f(\boldsymbol{x}_k;\boldsymbol{\xi}_i)+\lambda_k\boldsymbol{x}_k$, then updates $\boldsymbol{m}_k$ and $\boldsymbol{n}_k$ in (2), and $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta\boldsymbol{m}_k/\sqrt{\boldsymbol{n}_k+\delta}$. The decoupled weight decay in AdamW often achieves better generalization than ℓ_2-Adam on many networks, e.g., [12], [14].

Analysis Assumptions: Here we introduce necessary assumptions for analysis, which are commonly used in [1], [8], [19], [34], [35], [36].

Assumption 1 (L-smoothness): The function $f(\cdot,\cdot)$ is L-smooth w.r.t. the parameter, if $\exists L > 0$, for $\forall\boldsymbol{x}_1, \boldsymbol{x}_2$ and $\boldsymbol{\xi} \sim \mathcal{D}$, we have

$$\|\nabla f(\boldsymbol{x}_1,\boldsymbol{\zeta}) - \nabla f(\boldsymbol{x}_2,\boldsymbol{\zeta})\|_2 \le L\,\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2 .$$

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

$$\mathbb{E}[\boldsymbol{g}_k] = \nabla F(\boldsymbol{x}_k),\; \|\boldsymbol{g}_k\|_\infty \le c_\infty,\; \mathbb{E}[\|\nabla F(\boldsymbol{x}_k) - \boldsymbol{g}_k\|_2^2] \le \sigma^2.$$

When a nonconvex problem satisfies 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 $\Omega(\epsilon^{-4})$ [15]. Next, we introduce Polyak-Łojasiewicz (PŁ) condition which is widely used in deep network analysis, since as observed or proved in [37], [38], [39], [40], deep neural networks often satisfy PŁ condition at least around a local minimum.

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

IV. Convergence Analysis

Here we first use a specific least square problem to compare the convergence behavior of AdamW and ℓ_2-Adam. Next, we study the convergence of AdamW on general nonconvex problems and show its performance improvement on PŁ-conditioned problems.

A. Results on Specific Least Square Problems

Here we first use a specific least square problem (3) to analyze the different convergence performance of AdamW and ℓ_2-Adam:

$$\min_{\boldsymbol{x}\in\mathbb{R}} F(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{N}(0,1)}\frac{1}{2}\|a\boldsymbol{x} - \boldsymbol{\xi}\|_2^2, \tag{3}$$

where $a \neq 0$ is a constant. Then we state our main results in Theorem 1 whose proof can be found in Appendix G.1, available online.

Theorem 1: Suppose that stochastic gradient $\boldsymbol{g}_k$ is unbiased, $\mathbb{E}[\|\boldsymbol{g}_k\|_2] \le \tau$, and $\mathbb{E}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2 \le \Delta$. Then with learning rate $\eta_k = \mathcal{O}(\frac{1}{k})$ and $\lambda_k = \lambda = \mathcal{O}(\sqrt{k})$, the sequence $\{\boldsymbol{x}_k\}$ generated by AdamW obeys:

$$\mathbb{E}[\|\boldsymbol{x}_k - \boldsymbol{x}_*\|_2] \le \left(1 - 1/\sqrt{k}\right)^{\frac{3\tilde{k}}{2}}\Lambda + \frac{\tau}{k^{\frac{1}{2}+\alpha}},$$

where $\alpha>0$, $\Lambda=\eta_0+\Delta$. With learning rate $\eta_k=\mathcal{O}(\frac{1}{\sqrt{k}})$ and $\lambda_k=\lambda=\mathcal{O}(\sqrt{k})$, the sequence $\{\boldsymbol{x}_k\}$ generated by ℓ_2-Adam obeys:

$$\mathbb{E}[\|\boldsymbol{x}_k - \boldsymbol{x}_*\|_2] \le \left(1 - 1/\sqrt{k}\right)^{\frac{k}{2}}\Lambda + \frac{2\tau}{k^{\frac{1}{2}}}.$$

Theorem 1 shows that AdamW enjoys a faster convergence speed than ℓ_2-Adam on the least square problem in (3). Specifically, the first convergence term $(1 - 1/\sqrt{k})^{\frac{3\tilde{k}}{2}}\Lambda$ in AdamW converges much faster than the corresponding term $(1 - 1/\sqrt{k})^{\frac{k}{2}}\Lambda$ in ℓ_2-Adam. For the second term $\frac{\tau}{k^{\frac{1}{2}+\alpha}}$ in AdamW, it improves the corresponding term in ℓ_2-Adam by a factor of $2\tilde{\ }k^{\alpha}$ ($\alpha>0$). This comparison shows the superiority of AdamW over ℓ_2-Adam, and thus partially explains their different convergence behaviors.

B. Results on Nonconvex Problems

Now we move on to the general and also PŁ conditioned nonconvex problems. We first define a dynamic surrogate function $F_k(\boldsymbol{x})$ at the k-th iteration which is indeed the combination of the vanilla loss $F(\boldsymbol{x})$ in Eq. (1) and a dynamic regularization $\frac{\lambda}{2}\|\boldsymbol{x}\|^2_{\boldsymbol{v}_k}$:

$$F_k(\boldsymbol{x}) = F(\boldsymbol{x}) + \frac{\lambda_k}{2}\|\boldsymbol{x}\|^2_{\boldsymbol{v}_k} = \mathbb{E}_{\boldsymbol{\zeta}}[f(\boldsymbol{\theta};\boldsymbol{\zeta})] + \frac{\lambda_k}{2}\|\boldsymbol{x}\|^2_{\boldsymbol{v}_k}, \quad (4)$$

where $\boldsymbol{v}_k=\sqrt{\boldsymbol{n}_k+\delta}$ and $\|\boldsymbol{x}\|_{\boldsymbol{v}_k}=\sqrt{\langle \boldsymbol{x}, \boldsymbol{v}_k\odot\boldsymbol{x}\rangle}$ with element-wise product $\odot$. To minimize (4), one can approximate vanilla loss $F(\boldsymbol{x})$ by its Taylor expansion, and compute $\boldsymbol{x}_{k+1}$:

$$\boldsymbol{x}_{k+1}\approx \mathrm{argmin}_{\boldsymbol{x}} F(\boldsymbol{x}_k)+\langle\nabla F(\boldsymbol{x}_k), \boldsymbol{x}-\boldsymbol{x}_k\rangle+\frac{1}{2\eta}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2_{\boldsymbol{v}_k} + \frac{\lambda_k}{2}\|\boldsymbol{x}\|^2_{\boldsymbol{v}_k} = \frac{1}{1+\lambda_k\eta}(\boldsymbol{x}_k-\eta\nabla F(\boldsymbol{x}_k)/\boldsymbol{v}_k).$$

Then considering η is very small in practice, one can approximate $\frac{1}{1+\lambda_k\eta}\approx 1-\lambda_k\eta$, and the factor $\lambda_k\eta^2$ 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 full gradient $\nabla F(\boldsymbol{x}_k)$, and thus achieves $\boldsymbol{x}_{k+1}=(1-\lambda_k\eta)\boldsymbol{x}_k-\eta\boldsymbol{m}_k/\boldsymbol{v}_k$ which accords with the update (2) of AdamW. From this process, one can also observe that the dynamic regularizer $\frac{\lambda}{2}\|\boldsymbol{x}\|^2_{\boldsymbol{v}_k}$ is induced by the decoupled weight decay $-\lambda_k\eta\boldsymbol{x}_k$ in AdamW. In the following, we will show that AdamW indeed minimizes the dynamic function $F_k(\boldsymbol{x})$ instead of the vanilla loss $F(\boldsymbol{x})$.

C. Results on General Nonconvex Problems

Following many works which analyze adaptive gradient algorithms [16], [18], [21], [41], [42], we first provide the convergence results of AdamW by using a constant learning rate η.

Theorem 2: Suppose that Assumptions 1 and 2 hold. Let $\boldsymbol{x}_*\in\mathrm{argmin}_{\boldsymbol{x}}F(\boldsymbol{x})$, $\Delta=F(\boldsymbol{x}_0)-F(\boldsymbol{x}_*)$, $\eta\le\frac{\delta^{1.25}b\epsilon^2}{6(c_\infty^2+\delta)^{0.75}\sigma^2\tilde{\ }L}$, $\beta_1\le\frac{\delta^{0.5}b\epsilon^2}{3(c_\infty^2+\delta)^{0.5}\sigma^2}$ and $\beta_2\in(0,1)$ for all iterations, and $\lambda_k=\lambda(1-\frac{\beta_2 c_\infty^2}{\delta})^k$ with a constant λ. After $T=\mathcal{O}(\max(\frac{c_\infty^{2.5}L\Delta\sigma^2}{\delta^{1.25}b\epsilon^4},\frac{c_\infty^2\sigma^4}{\delta b^2\epsilon^4}))$ iterations, the sequence $\{\boldsymbol{x}_k\}_{k=0}^T$ of AdamW in (2) obeys

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\Big[\|\nabla F_k(\boldsymbol{x}_k)\|_2^2\Big]\le\epsilon^2,\quad \frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\Big[\|\boldsymbol{x}_k-\boldsymbol{x}_{k+1}\|^2_{\boldsymbol{v}_k}\Big]\le\frac{\eta^2\epsilon^2}{4},$$
$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\boldsymbol{m}_k-\nabla F(\boldsymbol{x}_k)\|_2^2\right]\le 8\epsilon^2. \quad (5)$$

Moreover, the total stochastic gradient complexity to achieve (5) is $\mathcal{O}(\max(\frac{c_\infty^{2.5}L\Delta\sigma^2}{\delta^{1.25}\epsilon^4},\frac{c_\infty^2\sigma^4}{\delta b\epsilon^4}))$.

See its proof in Appendix G.2, available online. Theorem 2 shows the convergence of AdamW on the nonconvex problems. Within $T=\mathcal{O}(\max(\frac{c_\infty^{2.5}L\Delta\sigma^2}{\delta^{1.25}b\epsilon^4},\frac{c_\infty^2\sigma^4}{\delta b^2\epsilon^4}))$ iterations, the average gradient $\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}[\|\nabla F_k(\boldsymbol{x}_k)\|_2^2]$ is smaller than ϵ^2, indicating the convergence of AdamW. Now we show small $\|\nabla F_k(\boldsymbol{x}_k)\|_2$ guarantees small $\|\nabla F(\boldsymbol{x}_k)\|_2$ in Corollary 1 with proof in Appendix G.3, available online.

Corollary 1: Assume that $\|\boldsymbol{v}_k\|_2\le\rho'\|\nabla F(\boldsymbol{x}_k)\|_2$ with a constant $\rho'>0$, and $1>\lambda_k\rho'\|\boldsymbol{x}_k\|_\infty$. We have $\|\nabla F(\boldsymbol{x}_k)\|_2\le\frac{1}{1-\lambda_k\rho'\|\boldsymbol{x}_k\|_\infty}\|\nabla F_k(\boldsymbol{x}_k)\|_2$.

The assumptions in Corollary 1 are mild. As $\boldsymbol{n}_k$ is the moving average of stochastic square version of full gradient $\nabla F(\boldsymbol{x}_k)$, one can assume $\|\boldsymbol{n}_k\|_2\le\rho\|\nabla F(\boldsymbol{x}_k)\|_2^2$, especially for the late training phase where $\boldsymbol{x}_k$ is updated very slowly. Indeed, this assumption is validated in Adam analysis works, e.g., [9]. Specifically, since δ is extremely small in $\boldsymbol{v}_k=\sqrt{\boldsymbol{n}_k+\delta}$, one can find a constant $\rho'\approx\rho$ so that $\|\boldsymbol{v}_k\|_2\le\|\nabla F(\boldsymbol{x}_k)\|_2$. For assumption $1>\lambda_k\rho'\|\boldsymbol{x}_k\|_\infty$, it is mild, since a) λ_k is often very small in practice, e.g., 10^{-4}, and b) the magnitude $\|\boldsymbol{x}_k\|_\infty$ of network parameter is not large as observed and proved in [43] because of the auto-adaptive tradeoff among the parameter magnitude at different layers. Also, we empirically find $\|\boldsymbol{x}_k\|_\infty\approx 8.0$ in the well-trained ViT-small across different training epoch numbers. Indeed, for ρ', Zhou et al. [9] empirically finds it around 1.0 on CNNs (see their Fig. 2).

The second inequality in (5) 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 Eq. (5) reveals that the exponential moving average (EMA) $\boldsymbol{m}_k$ of all historical stochastic gradient is close to the full gradient $\nabla F(\boldsymbol{x}_k)$ and explains the success of EMA gradient estimation.

Besides, in Theorem 2, to find an ϵ-accurate first-order stationary point (ϵ-ASP), the stochastic gradient complexity of AdamW is $\mathcal{O}(c_\infty^{2.5}\epsilon^{-4})$ which matches the lower bound $\Omega(\epsilon^{-4})$ in [15] (up to constant factors). Moreover, AdamW enjoys lower complexity than Adabelief [21] of $\mathcal{O}(c_2^6\epsilon^{-4})$ and LAMB [44] of $\mathcal{O}(c_2\sqrt{d}\epsilon^{-4})$, 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, its ℓ_∞-norm c_∞ is often much smaller than its ℓ_2-norm c_2, and can be $\sqrt{d}\times$ smaller for the best case. Appendix D, available online, discusses the proof technique differences among ours and the above works. One can extend the results in Theorem 2 to ℓ_2-Adam. See the proof of Corollary 2 in Appendix G.4, available online.

Corollary 2: With the same parameter settings in Theorem 2, to achieve (5), the total stochastic gradient complexity of Adam and ℓ_2-Adam is $\mathcal{O}(\max(\frac{c_\infty^{2.5}L\Delta\sigma^2}{\delta^{1.25}\epsilon^4},\frac{c_\infty^2\sigma^4}{\delta b\epsilon^4}))$.

Corollary 2 shows that the complexity of Adam and ℓ_2-Adam is $\mathcal{O}(c_\infty^{2.5}\epsilon^{-4})$, and is superior than the previously known complexity $\mathcal{O}(c_\infty\sqrt{d}\epsilon^{-4})$ of Adam-type optimizers analyzed in [16], [17], [18], e.g., (ℓ_2-)Adam, AdaGrad [34], AdaBound [8]. Though sharing the same complexity with Adam and ℓ_2-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., [12], and theoretically in our Section V.

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., [16], [21], [44]) except for [18]. Theorem 2 states our main results.

Theorem 3: Suppose that Assumptions 1 and 2 hold. Let $\eta_k=\frac{\gamma\delta^{0.75}}{2(c_\infty^2+\delta)^{0.25}L\sqrt{k+1}}$, $\beta_1\tilde{\ }_k=\frac{\gamma}{\sqrt{k+1}}$, $\beta_2\tilde{\ }_k=\beta_2\in(0,1)$ with $\gamma=\max(1,\frac{c_\infty^{0.25}L^{0.5}\Delta^{0.5}}{\delta^{0.125}\sigma})$, and $\lambda_k=\lambda(1-\frac{\beta_2c_\infty^2}{\delta})^k$ with a constant λ for the k-th training iteration. To achieve the results in (5) with η replaced by η_1, the stochastic gradient complexity of AdamW in (2) is $\mathcal{O}(\max(\frac{c_\infty^{1.25}L^{0.5}\Delta^{0.5}\sigma}{\delta^{0.625}\epsilon^4}\log(\frac{1}{\epsilon}),\frac{c_\infty\sigma^2}{\delta^{0.5}\epsilon^4}\log(\frac{1}{\epsilon})))$.

See its proof in Appendix G.5, available online. Theorem 3 shows that with decaying learning rate $\eta_k=\frac{1}{\sqrt{k+1}}$, AdamW converges and shares almost the same results in Theorem 2 where it uses constant learning rate. To achieve ϵ-ASP, the complexity of AdamW with decaying learning rate is $\mathcal{O}(\max(\frac{c_\infty^{1.25}L^{0.5}\Delta^{0.5}\sigma}{\delta^{0.625}\epsilon^4}\log(\frac{1}{\epsilon}), \frac{c_\infty\sigma^2}{\delta^{0.5}\epsilon^4}\log(\frac{1}{\epsilon})))$ and slightly differs from the one $\mathcal{O}(\max(\frac{c_\infty^{2.5}L\Delta\sigma^2}{\delta^{1.25}\epsilon^4}, \frac{c_\infty^2\sigma^4}{\delta b\epsilon^4}))$ of AdamW using constant learning rate. By comparing each complexity term, decaying 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 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 $\log(\frac{1}{\epsilon})$, as the ℓ_1-norm upper bound c_∞ of stochastic gradient is often not small and δ is very small, e.g., 10^{-4} by default, decaying learning rate is superior than constant one which accords with the practical observations. When 1) $\lambda_k = 0$ or 2) the loss $F(\boldsymbol{x})$ is a ℓ_2-regularized loss, Theorem 3 still holds. So the stochastic complexity in Theorem 3 is applicable to ℓ_2-Adam. Guo et al. [18] proved the complexity $\mathcal{O}(\max(\frac{c_\infty^{2.5}L^2\sigma^2}{\delta^{2.5}\epsilon^4}\log(\frac{1}{\epsilon}), \frac{c_\infty^2\sigma^4}{\delta^2\epsilon^4}\log(\frac{1}{\epsilon})))$ of Adam-type algorithms, e.g., Adam and ℓ_2-Adam, with decaying learning rate, which but is inferior than the complexity in this work, since as aforementioned, δ is often very small.

D. 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 [37], [38], deep learning models often satisfy PŁ condition at least around a local minimum. For this special nonconvex problem, we follow [18], 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 $\mathbb{E}[F_k(\boldsymbol{x}_k) - F_k(\boldsymbol{x}_*)] \le \epsilon_k$, where $\boldsymbol{x}_* \in \text{argmin}_{\boldsymbol{x}} F(\boldsymbol{x})$, $\boldsymbol{x}_k$ is the output of the k-stage and $\epsilon_k = \frac{1}{2^k}[F_0(\boldsymbol{x}_0) - F_0(\boldsymbol{x}_*)]$ denotes the optimization accuracy. See detailed Algorithm 2 in Appendix B, available online. At below, we provide the convergence results of AdamW under both settings of constant or decayed learning rate in Theorem 4 with proof in Appendix G.6, available online.

Theorem 4: Suppose Assumptions 1 and 2 hold, and $\boldsymbol{x}_* \in \text{argmin}_{\boldsymbol{x}} F(\boldsymbol{x})$. Assume the loss $F_k(\boldsymbol{x}_k)$ in (4) and $F_k(\boldsymbol{x}_*)$ satisfy the PŁ condition in Assumption 3.

1) For constant learning rate setting, assume a constant learning rate $\eta_k \le \frac{\delta^{1.25}\mu b\epsilon_k}{12(c_\infty^2+\delta)^{0.75}\sigma^{2\,\tilde{}}L}$, constant $\beta_{1\,\tilde{}\,k} \le \frac{\delta^{0.5}\mu b\epsilon_k}{6(c_\infty^2+\delta)^{0.5}\sigma^2}$, $\beta_{2\,\tilde{}\,k} \in (0,1)$ and $\lambda_k = \lambda(1-\frac{\beta_2 c_\infty^2}{\delta})^k$ at the k-th stage. We have:

1.1) For the k-th stage, AdamW runs at most $T_k = \mathcal{O}(\max(\frac{c_\infty^{2.5}L\sigma^2}{\mu^2\delta^{1.25}b\epsilon_k}, \frac{c_\infty^2\sigma^2}{\mu\delta b\epsilon_k}))$ iterations to achieve $\mathbb{E}[F_k(\boldsymbol{x}_k) - F_k(\boldsymbol{x}_*)] \le \epsilon_k$, 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 $\mathcal{O}(\max(\frac{c_\infty^{2.5}L\sigma^2}{\mu^2\delta^{1.25}\epsilon}, \frac{c_\infty^2\sigma^2}{\mu\delta\epsilon}))$ to achieve

$$\min_{1\le k\le K} \mathbb{E}\left[F_k(\boldsymbol{x}_k) - F_k(\boldsymbol{x}_*)\right] \le \epsilon. \tag{6}$$

2) For decaying learning rate setting, let $\eta_{k_i} \le \frac{\gamma\delta^{0.75}}{2(c_\infty^2+\delta)^{0.25}L\sqrt{i+1}}$, $\beta_{1k_i} \le \frac{\gamma}{\sqrt{i+1}}$, $\beta_{2k_i} = \beta_{2\,\tilde{}\,k} \in (0,1)$, $\lambda_{k_i} = \lambda(1-\frac{\beta_2 c_\infty^2}{\delta})^i$ at the i-th iteration of the k-th stage with $\gamma = \max(1, \frac{(c_\infty^2+\delta)^{0.125}L^{0.5}b^{0.5}\epsilon_k^{0.5}}{\delta^{0.125}\sigma})$.

2.1) For the k-th stage, AdamW runs at most $T_k = \mathcal{O}(\frac{c_\infty^{2.5}L\sigma^2}{\mu^2\delta^{1.25}b\epsilon})$ iterations to achieve $\mathbb{E}[F_k(\boldsymbol{x}_k) - F_k(\boldsymbol{x}_*)] \le \epsilon_k$, 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 distribution $\{\frac{\eta_{k_i}}{\sum_{j=1}^{T_k}\eta_{k_j}}\}_{i=1}^{T_k}$.

2.2) The total complexity is $\mathcal{O}(\frac{c_\infty^{2.5}L\sigma^2}{\mu^2\delta^{1.25}\epsilon})$ to achieve (6).

Theorem 4 shows that AdamW can converge under both constant and decaying learning rate settings. Moreover, by comparison, to achieve ϵ-ASP in (6), the decaying learning rate has the total complexity $\mathcal{O}(\frac{c_\infty^{2.5}L\sigma^2}{\mu^2\delta^{1.25}\epsilon})$, and could be better than the constant learning rate whose complexity is $\mathcal{O}(\max(\frac{c_\infty^{2.5}L\sigma^2}{\mu^2\delta^{1.25}\epsilon}, \frac{c_\infty^2\sigma^2}{\mu\delta\epsilon}))$. 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.

V. Generalization Analysis

A. Generalization Results

Analysis on hypothesis posterior: As shown in the classical PACBayesian framework [30], [45] 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 investigate the generalization error of AdamW. Specifically, following [9], [22], [23], [24], [46], we 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. Firstly, the updating rule of AdamW can be formulated as

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

where $\boldsymbol{u}_t = \nabla F(\boldsymbol{x}_t) - \boldsymbol{m}_t$ is gradient noise, $\boldsymbol{Q}_t = \text{diag}(\boldsymbol{n}_t^{-\frac{1}{2}})$ is a diagonal matrix. In (7), the small δ in (2) is ignored for convenience which does not affect our following results. Then following [23], [47], [48], we assume the gradient noise $\boldsymbol{u}_t$ obeys Gaussian distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{x}_t})$ because of the Central Limit theory. Accordingly, one can write the SDE of AdamW as

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

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

$$\boldsymbol{C}_{\boldsymbol{x}_t} = \frac{1}{b}\left[\frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla f(\boldsymbol{x}_t;\boldsymbol{\zeta}_i)\nabla f(\boldsymbol{x}_t;\boldsymbol{\zeta}_i)^\top - \nabla F(\boldsymbol{x}_t)\nabla F(\boldsymbol{x}_t)^\top\right],$$

where n is the training sample number, and b is minibatch size. For analysis, we make some necessary assumptions.

Assumption 4: a) Assume $\boldsymbol{C}_{\boldsymbol{x}_t}$ can approximate the Fisher matrix $\boldsymbol{F}_{\boldsymbol{x}_t} = \frac{1}{n}\sum_{i=1}^{n}\nabla F(\boldsymbol{x}_t;\boldsymbol{\zeta}_i)\nabla F(\boldsymbol{x}_t;\boldsymbol{\zeta}_i)^\top$, i.e., $\boldsymbol{C}_{\boldsymbol{x}_t} \approx \boldsymbol{F}_{\boldsymbol{x}_t}$. **b)** Assume $\boldsymbol{F}_{\boldsymbol{x}_t}$ can approximate the Hessian matrix $\boldsymbol{H}_{\boldsymbol{x}_t}$ near a minimum, i.e., $\boldsymbol{F}_{\boldsymbol{x}_t} \approx \boldsymbol{H}_{\boldsymbol{x}_t}$. **c)** Suppose $\boldsymbol{n}'_{t+1} = (1-\beta_2)\boldsymbol{n}'_t + \beta_2\boldsymbol{g}_t\boldsymbol{g}_t^\top$ (virtual sequence) with $\boldsymbol{n}'_0 = \boldsymbol{g}_0\boldsymbol{g}_0^\top$ is a good estimation to $\boldsymbol{F}_{\boldsymbol{x}_t}$, i.e., $\boldsymbol{n}'_{t+1} \approx \boldsymbol{F}_{\boldsymbol{x}_t}$.

Assumption 4 is widely used. Specifically, we follow [23], [47], [48], and approximate $\boldsymbol{C}_{\boldsymbol{x}_t} \approx \boldsymbol{F}_{\boldsymbol{x}_t}$, since we analyze the local convergence around an optimum, leading to 1) $\nabla F(\boldsymbol{x}_t) \approx 0$ and 2) a dominated variance of gradient noise. Assumption 4 b) is used in [24], [49] for analysis, and holds when $\boldsymbol{x}_t$ is around a minimum. Since most works analyze the generalization performance of an algorithm around a local minimum, e.g., [9], [23], [24], [46], [47], [47], [48], [50], Assumption 4 b) holds in their setting and thus is mild. For Assumption 4 c), Staib et al. [51] proved that the matrix-based second-order moment $\boldsymbol{n}_t'$ is a good estimation to the Fisher matrix $\boldsymbol{F}_{\boldsymbol{x}_t}$ after running a certain iteration number. Please refer to the theoretical details of Assumption 4 in Appendix E, available online. Then we can derive the hypothesis posterior learnt by AdamW.

Lemma 5: Assume the loss can be approximated by a second-order Taylor approximation, i.e., $F(\boldsymbol{x}) \approx F(\boldsymbol{x}^*) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^*)^\top \boldsymbol{H}_*(\boldsymbol{x}-\boldsymbol{x}^*)$ where $\boldsymbol{H}_*$ is systemic. With Assumption 4, the solution $\boldsymbol{x}_t$ of AdamW obeys a Gaussian distribution $\mathcal{N}(\boldsymbol{x}_*, \boldsymbol{M}_{\text{AdamW}})$ where the covariance matrix $\boldsymbol{M}_{\text{AdamW}} = \mathbb{E}[\boldsymbol{x}_t \boldsymbol{x}_t^\top]$ is defined as

$$\boldsymbol{M}_{\text{AdamW}} = \frac{\eta}{2b}(\boldsymbol{Q}\boldsymbol{H}_* + \lambda \boldsymbol{I})^{-1}\boldsymbol{Q}\boldsymbol{H}_*\boldsymbol{Q}.$$

where $\boldsymbol{Q} = \text{diag}[\boldsymbol{H}_{*(11)}^{-\frac{1}{2}}, \boldsymbol{H}_{*(22)}^{-\frac{1}{2}}, \ldots, \boldsymbol{H}_{*(dd)}^{-\frac{1}{2}}]$ is diagonal matrix.

See its proof in Appendix H.1, available online. Lemma 5 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 an SDE aspect. From the covariance matrix $\boldsymbol{M}_{\text{AdamW}}$, one can see that all singular values of $\boldsymbol{M}_{\text{AdamW}}$ become smaller when increases and is large enough to ensure $\boldsymbol{Q}\boldsymbol{H}_* + \lambda\boldsymbol{I} \succeq \boldsymbol{0}$. This indicates that proper weight decay in AdamW can stabilize the algorithm, and benefits its convergence to the minimizer $\boldsymbol{x}^*$.

Generalization analysis: Based on the above posterior analysis, we employ the PAC Bayesian framework [30] 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 5. 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 that the prior hypothesis satisfies Gaussian distribution $\mathcal{P}_{\text{pre}} \sim \mathcal{N}(\boldsymbol{0}, \rho\boldsymbol{I})$ [13], [50], [52], since we do not know any information on the posterior hypothesis. Based on Lemma 5, we can derive the generalization error bound of AdamW.

Theorem 6: Assume that $\boldsymbol{x}_0$ satisfies $\mathcal{P}_{\text{pre}} \sim \mathcal{N}(\boldsymbol{0}, \rho\boldsymbol{I})$. Then with at least probability $1-\tau$ ($\tau \in (0,1)$), the expected risk for the posterior hypothesis $\boldsymbol{x} \sim \mathcal{P}$ of AdamW learned on training dataset $\mathcal{D}_{\text{tr}} \sim \mathcal{D}$ with n samples holds

$$\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{D},\boldsymbol{x}\sim\mathcal{P}}[f(\boldsymbol{x},\boldsymbol{\xi})] - \mathbb{E}_{\boldsymbol{\xi}\in\mathcal{D}_{\text{tr}},\boldsymbol{x}\sim\mathcal{P}}[f(\boldsymbol{x},\boldsymbol{\xi})] \leq \Phi_{\text{AdamW}},$$

where $\Phi_{\text{AdamW}} = \frac{\sqrt{8}}{\sqrt{n}}(AdamW + c_0)^{\frac{1}{2}}$ with $AdamW = -\log\det(\boldsymbol{M}_{\text{AdamW}}) + \frac{\eta}{2\rho b}\text{Tr}(\boldsymbol{M}_{\text{AdamW}}) + d\log\frac{2b\rho}{\eta}$, $c_0 = \frac{1}{2\rho}\|\boldsymbol{x}_*\|^2 - \frac{d}{2} + 2\ln(\frac{2n}{\tau})$. Here $\det(M)$ and $\text{tr}(M)$ denote the determinant and trace of matrix M respectively.

See its proof in Appendix H.2, available online. Theorem 6 shows that the generalization error of AdamW is upper bounded by $\mathcal{O}(\frac{1}{\sqrt{n}})$ (up to other factors) which matches the error bound in [53], [54], [55], [56] derived from the PAC theory or stability aspects. When λ is large, the first term $-\log\det(\boldsymbol{M}_{\text{AdamW}})$ in $\boldsymbol{M}_{\text{AdamW}}$ becomes larger since the singular values of $\boldsymbol{M}_{\text{AdamW}}$ become small, and leads to small $\det(\boldsymbol{M}_{\text{AdamW}})$, while the second term $\frac{\eta}{2\rho b}\text{Tr}(\boldsymbol{M}_{\text{AdamW}})$ is small. But for small λ, the first term $-\log\det(\boldsymbol{M}_{\text{AdamW}})$ is small, while the second term becomes large. Though it is hard to precisely decide the best λ, from the above discussion, at least we know that tuning λ can yield smaller generalization error, partly explaining the better performance of AdamW over vanilla Adam ($\lambda = 0$).

B. Comparison With ℓ_2-Regularized Adam

Now we compare AdamW with ℓ_2-Adam. 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., [57], [58], set $\beta_1 = 1$ in (2) to focus on concurrent optimization process of adaptive algorithms. Here we follow this setting to investigate ℓ_2-Adam with updating rule:

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

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

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

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

Theorem 7: Assume $\boldsymbol{x}_0$ satisfies $\mathcal{P}_{\text{pre}} \sim \mathcal{N}(\boldsymbol{0}, \rho\boldsymbol{I})$. With at least probability $1-\tau$ and a constant c_0 in Theorem 6, the expected risk for the posterior hypothesis $\boldsymbol{x} \sim \mathcal{P}_{\ell_2\text{-Adam}}$ of ℓ_2-Adam learned on training dataset $\mathcal{D}_{\text{tr}} \sim \mathcal{D}$ with n samples can be upper bounded:

$$\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{D},\boldsymbol{x}\sim\mathcal{P}_{\ell_2\text{-Adam}}}[f(\boldsymbol{x},\boldsymbol{\xi})] - \mathbb{E}_{\boldsymbol{\xi}\in\mathcal{D}_{\text{tr}},\boldsymbol{x}\sim\mathcal{P}}[f(\boldsymbol{x},\boldsymbol{\xi})] \leq \Phi_{\ell_2\text{-Adam}},$$

where $\Phi_{\ell_2\text{-Adam}} = \frac{\sqrt{8}}{\sqrt{n}}(\ell_2\text{-Adam} + c_0)^{\frac{1}{2}}$ with $\ell_2\text{-Adam} = -\log\det(\boldsymbol{M}_{\text{AdamW}}) + \frac{\eta}{2\rho b}\text{Tr}(\boldsymbol{M}_{\ell_2\text{-Adam}}) + d\log\frac{2b\rho}{\eta}$.

See its proof in Appendix H.3, available online. Theorem 7 shows the generalization error bound $\mathcal{O}(\frac{1}{\sqrt{n}})$ of ℓ_2-Adam. Moreover, when $\lambda = 0$, AdamW and ℓ_2-Adam are exactly the same, and their error bounds are also the same as shown in Theorems 6 and 7.

Next, we compare the generalization error bounds of AdamW and ℓ_2-Adam. To this end, we follow the similar spirit in [9] and approximate $\boldsymbol{Q} \approx \boldsymbol{H}_*^{-\frac{1}{2}}$ to simplify Φ_{AdamW} and $\Phi_{\ell_2\text{-Adam}}$ in the Corollary 3 whose proof can be found in Appendix H.4, available online.

Corollary 3: Assume $\boldsymbol{Q} \approx \boldsymbol{H}_*^{-\frac{1}{2}}$. Then we have

$$\Phi_{\text{AdamW}} \approx \frac{\sqrt{8}}{\sqrt{n}}(\mathsf{err}_{\text{AdamW}} + c_0)^{\frac{1}{2}}, \quad \Phi_{\ell_2\text{-Adam}} \approx \frac{\sqrt{8}}{\sqrt{n}}(\mathsf{err}_{\ell_2\text{-Adam}} + c_0)^{\frac{1}{2}},$$

where $\mathsf{err}_{\text{AdamW}} = \sum_{i=1}^d h(x_{\text{AdamW}}^{(i)})$ with $x_{\text{AdamW}}^{(i)} = 2\eta^{-1}\rho b(\sigma_i^{\frac{1}{2}} + \lambda)$, $\mathsf{err}_{\ell_2\text{-Adam}} = \sum_{i=1}^d h(x_{\ell_2\text{-Adam}}^{(i)})$ with $x_{\ell_2\text{-Adam}}^{(i)} = 2\eta^{-1}\rho b(\sigma_i^{\frac{1}{2}} + \lambda\sigma_i^{-\frac{1}{2}})$. Here $h(x) = \log x + \frac{1}{x}$.

Then we only need to compare the different terms, i.e., $\mathsf{err}_{\text{AdamW}}$ and $\mathsf{err}_{\ell_2\text{-Adam}}$. For $h(x)$, since $h'(x) = \frac{x-1}{x^2}$, $h(x)$ will increase when $x \in (1, +\infty)$. Meanwhile, generally, we have $x_{\ell_2\text{-Adam}}^{(i)} > x_{\text{AdamW}}^{(i)} > 1$ for most $i \in [d]$ due to three 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 [49], [59], [60], [61] and our experimental results on ResNet50 and ViT-small in Fig. 1. 2) The learning rate when reaching the minimum is set to be very small in practice. 3) The minibatch size b is often thousand to train a modern network, and the variance ρ for the initialization distribution $\mathcal{P}_{\text{pre}} \sim \mathcal{N}(\boldsymbol{0}, \rho\boldsymbol{I})$ is often of the order $\mathcal{O}(1/\sqrt{d_i})$ [62], where d_i is input dimension. These factors indicate $x_{\ell_2\text{-Adam}}^{(i)} > x_{\text{AdamW}}^{(i)} > 1$. So the generalization error term $\mathsf{err}_{\text{AdamW}}$ is smaller than $\mathsf{err}_{\ell_2\text{-Adam}}$, testified by our experimental results on ResNet50 and ViT-small in Section VI. So AdamW often enjoys better generalization performance than ℓ_2-Adam, also validated in Section VI. Appendix C, available online, intuitively discusses the generalization benefits of coordinate-adaptive regularization in AdamW.

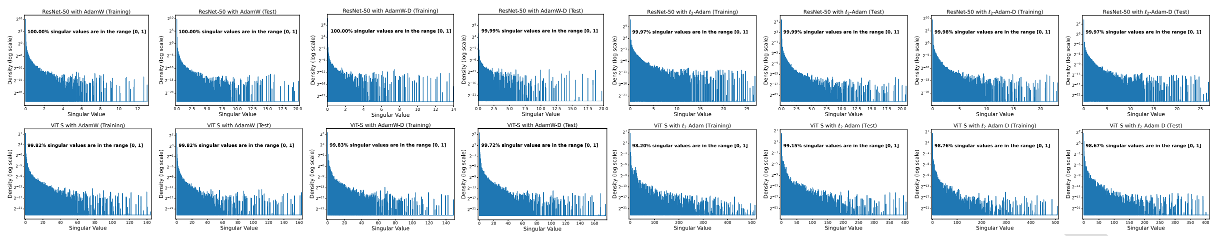

Fig. 1. Visualization of singular values in ResNet50 and ViT-small trained by AdamW (constant weight decay), AdamW-D (decreasing weight decay), ℓ_2-Adam (constant weight decay) and ℓ_2-Adam-D (decreasing weight decay). See more visualization results, e.g., ResNet18, in Fig. 7 of Appendix A, available online.

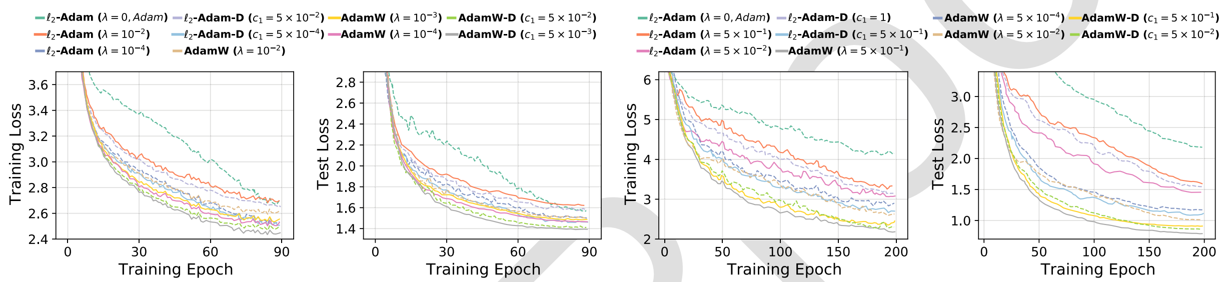

Fig. 2. Training and test curves of ℓ_2-Adam, ℓ_2-Adam-D, AdamW and AdamW-D on ImageNet. See more results in Appendix A, available online.

TABLE I
GENERALIZATION OF ADAMW (CONSTANT WEIGHT DECAY), ADAMW-D (DECAYING WEIGHT DECAY), ℓ_2-ADAM (CONSTANT WEIGHT DECAY) AND ℓ_2-ADAM-D (DECREASING WEIGHT DECAY) ON IMAGENET. ADAMW/-D DENOTES ADAMW/ADAMW-D; ℓ_2-ADAM/-D HAS THE SAME MEANING

model	ResNet18		ResNet50		ViT-small					
train epoch	90		100		100		200		300	
optimizer	AdamW/-D	ℓ_2-Adam/-D	AdamW/-D	ℓ_2-Adam/-D	AdamW/-D	ℓ_2-Adam/-D	AdamW/-D	ℓ_2-Adam/-D	AdamW/-D	ℓ_2-Adam/-D
err in bound	3.43 / 3.40	3.85 / 3.82	3.42 / 3.41	3.78 / 3.77	3.62 / 3.63	3.75 / 3.76	3.58 / 3.57	3.72 / 3.71	3.47 / 3.45	3.70 / 3.69
test acc. (%)	67.9 / 70.1	67.2 / 67.4	77.0 / 77.1	76.5 / 76.4	76.1 / 75.9	75.3 / 75.4	79.2 / 79.3	77.6 / 77.7	79.8 / 80.0	78.5 / 78.6

VI. EXPERIMENTS

Investigation on singular values of Hessian: We respectively use AdamW and ℓ_2-Adam to train two popular networks on ImageNet [63], i.e. ResNet50 [13] and vision transformer small (ViT-small) [3] for both 100 epochs. Then we adopt the method in [64] to estimate the singular values of these two trained networks. AdamW/ℓ_2-Adam uses constant weight decay λ_k, while AdamW-D/ℓ_2-Adam-D adopts exponentially-decaying weight decay $\lambda_k = c_1 \cdot \lambda^k$ with two constants $c_1 > 0$ and $\lambda \in (0,1)$. Fig. 1 plots the spectral density of these singular values on training/test data of ImageNet, and shows that there more than 99% singular values are in the range $[0,1]$ and are much smaller than one. This accords with the observations on AlexNet, VGG and ResNet in [49], [59], [60], [61]. All these observations support the results in Section V-B.

Investigation on generalization: To compute the key generalization error terms i.e., $\bar{\text{err}}_{\text{AdamW}}$ and $\bar{\text{err}}_{\ell_2-\text{Adam}}$ in Theorems 6 and 7, one needs to compute the full Hessian for matrix multiplication that however is prohibitively computable. So we compute their approximations $\text{err}_{\text{AdamW}}$ and $\text{err}_{\ell_2-\text{Adam}}$ in Corollary 3 to compare the generalization error bounds of AdamW and ℓ_2-Adam. For comprehension, we also compute $\text{err}_{\text{AdamW-D}}$ of AdamW-D and $\text{err}_{\ell_2-\text{Adam-D}}$ of ℓ_2-Adam-D which respectively share the same formulation with $\text{err}_{\text{AdamW}}$ and $\text{err}_{\ell_2-\text{Adam}}$ but performs computation on the models respectively trained by AdamW-D and ℓ_2-Adam-D with the above exponentially-decaying weight decay λ_k.

Then we receptively use AdamW, AdamW-D, ℓ_2-Adam and ℓ_2-Adam-D to train three models, i.e., ResNet18, ResNet50 and ViT-small, on ImageNet, and well tune their hyper-parameters, e.g., learning rate and weight decay parameter λ_k. Note, ℓ_2-Adam includes Adam by setting $\lambda_k = 0$. Next, we compute $\text{err}_{\text{AdamW}}$, $\text{err}_{\text{AdamW-D}}$, $\text{err}_{\ell_2-\text{Adam}}$ and $\text{err}_{\ell_2-\text{Adam-D}}$ on the test dataset of ImageNet, as test data can better reveal the generalization ability of an algorithm. Table I shows that on all test cases, $\text{err}_{\text{AdamW}}$ and $\text{err}_{\text{AdamW-D}}$ are smaller than $\text{err}_{\ell_2-\text{Adam}}$ and $\text{err}_{\ell_2-\text{Adam-D}}$ by a remarkable margin. $\text{err}_{\text{AdamW-D}}$ and $\text{err}_{\ell_2-\text{Adam-D}}$ respectively enjoy similar values with their corresponding $\text{err}_{\text{AdamW}}$ and $\text{err}_{\ell_2-\text{Adam}}$. These results empirically support the superior generalization error of AdamW over ℓ_2-Adam. Moreover, Table I also reveals that 1) AdamW and AdamW-D have higher test accuracy than ℓ_2- Adam and ℓ_2- Adam-D; 2) AdamW-D (ℓ_2- Adam-D) enjoys very similar performance as AdamW (ℓ_2- Adam). All these results accord with our theoretical results in Section V-B.

Investigation on convergence: We plot the training/test curves of AdamW, AdamW-D, ℓ_2-Adam and ℓ_2-Adam-D on ImageNet in Fig. 2. For AdamW-D and ℓ_2-Adam-D, we fix $\lambda = 0.99999$ and tune c_1 to compute its weight decay λ_k. One can find that on ResNet50 and ViT-small, 1) AdamW and AdamW-D show faster convergence speed than ℓ_2-Adam (including Adam via $\lambda = 0$) and ℓ_2-Adam-D when their weight decay parameter are well-tuned, e.g., $\lambda = 5 \times 10^{-1}$ for AdamW and ℓ_2-Adam, $c_1 = 5 \times 10^{-2}$ for AdamW-D on ViT-small; 2) AdamW and AdamW-D share similar convergence behaviors; 3) weight decay parameter greatly affects the convergence speed of the three optimizers. So under the same training cost, the faster convergence of AdamW could also partially explain its better generalization performance over ℓ_2-Adam.

VII. Conclusion

In this work, we first prove the convergence of AdamW using both constant and decaying 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 Adam and ℓ_2-Adam. Besides, for the first time, we quantitatively justify the generalization superiority of AdamW over both Adam and ℓ_2-Adam. Finally, experimental results validate the implications of our theory.

References

[1] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

[2] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 1951.

[3] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2020.

[4] P. Zhou, Y. Zhou, C. Si, W. Yu, T. K. Ng, and S. Yan, "Mugs: A multi-granular self-supervised learning framework," 2022, *arXiv:2203.14415*.

[5] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 10, pp. 1533–1545, Oct. 2014.

[6] E. Süli and D. F. Mayers, *An Introduction to Numerical Analysis*, Cambridge, U.K.: Cambridge Univ. Press, 2003.

[7] N. S. Keskar and R. Socher, "Improving generalization performance by switching from Adam to SGD," in *Proc. Int. Conf. Learn. Representations*, 2018.

[8] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive gradient methods with dynamic bound of learning rate," in *Proc. Int. Conf. Learn. Representations*, 2018.

[9] P. Zhou et al., "Towards theoretically understanding why SGD generalizes better than Adam in deep learning," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020, pp. 21285–21296.

[10] P. Zhou, X. Xie, and Y. Shuicheng, "Win: Weight-decay-integrated Nesterov acceleration for adaptive gradient algorithms," in *Proc. Int. Conf. Learn. Representations*, 2022.

[11] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Representations*, 2018.

[12] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[14] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11966–11976.

[15] Y. Arjevani, Y. Carmon, J. C. Duchi, D. J. Foster, N. Srebro, and B. Woodworth, "Lower bounds for non-convex stochastic optimization," *Math. Program.*, vol. 199, no. 1/2, pp. 165–214, 2023.

[16] D. Zhou, J. Chen, Y. Cao, Y. Tang, Z. Yang, and Q. Gu, "On the convergence of adaptive gradient methods for nonconvex optimization," in *Proc. Workshop Optim. Mach. Learn.*, 2020.

[17] J. Chen, D. Zhou, Y. Tang, Z. Yang, Y. Cao, and Q. Gu, "Closing the generalization gap of adaptive gradient methods in training deep neural networks," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 3267–3275.

[18] Z. Guo, Y. Xu, W. Yin, R. Jin, and T. Yang, "A novel convergence analysis for algorithms of the Adam family," in *Proc. Workshop Optim. Mach. Learn.*, 2023.

[19] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of Adam and beyond," in *Proc. Int. Conf. Learn. Representations*, 2019.

[20] X. Chen, S. Liu, R. Sun, and M. Hong, "On the convergence of a class of Adam-type algorithms for non-convex optimization," in *Proc. Int. Conf. Learn. Representations*, 2018.

[21] J. Zhuang et al., "Adabelief optimizer: Adapting stepsizes by the belief in observed gradients," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020, pp. 18795–18806.

[22] S. Mandt, M. Hoffman, and D. Blei, "A variational analysis of stochastic gradient algorithms," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 354–363.

[23] Z. Zhu, J. Wu, B. Yu, L. Wu, and J. Ma, "The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects," in *Proc. Int. Conf. Mach. Learn.*, 2018.

[24] S. Jastrzkebski et al., "Three factors influencing minima in SGD," in *Proc. Int. Conf. Learn. Representations*, 2017.

[25] Twan Van Laarhoven, "L2 regularization versus batch and weight normalization," 2017, *arXiv: 1706.05350*.

[26] G. Zhang, C. Wang, B. Xu, and R. Grosse, "Three mechanisms of weight decay regularization," in *Proc. Int. Conf. Learn. Representations*, 2018.

[27] E. Hoffer, R. Banner, I. Golan, and D. Soudry, "Norm matters: Efficient and accurate normalization schemes in deep networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[29] Y. Zhou, Y. Sun, and Z. Zhong, "FixNorm: Dissecting weight decay for training deep neural networks," 2021, *arXiv:2103.15345*.

[30] D. A. McAllester, "Some PCA-Bayesian theorems," *Mach. Learn.*, vol. 37, no. 3, pp. 355–363, 1999.

[31] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 9650–9660.

[32] S.-H. Gao, Y.-Q. Tan, M.-M. Cheng, C. Lu, Y. Chen, and S. Yan, "Highly efficient salient object detection with 100k parameters," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 702–721.

[33] J. Bjorck, K. Q. Weinberger, and C. Gomes, "Understanding decoupled and early weight decay," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 6777–6785.

[34] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," in *Proc. J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 2011.

[35] P. Zhou, X. Yuan, and J. Feng, "Faster first-order methods for stochastic non-convex optimization on Riemannian manifolds," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2019, pp. 459–472.

[36] P. Zhou, X. Yuan, Z. Lin, and S. Hoi, "A hybrid stochastic-deterministic minibatch proximal gradient method for efficient optimization and generalization," in *Proc. IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 5933–5946, Oct. 2022.

[37] T. M. Moritz Hardt, "Identity matters in deep learning," in *Proc. Int. Conf. Learn. Representations*, 2023.

[38] B. Xie, Y. Liang, and L. Song, "Diverse neural network learns true target functions," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1216–1224.

[39] Z. Charles and D. Papailiopoulos, "Stability and generalization of learning algorithms that converge to global optima," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 745–754.

[40] P. Zhou, H. Yan, X. Yuan, J. Feng, and S. Yan, "Towards understanding why lookahead generalizes better than SGD and beyond," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021.

[41] T. Tijmen and H. Geoffrey, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Netw. Mach. Learn.*, vol. 4, 2012.

[42] X. Xie, P. Zhou, H. Li, Z. Lin, and S. Yan, "Adan: Adaptive nesterov momentum algorithm for faster optimizing deep models," 2022, *arXiv:2208.06677*.

[43] S. S. Du, W. Hu, and J. D. Lee, "Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018.

[44] Y. You et al., "Large batch optimization for deep learning: Training BERT in 76 minutes," in *Proc. Int. Conf. Learn. Representations*, 2019.

[45] T. Liu, D. Tao, M. Song, and S. J. Maybank, "Algorithm-dependent generalization bounds for multi-task learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 227–241, Feb. 2017.

[46] Z. Xie, L. Yuan, Z. Zhu, and M. Sugiyama, "Positive-negative momentum: Manipulating stochastic gradient noise to improve generalization," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 11448–11458.

[47] M. Stephan et al., “Stochastic gradient descent as approximate Bayesian inference,” *J. Mach. Learn. Res.*, vol. 18, no. 134, pp. 1–35, 2017.

[48] S. L. Smith and Q. V. Le, “A Bayesian perspective on generalization and stochastic gradient descent,” in *Proc. Int. Conf. Learn. Representations*, 2018.

[49] B. Ghorbani, S. Krishnan, and Y. Xiao, “An investigation into neural net optimization via Hessian eigenvalue density,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2232–2241.

[50] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, *arXiv:1409.1556*.

[51] M. Staib, S. Reddi, S. Kale, S. Kumar, and S. Sra, “Escaping saddle points with adaptive gradient methods,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5956–5965.

[52] C. Szegedy et al., “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[53] V. Vapnik, *Estimation of Dependences Based on Empirical Data*, Berlin, Germany: Springer, 2006.

[54] M. Hardt, B. Recht, and Y. Singer, “Train faster, generalize better: Stability of stochastic gradient descent,” in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1225–1234.

[55] P. Zhou and J. Feng, “Empirical risk landscape analysis for understanding deep neural networks,” in *Proc. Int. Conf. Learn. Representations*, 2018.

[56] P. Zhou and J. Feng, “Understanding generalization and optimization performance of deep CNNs,” in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5960–5969.

[57] K. Lyu, Z. Li, and S. Arora, “Understanding the generalization benefit of normalization layers: Sharpness reduction,” in *Proc. Conf. Neural Inf. Process. Syst.*, 2022, pp. 34689–34708.

[58] S. Malladi, K. Lyu, A. Panigrahi, and S. Arora, “On the SDEs and scaling rules for adaptive gradient algorithms,” in *Proc. Conf. Neural Inf. Process. Syst.*, 2022, pp. 7697–7711.

[59] L. Sagun, L. Bottou, and Y. LeCun, “Eigenvalues of the Hessian in deep learning: Singularity and beyond,” 2016, *arXiv:1611.07476*.

[60] L. Sagun, U. Evci, V. U. Guney, Y. Dauphin, and L. Bottou, “Empirical analysis of the Hessian of over-parametrized neural networks,” 2017, *arXiv: 1706.04454*.

[61] A. R. Sankar, Y. Khasbage, R. Vigneswaran, and V. N. Balasubramanian, “A deeper look at the Hessian eigenspectrum of deep neural networks and its applications to regularization,” in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 9481–9488.

[62] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.

[63] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[64] Z. Yao, A. Gholami, K. Keutzer, and M. W. Mahoney, “PyHessian: Neural networks through the lens of the Hessian,” in *Proc. Int. conf. Big Data*, 2020, pp. 581–590.