Journal of Machine Learning Research 25 (2024) 1-74 Submitted 8/23; Published 3/24

Win: Weight-Decay-Integrated Nesterov Acceleration for Faster Network Training

Pan Zhou PANZHOU@SMU.EDU.SG
School of Computing and Information Systems, Singapore Management University, Singapore

Xingyu Xie XYXIE@PKU.EDU.CN
National Key Lab of General AI, School of Intelligence Science and Technology, Peking University, China

Zhouchen Lin ZLIN@PKU.EDU.CN
National Key Lab of General AI, School of Intelligence Science and Technology, Peking University, China
Institute for Artificial Intelligence, Peking University, China
Peng Cheng Laboratory, China

Kim-Chuan Toh MATTOHKC@NUS.EDU.SG
Department of Mathematics and Institute of Operations Research and Analytics, National University of Singapore, Singapore

Shuicheng Yan SHUICHENG.YAN@GMAIL.COM
Skywork AI

Editor: Sanjiv Kumar

Abstract

Training deep networks on large-scale datasets is computationally challenging. This work explores the problem of "*how to accelerate adaptive gradient algorithms in a general manner*", and proposes an effective Weight-decay-Integrated Nesterov acceleration (Win) to accelerate adaptive algorithms. Taking AdamW and Adam as examples, per iteration, we construct a dynamical loss that combines the vanilla training loss and a dynamic regularizer inspired by proximal point method, and respectively minimize the first- and second-order Taylor approximations of dynamical loss to update variable. This yields our Win acceleration that uses a conservative step and an aggressive step to update, and linearly combines these two updates for acceleration. Next, we extend Win into Win2 which uses multiple aggressive update steps for faster convergence. Then we apply Win and Win2 to the popular LAMB and SGD optimizers. Our transparent derivation could provide insights for other accelerated methods and their integration into adaptive algorithms. Besides, we theoretically justify the faster convergence of Win- and Win2-accelerated AdamW, Adam and LAMB to their non-accelerated counterparts. Experimental results demonstrates the faster convergence speed and superior performance of our Win- and Win2-accelerated AdamW, Adam, LAMB and SGD over their vanilla counterparts on vision classification and language modeling tasks.

Keywords: Accelerated Adaptive Gradient Algorithms, Deep Learning Optimizer, Network Optimization, Nesterov Acceleration in Deep Learning

1. Introduction

Deep neural networks (DNNs) are effective in modeling realistic data and have been successfully applied to many applications, *e.g.,* image classification (He et al., 2016) and speech recognition (Sainath

et al., 2013). Typically, their training models can be formulated as a nonconvex problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) := \mathbb{E}_{\zeta\sim\mathcal{D}}[f(\mathbf{x},\zeta)] + \frac{\lambda}{2}\|\mathbf{x}\|_2^2, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^d$ is the model parameter, sample ζ is drawn from a data distribution $\mathcal{D}$, the loss f is differentiable, and λ is a constant. Although there are various algorithms available, such as gradient descent (Cauchy et al., 1847) and variance-reduced algorithms (Rie Johnson, 2013), that can solve problem (1), stochastic gradient descent (SGD) (Robbins and Monro, 1951) leverages the compositional structure in (1) to efficiently estimate the gradient using minibatch data. As a result, SGD has emerged as a dominant algorithm for training deep neural networks (DNNs) because of its improved efficiency and effectiveness. Nevertheless, SGD encounters slow convergence speed on the sparse data or ill-conditioned problems (Duchi et al., 2011; Kingma and Ba, 2015), as it uniformly scales the gradient across all parameter coordinates, disregarding the problem-specific properties associated with each coordinate. To address this issue, recent research has introduced adaptive methods like Adam (Kingma and Ba, 2015) and AdamW (Loshchilov and Hutter, 2018), which scale each gradient coordinate based on the current geometry curvature of the loss function $F(\mathbf{x})$. This coordinate-wise scaling significantly accelerates optimization convergence, making methods like Adam and AdamW more popular for DNN training, particularly with transformer models.

Unfortunately, along with the increasing scale of both datasets and models, efficient DNN training even with SGD or adaptive algorithms has become increasingly challenging. In this work, we are particularly interested in the problem of "*how to accelerate the convergence of adaptive algorithms in a general manner*" because of their widespread popularity in various DNNs. While acceleration techniques, such as heavy ball acceleration (Polyak, 1964) and Nesterov acceleration (Nesterov, 2003), are commonly employed in SGD, their application to adaptive algorithms remains largely unexplored. Among the limited studies in this area, NAdam (Dozat, 2016) simplifies Nesterov acceleration by solely estimating the first moment of the gradient in Adam, disregarding the second-order moments, which is not exact Nesterov acceleration and may not inherit its full acceleration potential.

Contributions: In this work, based on a recent Nesterov-type acceleration formulation (Nesterov et al., 2018) and proximal point method (PPM) (Moreau, 1965), we propose a new *Weight-decay-Integrated Nesterov acceleration* (Win[1] for short) to accelerate adaptive algorithms, and also further analyze the convergence of Win-accelerated adaptive algorithms to justify their convergence superiority by taking AdamW, Adam and LAMB as examples. We also further extend Win into a more general version, Win2, for training networks more efficiently. Our main contributions are highlighted below.

Firstly, we use PPM to rigorously derive our Win acceleration for accelerating adaptive algorithms. By taking AdamW and Adam as examples, at the k-th iteration, we follow PPM spirit and minimize a dynamically regularized loss $F(\mathbf{x})+\frac{1}{2\eta_k^x}\|\mathbf{x}-\mathbf{x}_k\|^2_{\sqrt{\mathbf{v}_k+\nu}}$, where $\mathbf{x}_k$ is the previous solution, $\mathbf{v}_k$ is the second-order gradient moment, the small constant ν is to stabilize in AdamW and Adam, and $\|\mathbf{x}\|_{\mathbf{v}_k}=\sqrt{\langle\mathbf{x},\mathbf{v}_k\odot\mathbf{x}\rangle}$ with an element-wise product operation $\odot$. Then to introduce Nesterov-alike acceleration and also make the problem solvable iteratively, we respectively approximate $F(\mathbf{x})$ by its first- and second-order Taylor expansions to update the variable $\mathbf{x}$ twice while always fixing the above dynamic regularization and also an extra regularizer $\frac{1}{2\eta_k^x}\|\mathbf{x}\|^2_{\sqrt{\mathbf{v}_k+\nu}}$ induced by the weight

1. Code is released at `https://github.com/sail-sg/win`.

decay in AdamW. As a result, we arrive at our Win acceleration, a Nesterov-alike acceleration, for AdamW and Adam that uses a conservative step and an aggressive step to update twice and then linearly combines these two updates for acceleration. Since Win is simple and efficient, it brings negligible computational overhead for per iteration cost when plugging it into popular optimizers, *e.g.,* about 2% extra average training time per iteration on AdamW evaluated on ResNet as shown in Sec. 5.4. Then we extend this Win acceleration to LAMB (You et al., 2019) and SGD. The above acceleration derivation is transparent and general which could motivate other accelerations and serve as examples for introducing other accelerations into adaptive gradient algorithms.

Secondly, we prove the convergence of our Win-accelerated AdamW, Adam, LAMB and SGD. For Win-accelerated AdamW and Adam, to find an ϵ-approximate first-order stationary point, their stochastic gradient complexity is $\mathcal{O}\big(\frac{c_\infty^{2.5}}{\nu^{1.25}\epsilon^4}\big)$ and this matches the lower bound $\Omega(\frac{1}{\epsilon^4})$ in (Arjevani et al., 2022, 2020) (up to constant factors) under the same conditions, where c_∞ upper bounds the ℓ_∞ norm of stochastic gradient. Moreover, this complexity improves a factor of $\mathcal{O}(\frac{d}{c_\infty^{0.5}})$ over the complexity $\mathcal{O}(\frac{c_\infty^2 d}{\nu^{1.25}\epsilon^4})$ of Adam-type optimizers in (Zhou et al., 2018; Guo et al., 2021), *e.g.,* Adam, AdaGrad (Duchi et al., 2011), AdaBound (Luo et al., 2018), since the network parameter dimension d is often much larger than $c_\infty^{0.5}$, especially for over-parameterized networks. Indeed, Win-accelerated Adam and AdamW also enjoy superior complexity to other Adam variants, *e.g.,* Adabelief (Zhuang et al., 2020) with compelxity $\mathcal{O}(\frac{c_2^6}{\nu^2\epsilon^4})$, especially on over-parameterized networks, where c_2 is the maximum ℓ_2-norm of stochastic gradient. We also show that Win-accelerated LAMB improves the complexity of LAMB by a factor of $\mathcal{O}\big(\frac{d^{2.5}}{c_\infty^{1.5}}\big)$ which is often large, especially for over-parameterized networks. For Win-accelerated SGD, it enjoys the complexity of $\mathcal{O}(\frac{1}{\epsilon^4})$ which matches the lower bound in (Arjevani et al., 2022, 2020).

Thirdly, we develop a more general Win acceleration, Win2 for short, which extends the parameter update from two steps in Win to multiple steps. To minimize the dynamically regularized loss $F(\mathbf{x})+\frac{1}{2\eta_k^x}\|\mathbf{x}-\mathbf{x}_k\|^2_{\sqrt{\mathbf{v}_k+\nu}}$, Win2 also approximates the vanilla $F(\mathbf{x})$ by its Taylor expansions but at multiple different points. Accordingly, Win2 needs to update the parameter multiple times, and then linearly combines these multiple updates for acceleration. Since multiple updates yield more stable linear combination and thus better stabilize the training, Win2 can use more aggressive stepsize than Win to achieve faster convergence speed as empirically shown in Sec. 5. Besides, we prove that Win2-accelerated AdamW, Adam and LAMB respectively enjoy superior stochastic gradient complexity to vanilla AdamW, Adam and LAMB. For Win2-accelerated SGD, its stochastic gradient complexity also accords with the lower complexity bound in (Arjevani et al., 2022, 2020).

Finally, extensive experimental results on both vision tasks (*e.g.,* classification and segmentation) and language modeling tasks show that our Win- and Win2-accelerated algorithms, *i.e.* accelerated AdamW, Adam, LAMB and SGD, can accelerate the convergence speed and also improve the performance of their corresponding non-accelerated counterparts by a remarkable margin on both CNN and transformer architectures. Moreover, Win2 also shows better empirical acceleration effects and also higher empirical performance than Win. All these results show the strong compatibility, generalization and superiority of our acceleration techniques.

Comparison with our conference work. This paper is an extension of our previous work (Zhou et al., 2023) which proposes Win and analyzes the convergence performance of Win-accelerated AdamW, Adam, and SGD on the stochastic nonconvex problem (1). Compared with its shorter version, this paper makes the following changes. **1)** Our previous work (Zhou et al., 2023) only analyzes Win-accelerated AdamW, Adam and SGD, while this work further analyzes Win-accelerated

LAMB and shows its superior complexity to vanilla LAMB. In practice, LAMB is a very popular optimizer for large minibatch training, and is more complex due to its scaling operation, imposing more challenges for analysis. **2)** It proposes a more general acceleration framework, Win2, which extends the parameter update from two steps in Win to multiple steps, and achieves faster convergence speed empirically. **3)** This work also proves that Win2-accelerated AdamW, Adam and LAMB reveal superior complexity to vanilla AdamW, Adam and LAMB, and Win2-accelerated SGD enjoys a complexity which matches the lower complexity bound. **4)** This work conducts comprehensive experiments on additional tasks, *e.g.,* instance segmentation which includes object detection and mask segmentation, to evaluates the performance of Win, and also Win2 on image classification, detection, segmentation and language modeling tasks.

2. Related Work

In the context of deep learning, when considering efficiency and generalization, one often prefers to employ SGD and adaptive gradient algorithms, *e.g.,* Adam (Kingma and Ba, 2015) , instead of other algorithms, *e.g.,* variance-reduced algorithms (Rie Johnson, 2013), to solve problem (1). But, in practice and theory, adaptive gradient algorithms often suffer from inferior generalization performance than SGD (Zhou et al., 2020a,b). To solve this issue, AdamW (Loshchilov and Hutter, 2018) proposes a decoupled weight decay which introduces an ℓ_2-like regularization into Adam to decay the network weight iteratively, and its effectiveness is widely validated on vision transformers, *e.g.,* ViTs (Touvron et al., 2021), and CNN, *e.g.,* ResNet (He et al., 2016; Touvron et al., 2021; Zhou et al., 2024). Later, to train DNNs with a large batch, LAMB (You et al., 2019) scales the update in AdamW to the weight magnitude for getting rid of too large or small update for faster convergence. But in practice, LAMB suffers unsatisfactory performance on small batch. In this work, we hope to design a general acceleration approach to accelerate the convergence of these algorithms.

Heavy-ball acceleration (Polyak, 1964) and Nesterov acceleration (Nesterov, 2003) are two classical acceleration techniques, and their effectiveness in SGD is well testified. Heavy-ball acceleration moving averages stochastic gradient in SGD for faster convergence, while Nesterov acceleration runs a step along the moving gradient average and then computes gradient at the new point to look ahead for correction. Typically, Nesterov acceleration (Nesterov, 2003) converges faster both empirically and theoretically at least on convex problems, and also has superior generalization on DNNs (Foret et al., 2021; Kwon et al., 2021). Later, NAdam (Dozat, 2016) integrates Nesterov acceleration into the first-order gradient moment estimation but ignores the second-order gradient moments which harms the acceleration effect. Some works (Anil et al., 2022, 2020) also explore Nesterov acceleration for second-order algorithms, *e.g.,* shampoo (Gupta et al., 2018). Recently, for full gradient decent algorithm, a new general Nesterov-type acceleration (Nesterov et al., 2018) directly interpolates two variables to look ahead for correction, and is more flexible than vanilla Nesterov acceleration (Nesterov, 2003) which interpolates the variable and gradient. See discussion in Sec. 3.2. Here we use proximal point method to introduce this new acceleration into adaptive algorithms by a rigorous and transparent derivation, which could provide insights for other accelerated methods and their integration into adaptive gradient algorithms.

3. Weight-decay-Integrated Nesterov Acceleration

In this section, we first use AdamW and Adam as two examples to elaborate on our Weight-decay-Integrated Nesterov (Win) acceleration and also derive Win-accelerated AdamW and Adam in

Sec. 3.1. Then, we extend this acceleration technique to LAMB and SGD in Sec. 3.2. Finally, we analyze the convergence behvaiors of Win-accelerated AdamW, Adam, LAMB and SGD in Sec. 3.3.

To accelerate full gradient descent algorithm, given a full gradient $\nabla F(\mathbf{x}_k)$ of problem (1) at the k-th iteration, Nesterov-type acceleration (Nesterov et al., 2018) generally uses a conservative step η_k^x and an aggressive step η_k^y to update two sequences $\mathbf{x}_{k+1}$ and $\mathbf{y}'_{k+1}$ respectively, and then linearly combines them to update the variable $\mathbf{x}_{k+1}$ of the problem. Similar formulations are also observed and proved in recent works, *e.g.,* (Allen-Zhu and Orecchia, 2017; Bansal and Gupta, 2019; Ahn and Sra, 2022). In general, their acceleration formulation can be formally written as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k^x \nabla F(\mathbf{x}_k), \quad \mathbf{y}'_{k+1} = \mathbf{y}_k - \eta_k^y \nabla F(\mathbf{x}_k), \quad \mathbf{y}_{k+1} = \rho_k^y \mathbf{x}_{k+1} + (1-\rho_k^y)\mathbf{y}'_{k+1}, \quad (2)$$

where $\rho_k^y \in [0,1]$ is a constant. This acceleration enjoys provably faster convergence rate for the full gradient descent method on convex problems (Beck and Teboulle, 2009; Nesterov et al., 2018), and is also empirically validated in many convex and nonconvex cases, *e.g.,* (Wilson et al., 2017; Nado et al., 2021). Despite its effectiveness, such an acceleration is rarely explored in adaptive gradient algorithms, particularly in the realm of network training. In deterministic optimization setting, another widely used optimization stabilization and acceleration approach is the proximal point method (PPM) (Moreau, 1965; Rockafellar, 1976). At the k-th iteration, PPM optimizes an ℓ_2-regularized loss

$$F(\mathbf{x}) + \frac{1}{2\eta_k^x}\|\mathbf{x}-\mathbf{x}_{k-1}\|_2^2$$

instead of the vanilla loss $F(\mathbf{x})$. This small change enhances the convexity of the problem, accelerating and also stabilizing the optimization process (Kim et al., 2022; Zhou et al., 2021). To enable iterative solvability of the ℓ_2-regularized problem, PPM approximates the loss $F(\mathbf{x})$ using either its first- or second-order Taylor expansion, ensuring that each iteration has a closed-form solution (see below). In this work, we draw inspiration from PPM to induce a *Weight-decay-Integrated Nesterov acceleration* (Win) for adaptive gradient algorithms by using AdamW and Adam as examples in Sec. 3.1, and then extend this acceleration technique to LAMB and SGD in Sec. 3.2.

3.1 Win-Accelerated AdamW and Adam

To begin with, following most adaptive gradient algorithms, *e.g.,* Adam and AdamW, we estimate the first- and second-order moments $\mathbf{m}_k$ and $\mathbf{v}_k$ of gradient as follows:

$$\mathbf{g}_k = \frac{1}{b}\sum\nolimits_{i=1}^{b} \nabla f(\mathbf{y}_k;\zeta_i), \quad \mathbf{m}_k = (1-\beta_1)\mathbf{m}_{k-1} + \beta_1 \mathbf{g}_k, \quad \mathbf{v}_k = (1-\beta_2)\mathbf{v}_{k-1} + \beta_2 \mathbf{g}_k^2, \quad (3)$$

where $\mathbf{g}_k$ is the average gradient on a minibatch data of size b, $\beta_1 \in [0,1]$ and $\beta_2 \in [0,1]$. For the initialization, we set $\mathbf{m}_0 = \mathbf{g}_0$, $\mathbf{v}_0 = \mathbf{g}_0^2$. For brevity, with a small scaler $\nu > 0$, we define

$$\mathbf{s}_k = \sqrt{\mathbf{v}_k + \nu}, \qquad \mathbf{u}_k = \mathbf{m}_k / \sqrt{\mathbf{v}_k + \nu}. \quad (4)$$

Then following the spirit of PPM, at the k-th iteration, we minimize a regularized loss $F(\mathbf{x}) + \frac{1}{2\eta_k^x}\|\mathbf{x}-\mathbf{x}_k\|_{\mathbf{s}_k}^2$, where $\|\mathbf{x}\|_{\mathbf{s}_k} = \sqrt{\langle \mathbf{x}, \mathbf{s}_k \odot \mathbf{x}\rangle}$ with an element-wise product operation $\odot$. Here we use the regularizer $\|\mathbf{x}-\mathbf{x}_k\|_{\mathbf{s}_k}^2$ instead of the ℓ_2-regularization $\|\mathbf{x}-\mathbf{x}_k\|_2^2$, since 1) this new regularization can induce adaptive gradient algorithms as shown below in Eqn. (5); and 2) it increases the convexity of the problem and further considers different sharpness property of each coordinate in $\mathbf{s}_k$ to accelerate

Algorithm 1: Win-Accelerated AdamW, Adam, LAMB and SGD

Input: initialization $\mathbf{x}_0 = \mathbf{y}_0$, stepsize $\{(\eta_k^x, \eta_k^y)\}_{k=0}^{T-1}$, weight decay parameter $\{\lambda_k\}_{k=0}^{T-1}$, moment parameter $\{(\beta_1, \beta_2)\}$, moment parameter β_1' in SGD.

Output: $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ uniformly seleted from $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=0}^{T}$.

while $k < T$ **do**

$\mathbf{g}_k = \frac{1}{b}\sum_{i=1}^{b} \nabla f(\mathbf{y}_k; \zeta_i)$

$\mathbf{m}_k = (1-\beta_1)\mathbf{m}_{k-1} + \beta_1' \mathbf{g}_k$ where $\beta_1' = \beta_1$ except SGD /* $\mathbf{m}_0 = \mathbf{g}_0$ */

$\mathbf{v}_k = (1-\beta_2)\mathbf{v}_{k-1} + \beta_2 \mathbf{g}_k^2$ /* $\mathbf{v}_0 = \mathbf{g}_0^2$ */

compute parameter update $\mathbf{u}_k$:

$$\mathbf{u}_k = \begin{cases} \frac{\mathbf{m}_k}{\sqrt{\mathbf{v}_k + \nu}}, & \text{for AdamW and Adam} \\ \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\sqrt{\mathbf{v}_k+\nu} + \lambda_k \mathbf{x}_k\|_2}\left(\frac{\mathbf{m}_k}{\sqrt{\mathbf{v}_k+\nu}} + \lambda_k \mathbf{x}_k\right), & \text{for LAMB} \\ \mathbf{m}_k, & \text{for SGD} \end{cases}$$

set $\lambda_k' = \lambda_k$ for AdamW, Adam and SGD, and $\lambda_k' = 0$ for LAMB as vanilla LAMB uses weight decay in Step 5

$\mathbf{x}_{k+1} = \frac{1}{1+\lambda_k' \eta_k^x}(\mathbf{x}_k - \eta_k^x \mathbf{u}_k)$

$\mathbf{y}_{k+1} = \eta_k^y \tau_k \mathbf{x}_{k+1} + \eta_k^x \tau_k \left(\mathbf{y}_k - \eta_k^y \mathbf{u}_k\right)$ with $\tau_k = \frac{1}{\eta_k^x + \eta_k^y + \lambda_k' \eta_k^x \eta_k^y}$

end while

the convergence speed. To make the problem solvable iteratively, we approximate the vanilla loss $F(\mathbf{x})$ in the PPM-inspired regularized loss $F(\mathbf{x}) + \frac{1}{2\eta_k^x}\|\mathbf{x} - \mathbf{x}_k\|_{\mathbf{s}_k}^2$ by its first-order Taylor expansion at the point $\mathbf{x}_k$, and update the parameter $\mathbf{x}_{k+1}$ as follows:

$$\mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}} F(\mathbf{x}_k) + \langle \mathbf{m}_k, \mathbf{x} - \mathbf{x}_k \rangle + \frac{1}{2\eta_k^x}\|\mathbf{x} - \mathbf{x}_k\|_{\mathbf{s}_k}^2 + \frac{\lambda_k}{2}\|\mathbf{x}\|_{\mathbf{s}_k}^2 = \frac{1}{1+\lambda_k \eta_k^x}(\mathbf{x}_k - \eta_k^x \mathbf{u}_k), \quad (5)$$

where $\mathbf{m}_k$ is used to approximate the full gradient $\nabla F(\mathbf{x}_k)$. In Eqn. (5), we add a small regularization $\frac{\lambda_k}{2}\|\mathbf{x}\|_{\mathbf{s}_k}^2$, since 1) it can largely improve the generalization performance in practice (Loshchilov and Hutter, 2018; Touvron et al., 2021); 2) it allows us to derive Adam ($\lambda_k = 0$) and AdamW ($\lambda_k > 0$). Here λ_k can be fixed as a constant or evolves with iteration number k. In practice, an evolving λ_k often enjoys better performance than a fixed one (Caron et al., 2021; Zhou et al., 2022). When $\lambda_k = 0$, the updating scheme (5) becomes the exact Adam. If $\lambda_k > 0$, the updating (5) can approximate the updating rule $\mathbf{x}_{k+1} = (1 - \lambda_k \eta_k^x)\mathbf{x}_k - \eta_k^x \mathbf{u}_k$ of AdamW. This is because as $\lambda_k \eta_k^x$ is small in practice, we can approximate $(1 + \lambda_k \eta_k^x)^{-1} = 1 - \lambda_k \eta_k^x + \mathcal{O}(\lambda_k^2 (\eta_k^x)^2)$ and thus

$$\frac{1}{1+\lambda_k \eta_k^x}(\mathbf{x}_k - \eta_k^x \mathbf{u}_k) = \left[1 - \lambda_k \eta_k^x + \mathcal{O}(\lambda_k^2 (\eta_k^x)^2)\right]\mathbf{x}_k - \left[\eta_k^x - \mathcal{O}(\lambda_k (\eta_k^x)^2) + \mathcal{O}(\lambda_k^2 (\eta_k^x)^3)\right]\mathbf{u}_k$$

which becomes AdamW by ignoring the very small terms $\mathcal{O}((\eta_k^x)^2)$ or $\mathcal{O}((\eta_k^x)^3)$. This is also one reason that we adopt the regularizer $\|\mathbf{x} - \mathbf{x}_k\|_{\mathbf{s}_k}^2$ in (5) instead of the ℓ_2-regularization in PPM, since we can flexibly derive Adam and AdamW.

Similarly, we minimize a regularized loss $F(\mathbf{y})+\frac{1}{2\eta_k^x}\|\mathbf{y}-\mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2$ again, and further approximate $F(\mathbf{y})$ by its second-order approximation $F(\mathbf{y}_k)+\langle\mathbf{m}_k,\mathbf{y}-\mathbf{y}_k\rangle+\frac{1}{2\eta_k^y}\|\mathbf{y}-\mathbf{y}_k\|_{\mathbf{s}_k}^2$:

$$\begin{aligned}\mathbf{y}_{k+1}=&\operatorname{argmin}_{\mathbf{y}} F(\mathbf{y}_k)+\langle\mathbf{m}_k,\mathbf{y}-\mathbf{y}_k\rangle+\frac{1}{2\eta_k^y}\|\mathbf{y}-\mathbf{y}_k\|_{\mathbf{s}_k}^2+\frac{1}{2\eta_k^x}\|\mathbf{y}-\mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2+\frac{\lambda_k}{2}\|\mathbf{y}\|_{\mathbf{s}_k}^2\\ =&\eta_k^y\tau_k\mathbf{x}_{k+1}+\eta_k^x\tau_k\big(\mathbf{y}_k-\eta_k^y\mathbf{u}_k\big),\end{aligned}\tag{6}$$

where $\tau_k=\frac{1}{\eta_k^x+\eta_k^y+\lambda_k\eta_k^x\eta_k^y}$, $\mathbf{m}_k$ is used to approximate $\nabla F(\mathbf{y}_k)$ as guaranteed by Theorem 1 in Sec. 3.3, η_k^y approximates the inverse of the local smoothness parameter of $F(\mathbf{y})$ around $\mathbf{y}_k$. Here we use a regularizer $\|\mathbf{y}-\mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2$ with the latest update $\mathbf{x}_{k+1}$ instead of $\mathbf{x}_k$ as an anchor point, since the latest update $\mathbf{x}_{k+1}$ could often provide better regularization for the concurrent optimization.

Now we have used PPM to rigorously derive our Win-accelerated AdamW and Adam in Eqns. (3), (5) and (6). For more clarity, we summarize the algorithmic steps of Win-accelerated AdamW and Adam in Algorithm 1, omitting the bias-correction term for simplicity. When $\lambda_k=0$, it is Win-accelerated Adam; if $\lambda_k>0$, it corresponds to Win-accelerated AdamW. Generally, AdamW can greatly improve the generalization performance of Adam by simply adding a weight decay (*i.e.* the regularizer $\frac{\lambda_k}{2}\|\cdot\|_{\mathbf{s}_k}^2$) into Adam as observed in many works, *e.g.,* (Loshchilov and Hutter, 2018; Touvron et al., 2021). Our Win-acceleration is simple and efficient, since our accelerated AdamW/Adam merely adds one extra simple algorithmic step, *i.e.* the eighth step in Algorithm 1, on vanilla AdamW/Adam, and brings negligible extra computational overhead into the vanilla optimizer, *e.g.,* about 2% extra average training time per iteration on AdamW evaluated on ResNet as shown in Sec. 5.4. Moreover, regarding the extra hyperparameter, namely, the aggressive step η_k^y, in Algorithm 1 over AdamW/Adam, we always set it to be 2 times larger than the conservative step η_k^x for all iterations, *i.e.* $\eta_k^y=2\eta_k^x$, which works well in all our experiments.

Now we discuss the relations between Nesterov-type acceleration (2) and our Win acceleration (6). For comparison, we introduce a virtual sequence $\mathbf{y}'_{k+1}=\mathbf{y}_k-\eta_k^y\mathbf{u}_k$ in Win, and rewrite (6) as

$$\mathbf{x}_{k+1}=(1+\lambda_k\eta_k^x)^{-1}\left(\mathbf{x}_k-\eta_k^x\mathbf{u}_k\right),\ \ \mathbf{y}'_{k+1}=\mathbf{y}_k-\eta_k^y\mathbf{u}_k,\ \ \mathbf{y}_{k+1}=\eta_k^y\tau_k\mathbf{x}_{k+1}+\eta_k^x\tau_k\mathbf{y}'_{k+1},\tag{7}$$

where $\mathbf{u}_k$ is defined in (4). By comparing Nesterov-type acceleration (2) with our Win acceleration (7), one can observe some similarities and also differences as well. For similarity, both methods use a conservative step η_k^x and an aggressive step η_k^y to update $\mathbf{x}_{k+1}$ and $\mathbf{y}'_{k+1}$ respectively, and then linearly combine $\mathbf{x}_{k+1}$ and $\mathbf{y}'_{k+1}$ to obtain $\mathbf{y}_{k+1}$. Regarding the differences, the first distinction is that Win incorporates a weight-decay-alike factor $\frac{1}{1+\lambda_k\eta_k^x}$ in (7) which slightly decays the variable $\mathbf{x}_k$ like AdamW and also the update $\mathbf{u}_k$, while Nesterov acceleration does not. Notably, weight decay has demonstrated significant benefits for generalization in practical applications, as observed in various studies, *e.g.,* (Loshchilov and Hutter, 2018; Touvron et al., 2021; Liu et al., 2021). Another difference is that for almost all acceleration techniques, including Nesterov-type acceleration (2), the sum of their linear combination factors (*e.g.,* ρ_k^y and $1-\rho_k^y$ in (2)) is always one. In contrast, in Eqn. (7), Win uses $\eta_k^y\tau_k+\eta_k^x\tau_k=1-\frac{\lambda_k\eta_k^x\eta_k^y}{\eta_k^x+\eta_k^y+\lambda_k\eta_k^x\eta_k^y}<1$ when $\lambda_k>0$, which introduces an additional weight decay effect. Since these two differences arise from the presence of weight decay, we refer to our acceleration technique as "weight-decay-integrated Nesterov acceleration" (Win for brevity). Besides, the empirical results on ResNet in Sec. 5.4 also show that Win acceleration often brings more performance improvement than Nesterov-type acceleration (2).

3.2 Extension to LAMB and SGD

Here we generalize Win acceleration to LAMB (You et al., 2019) and SGD (Robbins and Monro, 1951). For LAMB, it scales the update $\mathbf{u}_k$ of AdamW in Eqn. (4) so that $\mathbf{u}_k$ is of the same magnitude as the network weight $\mathbf{x}_k$. That is, it changes the update rule $\mathbf{x}_{k+1} = (1-\lambda_k\eta_k^x)\mathbf{x}_k - \eta_k^x \mathbf{m}_k/\mathbf{s}_k$ in AdamW to $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k^x \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{r}_k+\lambda_k\mathbf{x}_k\|_2}(\mathbf{r}_k+\lambda_k\mathbf{x}_k)$ where $\mathbf{r}_k = \mathbf{m}_k/\mathbf{s}_k$. This modification is to avoid too large or small update to improved the optimization efficiency. To extend Win acceleration to LAMB, we inherit this scaling spirit, and scale the update $\mathbf{u}_k$ in (4) to the following:

$$\mathbf{u}_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{r}_k+\lambda_k\mathbf{x}_k\|_2}(\mathbf{r}_k+\lambda_k\mathbf{x}_k). \tag{8}$$

Next, we can respectively follow Eqn. (5) and (6) to update the two sequences $\mathbf{x}_k$ and $\mathbf{y}_k$. See the detailed steps of Win-accelerated LAMB in Algorithm 1. Since vanilla LAMB uses weight decay in scaling operation already (namely computing $\mathbf{u}_k$), it does not use extra weight decay in updating $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k^x\mathbf{u}_k$. Following this spirit, we also do not use extra weight decay in updating $\mathbf{x}_{k+1}$ and $\mathbf{y}_{k+1}$ as shown in Steps 7 and 8 in Algorithm 1.

For SGD, applying Win acceleration to it is quite straightforward. Specifically, the only algorithmic difference between SGD and AdamW on the ℓ_2-regularized problem is that SGD lacks the second-order moment $\mathbf{v}_k$ in AdamW. So we can leverage the acceleration framework of AdamW described in Sec. 3.1 to accelerate SGD. By setting $\mathbf{s}_k = \mathbf{1} \in \mathbb{R}^d$ in Eqn. (4), (5) and (6), we can obtain Win-accelerated SGD:

$$\mathbf{m}_k = \beta_1\mathbf{m}_{k-1} + \beta_1'\mathbf{g}_k,\ \mathbf{x}_{k+1} = \frac{1}{1+\lambda_k\eta_k^x}(\mathbf{x}_k - \eta_k^x\mathbf{m}_k),\ \mathbf{y}_{k+1} = \eta_k^y\tau_k\mathbf{x}_{k+1} + \eta_k^x\tau_k\big(\mathbf{y}_k - \eta_k^y\mathbf{m}_k\big), \tag{9}$$

where $\beta_1' \in [0,1]$ is dampening parameter. Here we slightly modify the moment $\mathbf{m}_k$ to accord with the one used in Nesterov-accelerated SGD (*e.g.,* SGD-N in Pytorch) whose updating steps are

$$\mathbf{m}_k = \beta_1\mathbf{m}_{k-1} + \beta_1'(\mathbf{g}_k + \lambda_k\mathbf{x}_k), \quad \mathbf{x}_{k+1} = (1-\lambda_k\eta_k^x)\mathbf{x}_k - \eta_k^x(\mathbf{g}_k + \beta_2\mathbf{m}_k). \tag{10}$$

By comparing Win-accelerated SGD and SGD-N in (10), one can find their big differences primarily stemming from their distinct acceleration strategies and approaches to handling weight decay. Win-accelerated SGD is derived from PPM and a recently proposed acceleration (2), while SGD-N modifies another previous Nesterov-type acceleration (Nesterov, 2003) (namely, $\mathbf{m}_k = \beta_1\mathbf{m}_{k-1} - \frac{\eta_k^x}{b}\sum_{i=1}^b \nabla f(\mathbf{x}_k + \eta_k^x\mathbf{m}_{k-1};\zeta_i)$ and $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{m}_k$) to better train networks. See more mechanisms of previous Nesterov acceleration and (10) in (Sutskever et al., 2013; Bengio et al., 2013). Sec. 5.4 also empirically compares Win-accelerated SGD and Nesterov-accelerated SGD on image classification tasks, and shows the superiority of Win-accelerated SGD.

3.3 Convergence Analysis

Here we investigate the convergence performance of Win-accelerated algorithms by taking accelerated AdamW, Adam, LAMB and SGD as examples, as these algorithms are commonly used in the deep learning field. Moreover, since we aim to accelerate deep network training which is a highly nonconvex problem, we focus on analyzing nonconvex problems to align with the practical scenarios. For analysis, we follow previous optimization works, *e.g.,* (Kingma and Ba, 2015; Reddi et al., 2018; Duchi et al., 2011; Zhou et al., 2021; Xie et al., 2022), to introduce necessary assumptions.

Assumption 1 (L-smoothness) *We say a function $f(\mathbf{x},\cdot)$ to be L-smooth w.r.t. $\mathbf{x}$, if for $\forall \mathbf{x}_1,\mathbf{x}_2$ and $\forall \boldsymbol{\zeta}\sim\mathcal{D}$, we have $\|\nabla f(\mathbf{x}_1,\boldsymbol{\zeta})-\nabla f(\mathbf{x}_2,\boldsymbol{\zeta})\|_2\le L\|\mathbf{x}_1-\mathbf{x}_2\|_2$ with a universal constant L.*

Assumption 2 (Unbiased and bounded gradient estimation) *Assume the gradient estimation $\mathbf{g}_k=\frac{1}{b}\sum_{i=1}^b\nabla f(\mathbf{x}_k;\zeta_i)$ is unbiased,* i.e.*, $\mathbb{E}[\mathbf{g}_k]=\nabla F(\mathbf{x}_k)$, and its magnitude and variance are bounded,* i.e.*, $\|\mathbf{g}_k\|_\infty\le c_\infty$ and $\mathbb{E}[\|\nabla F(\mathbf{x}_k)-\mathbf{g}_k\|_2^2]\le\sigma^2$ $(\forall k)$ with two universal constants c_∞ and σ.*

Next, we first define a dynamic function $F_k(\mathbf{x})$ at the k-th iteration which is the real loss minimized by our algorithms. It combines the vanilla loss $F(\mathbf{x})$ in (1) and a dynamic regularization $\frac{\lambda_k}{2}\|\mathbf{x}\|_{\mathbf{s}_k}^2$:

$$F_k(\mathbf{x})=F(\mathbf{x})+\frac{\lambda_k}{2}\|\mathbf{x}\|_{\mathbf{s}_k}^2=\mathbb{E}_\zeta[f(\mathbf{x};\zeta)]+\frac{\lambda_k}{2}\|\mathbf{x}\|_{\mathbf{s}_k}^2, \tag{11}$$

where $\mathbf{s}_k$ is given in (4). To obtain (11), following PPM spirit and Eqn. (5), one can approximate $F(\mathbf{x})$ by its first-order Taylor expansion, and obtain Eqn. (5) to update $\mathbf{x}_{k+1}=\frac{1}{1+\lambda_k\eta_k^x}(\mathbf{x}_k-\eta_k^x\mathbf{m}_k/\mathbf{s}_k)$. Since $\lambda_k\eta_k^x$ is very small, one can follow the discussion below Eqn. (5) and approximate $\mathbf{x}_{k+1}$ as $\mathbf{x}_{k+1}=(1-\lambda_k\eta_k^x)\mathbf{x}_k-\eta_k^x\mathbf{m}_k/\mathbf{s}_k$ which becomes the update rule of AdamW. This is the reason why our analysis on Win-accelerated AdamW involves a dynamic loss $F_k(\mathbf{x})$ in (11). Note, for Win-accelerated Adam ($\lambda_k=0$), $F_k(\mathbf{x})$ degenerates to the vanilla objective loss $F(\mathbf{x})$.

Convergence Analysis of Win-accelerated AdamW and Adam. With these assumptions, we analyze the convergence behaviors of our accelerated algorithms on general nonconvex problems, and summarize our main results in Theorem 1 with its proof given in Appendix D.1. For brevity, we use the notation $\frac{1}{T}\sum_{i=0}^T\{a_i,b_i\}\le\{a,b\}$ to denote $\frac{1}{T}\sum_{i=0}^T a_i\le a$ and $\frac{1}{T}\sum_{i=0}^T b_i\le b$.

Theorem 1 *Suppose that Assumptions 1 and 2 hold, $\mathbf{x}_\star\in\operatorname{argmin}_{\mathbf{x}}F(\mathbf{x})$, and $\eta\le\mathcal{O}\big(\frac{\nu^{1.25}b\epsilon^2}{c^{1.5}\gamma^{1.5}\sigma^2L}\big)$. Let $\eta_k^y=\gamma\eta_k^x$ $(\gamma>1)$, $\eta_k^x=\eta$, $\beta_1\le\mathcal{O}\big(\frac{\nu^{0.5}b\epsilon^2}{c\sigma^2}\big)$, $\beta_2\in(0,1)$, $c=(c_\infty^2+\nu)^{0.5}$, $\lambda_k=\lambda(1-\frac{\beta_2c_\infty^2}{\nu})^k$ $(k>0)$ and $\lambda_0=0$ with a constant $\lambda>0$. Then after $T=\mathcal{O}\big(\frac{c_\infty^{2.5}\gamma^{1.5}\sigma^2L\Delta}{\nu^{1.25}b\epsilon^4}\big)$ iterations with minibatch size b and $\Delta=F(\mathbf{x}_0)-F(\mathbf{x}_\star)$, the sequence $\{(\mathbf{x}_k,\mathbf{y}_k)\}_{k=0}^T$ generated by Win-accelerated AdamW and Adam in Algorithm 1 satisfies the following four properties.*

a) *The gradient $\nabla F_k(\mathbf{x}_k)$ of the sequence $\{\mathbf{x}_k\}_{k=0}^T$ can be upper bounded by*

$$\frac{1}{T}\sum\nolimits_{k=0}^{T-1}\mathbb{E}\Big[\|\nabla F_k(\mathbf{x}_k)\|_2^2+\frac{1}{4}\|\mathbf{m}_k+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\|_2^2\Big]\le\epsilon^2.$$

b) *The gradient moment $\mathbf{m}_k$ can well estimate the full gradient $\nabla F(\mathbf{x}_k)$ and $\nabla F(\mathbf{y}_k)$:*

$$\frac{1}{T}\sum\nolimits_{k=0}^{T-1}\max\Big\{\mathbb{E}\|\mathbf{m}_k-\nabla F(\mathbf{x}_k)\|_2^2,\mathbb{E}\|\mathbf{m}_k-\nabla F(\mathbf{y}_k)\|_2^2\Big\}\le16\epsilon^2+\frac{8\eta^2\gamma^2L^2}{\nu^2}\epsilon^2.$$

c) *The sequence $\{(\mathbf{x}_k,\mathbf{y}_k)\}$ satisfies*

$$\frac{1}{T}\sum\nolimits_{k=0}^{T-1}\Big\{\mathbb{E}\|\mathbf{x}_k-\mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2,\mathbb{E}\|\mathbf{y}_k-\mathbf{x}_k\|_2^2\Big\}\le\Big\{4\eta^2\epsilon^2,\frac{4\eta^2\gamma^2}{\nu^2}\epsilon^2\Big\}.$$

d) *The stochastic gradient complexity to achieve the above three properties is $\mathcal{O}\big(\frac{c_\infty^{2.5}\Delta\sigma^2L}{\nu^{1.25}\epsilon^4}\big)$, where stochastic gradient complexity is the total evaluation number of the gradient on a single sample.*

Theorem 1 guarantees the convergence of Win-accelerated AdamW and Adam in Algorithm 1 on nonconvex problems. When $\lambda_k > 0$, Algorithm 1 corresponds to Win-accelerated AdamW, and if $\lambda_k = 0$, it becomes Win-accelerated Adam. For both cases, Theorem 1 holds. Theorem 1 a) shows that by running at most $T = \mathcal{O}\big(\frac{c_\infty^{2.5}\Delta\sigma^2 L}{\nu^{1.25}b\epsilon^4}\big)$ iterations, the average gradient $\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\big[\,\|\nabla F_k(\mathbf{x}_k)\|_2^2\,\big]$ is upper bounded by ϵ^2, guaranteeing the algorithmic convergence. Theorem 1 b) indicates that the gradient moment $\mathbf{m}_k$ can well estimate the full gradient $\nabla F(\mathbf{y}_k)$ and also $\nabla F(\mathbf{x}_k)$ because of their small distances, guaranteeing the good Taylor approximation used in Eqns. (5) and (6). Moreover, in Theorem 1 c), one can find that although Algorithm 1 uses a conservative step η_k^x and an aggressive step $\eta_k^y = \gamma\eta_k^x$ $(\forall\gamma > 1)$ to update, the two sequences $\mathbf{x}_{k+1}$ and $\mathbf{y}_{k+1}$ can converge to each other, which could be the key for the good convergence behavior of both Win-accelerated AdamW and Adam.

Now we discuss the stochastic gradient complexity of Win-accelerated Adam and AdamW. Theorem 1 d) shows that to find an ϵ-approximate first-order stationary point, both Win-accelerated Adam and AdamW have the complexity of $\mathcal{O}\big(\frac{c_\infty^{2.5}\sigma^2 L}{\nu^{1.25}\epsilon^4}\big)$ when ignoring some other constant factors like other algorithms (Zhuang et al., 2020; Guo et al., 2021; Xie et al., 2022). This complexity matches the lower bound of $\Omega(\frac{1}{\epsilon^4})$ in (Arjevani et al., 2022, 2020) (up to constant factors). Our accelerated Adam and AdamW enjoy superior complexity to Adam-type optimizers, *e.g.,* Adam, AdaGrad (Duchi et al., 2011), AdaBound (Luo et al., 2018), whose previously best known complexity under the same assumptions is $\mathcal{O}(\frac{c_\infty^2 d\sigma^2 L}{\nu^{1.25}\epsilon^4})$ in (Zhou et al., 2018; Chen et al., 2021; Guo et al., 2021). By comparison, both accelerated Adam and AdamW improve their complexity by a factor $\mathcal{O}(\frac{d}{c_\infty^{0.5}})$, where the network parameter dimension d is often much larger than $c_\infty^{0.5}$, especially for over-parameterized neural networks. Moreover, the complexity of Win-accelerated Adam and AdamW is also lower than $\mathcal{O}(\frac{c_2^6\sigma^2 L}{\nu^2\epsilon^4})$ of Adabelief (Zhuang et al., 2020) and $\mathcal{O}(\frac{c_\infty^{0.5}d^{0.5}\sigma^2 L}{\nu\epsilon^4})$ of RMSProp (Tijmen and Geoffrey, 2012; Zhou et al., 2018), especially on over-parameterized networks, since for a d-dimensional gradient, its ℓ_2-norm upper bound c_2 is often much larger than the ℓ_∞-norm c_∞ and can be $\sqrt{d}$ times larger in the worse case.

Convergence Analysis of Win-accelerated LAMB. Next we analyze another important optimizer, LAMB, which is also widely used in vision transformer training. For analysis, we first define the scaling factor $\alpha_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\sqrt{\mathbf{v}_k+\nu}+\lambda_k\mathbf{x}_k\|_2}$ and follow the analysis of LAMB to assume it to be bounded, namely, $0 < \alpha_s \le \alpha_k \le \alpha_l$, where α_s and α_l are two universal constants. Based on these assumptions, we can analyze the convergence behavior of Win-accelerated LAMB and present the main results in Theorem 2, whose proof can be found in Appendix D.2.

Theorem 2 *Suppose that Assumptions 1 and 2 hold,* $\mathbf{x}_\star \in \operatorname{argmin}_{\mathbf{x}} F(\mathbf{x})$, $0 < \alpha_s \le \alpha_k \le \alpha_l$, *and* $\eta \le \mathcal{O}\big(\frac{\nu^{1.25}b\epsilon^2}{\alpha_l c^{1.5}\gamma^{1.5}\sigma^2 L}\big)$. *Let* $\eta_k^y = \gamma\eta_k^x$ $(\gamma > 1)$, $\eta_k^x = \eta$, $\beta_1 \le \mathcal{O}\big(\frac{\alpha_s\nu^{0.5}b\epsilon^2}{\alpha_l c\sigma^2}\big)$, $\beta_2 \in (0,1)$, $c = (c_\infty^2+\nu)^{0.5}$, $\lambda_k = 0$ $(k \ge 0)$. *Then after* $T = \mathcal{O}\big(\frac{\alpha_l c_\infty^{2.5}\gamma^{1.5}\sigma^2 L\Delta}{\alpha_s\nu^{1.25}b\epsilon^4}\big)$ *iterations with minibatch size* b *and* $\Delta = F(\mathbf{x}_0) - F(\mathbf{x}_\star)$, *the sequence* $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=0}^T$ *generated by Win-accelerated LAMB in Algorithm 1 satisfies the following four properties.*

a) *The gradient* $\nabla F_k(\mathbf{x}_k)$ *of the sequence* $\{\mathbf{x}_k\}_{k=0}^T$ *can be upper bounded by*

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\Big[\,\|\nabla F_k(\mathbf{x}_k)\|_2^2 + \frac{1}{4\alpha_s\alpha_l}\|\mathbf{m}_k + \lambda_k\mathbf{x}_k \odot \mathbf{s}_k\|_2^2\,\Big] \le \epsilon^2.$$

b) *The gradient moment* $\mathbf{m}_k$ *can well estimate the full gradient* $\nabla F(\mathbf{x}_k)$ *and* $\nabla F(\mathbf{y}_k)$*:*

$$\frac{1}{T}\sum\nolimits_{k=0}^{T-1}\max\Big\{\mathbb{E}\|\mathbf{m}_k-\nabla F(\mathbf{x}_k)\|_2^2,\mathbb{E}\|\mathbf{m}_k-\nabla F(\mathbf{y}_k)\|_2^2\Big\}\leq\frac{20\alpha_l}{\alpha_s}\epsilon^2+\frac{8\alpha_s\alpha_l\eta^2\gamma^2L^2}{\nu^2}\epsilon^2.$$

c) *The sequence* $\{(\mathbf{x}_k,\mathbf{y}_k)\}$ *satisfies*

$$\frac{1}{T}\sum\nolimits_{k=0}^{T-1}\Big\{\mathbb{E}\|\mathbf{x}_k-\mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2,\mathbb{E}\|\mathbf{y}_k-\mathbf{x}_k\|_2^2\Big\}\leq\Big\{4\alpha_s\alpha_l\eta^2\epsilon^2,\frac{4\alpha_s\alpha_l\gamma^2\eta^2}{\nu^2}\epsilon^2\Big\}.$$

d) *The total stochastic gradient complexity to achieve the above three properties is* $\mathcal{O}\big(\frac{\alpha_l c_\infty^{2.5}\Delta\sigma^2L}{\alpha_s\nu^{1.25}\epsilon^4}\big)$.

From Theorem 2 a), one can observe that on the nonconvex problems, Win-accelerated LAMB optimizer can also converge, because its average gradient $\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\big[\,\|\nabla F_k(\mathbf{x}_k)\|_2^2\,\big]$ can be bounded by ϵ^2 after running at most $T=\mathcal{O}\big(\frac{\alpha_l c_\infty^{2.5}\gamma^{1.5}\sigma^2L\Delta}{\alpha_s\nu^{1.25}b\epsilon^4}\big)$ iterations. Similar to Theorem 1 b), Theorem 2 b) also reveals that the first-order moment $\mathbf{m}_k$ is a good estimation to the full gradient $\nabla F(\mathbf{x}_k)$ and $\nabla F(\mathbf{y}_k)$, validating the Taylor approximation in Eqns. (5) and (6). Moreover, Theorem 1 c) shows small distance between the two points $\mathbf{x}_k$ and $\mathbf{y}_k$ which also guarantees the convergence of Win-accelerated LAMB optimizer.

Now we compare the stochastic gradient complexity of LAMB and Win-accelerated LAMB. To compute an ϵ-approximate first-order stationary point, You et al. (2019) showed the complexity of vanilla LAMB optimizer is at the order of $\mathcal{O}\big(\frac{c_\infty d^{2.5}}{\epsilon^4}\big)$ (see their Theorem 3). In contrast, as shown in Theorem 2 d), Win-accelerated LAMB has the complexity of $\mathcal{O}\big(\frac{c_\infty^{2.5}}{\epsilon^4}\big)$ and improves LAMB by a factor $\mathcal{O}\big(\frac{d^{2.5}}{c_\infty^{1.5}}\big)$ which is often large, especially for over-parameterized networks where parameter dimension d is huge. This shows the superiority of Win-accelerated LAMB in terms of the efficiency.

Convergence Analysis of Win-accelerated SGD. Now we discuss the convergence performance of Win-accelerated SGD in Theorem 3 with its proof given in Appendix D.3.

Theorem 3 *Suppose that Assumptions 1 and 2 hold, and* $\mathbf{x}_\star\in\operatorname{argmin}_{\mathbf{x}}F(\mathbf{x})$*. Let* $\eta_k^y=\gamma\eta_k^x,\gamma>1,\eta_k^x=\eta\leq\mathcal{O}\big(\frac{b\epsilon^2}{c^{1.5}\gamma^{2.5}\sigma^2L}\big)$*,* $\beta_1\leq\mathcal{O}\big(\frac{b\epsilon^2}{c\sigma^2}\big)$*,* $\beta_1'=1-\beta_1$*,* $\lambda_k=\lambda(1-\frac{\beta_2c_\infty^2}{\nu})^k$ $(k>0)$*,* $\lambda_0=0$*. After* $T=\mathcal{O}\big(\frac{\Delta\sigma^2L}{b\epsilon^4}\big)$ *iterations with minibatch size* b *and* $\Delta=F(\mathbf{x}_0)-F(\mathbf{x}_\star)$*, the sequence* $\{(\mathbf{x}_k,\mathbf{y}_k)\}_{k=0}^T$ *generated by Win-accelerated SGD in* (9) *satisfies the four properties in Theorem 1 with* $\nu=c_\infty=c=1$ *and* $\mathbf{s}_k=\mathbf{1}\in\mathbb{R}^d$*.*

Theorem 3 also guarantees the convergence of Win-accelerated SGD. By using the hyper-parameter settings in Theorem 3, the sequence $\{(\mathbf{x}_k,\mathbf{y}_k)\}_{k=0}^T$ generated by Win-accelerated SGD satisfies the four properties in Theorem 1 with $\nu=c_\infty=c=1$ and $\mathbf{s}_k=\mathbf{1}$. It shows the complexity of $\mathcal{O}(\frac{L\sigma^2}{\epsilon^4})$ of the Win-accelerated SGD which also matches the lower bound of $\Omega(\frac{1}{\epsilon^4})$ in (Arjevani et al., 2022, 2020) (up to constant factors) under Assumptions 1 and 2.

4. Win2: A More General Win Acceleration

In Sec. 3, we have integrated PPM and Nestrov acceleration to develop Win acceleration in which it respectively uses a conservative step and an aggressive step to update parameters, and then linearly combines these two updates for acceleration. The effectiveness and simplicity of Win inspires us to consider the problem of how to extend the parameter update from two updating steps in Win to

multiple updating steps, yielding a more general Win acceleration version called "Win2". Compared with the combination of two updates in Win, the multiple updates in Win2 should achieve a more stable linear combination, and thus should better stabilize the training (see more discussion below Eqn. (15)). In this way, Win2 can use more aggressive stepsize than Win which can often help to achieve faster convergence speed as empirically shown in Sec. 5. In the following, to provide a clear and intuitive example of Win2, we first derive the formulation for three updating steps, and then extend it to q updating steps ($\forall q \geq 4$) in Sec. 4.1. We also apply Win2 into LAMB and SGD in Sec. 4.2, and finally provide convergence analysis of Win2-accelerated optimizers in Sec. 4.3.

4.1 Win2-Accelerated AdamW and Adam

Since Win2 uses a conservative step, an aggressive step and a more aggressive step for parameter update, it needs three sequences denoted by $\{(\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k)\}$ which respectively correspond to the three steps. Accordingly, we define minibatch gradient $\mathbf{g}_k$ at the point $\mathbf{z}_k$ instead of $\mathbf{y}_k$ as used in Win, the first- and second-order moments $\mathbf{m}_k$ and $\mathbf{v}_k$ as follows:

$$\mathbf{g}_k = \frac{1}{b}\sum\nolimits_{i=1}^{b} \nabla f(\mathbf{z}_k;\zeta_i), \quad \mathbf{m}_k = (1-\beta_1)\mathbf{m}_{k-1} + \beta_1 \mathbf{g}_k, \quad \mathbf{v}_k = (1-\beta_2)\mathbf{v}_{k-1} + \beta_2 \mathbf{g}_k^2. \tag{12}$$

Then the same as in Eqn. (4), we also define $\mathbf{s}_k = \sqrt{\mathbf{v}_k + \nu}$ and $\mathbf{u}_k = \mathbf{m}_k/\sqrt{\mathbf{v}_k+\nu}$ for brevity. Next, at the k-th iteration, to update the sequence $\mathbf{x}_k$, following (5) in Win, we minimize a regularized loss $F(\mathbf{x}) + \frac{1}{2\eta_k^x}\|\mathbf{x}-\mathbf{x}_k\|_{\mathbf{s}_k}^2$, and further approximate the vanilla loss $F(\mathbf{x})$ by its first-order Taylor expansion at the point $\mathbf{x}_k$ to compute the close-form solution. In this way, we can follow Eqn. (5) in Win to update $\mathbf{x}_k$ as

$$\mathbf{x}_{k+1} = \mathrm{argmin}_{\mathbf{x}} F(\mathbf{x}_k) + \langle \mathbf{m}_k, \mathbf{x}-\mathbf{x}_k\rangle + \frac{1}{2\eta_k^x}\|\mathbf{x}-\mathbf{x}_k\|_{\mathbf{s}_k}^2 + \frac{\lambda_k}{2}\|\mathbf{x}\|_{\mathbf{s}_k}^2 = \frac{1}{1+\lambda_k\eta_k^x}(\mathbf{x}_k - \eta_k^x \mathbf{u}_k), \tag{13}$$

where η_k^x is small and yields a conservative step in (13). Next, we minimize a regularized loss $F(\mathbf{y}) + \frac{1}{2\eta_k^x}\|\mathbf{y}-\mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2$ again, and further approximate $F(\mathbf{y})$ by its second-order approximation $F(\mathbf{y}_k) + \langle \mathbf{m}_k, \mathbf{y}-\mathbf{y}_k\rangle + \frac{1}{2\eta_k^y}\|\mathbf{y}-\mathbf{y}_k\|_{\mathbf{s}_k}^2$ at the point $\mathbf{y}_k$. This gives the following update of $\mathbf{y}_k$:

$$\begin{aligned}\mathbf{y}_{k+1} =& \mathrm{argmin}_{\mathbf{y}}\, F(\mathbf{y}_k) + \langle \mathbf{m}_k, \mathbf{y}-\mathbf{y}_k\rangle + \frac{1}{2\eta_k^y}\|\mathbf{y}-\mathbf{y}_k\|_{\mathbf{s}_k}^2 + \frac{1}{2\eta_k^x}\|\mathbf{y}-\mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2 + \frac{\lambda_k}{2}\|\mathbf{y}\|_{\mathbf{s}_k}^2 \\ =& \xi_k^x \delta_k^y \mathbf{x}_{k+1} + \xi_k^y \delta_k^y \big(\mathbf{y}_k - \eta_k^y \mathbf{u}_k\big),\end{aligned} \tag{14}$$

where $\xi_k^x = \frac{1}{\eta_k^x}$, $\xi_k^y = \frac{1}{\eta_k^y}$, and $\delta_k^y = \frac{1}{1/\eta_k^x + 1/\eta_k^y + \lambda_k}$. For the stepsize η_k^y, it is larger than η_k^x and corresponds to an aggressive step for boosting the convergence speed.

Finally, we use a similar technique in updating $\mathbf{y}_k$ to update $\mathbf{z}_k$, but add two addiontal local regularization terms $\frac{1}{2\eta_k^x}\|\mathbf{z}-\mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2$ and $\frac{1}{2\eta_k^y}\|\mathbf{z}-\mathbf{y}_{k+1}\|_{\mathbf{s}_k}^2$. This is because 1) these two local regularization terms can enhance the convexity of the problem like PPM method; 2) they can prevent $\mathbf{z}_{k+1}$ from being too far from $\mathbf{x}_{k+1}$ and $\mathbf{y}_{k+1}$, since η_k^z is much larger than η_k^x, *e.g.,* $8\eta_k^x$ in all our experiments, and could results in a large but undesired update. Accordingly, we can update $\mathbf{z}_k$ by

Algorithm 2: Win2-Accelerated AdamW, Adam, LAMB and SGD

Input: initialization $\mathbf{x}_0 = \mathbf{y}_0 = \mathbf{z}_0$, stepsize $\{(\eta_k^x, \eta_k^y, \eta_k^z)\}_{k=0}^{T-1}$, weight decat $\{\lambda_k\}_{k=0}^{T-1}$, weight decay parameter $\{\lambda_k\}_{k=0}^{T-1}$, moment parameter $\{(\beta_1, \beta_2)\}$, moment parameter β_1' in SGD.

Output: $(\bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\mathbf{z}})$ uniformly seleted from $\{(\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k)\}_{k=0}^{T}$.

1 **while** $k < T$ **do**

2 $\quad \mathbf{g}_k = \frac{1}{b}\sum_{i=1}^{b} \nabla f(\mathbf{z}_k; \zeta_i)$

3 $\quad \mathbf{m}_k = (1-\beta_1)\mathbf{m}_{k-1} + \beta_1'\mathbf{g}_k$ where $\beta_1' = \beta_1$ except SGD $\qquad$ /* $\mathbf{m}_0 = \mathbf{g}_0$ */

4 $\quad \mathbf{v}_k = (1-\beta_2)\mathbf{v}_{k-1} + \beta_2\mathbf{g}_k^2$ $\qquad$ /* $\mathbf{v}_0 = \mathbf{g}_0^2$ */

5 $\quad$ compute parameter update $\mathbf{u}_k$:

$$\mathbf{u}_k = \begin{cases} \frac{\mathbf{m}_k}{\sqrt{\mathbf{v}_k+\nu}}, & \text{for AdamW and Adam} \\ \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\sqrt{\mathbf{v}_k+\nu}+\lambda_k\mathbf{x}_k\|_2}\big(\frac{\mathbf{m}_k}{\sqrt{\mathbf{v}_k+\nu}} + \lambda_k\mathbf{x}_k\big), & \text{for LAMB} \\ \mathbf{m}_k, & \text{for SGD} \end{cases}$$

6 $\quad$ set $\lambda_k' = \lambda_k$ for AdamW, Adam and SGD, and $\lambda_k' = 0$ for LAMB as vanilla LAMB uses weight decay in Step 5

7 $\quad \mathbf{x}_{k+1} = \frac{1}{1+\lambda_k\eta_k^x}(\mathbf{x}_k - \eta_k^x\mathbf{u}_k)$

8 $\quad \mathbf{y}_{k+1} = \xi_k^x\delta_k^y\mathbf{x}_{k+1} + \xi_k^y\delta_k^y\big(\mathbf{y}_k - \eta_k^y\mathbf{u}_k\big)$ with $\xi_k^x = \frac{1}{\eta_k^x}$, $\xi_k^y = \frac{1}{\eta_k^y}$, $\delta_k^y = \frac{1}{\xi_k^x+\xi_k^y+\lambda_k}$

9 $\quad \mathbf{z}_{k+1} = \xi_k^x\delta_k^z\mathbf{x}_{k+1} + \xi_k^y\delta_k^z\mathbf{y}_{k+1} + \xi_k^z\delta_k^z\big(\mathbf{z}_k - \eta_k^z\mathbf{u}_k\big)$ with $\xi_k^z = \frac{1}{\eta_k^z}$, $\delta_k^z = \frac{1}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}$.

10 **end while**

solving the following subproblem:

$$\mathbf{z}_{k+1} = \operatorname{argmin}_{\mathbf{z}}\Big\{F(\mathbf{z}_k) + \langle\mathbf{m}_k, \mathbf{z}-\mathbf{z}_k\rangle + \frac{1}{2\eta_k^z}\|\mathbf{z}-\mathbf{z}_k\|_{\mathbf{s}_k}^2 + \frac{1}{2\eta_k^x}\|\mathbf{z}-\mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2 + \frac{1}{2\eta_k^y}\|\mathbf{z}-\mathbf{y}_{k+1}\|_{\mathbf{s}_k}^2 + \frac{\lambda_k}{2}\|\mathbf{z}\|_{\mathbf{s}_k}^2\Big\} = \xi_k^x\delta_k^z\mathbf{x}_{k+1} + \xi_k^y\delta_k^z\mathbf{y}_{k+1} + \xi_k^z\delta_k^z\big(\mathbf{z}_k - \eta_k^z\mathbf{u}_k\big), \tag{15}$$

where $\xi_k^z = \frac{1}{\eta_k^z}$ and $\delta_k^z = \frac{1}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}$, $\mathbf{m}_k$ is used to approximate $\nabla F(\mathbf{z}_k)$, and η_k^z approximates the inverse of the local smoothness parameter of $F(\mathbf{z})$ around $\mathbf{z}_k$.

In this way, by combining Eqns. (12), (13), (14) and (15), we can obtain our Win2-accelerated AdamW and Adam. Algorithm 2 also summarizes their algorithmic steps in which the bias-correction term is omitted for simplicity. Similar to Algorithm 1, if $\lambda_k = 0$, Algorithm 2 represents Win2-accelerated Adam, while if $\lambda_k > 0$, it corresponds to Win2-accelerated AdamW. Compared with vanilla optimziers, *e.g.,* AdamW and Adam, our Win2-acceleration introduces only two extra steps in them, namely *i.e.* the eighth and ninth steps in Algorithm 2, and thus is very simple. Moreover, for the two extra hyper-parameters, the aggressive stepsize η_k^y and η_k^z, in Algorithm 2 over AdamW/Adam, we always set $\eta_k^y = 2\eta_k^x$ and $\eta_k^z = 8\eta_k^x$ which achieve good performance in all our experiments and thus do not introduce extra hyper-parameter tuning cost. Compared with Win whose aggressive stepsize is $\eta_k^y = 2\eta_k^x$, Win2 uses a more aggressive stepsize $\eta_k^z = 8\eta_k^x$ to pursue faster convergence speed while ensuring stable training. This is because as shown in Eqn. (15), Win2 combines three updates $\mathbf{z}_{k+1} = \xi_k^x\delta_k^z\mathbf{x}_{k+1} + \xi_k^y\delta_k^z\mathbf{y}_{k+1} + \xi_k^z\delta_k^z\big(\mathbf{z}_k - \eta_k^z\mathbf{u}_k\big)$ in which $\mathbf{x}_{k+1}$ and $\mathbf{y}_{k+1}$ computed by using less

aggressive stepsizes can effectively stabilize the training, even though $\mathbf{z}_k$ uses a much more aggressive stepsize. In contrast, Win updates the aggressive step as $\mathbf{y}_{k+1} = \eta_k^y \tau_k \mathbf{x}_{k+1} + \eta_k^x \tau_k \big(\mathbf{y}_k - \eta_k^y \mathbf{u}_k\big)$ in Eqn. (6), and only uses a single $\mathbf{x}_{k+1}$ to stabilize the training, which limits the aggressive stepsize η_k^y.

Following the above three updating steps ($\mathbf{x}_k, \mathbf{y}_k$ and $\mathbf{z}_k$), one can extend it to q-updating steps denoted by $\{\mathbf{x}_k^{(i)}\}_{i=1}^q$ ($q \geq 4$). At the $(k+1)$-th iteration, after computing $\{\mathbf{x}_{k+1}^{(i)}\}_{i=1}^s$, to update $\mathbf{x}_{k+1}^{(s+1)}$, we minimize a regularized loss $F(\mathbf{x}) + \frac{\lambda_k}{2}\|\mathbf{x}\|_{\mathbf{s}_k}^2 + \sum_{i=1}^s \frac{1}{2\eta_k^{(i)}}\|\mathbf{x} - \mathbf{x}_k^{(i)}\|_{\mathbf{s}_k}^2$, and approximate $F(\mathbf{x})$ by its second-order estimation $F(\mathbf{x}_k^{(s+1)}) + \langle \mathbf{m}_k, \mathbf{x} - \mathbf{x}_k^{(s+1)}\rangle + \frac{1}{2\eta_k^{(s+1)}}\|\mathbf{x} - \mathbf{x}_k^{(s+1)}\|_{\mathbf{s}_k}^2$:

$$\begin{aligned}\mathbf{x}_{k+1}^{(s+1)} =& \operatorname{argmin}_{\mathbf{x}} \left\{ F(\mathbf{x}_k^{(s+1)}) + \langle \mathbf{m}_k, \mathbf{x} - \mathbf{x}_k^{(s+1)}\rangle + \sum_{i=1}^{s+1} \frac{1}{2\eta_k^{(i)}}\|\mathbf{x} - \mathbf{x}_k^{(i)}\|_{\mathbf{s}_k}^2 + \frac{\lambda_k}{2}\|\mathbf{x}\|_{\mathbf{s}_k}^2 \right\} \\ =& \xi_k^{(s+1)} \delta_k^{(s+1)} \big(\mathbf{x}_k^{(s+1)} - \eta_k^{(s+1)} \mathbf{u}_k\big) + \sum_{i=1}^{s} \xi_k^{(i)} \delta_k^{(s+1)} \mathbf{x}_k^{(i)},\end{aligned}$$

where $\xi_k^{(i)} = \frac{1}{\eta_k^{(i)}}$ and $\delta_k^{(s+1)} = \frac{1}{\lambda_k + \sum_{i=1}^{s+1} \xi_k^{(i)}}$. For $\mathbf{x}_{k+1}^{(1)}$, we update it as $\mathbf{x}_{k+1}^{(1)} = \xi_k^{(1)} \delta_k^{(1)} \big(\mathbf{x}_k^{(1)} - \eta_k^{(1)} \mathbf{u}_k\big)$, which accords with Eqn. (13) when $q = 3$. In practice, as shown in Sec. 5.4, large q (e.g., $q \geq 4$) actually does not bring significant improvement but extra memory cost. This is because for Win2 with three updating steps, it already uses very aggressive stepsizes in practice, *e.g.,* $\eta_k^{(2)} = 2\eta_k^{(1)}$, $\eta_k^{(3)} = 8\eta_k^{(1)}$ in all our experiments. So for more updating steps, *e.g.,* $q \geq 4$, it is hard to use more aggressive stepsize $\eta_k^{(4)} = a\eta_k^{(1)}$ ($a > 8$) while enjoying very stable training. So in the following, we focus more on Win2 with three updating steps unless otherwise specified.

4.2 Extension to LAMB and SGD

Based on Win-accelerated LAMB and SGD, we can extend Win2 acceleration to these two algorithms easily. For LAMB, we can follow Eqn. (8) in Win to scale the update $\mathbf{u}_k$ in (4) so that $\mathbf{u}_k$ is at the same magnitude as the network weight $\mathbf{x}_k$. Then we can update $\mathbf{x}_k$, $\mathbf{y}_k$ and $\mathbf{z}_k$ by respectively using Eqn. (13), (14) and (15). Accordingly, we can obtain Win2-accelerated LAMB. For more clarity, Algorithm 2 summarizes the detailed algorithmic steps of Win2-accelerated LAMB.

For SGD, its only algorithmic difference with AdamW on the ℓ_2-regularized problem is that SGD has no second-order moment $\mathbf{v}_k$, while AdamW has. In this way, one can set $\mathbf{s}_k = \mathbf{1} \in \mathbb{R}^d$ in Eqn. (4), (13), (14) and (15) which directly yields the Win2-accelerated SGD. To obtain exact SGD, one should update the first-order moment $\mathbf{m}_k$ in SGD as $\mathbf{m}_k = \beta_1 \mathbf{m}_{k-1} + \beta_1' \mathbf{g}_k$, where $\beta_1' \in [0, 1]$ is a dampening parameter. By these modifications, one can obtain the desired Win2-accelerated SGD as shown in Algorithm 2.

4.3 Convergence Analysis

In this subsection, we provide theoretical convergence analysis for Win2-accelerated AdamW, Adam, LAMB and SGD. Here we also analyze the highly nonconvex network training problems to accord with the practical setting. For the convergence analysis, we also need to borrow the dynamic function $F_k(\mathbf{x}) = F(\mathbf{x}) + \frac{\lambda_k}{2}\|\mathbf{x}\|_{\mathbf{s}_k}^2 = \mathbb{E}_\zeta[f(\mathbf{x};\zeta)] + \frac{\lambda_k}{2}\|\mathbf{x}\|_{\mathbf{s}_k}^2$ at the k-th iteration which is defined in (11). It combines the vanilla loss $F(\mathbf{x})$ in (1) and a dynamic regularization $\frac{\lambda_k}{2}\|\mathbf{x}\|_{\mathbf{s}_k}^2$ which induces the decoupled weight decay and often improves the generalization performance in practice. For

Win2-accelerated Adam ($\lambda_k=0$), $F_k(\mathbf{x})$ degenerates to the vanilla loss $F(\mathbf{x})$. For the reasons why our analysis on Win2-accelerated AdamW involves a dynamic loss $F_k(\mathbf{x})$, please refer to the detailed discussion in Sec. 3.3. In the following, we will analyze AdamW, Adam, LAMB and SGD in turn.

Convergence Analysis of Win2-accelerated Adam and AdamW. Based on Assumptions 1 and 2 in Sec. 3.3, we are ready to provide the theoretical results of Win2-accelerated Adam and AdamW in Theorem 4 whose proof can be found in Appendix E.1.

Theorem 4 *Suppose that Assumptions 1 and 2 hold,* $\mathbf{x}_\star \in \operatorname{argmin}_{\mathbf{x}} F(\mathbf{x})$*, and* $\eta \leq \mathcal{O}\big(\frac{\nu^{1.25} b\epsilon^2}{c^{1.5}(\gamma_y^{1.5}+\gamma_z^{1.5})\sigma^2 L}\big)$*. Let* $\eta_k^x = \eta, \eta_k^y = \gamma_y\eta, \eta_k^z = \gamma_z\eta, \gamma_z > \gamma_y > 1$*,* $\beta_1 \leq \mathcal{O}\big(\frac{\nu^{0.5} b\epsilon^2}{c\sigma^2}\big)$*,* $\beta_2 \in (0,1)$*,* $c = (c_\infty^2+\nu)^{0.5}$*,* $\lambda_k = \lambda(1-\frac{\beta_2 c_\infty^2}{\nu})^k$ $(k>0)$ *and* $\lambda_0 = 0$ *with a constant* $\lambda > 0$*. Then after* $T = \mathcal{O}\big(\frac{c_\infty^{2.5}\sigma^2 L\Delta}{\nu^{1.25} b\epsilon^4}\big)$ *iterations with minibatch size* b *and* $\Delta = F(\mathbf{x}_0) - F(\mathbf{x}_\star)$*, the sequence* $\{(\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k)\}_{k=0}^T$ *generated by Win2-accelerated AdamW and Adam in Algorithm 2 satisfies the following four properties.*
a) *The gradient* $\nabla F_k(\mathbf{x}_k)$ *of the sequence* $\{\mathbf{x}_k\}_{k=0}^T$ *can be upper bounded by*

$$\frac{1}{T}\sum\nolimits_{k=0}^{T-1}\mathbb{E}\Big[\,\|\nabla F_k(\mathbf{x}_k)\|_2^2 + \frac{1}{4}\,\|\mathbf{m}_k + \lambda_k\mathbf{x}_k \odot \mathbf{s}_k\|_2^2\Big] \leq \epsilon^2.$$

b) *The gradient moment* $\mathbf{m}_k$ *can well estimate the full gradient* $\nabla F(\mathbf{x}_k)$ *and* $\nabla F(\mathbf{z}_k)$*:*

$$\frac{1}{T}\sum\nolimits_{k=0}^{T-1}\max\Big\{\mathbb{E}\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\|_2^2, \mathbb{E}\|\mathbf{m}_k - \nabla F(\mathbf{z}_k)\|_2^2\Big\} \leq 16\epsilon^2 + \frac{8(\gamma_y^3+\gamma_z^3)\eta^2}{\nu^2}\epsilon^2.$$

c) *The sequence* $\{(\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k)\}$ *satisfies*

$$\frac{1}{T}\sum\nolimits_{k=0}^{T-1}\Big\{\mathbb{E}\|\mathbf{x}_k - \mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2, \mathbb{E}\|\mathbf{y}_k - \mathbf{x}_k\|_2^2, \mathbb{E}\|\mathbf{z}_k - \mathbf{x}_k\|_2^2\Big\} \leq \Big\{4\eta^2\gamma_y^2\epsilon^2, 4\eta\gamma_y^2\epsilon^2, \frac{8(\gamma_y^3+\gamma_z^3)\eta^2}{\nu^2}\epsilon^2\Big\}.$$

d) *The total stochastic gradient complexity to achieve the above three properties is* $\mathcal{O}\big(\frac{c_2^{2.5}\sigma^2 L\Delta}{\nu^{1.25}\epsilon^4}\big)$*.*

By inspecting Theorem 4, one can observe that on the nonconvex problem, Win2-accelerated AdamW and Adam in Algorithm 2 can converge. Specifically, as shown in Theorem 4 a), with at most $T = \mathcal{O}\big(\frac{c_\infty^{2.5}\Delta\sigma^2 L}{\nu^{1.25} b\epsilon^4}\big)$ iterations, one can bound the average gradient $\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\big[\,\|\nabla F_k(\mathbf{x}_k)\|_2^2\big] \leq \epsilon^2$. Theorem 1 b) indicates that the gradient moment $\mathbf{m}_k$ is very close to the full gradient $\nabla F(\mathbf{z}_k)$ and also $\nabla F(\mathbf{x}_k)$, and thus is a good estimation to $\nabla F(\mathbf{z}_k)$ and $\nabla F(\mathbf{x}_k)$ used in Eqn. (13)–(15). Theorem 1 c) shows that the three sequences $\mathbf{x}_k$, $\mathbf{y}_k$ and $\mathbf{z}_k$ can converge to each other even though they use different stepsizes, guaranteeing the good convergence behavior of Win2-accelerated AdamW and Adam. All these convergence properties are very similar to Win-accelerated AdamW and Adam.

To find an ϵ-approximate first-order stationary point, Theorem 1 d) shows that stochastic gradient complexity of Win2-accelerated Adam and AdamW is at the order of $\mathcal{O}\big(\frac{c_\infty^{2.5}\sigma^2 L}{\nu^{1.25}\epsilon^4}\big)$ which accords with the lower bound $\Omega(\frac{1}{\epsilon^4})$ in (Arjevani et al., 2022, 2020) (up to constant factors). By comparing Theorem 1 and 4, one can observe that Win2-accelerated Adam and AdamW share the same complexity $\mathcal{O}\big(\frac{c_\infty^{2.5}\sigma^2 L}{\nu^{1.25}\epsilon^4}\big)$ with Win-accelerated Adam and AdamW, but achieves faster empirical convergence speed on deep network training tasks as shown in Sec. 5. Accordingly, like Win-accelerated Adam and AdamW, Win2-accelerated Adam and AdamW also reveal superior complexity to previous network optimizers, including Adam-type optimizers (*e.g.,* Adam, AdaGrad, AdaBound), Adabelief and RMSProp. Please see the detailed comparison in Sec. 3.3.

Convergence Analysis of Win2-accelerated LAMB. Here we also define the scaling factor $\alpha_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\sqrt{\mathbf{v}_k+\nu}+\lambda_k\mathbf{x}_k\|_2}$ for the analysis of Win2-accelerated LAMB. Theorem 5 summarizes the main convergence results of Win2-accelerated LAMB. See its proof in Appendix E.3.

Theorem 5 *Suppose that Assumptions 1 and 2 hold,* $\mathbf{x}_\star \in \mathrm{argmin}_{\mathbf{x}} F(\mathbf{x})$, $0 < \alpha_s \le \alpha_k \le \alpha_l$, *and* $\eta \le \mathcal{O}\Big(\frac{\nu^{1.25}b\epsilon^2}{\alpha_l c^{1.5}(\gamma_y^{1.5}+\gamma_z^{1.5})\sigma^2 L}\Big)$. *Let* $\eta_k^x = \eta$, $\eta_k^y = \gamma_y\eta_k^x$, $\eta_k^z = \gamma_z\eta_k^x$, $\gamma_z > \gamma_y > 1$, $\beta_1 \le \mathcal{O}\big(\frac{\alpha_s\nu^{0.5}b\epsilon^2}{\alpha_l c\sigma^2}\big)$, $\beta_2 \in (0,1)$, $c = (c_\infty^2+\nu)^{0.5}$, $\lambda_k = \lambda(1-\frac{\beta_2 c_\infty^2}{\nu})^k$ $(k>0)$ *and* $\lambda_0 = 0$ *with a constant* $\lambda > 0$. *Then after* $T = \mathcal{O}\big(\frac{\alpha_l c_\infty^{2.5}\sigma^2 L\Delta}{\alpha_s\nu^{1.25}b\epsilon^4}\big)$ *iterations with minibatch size* b *and* $\Delta = F(\mathbf{x}_0) - F(\mathbf{x}_\star)$, *the sequence* $\{(\mathbf{x}_k,\mathbf{y}_k,\mathbf{z}_k)\}_{k=0}^T$ *generated by Win2-accelerated LAMB in Algorithm 2 satisfies the following four properties.*

a) *The gradient* $\nabla F_k(\mathbf{x}_k)$ *of the sequence* $\{\mathbf{x}_k\}_{k=0}^T$ *can be upper bounded by*

$$\frac{1}{T}\sum\nolimits_{k=0}^{T-1}\mathbb{E}\Big[\,\|\nabla F_k(\mathbf{x}_k)\|_2^2 + \frac{1}{4\alpha_s\alpha_l}\|\mathbf{m}_k + \lambda_k\mathbf{x}_k\odot\mathbf{s}_k\|_2^2\Big] \le \epsilon^2.$$

b) *The gradient moment* $\mathbf{m}_k$ *can well estimate the full gradient* $\nabla F(\mathbf{x}_k)$ *and* $\nabla F(\mathbf{z}_k)$*:*

$$\frac{1}{T}\sum\nolimits_{k=0}^{T-1}\max\Big\{\mathbb{E}\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\|_2^2, \mathbb{E}\|\mathbf{m}_k - \nabla F(\mathbf{z}_k)\|_2^2\Big\} \le \frac{20\alpha_l}{\alpha_s}\epsilon^2 + \frac{8\alpha_s\alpha_l(\gamma_y^3+\gamma_z^3)\eta^2}{c_1^2}\epsilon^2.$$

c) *The sequence* $\{(\mathbf{x}_k,\mathbf{y}_k,\mathbf{z}_k)\}$ *satisfies*

$$\frac{1}{T}\sum\nolimits_{k=0}^{T-1}\Big\{\mathbb{E}\|\mathbf{x}_k-\mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2, \mathbb{E}\|\mathbf{y}_k-\mathbf{x}_k\|_2^2, \mathbb{E}\|\mathbf{z}_k-\mathbf{x}_k\|_2^2\Big\} \le 4\alpha_s\alpha_l\cdot\Big\{\eta^2\epsilon^2, \eta\gamma_y^2\epsilon^2, \frac{2(\gamma_y^3+\gamma_z^3)\eta^2}{\nu^2}\epsilon^2\Big\}.$$

d) *The total stochastic gradient complexity to achieve the above three properties is* $\mathcal{O}\Big(\frac{\alpha_l c_2^{2.5}\sigma^2 L\Delta}{\alpha_s\nu^{1.25}\epsilon^4}\Big)$.

Theorem 5 shows the convergence of Win2-accelerated LAMB optimizer. Specifically, it proves that the average gradient $\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\big[\,\|\nabla F_k(\mathbf{x}_k)\|_2^2\,\big]$ can be bounded by ϵ^2, and the moment $\mathbf{m}_k$ is very close to the full gradient $\nabla F(\mathbf{x}_k)$ and $\nabla F(\mathbf{z}_k)$. Furthermore, it also reveals the small distance between the three sequences $\mathbf{x}_k$, $\mathbf{y}_k$ and $\mathbf{z}_k$. For the stochastic gradient complexity to compute an ϵ-approximate first-order stationary point, Win2-accelerated LAMB shares the same complexity of $\mathcal{O}\big(\frac{c_\infty^{2.5}}{\epsilon^4}\big)$ with Win-accelerated LAMB, but reveals faster empirical convergence speed and better performance as shown in Sec. 5. In this way, Win2-accelerated LAMB has also lower complexity than vanilla LAMB optimizer whose complexity is $\mathcal{O}\big(\frac{c_\infty d^{2.5}}{\epsilon^4}\big)$, and makes an improvement by a factor of $\mathcal{O}\big(\frac{d^{2.5}}{c_\infty^{1.5}}\big)$ which is indeed large for modern over-parameterized networks. All these results are similar and also consistent with the results in Theorem 2.

Convergence Analysis of Win2-accelerated SGD. Now we discuss the convergence performance of Win-accelerated SGD in Theorem 6, whose proof is given in Appendix D.3.

Theorem 6 *Suppose that Assumptions 1 and 2 hold, and* $\mathbf{x}_\star \in \mathrm{argmin}_{\mathbf{x}} F(\mathbf{x})$. *Assume* $\eta \le \mathcal{O}\big(\frac{\nu^{1.25}b\epsilon^2}{c^{1.5}(\gamma_y^{1.5}+\gamma_z^{1.5})\sigma^2 L}\big)$. *Let* $\eta_k^x = \eta, \eta_k^y = \gamma_y\eta, \eta_k^z = \gamma_z\eta, \gamma_z > \gamma_y > 1$, $\beta_1 \le \mathcal{O}\big(\frac{\nu^{0.5}b\epsilon^2}{c\sigma^2}\big)$, $\beta_2 \in (0,1)$, $c = (c_\infty^2+\nu)^{0.5}$, $\lambda_k = \lambda(1-\frac{\beta_2 c_\infty^2}{\nu})^k$ $(k>0)$ *and* $\lambda_0 = 0$ *with a constant* $\lambda > 0$. *Then after* $T = \mathcal{O}\big(\frac{c_\infty^{2.5}\sigma^2 L\Delta}{\nu^{1.25}b\epsilon^4}\big)$ *iterations with minibatch size* b *and* $\Delta = F(\mathbf{x}_0) - F(\mathbf{x}_\star)$, *the sequence* $\{(\mathbf{x}_k,\mathbf{y}_k,\mathbf{z}_k)\}_{k=0}^T$ *generated by Win2-accelerated SGD in Algorithm 2 satisfies the four properties in Theorem 4 with* $\nu = c_\infty = c = 1$ *and* $\mathbf{s}_k = \mathbf{1} \in \mathbb{R}^d$.

From Theorem 6, one can observe that with the same hyper-parameter settings as in Theorem 4, the sequence $\{(\mathbf{x}_k,\mathbf{y}_k,\mathbf{z}_k)\}_{k=0}^T$ generated by Win2-accelerated SGD satisfies the four properties in

Table 1: ImageNet top-1 accuracy (%) of ResNet18. $*$, † and ‡ are respectively reported in (Chen et al., 2021), (Zhuang et al., 2020) and (Liu et al., 2019).

AdaBound	68.1*	Radam	67.7*
Yogi	68.2*	Padam	70.1*
Nadam	68.8	AdaBelief	70.1†
SGD-H	67.3	Yogi	68.2*
SGD-N	70.2*	Adam	66.5‡
SGD-Win	$70.7_{+0.5}$	Adam-Win	$69.3_{+2.8}$
SGD-Win2	$70.8_{+0.6}$	Adam-Win2	$69.9_{+3.4}$
AdamW	67.9*	LAMB	68.5
AdamW-Win	$71.0_{+3.1}$	LAMB-Win	$71.1_{+2.6}$
AdamW-Win2	$71.2_{+3.3}$	LAMB-Win2	$71.3_{+2.8}$

Theorem 4 with $\nu = c_\infty = c = 1$ and $\mathbf{s}_k = \mathbf{1}$. In this way, the complexity of Win2-accelerated SGD is $\mathcal{O}(\frac{L\sigma^2}{\epsilon^4})$, and accords with the lower bound $\Omega(\frac{1}{\epsilon^4})$ in (Arjevani et al., 2022, 2020) (up to constant factors) under Assumptions 1 and 2. This also shows the efficiency of Win2-accelerated SGD.

5. Experiments

Here we evaluate our accelerated algorithms on three representative tasks, including vision classification tasks, instance segmentation tasks, and natural language modeling tasks. For classification tasks, we conduct experiments using Convolutional Neural Networks (CNNs) such as ResNet (He et al., 2016), as well as Vision Transformers (ViTs), including ViT (Dosovitskiy et al., 2021) and PoolFormer (Yu et al., 2022a,b). Regarding instance segmentation, Mask R-CNN (He et al., 2017) with Swin transformer (Liu et al., 2021) as backbone is used for evaluation. For language modeling tasks, we employ LSTM (Hochreiter and Schmidhuber, 1997) and Transformer-XL (Dai et al., 2019) for evaluation. Moreover, we also compare Win and Nesterov in Sec. 5.4.

For clarity, we call our accelerated algorithm "X-Win" or "X-Win2" , where "X" denotes vanilla optimizers, *e.g.,* Adam. For Win2, it always uses three updating steps as shown in Algorithm 2, because of its good trade-off performance and GPU memory cost (see the experiments in Sec. 5.4). In all experiments, we do not change model architectures and data augmentations, and only replace the default optimizer with ours. Moreover, for all experiments, our accelerated algorithms, *e.g.,* AdamW-Win and AdamW-Win2, always use the default optimizer-inherent hyper-parameters of the vanilla optimizers, *e.g.,* first- and second-order moment parameters β_1 and β_2 in AdamW; and their aggressive steps η_k^y and η_k^z always satisfies $\eta_k^y = 2\eta_k^x$ and $\eta_k^z = 8\eta_k^x$. These settings well reduce the parameter-tuning cost of our algorithms. In the experiments, same with other optimizers, we only slightly tune other widely tuned hyper-parameters around the default ones used in the vanilla optimizers, *e.g.,* stepsize and warm-up epochs. This is reasonable, as our accelerated algorithms have two or three stepsizes, while vanilla optimizers often have a single step size which may not be suitable for ours.

5.1 Results on Vision Classification Tasks

Results on ResNet18. Here we follow the conventional supervised training setting commonly used in ResNets (He et al., 2016) and evaluate our accelerated algorithms on the ImageNet dataset (Fei-Fei, 2009). We defer the hyper-parameter settings of the four accelerated algorithms in Table 1 into Appendix A.

Table 2: ImageNet top-1 accuracy (%) of ResNet50&101 whose official optimizer is LAMB due to the stronger data augmentation for better performance. $*$ is reported in (Wightman et al., 2021).

	ResNet50				ResNet101			
Epoch	100	200	300	average	100	200	300	average
SAM	77.3	78.7	79.4	78.5	79.5	81.1	81.6	80.7
SGD-H	75.3	76.9	77.2	76.5	77.7	78.6	78.8	78.4
SGD-N	77.0	78.6	79.3	78.3	79.3	81.0	81.4	80.6
SGD-Win	78.0	79.2	79.7	$79.0_{+0.7}$	80.1	81.2	81.6	$81.0_{+0.4}$
SGD-Win2	78.1	79.3	79.8	$79.1_{+0.8}$	80.3	81.4	81.8	$81.2_{+0.6}$
Adam	76.9	78.4	78.8	78.1	78.4	80.2	80.6	79.7
Adam-Win	77.4	78.8	79.3	$78.5_{+0.4}$	79.2	80.6	81.0	$80.3_{+0.6}$
Adam-Win2	77.7	79.1	79.4	$78.8_{+0.6}$	79.2	80.6	81.3	$80.4_{+0.7}$
AdamW	77.0	78.9	79.3	78.4	78.9	79.9	80.4	79.7
AdamW-Win	78.0	79.3	79.9	$79.1_{+0.7}$	80.2	81.1	81.3	$80.9_{+1.2}$
AdamW-Win2	78.2	79.5	79.9	$79.2_{+0.8}$	80.4	81.4	81.7	$81.2_{+1.5}$
LAMB	77.0	79.2	79.8^*	78.7	79.4	81.1	81.3^*	80.6
LAMB-Win	78.4	79.7	80.1	$79.4_{+0.7}$	80.6	81.5	81.7	$81.3_{+0.7}$
LAMB-Win2	78.6	79.7	80.2	$79.5_{+0.8}$	80.6	81.6	81.9	$81.4_{+0.8}$

From the results in Table 1, one can observe that our Win- and Win2-accelerated algorithms can improve the corresponding non-accelerated versions by a remarkable margin. For instance, AdamW-Win, Adam-Win and LAMB-Win respectively make 3.1%, 2.8% and 2.6% improvement over their corresponding non-accelerated counterparts, AdamW, Adam and LAMB. Moreover, AdamW-Win2, Adam-Win2 and LAMB-Win2 make more improvements, and respectively improve the corresponding vanilla optimizers by 3.3%, 3.4% and 2.8% in accuracy. For SGD optimizer, SGD-Win improves SGD-H (*i.e.* SGD + heavy ball) by 3.4%, and also surpasses SGD-N (Nesterov-accelerated SGD in Sec. 3.2) by 0.5%, thus validating the superiority of our Win acceleration. SGD-Win2 also outperforms SGD-H by 3.5% and SGD-N by 0.6%.

Notably, our Win2- and Win-accelerated algorithms, *i.e.* SGD-Win2, AdamW-Win2 and LAMB-Win2, beat several other optimizers, *e.g.,* AdaBound, Radam (Liu et al., 2019), Nadam (Dozat, 2016), Padam (Chen et al., 2021), AdaBelief, in which Nadam uses vanilla Nesterov acceleration in Adam to estimate its first-order gradient moment. Actually, LAMB-Win2 sets a new SoTA top-1 accuracy of 71.3% on ResNet18. All these results show the strong compatibility and superiority of our Win- and Win2-acceleration in adaptive algorithms.

Results on ResNet50 & 101. Here we adopt the training setting in (Wightman et al., 2021) to train ResNet50 and ResNet101, because this setting uses stronger data augmentation and largely improves CNNs' performance. Specifically, this setting uses not only the conventional augmentations in (He et al., 2016), *e.g.,* random crop and horizontal flipping, but also other advanced augmentations, *e.g.,* RandAugment (Cubuk et al., 2020); see the augmentation details and our algorithmic hyper-parameter settings in Appendix A. Here LAMB is the default optimizer because of its higher performance than other optimizers caused by the stronger augmentations (Wightman et al., 2021). All optimizers in Table 2 are under this setting.

Table 2 reports the top-1 accuracy of the compared optimizers on ImageNet. By comparison, one can observe that our accelerated algorithms consistently outperform their corresponding non-accelerated version. For example, across the three training epoch settings on ResNet50 / ResNet101, LAMB-Win and LAMB-Win2 always achieve remarkable improvement over the official optimizer

Table 3: ImageNet top-1 accuracy (%) of ViT and PoolFormer whose default optimizers are both AdamW. $*$ and $\diamond$ are respectively reported in (Touvron et al., 2021) and (Yu et al., 2022a).

	ViT-S			ViT-B			PoolFormer-S12		
Epoch	150	300	average	150	300	average	150	300	average
SGD-N	77.4	79.4	78.4	79.6	80.0	79.8	69.7	74.3	72.0
SGD-Win	78.1	80.1	$79.1_{+0.7}$	80.4	80.8	$80.6_{+0.8}$	71.1	74.5	$72.8_{+0.8}$
SGD-Win2	78.2	80.3	$79.3_{+0.9}$	80.6	81.4	$81.0_{+1.2}$	71.4	74.7	$73.1_{+1.1}$
Adam	77.3	79.3	78.3	79.0	79.7	79.4	74.3	76.3	75.3
Adam-Win	78.6	80.2	$79.4_{+1.1}$	80.0	80.5	$80.3_{+0.9}$	75.6	77.1	$76.4_{+1.1}$
Adam-Win2	79.1	80.6	$79.9_{+1.6}$	80.6	81.1	$80.9_{+1.5}$	76.2	77.6	$76.9_{+1.6}$
AdamW	78.3	79.8*	79.1	79.5	81.8*	80.7	75.2	77.1*	76.2
AdamW-Win	79.3	81.0	$80.2_{+1.1}$	81.0	82.3	$81.7_{+1.0}$	76.7	77.6	$77.2_{+1.0}$
AdamW-Win2	79.5	81.4	$80.5_{+1.4}$	81.1	82.6	$81.9_{+1.2}$	77.0	78.3	$77.7_{+1.5}$
LAMB	78.0	79.6	78.8	80.3	80.8	80.6	75.4	77.4	76.4
LAMB-Win	79.3	80.6	$80.0_{+1.2}$	81.0	81.4	$81.2_{+0.6}$	76.7	78.0	$77.4_{+1.0}$
LAMB-Win2	79.4	81.0	$80.2_{+1.4}$	81.3	81.9	$81.6_{+1.0}$	77.3	78.4	$77.9_{+1.5}$

LAMB for this training recipe. Specifically, LAMB-Win makes 0.7% average improvement over LAMB on both ResNet50 / ResNet101. For AdamW-Win and Adam-Win, they also respectively improve their vanilla counterparts by 0.7% and 0.4% on ResNet50, 1.2% and 0.6% on ResNet101. SGD-Win also makes 2.5% and 0.7% overall improvement over heavy-ball accelerated SGD (SGD-H) and Nesterov accelerated SGD (SGD-N) on ResNet50, and also has similar advantage on ResNet101. Besides, Win2-accelerated optimizers show further improvement as demonstrated by the overall 0.8%, 0.8%, 0.6%, and 0.8% improvement of LAMB-Win2, AdamW-Win2, Adam-Win2 and SGD-Win2 over their corresponding vanilla counterparts on ResNet50. For ResNet101. One can also observe very similar improvement of Win2-accelerated optimizers.

The above improvements achieved by Win and Win2 are not trivial because of the following two reasons. 1) Since the performance is already high and may approach the model limit, it is already very hard to make large improvement. This is testified by the fact that in (Wightman et al., 2021), using LAMB to train ResNet50 for 600 epochs only gives 80.4% top-1 accuracy. In contrast, our accelerated LAMB-Win uses 300 epochs (half training cost) to achieve 80.2%. 2) By comparing the previous optimizers, including SAM, SGD-N, Adam, AdamW and LAMB, one can observe smaller accuracy gap ($\leq 0.2\%$) between the best optimizer and the runner-up. For example, on ResNet101, the SoTA optimizer, *i.e.* SAM, only makes 0.1% average improvement over the runner-up LAMB. All these comparisons show the non-travail improvement of our accelerated algorithms over their corresponding counterparts.

Results on ViTs. We follow the widely used official training setting of ViTs (Touvron et al., 2021; Yu et al., 2022a). To evaluate the performance of our accelerated algorithms, we select two popular and representative ViT architures, including ViT (Dosovitskiy et al., 2021) and PoolFormer (Yu et al., 2022a) whose official optimziers are both AdamW. We refere the reader to the training setting and our hyper-parameter settings in Appendix A.

We test our accelerated algorithms under different model sizes and different training epochs, and report the results in Table 3. One can find that since AdamW and LAMB use the decoupled weight decay, they often enjoy better performance than SGD and Adam, which is also observed in other works, *e.g.,* (Xiao et al., 2021; Nado et al., 2021). Moreover, under different training settings, our accelerated algorithms consistently outperform the corresponding non-accelerated

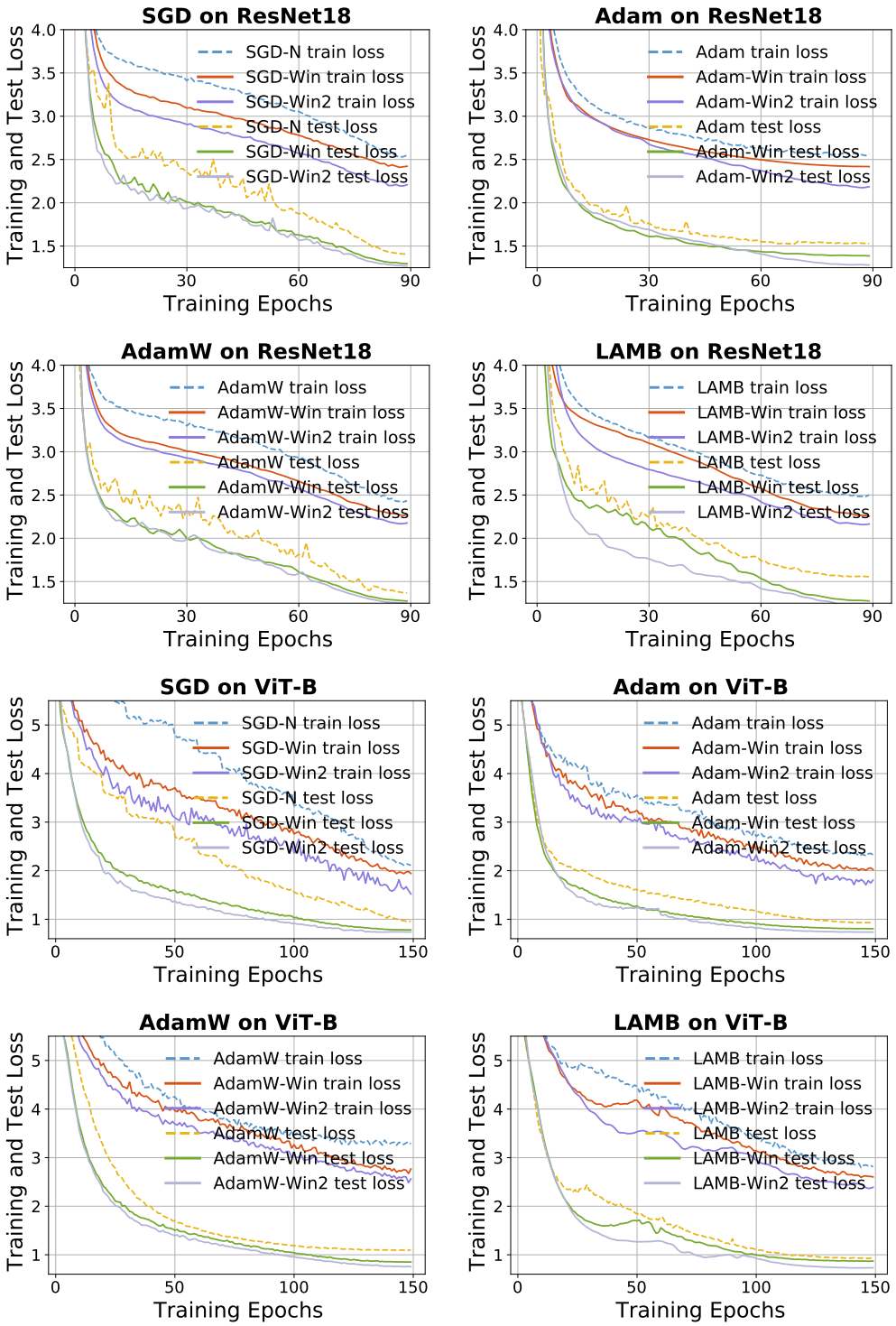

Figure 1: Visualization of training and test losses on ImageNet. In all figures, training loss is larger than test one, as training data use random augmentations, *e.g.,* random crop and clip, while test data only adopt the centralization crop which eases the recognition difficulty and thus has small loss.

counterparts. Specifically, compared with the default AdamW optimizer on both ViT and PoolFormer, our accelerated AdamW-Win respectively makes about 1.1%, 1.0%, 1.0% average improvement under the two training epoch settings on ViT-S, ViT-B and PoolFormer-S12. For Adam-Win and LAMB-Win, one can also observe their remarkable improvements on the three ViT backbones. Moreover, our accelerated SGD-Win also outperforms the Nesterov-accelerated SGD denoted as "SGD-N" by non-trivial margins under all settings.

For Win2-accelerated optimizers, they can further improve the vanilla optimizers and also Win-accelerated optimizers. Specially, compared with vanilla AdamW, AdamW-Win2 also respectively brings overall 1.4%, 1.2%, 1.5% accuracy improvement on ViT-S, ViT-B and PoolFormer-S12. For Adam-Win2, it also improves vanilla Adam by 1.6%, 1.5%, 1.6% on the three models. Indeed, from the results in Table 3, one can also observe very consistent improvement made by LAMB-Win2 and SGD-Win2 on the ViT and PoolFormer models. All these results are consistent with the observations on ResNets, and they together demonstrate the advantage of our accelerated optimizers for deep network training.

Results Analysis. Here we investigate the convergence behaviors of our accelerated algorithms, and aim to explain their better test performance over their non-accelerated counterparts. In Fig. 1, we plot the curves of training and test losses along with the training epochs on ResNet18 and ViT-B. One can find that our accelerated algorithms, including Win- and Win2-accelerated optimizers, show much faster convergence behaviors than their non-accelerated counterparts, *e.g.,* AdamW. Moreover, Win2 also outperforms Win in terms of convergence speed, especially on the training loss. Besides, SGD-Win and SGD-Win2 also converge faster than Nesterove-accelerated SGD, *i.e.* SGD-N. So these faster convergence behaviors could contribute to our accelerated algorithms for their higher performance over non-accelerated counterparts under the same computational cost.

5.2 Results on Instance Segmentation

Here we evaluate our Win- and Win2-accelerated algorithms on the instance segmentation task which consists of 1) bounding box detection to detect the whole object and also 2) mask segmentation to segment the object in the bounding box. In this sense, instance segmentation indeed includes object detection and also segmentation tasks. For evaluation, we employ the widely used large-scale COCO dataset (Lin et al., 2014) for evaluation and adopt Mask R-CNN (He et al., 2017) framework with the Swin transformer (Liu et al., 2021) as the backbone. For fairness, we adopt the setting in MMdection to test all the optimizers and train the models for 12 epochs. The official optimizer is AdamW whose results are quoted from MMdection (Chen et al., 2019).

Table 4 reports the box Average Precision (AP^b) and mask AP (AP^m) to respectively evaluate the performance of the bounding box detection sub-task and mask segmentation sub-task in the instance segmentation task. By comparison, one can observe that both Win- and Win2-accelerated optimizers can improve vanilla optimizers. Specially, on the official AdamW optimizer, AdamW-Win surpasses it by 0.2 average AP^b and 0.1 average AP^m, and AdamW-Win2 also makes 0.3 average AP^b and 0.2 average AP^m. For SGD-N, SGD-Win improves it by 0.8 average AP^b and 0.5 average AP^m, and SGD-Win2 brings 1.1% AP^b and 0.7% AP^m improvement on average. One can also observe very consistent improvement made by Win and Win2 on both Adam and LAMB.

Table 4: Instance segmentation box/mask-AP (↑) of Swin-based Mask-RCNN (He et al., 2017) on COCO (Lin et al., 2014) dataset, where AdamW is the official optimizer. $*$ is from (Chen et al., 2019).

	Object Bounding Box Detection				Object Mask Segmentation			
	AP^b	AP^b_{50}	AP^b_{75}	average	AP^m	AP^m_{50}	AP^m_{75}	average
SGD-N	41.4	63.9	44.6	50.0	38.1	60.8	40.9	46.6
SGD-Win	42.1	64.2	46.0	$50.8_{+0.8}$	38.7	61.2	41.5	$47.1_{+0.5}$
SGD-Win2	42.3	64.6	46.4	$51.1_{+1.1}$	38.8	61.5	41.7	$47.3_{+0.7}$
Adam	42.4	64.6	46.2	51.1	39.1	61.6	42.0	47.5
Adam-Win	42.9	65.2	47.0	$51.7_{+0.6}$	39.4	61.9	42.3	$47.9_{+0.4}$
Adam-Win2	43.0	65.3	47.1	$51.8_{+0.7}$	39.4	62.2	42.4	$48.0_{+0.5}$
AdamW	42.7*	65.2*	46.8*	51.5	39.3*	62.2*	42.2*	47.9
AdamW-Win	42.8	65.4	46.8	$51.7_{+0.2}$	39.5	62.2	42.2	$48.0_{+0.1}$
AdamW-Win2	43.0	65.5	47.1	$51.8_{+0.3}$	39.5	62.5	42.4	$48.1_{+0.2}$
LAMB	42.5	64.9	46.3	51.2	39.1	61.7	41.9	47.5
LAMB-Win	42.7	65.0	46.7	$51.5_{+0.3}$	39.3	61.8	42.4	$47.8_{+0.3}$
LAMB-Win2	42.8	65.1	46.7	$51.5_{+0.3}$	39.4	62.2	42.4	$48.0_{+0.5}$

Table 5: Test perplexity (↓) of LSTM on Penn Treebank. $*$ is reported by AdaBelief (Zhuang et al., 2020).

AdaBound	63.6*	Radam	70.0*
Yogi	67.5*	AdaBelief	61.2*
fromage	68.0*	MSVAG	65.3*
SGD-H	67.4	Padam	63.2*
SGD-N	63.8*	Adam	64.3*
SGD-Win	$61.6_{+2.2}$	Adam-Win	$62.7_{+1.6}$
SGD-Win2	$61.0_{+2.8}$	Adam-Win2	$61.8_{+2.5}$
AdamW	67.0*	LAMB	66.8
AdamW-Win	$66.5_{+0.5}$	LAMB-Win	$66.2_{+0.6}$
AdamW-Win2	$64.4_{+1.6}$	LAMB-Win2	$64.1_{+1.7}$

5.3 Results on Natural Language Modeling Tasks

Results on LSTM. We follow AdaBelief to test our accelerated algorithms via training three-layered LSTM (Hochreiter and Schmidhuber, 1997) on the Penn TreeBank dataset (Marcinkiewicz, 1994) for 200 epochs. See optimization and training details in Appendix A. From Table 5, one can observe that our Win-accelerated optimizers consistently surpass the corresponding non-accelerated counterparts, and actually bring 1.2 overall average perplexity improvement over the four non-accelerated counterparts. Win2-accelerated algorithms further improves Win, and respectively makes 2.8, 2.5, 1.6 and 1.7 on the four corresponding vanilla optimizers.

Results on Transformer-XL. We adopt a widely used language sequence model, *i.e.* Transformer-XL (Dai et al., 2019), to further evaluate the performance of our accelerated algorithms. Since 1)

Table 6: Test PPL (↑) of Transformer-XL-base on WikiText-103 where Adam is the official optimizer. * is reported in the official implementation.

Transformer-XL	Training Steps			
	50k	100k	200k	average
Adam	28.5	25.5	24.2*	26.7
Adam-Win	26.7	25.0	24.0	$25.2_{+1.5}$
Adam-Win2	26.4	24.9	23.8	$25.0_{+1.7}$

Table 7: ImageNet top-1 accuracy (%) on ResNet18 (left) and ResNet50 (right).

SGD-H	67.3	Adam	66.5
SGD-N	$70.2_{+2.9}$	Adam-N	$68.8_{+2.3}$
SGD-Win	$70.7_{+3.4}$	Adam-Win	$69.3_{+2.8}$
SGD-Win2	$70.8_{+3.5}$	Adam-Win2	$69.9_{+3.4}$
AdamW	67.9	LAMB	68.5
AdamW-N	$69.3_{+1.4}$	LAMB-N	$69.7_{+1.2}$
AdamW-Win	$71.0_{+3.1}$	LAMB-Win	$71.1_{+2.6}$
AdamW-Win2	$71.2_{+3.3}$	LAMB-Win2	$71.3_{+2.8}$

SGD-H	75.3	Adam	76.9
SGD-N	$77.0_{+1.7}$	Adam-N	$77.0_{+0.1}$
SGD-Win	$78.0_{+2.7}$	Adam-Win	$77.4_{+0.5}$
SGD-Win2	$78.1_{+2.8}$	Adam-Win2	$77.7_{+0.8}$
AdamW	77.0	LAMB	77.0
AdamW-N	$78.4_{+0.4}$	LAMB-N	$77.5_{+0.5}$
AdamW-Win	$78.0_{+1.0}$	LAMB-Win	$78.4_{+1.4}$
AdamW-Win2	$78.2_{+1.2}$	LAMB-Win2	$78.6_{+1.6}$

Adam is the most popular and used optimizer in NLP models, including Transformer-XL, and 2) our limited resource cannot well tune the hyper-parameters of other optimizers in Sec. 5.1, we take Adam as an example to show the superiority of our accelerated algorithms. Follow the official setting of Transformer-XL-base, we use Adam-Win and Adam-Win2 with the default hyper-parameters of Adam on the WikiText-103 dataset. See more details in Appendix A.

Table 6 shows that under different training steps, our accelerated Adam-Win and Adam-Win2 always achieve lower test PPL than the official Adam optimizer. Specifically, Adam-Win and Adam-Win2 respectively improve 1.5 and 1.7 average test PPL over the official Adam optimizer on the three test cases. All these results are consistent with observations on vision tasks, and they together demonstrate the advantages of our accelerated algorithms.

5.4 Ablation Study

Comparison with Nesterov acceleration. Here we empirically compare Nesterov acceleration with our Win and Win2. Regarding Nesterov acceleration, NAdam (Dozat, 2016) introduces it into Adam for acceleration. So we follow Nadam to implement Nesterov-accelerated AdamW and LAMB. For Nesterov-accelerated SGD, we use the one implemented in PyTorch. For brevity, we call Nesterov-accelerated optimizer "X-N", where "X" denotes the vanilla optimizer, *e.g.,* SGD and Adam. Note, Adam-N denotes the vanilla NAdam. Then we use the same settings in Sec 5.1 to train ResNet18 for 90 epochs and ResNet50 for 100 epochs, and evaluate them on the ImageNet dataset.

Table 7 reports the classification results. One can observe that for SGD, Adam, AdamW and LAMB, Win- and Win2-accelerated optimizers still achieve higher classification accuracy than the corresponding Nesterov accelerated counterparts on both ResNet18 and ResNet50. This shows the superiority of our Win and Win2 acceleration on the deep network training tasks.

Robustness Analysis. Compared with the vanilla optimizer, Win-accelerated algorithm only introduces the only extra hyper-parameter η_k^y, and Win2-accelerated optimizer adds two extra hyper-parameter η_k^y and η_k^z. Now we investigate the robustness of Win- and Win2-accelerated algorithms to these hyper-parameters. For convenience, in all experiments, Win-accelerated algorithms always set $\eta_k^y = \gamma_1 \eta_k^x$, where $\gamma_1 = 2$. Here we first investigate the effects of γ_1 to Win-accelerated algorithms on ResNet50 by taking AdamW-Win and LAMB-Win as examples because of their superior performance. We train both AdamW-Win and LAMB-Win for 100 epochs. Table 8 shows the stable performance of AdamW-Win and LAMB-Win when tuning γ_1 in a relatively large range, thus validating their robustness to the hyper-parameter γ_1.

Then we investigate Win2-accelerated optimizers which always uses $\eta_k^y = \gamma_1 \eta_k^x$ and $\eta_k^z = \gamma_2 \eta_k^x$ in all experiments. For convenience, we fix $\gamma_1 = 2$ and then investigate the effects of hyper-parameter γ_2 to the performance. From Table 8, one can observe that even though γ_2 varies in a relatively

Table 8: Effects of γ_1 and γ_2 to top-1 accuracy (%) of Win- and Win2-accelerated AdamW and LAMB on ResNet50.

γ_1	1.5	2	3	4	6	8	γ_2	4	6	8	10	12
AdamW-Win	77.9	78.0	78.0	77.9	78.1	78.0	AdamW-Win2	78.1	78.2	78.2	78.2	78.1
LAMB-Win	78.3	78.4	78.4	78.4	78.5	78.3	LAMB-Win2	78.6	78.7	78.6	78.5	78.4

Table 9: Effects of the updating step number q to top-1 accuracy (%), GPU peak memory cost (M) and also running time (minute) per epochs of Win2-accelerated AdamW and LAMB on ResNet18. Note, $q = 1$ corresponds to the vanilla optimizer, and $q = 2$ denotes the Win-accelerated optimizer.

	AdamW-Win2						LAMB-Win2					
q	1	2	3	4	5	6	1	2	3	4	5	6
Accuracy (%)	67.9	71.0	71.2	71.3	71.3	71.0	68.5	71.1	71.3	71.4	71.2	71.1
Peak Memory (M)	7880	7978	8094	8288	8478	8588	7882	7980	8080	8252	8564	8808
Running Time per Epoch (minute)	6.64	6.78	6.87	7.02	7.20	7.42	6.82	6.90	7.07	7.23	7.34	7.58

large range, AdamW-Win2 and LAMB-Win2 are always stable, and reveals strong robustness to hyper-parameter γ_2.

Analysis on Multiple Updates in Win2. In Sec. 4.1, we already develop a more general Win acceleration, Win2 for short, which extends the parameter update from two steps in Win to q updating steps for brevity ($q \geq 3$). Here we investigate the effects of the updating step number q in Win2 to 1) final performance (after training) and 2) GPU memory and running time (minute) per epochs (during training). We use both AdamW-Win and LAMB-Win to train ResNet18 for 90 epochs with minibatch size 512 on two A100 GPUs. For $q = 1$, the optimizer denotes the vanilla AdamW or LAMB optimizer and its stepsize is $\eta_k^{(1)}$. When $q = 2$, it corresponds to Win-accelerated AdamW or LAMB which uses the aggressive stepsize $\eta_k^{(2)} = 2\eta_k^{(1)}$. For Win2 with three updating steps ($q = 3$), we set $\eta_k^{(2)} = 2\eta_k^{(1)}$, $\eta_k^{(3)} = 8\eta_k^{(1)}$. When $q = 4,\ 5$ and 6, we use $\eta_k^{(2)} = 2\eta_k^{(1)}$, $\eta_k^{(3)} = 4\eta_k^{(1)}$, $\eta_k^{(4)} = 8\eta_k^{(1)}$, $\eta_k^{(5)} = 10\eta_k^{(1)}$, and $\eta_k^{(6)} = 12\eta_k^{(1)}$ for brevity.

Table 9 reports the empirical results for which we have several important observations. Firstly, in most cases, large q (e.g., $q \geq 4$) actually does not bring significant improvement in terms of classification accuracy, but indeed results in more extra memory cost during training. This is because for Win2 with three updating steps ($q = 3$), it already uses very aggressive stepsizes in practice, *e.g.,* $\eta_k^{(2)} = 2\eta_k^{(1)}$, $\eta_k^{(3)} = 8\eta_k^{(1)}$ in all our experiments. In this way, even though the updating step number q becomes larger, *e.g.,* $q \geq 4$, it is already hard to use more aggressive stepsize $\eta_k^{(q)} = a\eta_k^{(1)}$ ($a > 8$) while enjoying very stable training. This is the key reason why we often use Win2 with three updating steps ($q = 3$) in our experiments.

Secondly, from Table 9, one can find that Win (*i.e.* $q = 1$) and Win2 (*i.e.* $q = 2$) actually do not bring much extra memory cost and also computational ahead. For example, compared with vanilla AdamW, AdamW-Win brings extra 1.2% memory cost and also extra 2% computational ahead for each epoch. Similarly, AdamW-Win2 also only brings extra 2.7% memory cost and extra 3.4% computational ahead. One can also observe similar comparison results between accelerated LAMB and vanilla LAMB. Since Win and Win2 respectively introduce one and two simple and efficient algorithmic steps on vanilla optimizers, *e.g.,* the eighth step in Algorithm 1, they bring negligible extra computational overhead into the vanilla optimizers. Regarding the memory cost, in network training, one needs to store all temperate variables (*e.g.,* activation states) and feature maps for

back-propagation which often use much more GPU memory than the model parameters introduced by Win and Win2. This explains why Win and Win2 only bring very small extra memory cost.

6. Conclusion

In this work, we adopt the proximal point method to derive a weight-decay-integrated Nesterov acceleration for AdamW and Adam, and extend it to LAMB and SGD. Moreover, we prove the convergence of our accelerated algorithms, *i.e.* accelerated AdamW, Adam and SGD, and observe the superiority of the accelerated Adam-type algorithm over the vanilla ones in terms of stochastic gradient complexity. Finally, experimental results validate the advantages of our accelerated algorithms. We hope that Win could become a default acceleration option for all popular optimizers in the deep learning community to improve the training efficiency.

Acknowledgements

Pan Zhou was supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant. Zhouchen Lin was supported by the NSF China (No. 62276004) and the major key project of PCL, China (No. PCL2021A12). The authors sincerely thank the editor and anonymous reviewers for their constructive comments on this work.

This appendix is structured as follows. Appendix A provides more experimental details, such as hyper-parameter settings of the four accelerated algorithms and the official data augmentations. In Appendix B, we define some necessary notations for our analysis. Then Appendix C provides some auxiliary lemmas throughout this document. Next, Appendix D presents the proof of the convergence results in Sec. 1, *i.e.*, the proof of Theorems 1 in Appendix D.1, Theorems 2 in Appendix D.2, and Theorems 3 in Appendix D.3. Similarly, Appendix E presents the proof of the convergence results in Sec. 4, *i.e.*, the proof of Theorems 4 in Appendix E.1, Theorems 5 in Appendix E.2, and Theorems 6 in Appendix E.3. Finally, Appendix F provides the proofs of some auxiliary lemmas in Appendix C.

Appendix A. More Experimental Details

Due to space limitation, we defer the experimental details, such as hyper-parameter settings of the four accelerated algorithms, and their official augmentations in (He et al., 2016) and (Wightman et al., 2021), to this section.

For Win-accelerated algorithms, including AdamW-Win, LAMB-Win, Adam-Win, and SGD-Win, always share the default optimizer-inherent hyper-parameters of the vanilla optimizers and its aggressive step η_k^y is always $2\times$ larger than its conservative step η_k^x for all iterations, *i.e.* $\eta_k^y = 2\eta_k^x$. For Win2-accelerated AdamW, Adam, SGD and LAMB, we also always set $\eta_k^y = 2\eta_k^x$ and $\eta_k^z = 8\eta_k^x$. For all Win- and Win2-accelerated optimizers, their first- and second-order moment parameters β_1 and β_2 are set to the default values $\beta_1 = 0.9$ and $\beta_2 = 0.999$ used in AdamW, LAMB and Adam. For LAMB-Win and LAMB-Win2, their other key parameters, such as "grad averaging" and "trust clip", also adopt the default ones in vanilla LAMB. For SGD-Win and SGD-Win2, they use the default momentum parameter 0.9 and set dampening parameter as 0.0 used in vanilla SGD.

Settings on ResNet18. Here we follow the conventional supervised training setting used in ResNets (He et al., 2016) and evaluate our accelerated algorithms on ImageNet (Fei-Fei, 2009). For data augmentation in (He et al., 2016), it uses random crop and horizontal flipping with probability 0.5. For warm-up epochs, for all four accelerated algorithms, we set it as 5.0. For base learning rate, we respectively set it as 3×10^{-3}, 5×10^{-3}, 3×10^{-3}, and 1.2 for AdamW-Win, LAMB-Win, Adam-Win and SGD-Win. Moreover, we follow the default setting and use cosine learning rate decay. For weight decay, we respectively set it as 5×10^{-2}, 5×10^{-2}, 10^{-6}, and 10^{-3} for AdamW-Win, LAMB-Win, Adam-Win and SGD-Win. On ResNet18, all algorithms are trained for 90 epochs with minibatch size 512 by following the conventional setting. Win2-accelerated optimizer uses the same setting as Win on ResNet18.

Settings on ResNet50&101. For these two networks, we use "A2 training recipe" in (Wightman et al., 2021) to train them, since this training setting uses stronger data augmentation and largely improves CNNs' performance. Specifically, the data augmentation in (Wightman et al., 2021) uses random crop, horizontal flipping with probability, Mixup with parameter 0.1 (Zhang et al., 2018), CutMix with parameter 1.0 and probability 0.5 (Yun et al., 2019), and RandAugment (Cubuk et al., 2020) with $M = 7, N = 2$ and MSTD $= 0.5$. Moreover, it often use binary cross-entropy (BCE) loss for training.

For both ResNet50 and ResNet101, we release the hyper-parameter settings of Win and Win2-accelerated optimizers at our Github page [1]. You can find all the training hyper-parameters, *e.g.,* base learning rate, learning rate decay, weight decay and warm-up epoch number, from the training commands, and also the training logs.

1. Github project: `https://github.com/sail-sg/win`.

Settings on ViT and PoolFormer. We follow the widely used official training setting of ViTs (Touvron et al., 2021; Yu et al., 2022a). For this setting, data augmentation includes random crop, horizontal flipping with probability, Mixup with parameter 0.8 (Zhang et al., 2018), CutMix with parameter 1.0 and probability 0.5 (Yun et al., 2019), RandAugment (Cubuk et al., 2020) with $M = 9, N = 2$ and MSTD $= 0.5$, and Random Erasing with parameter $p = 0.25$. For training loss, we use cross entropy loss.

For ViT-S, ViT-B and PoolFormer, we release the hyper-parameter settings of Win and Win2-accelerated optimizers at our Github page[1]. You can find all the training hyper-parameters, *e.g.,* base learning rate, learning rate decay, weight decay and warm-up epoch number, from the training commands, and also the training logs.

Settings on LSTM. On LSTM, for base learning rate, we respectively set it as 1×10^{-3}, 1×10^{-2}, 1×10^{-2}, and 15.0 for AdamW-Win, LAMB-Win, Adam-Win and SGD-Win. For weight decay, we set it as 2×10^{-2}, 5×10^{-2}, 1.8×10^{-6} and 2×10^{-5} for AdamW-Win, LAMB-Win, Adam-Win, and SGD-Win. As for Win2, we, respectively, set the base learning rates as 2×10^{-3}, 2×10^{-3}, 5×10^{-2} and 15.0 for AdamW-Win2, LAMB-Win2, Adam-Win2, and SGD-Win2. The weight decay is 4×10^{-2}, 4×10^{-2}, 2.0×10^{-6} and 2×10^{-5} for AdamW-Win2, LAMB-Win2, Adam-Win2, and SGD-Win2. Moreover, we follow the default setting and divide the learning rate by 10 at epoch 100 and 145. We do not utilize the warmup strategy in this experiment. Following the default setting, we set minibatch size as 20.

Settings on Transformer-XL. On Transformer-XL, for the base learning rates, we set them as 4×10^{-4} and 8×10^{-4} for Adam-Win and Adam-Win2, respectively. Moreover, we follow the default setting and use cosine learning rate decay. For both Adam-Win and Adam-Win2, we set weight decay as 10^{-6} for and set warm-up steps as 2000. Following the default setting, we set the minibatch size as 60×4.

Appendix B. Notations

Here we first give some important notations used in this document. For brevity, we let

$$\mathbf{s}_k = \sqrt{\mathbf{v}_k + \nu}.$$

Since we have $\|\mathbf{m}_k\|_\infty \leq c_\infty$ and $\nu \leq \|\mathbf{v}_i + \nu\|_\infty \leq c_\infty^2 + \nu$ in Lemma 7 (see Appendix C), for brevity, let

$$c_1 := \nu^{0.5} \leq \|\mathbf{s}_k\|_\infty \leq c_2 := (c_\infty^2 + \nu)^{0.5}.$$

For Win- and Win2-accelerated AdamW and Adam, we define

$$\mathbf{w}_k := \mathbf{m}_k + \lambda_k \mathbf{x}_k \odot \mathbf{s}_k, \qquad \mathbf{x}_{k+1} - \mathbf{x}_k = -\frac{\eta_k^x}{1 + \lambda_k \eta_k^x} \frac{\mathbf{m}_k + \lambda_k \mathbf{x}_k \odot \mathbf{s}_k}{\mathbf{s}_k} = -\frac{\eta_k^x}{1 + \lambda_k \eta_k^x} \frac{\mathbf{w}_k}{\mathbf{s}_k}.$$

For Win- and Win2-accelerated LAMB, we define

$$\begin{aligned} \mathbf{w}_k :=& \alpha_k \mathbf{m}_k + (1 + \alpha_k) \lambda_k \mathbf{x}_k \odot \mathbf{s}_k, \\ \mathbf{x}_{k+1} - \mathbf{x}_k =& -\frac{\eta_k^x}{1 + \lambda_k \eta_k^x} \frac{\alpha_k \mathbf{m}_k + (1 + \alpha_k) \lambda_k \mathbf{x}_k \odot \mathbf{s}_k}{\mathbf{s}_k} = -\frac{\eta_k^x}{1 + \lambda_k \eta_k^x} \frac{\mathbf{w}_k}{\mathbf{s}_k}. \end{aligned}$$

where $\alpha_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k / \sqrt{\mathbf{v}_k + \nu} + \lambda_k \mathbf{x}_k\|_2}$.

Next, we introduce an virtual sequence $\{\mathbf{y}'_k\}$ into the algorithm. In this way, we can rewrite the update steps in Algorithm 1 in the manuscript as its equivalent form (16):

$$\begin{cases} \mathbf{g}_k = \frac{1}{b}\sum_{i=1}^{b}\nabla f(\mathbf{y}_k;\zeta_i); \\ \mathbf{m}_k = (1-\beta_1)\mathbf{m}_{k-1} + \beta_1\mathbf{g}_k; \\ \mathbf{v}_k = (1-\beta_2)\mathbf{v}_{k-1} + \beta_2\mathbf{g}_k^2; \\ \mathbf{x}_{k+1} = \frac{1}{1+\lambda_k\eta_k^x}\left(\mathbf{x}_k - \eta_k^x\mathbf{u}_k\right) \\ \mathbf{y}'_{k+1} = \mathbf{z}_k - \eta_k^y\mathbf{u}_k \\ \mathbf{y}_{k+1} = \frac{\xi_k^y}{\xi_k^x+\xi_k^y+\lambda_k}\mathbf{y}'_{k+1} + \frac{\xi_k^x}{\xi_k^x+\xi_k^y+\lambda_k}\mathbf{x}_{k+1} = \eta_k^y\tau_k^y\mathbf{x}_{k+1} + \eta_k^x\tau_k^y\mathbf{y}'_{k+1} \end{cases} \tag{16}$$

where $\mathbf{m}_0 = \mathbf{g}_0$, $\mathbf{v}_0 = \mathbf{g}_0^2$, $\xi_k^x = \frac{1}{\eta_k^x}$, $\xi_k^y = \frac{1}{\eta_k^y}$, $\delta_i^y = \frac{1}{\xi_k^x+\xi_k^y+\lambda_k}$, $\delta_i^z = \frac{1}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}$, $\tau_k^y = \frac{1}{\eta_k^x+\eta_k^y+\lambda_k\eta_k^x\eta_k^y}$, $\tau_k^z = \frac{1}{\eta_k^x+\eta_k^z+\lambda_k\eta_k^x\eta_k^z}$. Moreover, $\mathbf{u}_k = \frac{\mathbf{m}_k}{\mathbf{s}_k}$ in Win-accelerated AdamW and Adam, $\mathbf{u}_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\mathbf{s}_k+\lambda_k\mathbf{x}_k\|_2}\left(\frac{\mathbf{m}_k}{\mathbf{s}_k}+\lambda_k\mathbf{x}_k\right)$ in Win-accelerated LAMB.

For Win2, we have the updating rule as follows:

$$\begin{cases} \mathbf{g}_k = \frac{1}{b}\sum_{i=1}^{b}\nabla f(\mathbf{z}_k;\zeta_i); \\ \mathbf{m}_k = (1-\beta_1)\mathbf{m}_{k-1} + \beta_1\mathbf{g}_k; \\ \mathbf{v}_k = (1-\beta_2)\mathbf{v}_{k-1} + \beta_2\mathbf{g}_k^2; \\ \mathbf{x}_{k+1} = \frac{1}{1+\lambda_k\eta_k^x}\left(\mathbf{x}_k - \eta_k^x\mathbf{u}_k\right) = \frac{\xi_k^x}{\xi_k^x+\lambda_k}\mathbf{x}_{k+1} - \frac{1}{\xi_k^x+\lambda_k}\mathbf{u}_k; \\ \mathbf{y}'_{k+1} = \mathbf{y}_k - \eta_k^y\mathbf{u}_k \\ \mathbf{y}_{k+1} = \frac{\eta_k^x}{\eta_k^x+\eta_k^y+\lambda_k\eta_k^x\eta_k^y}\mathbf{y}'_{k+1} + \frac{\eta_k^y}{\eta_k^x+\eta_k^y+\lambda_k\eta_k^x\eta_k^y}\mathbf{x}_{k+1} = \frac{\xi_k^y}{\xi_k^x+\xi_k^y+\lambda_k}\mathbf{y}'_{k+1} + \frac{\xi_k^x}{\xi_k^x+\xi_k^y+\lambda_k}\mathbf{x}_{k+1} \\ \mathbf{z}'_{k+1} = \mathbf{z}_k - \eta_k^y\mathbf{u}_k \\ \mathbf{z}_{k+1} = \frac{\frac{1}{\eta_k^z}}{\frac{1}{\eta_k^x}+\frac{1}{\eta_k^y}+\frac{1}{\eta_k^z}+\lambda_k}\mathbf{z}'_k + \frac{\frac{1}{\eta_k^y}}{\frac{1}{\eta_k^x}+\frac{1}{\eta_k^y}+\frac{1}{\eta_k^z}+\lambda_k}\mathbf{y}_{k+1} + \frac{\frac{1}{\eta_k^x}}{\frac{1}{\eta_k^x}+\frac{1}{\eta_k^y}+\frac{1}{\eta_k^z}+\lambda_k}\mathbf{x}_{k+1} \\ \qquad = \frac{\xi_k^z}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}\mathbf{z}'_k + \frac{\xi_k^y}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}\mathbf{y}_{k+1} + \frac{\xi_k^x}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}\mathbf{x}_{k+1} \end{cases} \tag{17}$$

where $\xi_k^x = \frac{1}{\eta_k^x}$, $\xi_k^y = \frac{1}{\eta_k^y}$ and $\xi_k^z = \frac{1}{\eta_k^z}$. $\mathbf{u}_k = \frac{\mathbf{m}_k}{\mathbf{s}_k}$ in Win2-accelerated AdamW and Adam, $\mathbf{u}_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\mathbf{s}_k+\lambda_k\mathbf{x}_k\|_2}\left(\frac{\mathbf{m}_k}{\mathbf{s}_k}+\lambda_k\mathbf{x}_k\right)$ in Win2-accelerated LAMB.

For analysis, we further define

$$F_k(\mathbf{x}) = F(\mathbf{x}) + \frac{\lambda_k}{2}\|\mathbf{x}\|_{\mathbf{s}_k}^2 = \mathbb{E}_\zeta[f(\mathbf{x};\zeta)] + \frac{\lambda_k}{2}\|\mathbf{x}\|_{\mathbf{s}_k}^2, \tag{18}$$

where $\lambda_k = \lambda(1-\mu)^k$ in which $\mu = \frac{\beta_2 c_\infty^2}{\nu}$. In the following, we mainly use these notations to finish our proofs.

Appendix C. Auxiliary Lemmas

Before giving our analysis, we first provide some important lemmas for both Win and Win2.

Lemma 7 *For Win- and Win2-accelerated Adam, AdamW and LAMB, their* $\{(\mathbf{m}_k, \mathbf{s}_k)\}$ *satisfies*

$$\|\mathbf{m}_k\|_\infty \le c_\infty,\ \|\mathbf{v}_i+\nu\|_\infty \le c_\infty^2+\nu,\ \frac{\mu}{2} \le \left\|\frac{\mathbf{s}_k}{\mathbf{s}_{k+1}}\right\|_\infty < 1+\frac{\mu}{2},$$

where $\mu = \frac{\beta_2 c_\infty^2}{\nu}$*, and* $c_{s,\infty} \geq 0$ *can lower bound the values in* $\mathbf{u}_k$*,* i.e. $c_{s,\infty} = \min_i \sqrt{\mathbf{v}_{k,i}} \geq 0$ *in which* $\mathbf{v}_{k,i}$ *denotes the* i*-th entry in* $\mathbf{u}_k$*. Note, in the following proof, we directly use* $c_{s,\infty} = 0$ *to consider the worse case.*

See its proof in Appendix F.1.

C.1 Auxiliary Lemmas for Win

Before giving our analysis, we provide some important lemmas for Win.

Lemma 8 *Suppose the sequence* $\{\mathbf{x}_k, \mathbf{y}'_k, \mathbf{y}_k\}$ *are updated by Eqn.* (16)*. Then* $\{\mathbf{x}_k, \mathbf{y}'_k, \mathbf{y}_k\}$ *for Win-accelerated Adam, AdamW and LAMB satisfies*

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{y}_k)\|^2\right] \leq & (1-\beta_1)\mathbb{E}\left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\|^2\right] \\ & + \frac{(1-\beta_1)^2 L^2}{\beta_1}\mathbb{E}\left[\|\mathbf{y}_k - \mathbf{y}_{k-1}\|^2\right] + \frac{\beta_1^2\sigma^2}{b}.\end{aligned}$$

See its proof in Appendix F.2.

Lemma 9 *Assume* $\rho_{k+1}^y = \eta\tau_{k-1}^y\rho_k^y$*,* $\rho_1^y = 1$ *and* $\rho_0^y = 0$*. Then we have*

$$\begin{aligned}\phi(y) :=& \sum_{k=0}^{T-1}\rho_k^y(\tau_{k-1}^y)^2(1+\lambda_{k-1}\eta)^2\left[\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\|\mathbf{w}_i\|^2\right] \\ \leq& \frac{a^2\tau}{\eta(1-\eta\tau)^2}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right], \\ \psi(y) :=& \sum_{k=0}^{T-1}\tau_{k-1}^y\rho_k^y\left[\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\|\mathbf{w}_i\|^2\right] \leq \frac{1}{\eta(1-\eta\tau)^2}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right], \\ h(y) :=& \sum_{k=0}^{T-1}\rho_k^y\left[\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\|\mathbf{w}_i\|^2\right] \leq \frac{1}{\eta\tau(1-\eta\tau)^2}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right],\end{aligned} \tag{19}$$

where $a \leq \frac{1}{1-\mu}$*,* $\tau = \frac{1}{\eta+\eta^y}$*. Moreover, if* $\rho_{k+1}^z = \eta\tau_{k-1}^z\rho_k^z$*,* $\rho_1^z = 1$ *and* $\rho_0^z = 0$*, we also have*

$$h(z) := \sum_{k=0}^{T-1}\rho_k^z\left[\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^z(1-\eta\tau_{i-1}^z)(1+\lambda_i\eta)^2}\|\mathbf{w}_i\|^2\right] \leq \frac{1}{\eta\tau_z(1-\eta\tau_z)^2}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right], \tag{20}$$

where $\tau_z = \frac{1}{\eta+\eta^z}$*.*

See its proof in Appendix F.3

Lemma 10 *Suppose the sequence* $\{\mathbf{x}_k, \mathbf{y}'_k, \mathbf{y}_k\}$ *are updated by Eqn.* (16)*. By setting* $\eta_k^x = \eta^x$*,* $\eta_k^y = \eta^y$*,* $\xi_k^x = \xi^x := \frac{1}{\eta^x}$*,* $\xi_k^y = \xi^y := \frac{1}{\eta^y}$*, then* $\{\mathbf{x}_k, \mathbf{y}'_k, \mathbf{y}_k\}$ *for Win-accelerated Adam, AdamW and*

LAMB satisfies

$$\begin{aligned}\|\mathbf{y}_{k+1}-\mathbf{x}_{k+1}\|^2 \leq & \tau_k^y \rho_{k+1}^y \eta(\eta^y-\eta)^2 \sum_{i=0}^{k} \frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2 \\ \|\mathbf{y}_{k+1}-\mathbf{y}_k\|^2 \leq & \frac{2(\eta^y)^2}{(1+\lambda_k\eta)^2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|^2 \\ & + 2\rho_{k+1}^y(\eta^y)^2(\eta^y-\eta)^2(\tau_k^y)^2(1+\lambda_k\eta)^2 \sum_{i=0}^{k} \frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\end{aligned}$$

where $\rho_{k+1}^y = \eta^x\tau_{k-1}^y\rho_k^y$, $\rho_1^y = 1$ *and* $\rho_0^y = 0$; $\tau_k^y = \frac{1}{\eta^x+\eta^y+\lambda_k\eta^x\eta^y}$, $\delta_k^y = \frac{1}{\xi^x+\xi^y+\lambda_k}$, $\xi^x = \frac{1}{\eta^x}$, $\xi^y = \frac{1}{\eta^y}$. *Here,* $\mathbf{w}_k := \mathbf{m}_k + \lambda_k\mathbf{x}_k \odot \mathbf{s}_k$ *in Win-accelerated AdamW and Adam, and* $\mathbf{w}_k := \alpha_k\mathbf{m}_k + (1+\alpha_k)\lambda_k\mathbf{x}_k \odot \mathbf{s}_k$*in Win-accelerated LAMB, where* $\alpha_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\sqrt{\mathbf{v}_k+\nu}+\lambda_k\mathbf{x}_k\|_2}$.

See its proof in Appendix F.4.

Lemma 11 *Suppose the sequence* $\{\mathbf{x}_k, \mathbf{y}_k', \mathbf{y}_k\}$ *are updated by Eqn.* (16). *By setting* $\eta_k^x = \eta^x$, $\eta_k^y = \eta^y$, $\xi_k^x = \xi^x := \frac{1}{\eta^x}$, $\xi_k^y = \xi^y := \frac{1}{\eta^y}$, $\beta_{1,k} = \beta_1$ *and* $\beta_{2,k} = \beta_2$, *then* $\{\mathbf{x}_k, \mathbf{y}_k', \mathbf{y}_k\}$ *for Win2-accelerated Adam, AdamW and LAMB satisfies*

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\|^2\right] \leq & 2(1-\beta_1)\mathbb{E}\left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\|^2\right] \\ & + \frac{2\Pi_{1,k}^y(1-\beta_1)^2L^2}{\beta_1} + \frac{2\beta_1^2\sigma^2}{b} + 2L\Pi_{2,k}^y,\end{aligned}$$

where

$$\begin{aligned}\Pi_{1,k}^y := & \frac{2(\eta^y)^2}{(1+\lambda_{k-1}\eta)^2}\left\|\frac{\mathbf{w}_{k-1}}{\mathbf{s}_{k-1}}\right\|^2 \\ & + 2\rho_k^y(\eta^y)^2(\eta^y-\eta)^2(\tau_{k-1}^y)^2(1+\lambda_{k-1}\eta)^2 \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2 \\ \Pi_{2,k}^y := & \tau_{k-1}^y\rho_k^y\eta(\eta^y-\eta)^2 \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\end{aligned}$$

in which $\rho_{k+1}^y = \eta^x\tau_{k-1}^y\rho_k^y$, $\rho_1^y = 1$ *and* $\rho_0^y = 0$; $\tau_k^y = \frac{1}{\eta^x+\eta^y+\lambda_k\eta^x\eta^y}$, $\delta_k^y = \frac{1}{\xi^x+\xi^y+\lambda_k}$, $\xi^x = \frac{1}{\eta^x}$, $\xi^y = \frac{1}{\eta^y}$. *Here,* $\mathbf{w}_k := \mathbf{m}_k + \lambda_k\mathbf{x}_k \odot \mathbf{s}_k$ *in Win-accelerated AdamW and Adam, and* $\mathbf{w}_k := \alpha_k\mathbf{m}_k + (1+\alpha_k)\lambda_k\mathbf{x}_k \odot \mathbf{s}_k$*in Win-accelerated LAMB, where* $\alpha_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\sqrt{\mathbf{v}_k+\nu}+\lambda_k\mathbf{x}_k\|_2}$.

see its proof in Appendix F.8.

C.2 Auxiliary Lemmas for Win2

Then we provide some important lemmas for Win2.

Lemma 12 *Suppose the sequence $\{\mathbf{x}_k, \mathbf{y}'_k, \mathbf{y}_k, \mathbf{z}'_k, \mathbf{z}_k\}$ are updated by Eqn. (17). Then $\{\mathbf{x}_k, \mathbf{y}'_k, \mathbf{y}_k, \mathbf{z}'_k, \mathbf{z}_k\}$ for Win2-accelerated Adam, AdamW and LAMB satisfies*

$$\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{z}_k)\|^2\right] \leq (1-\beta_1)\mathbb{E}\left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{z}_{k-1})\|^2\right] + \frac{(1-\beta_1)^2 L^2}{\beta_1}\mathbb{E}\left[\|\mathbf{z}_k - \mathbf{z}_{k-1}\|^2\right] + \frac{\beta_1^2\sigma^2}{b}.$$

See its proof in Appendix F.6.

Lemma 13 *Suppose the sequence $\{\mathbf{x}_k, \mathbf{y}'_k, \mathbf{y}_k, \mathbf{z}'_k, \mathbf{z}_k\}$ are updated by Eqn. (17). By setting $\eta_k^x = \eta^x$, $\eta_k^y = \eta^y$, $\eta_k^z = \eta^z$, $\xi_k^x = \xi^x := \frac{1}{\eta^x}$, $\xi_k^y = \xi^y := \frac{1}{\eta^y}$, $\xi_k^z = \xi^z := \frac{1}{\eta^z}$, then $\{\mathbf{x}_k, \mathbf{y}'_k, \mathbf{y}_k, \mathbf{z}'_k, \mathbf{z}_k\}$ for Win2-accelerated Adam, AdamW and LAMB satisfies*

$$\begin{aligned}
\left\|\mathbf{y}'_{k+1} - (1+\lambda_k\eta^y)\mathbf{x}_{k+1}\right\|^2 \leq& \rho_{k+1}^y(\eta^y-\eta^x)^2\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2,\\
\|\mathbf{y}_k - \mathbf{x}_k\|^2 \leq& \tau_{k-1}^y\rho_k^y\eta(\eta^y-\eta)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\
\left\|\mathbf{z}'_{k+1} - (1+\lambda_k\eta^z)\mathbf{x}_{k+1}\right\|^2 \leq& \rho_{k+1}^z(\eta^z-\eta^x)^2\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2,
\end{aligned}$$

where $\rho_{k+1}^y = \eta^x\tau_{k-1}^y\rho_k^y$, $\rho_1^y = 1$ and $\rho_0^y = 0$; $\rho_{k+1}^z = \eta^x\tau_{k-1}^z\rho_k^z$, $\rho_1^z = 1$ and $\rho_0^z = 0$; $\tau_k^y = \frac{1}{\eta^x+\eta^y+\lambda_k\eta^x\eta^y}$, $\tau_k^z = \frac{1}{\eta^x+\eta^z+\lambda_k\eta^x\eta^z}$, $\delta_k^y = \frac{1}{\xi^x+\xi^y+\lambda_k}$, $\delta_k^z = \frac{1}{\xi^x+\xi^y+\xi^z+\lambda_k}$. Here, $\mathbf{w}_k := \mathbf{m}_k + \lambda_k\mathbf{x}_k \odot \mathbf{s}_k$ in Win2-accelerated AdamW and Adam, and $\mathbf{w}_k := \alpha_k\mathbf{m}_k + (1+\alpha_k)\lambda_k\mathbf{x}_k \odot \mathbf{s}_k$ in Win2-accelerated LAMB, where $\alpha_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\sqrt{\mathbf{v}_k+\nu}+\lambda_k\mathbf{x}_k\|_2}$. Moreover,

$$\begin{aligned}
\|\mathbf{z}_{k+1} - \mathbf{x}_{k+1}\|^2 \leq& 2(\xi^z\delta_k^z)^2\rho_{k+1}^z(\eta^z-\eta^x)^2\mathbb{E}\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\
&+ 2(\xi^y)^4(\delta_k^z\delta_k^y)^2\rho_{k+1}^y(\eta^y-\eta^x)^2\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2.\\
\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 \leq& \frac{3(\xi^x)^2}{(\xi^z)^2(\lambda_k+\xi^x)^2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|^2\\
&+ 3(\xi^x+\xi^y+\lambda_k)^2(\delta_k^z)^2\rho_{k+1}^z(\eta^z-\eta^x)^2\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\
&+ 3(\xi^y)^4(\delta_k^y)^2(\delta_k^z)^2\rho_{k+1}^y(\eta^y-\eta^x)^2\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2.
\end{aligned}$$

See its proof in Appendix F.7.

Lemma 14 *Suppose the sequence $\{\mathbf{x}_k, \mathbf{y}'_k, \mathbf{y}_k, \mathbf{z}'_k, \mathbf{z}_k\}$ are updated by Eqn. (17). By setting $\eta_k^x = \eta^x$, $\eta_k^y = \eta^y$, $\eta_k^z = \eta^z$, $\xi_k^x = \xi^x := \frac{1}{\eta^x}$, $\xi_k^y = \xi^y := \frac{1}{\eta^y}$, $\xi_k^z = \xi^z := \frac{1}{\eta^z}$, $\beta_{1,k} = \beta_1$ and $\beta_{2,k} = \beta_2$, then*

$\{\mathbf{x}_k, \mathbf{y}'_k, \mathbf{y}_k, \mathbf{z}'_k, \mathbf{z}_k\}$ for Win2-accelerated Adam, AdamW and LAMB satisfies

$$\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\|^2\right] \leq 2(1-\beta_{1,k})\mathbb{E}\left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{z}_{k-1})\|^2\right] + \frac{2\Pi_{1,k}^z(1-\beta_{1,k})^2L^2}{\beta_{1,k}} + \frac{2\beta_{1,k}^2\sigma^2}{b} + 2L\Pi_{2,k}^z,$$

where $\mathbf{w}_k := \mathbf{m}_k + \lambda_k\mathbf{x}_k \odot \mathbf{s}_k$ in Win-accelerated AdamW and Adam, $\mathbf{w}_k := \alpha_k\mathbf{m}_k + (1+\alpha_k)\lambda_k\mathbf{x}_k \odot \mathbf{s}_k$in Win-accelerated LAMB with $\alpha_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\sqrt{\mathbf{v}_k+\nu}+\lambda_k\mathbf{x}_k\|_2}$, and

$$\begin{aligned}
\Pi_{1,k}^z :=& \frac{3(\xi^x)^2}{(\xi^z)^2(\lambda_{k-1}+\xi^x)^2}\left\|\frac{\mathbf{w}_{k-1}}{\mathbf{s}_{k-1}}\right\|^2 \\
&+ 3(\xi^x+\xi^y+\lambda_{k-1})^2(\delta_{k-1}^z)^2\rho_k^z(\eta^z-\eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2 \\
&+ 3(\xi^y)^4(\delta_{k-1}^y)^2(\delta_{k-1}^z)^2\rho_k^y(\eta^y-\eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2, \\
\Pi_{2,k}^z :=& 2(\xi^z\delta_{k-1}^z)^2\rho_k^z(\eta^z-\eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2 \\
&+ 2(\xi^y)^4(\delta_{k-1}^z\delta_{k-1}^y)^2\rho_k^y(\eta^y-\eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2.
\end{aligned}$$

where $\rho_{k+1}^y = \eta^x\tau_{k-1}^y\rho_k^y$, $\rho_1^y = 1$ and $\rho_0^y = 0$; $\rho_{k+1}^z = \eta^x\tau_{k-1}^z\rho_k^z$, $\rho_1^z = 1$ and $\rho_0^z = 0$; $\tau_k^y = \frac{1}{\eta^x+\eta^y+\lambda_k\eta^x\eta^y}$, $\tau_k^z = \frac{1}{\eta^x+\eta^z+\lambda_k\eta^x\eta^z}$, $\delta_k^y = \frac{1}{\xi^x+\xi^y+\lambda_k}$, $\delta_k^z = \frac{1}{\xi^x+\xi^y+\xi^z+\lambda_k}$. Here, $\mathbf{w}_k := \mathbf{m}_k + \lambda_k\mathbf{x}_k \odot \mathbf{s}_k$ in Win2-accelerated AdamW and Adam, and $\mathbf{w}_k := \alpha_k\mathbf{m}_k + (1+\alpha_k)\lambda_k\mathbf{x}_k \odot \mathbf{s}_k$in Win2-accelerated LAMB, where $\alpha_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\sqrt{\mathbf{v}_k+\nu}+\lambda_k\mathbf{x}_k\|_2}$.

See its proof in Appendix F.8.

Appendix D. Proofs of Main Results in Sec. 3

Here we provide proofs of the main results in Sec. 3, including Theorem 1, 2 and 3.

D.1 Proof of Theorem 1

Proof Recall our definition $F_k(\mathbf{y}_k) = F(\mathbf{z}) + \frac{\lambda_k}{2}\|\mathbf{z}\|_{\mathbf{s}_k}^2 = \mathbb{E}_\zeta[f(\mathbf{z};\zeta)] + \frac{\lambda_k}{2}\|\mathbf{z}\|_{\mathbf{s}_k}^2$, in the (18). By using the smoothness of $f(\theta;\zeta)$, we can obtain

$$
\begin{aligned}
&F_{k+1}(\mathbf{x}_{k+1})\\
\leq& F(\mathbf{x}_k) + \langle \nabla F(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k\rangle + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + \frac{\lambda_{k+1}}{2}\|\mathbf{x}_{k+1}\|_{\mathbf{s}_{k+1}}^2\\
\overset{①}{\leq}& F(\mathbf{x}_k) + \langle \nabla F(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k\rangle + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + \frac{\lambda_{k+1}}{2(1-\mu)}\|\mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2\\
\overset{②}{\leq}& F(\mathbf{x}_k) + \frac{\lambda_k}{2}\|\mathbf{x}_k\|_{\mathbf{s}_k}^2 + \langle \nabla F(\mathbf{x}_k) + \lambda_k \mathbf{x}_k \odot \mathbf{s}_k, \mathbf{x}_{k+1} - \mathbf{x}_k\rangle + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + \frac{\lambda_k}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_{\mathbf{s}_k}^2\\
\overset{③}{\leq}& F_k(\mathbf{x}_k) - \frac{\eta_k^x}{1+\lambda_k\eta_k^x}\left\langle \nabla F(\mathbf{x}_k) + \lambda_k \mathbf{x}_k \odot \mathbf{s}_k, \frac{\mathbf{w}_k}{\mathbf{s}_k}\right\rangle + \frac{L(\eta_k^x)^2}{2(1+\lambda_k\eta_k^x)^2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|^2 + \frac{\lambda_k(\eta_k^x)^2}{2(1+\lambda_k\eta_k^x)^2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|_{\mathbf{s}_k}^2
\end{aligned}
$$

where ① holds since Lemma 7 proves $\left\|\frac{\mathbf{s}_k}{\mathbf{s}_{k+1}}\right\|_\infty \in [1-\mu, 1+\mu]$ $(\forall p \in [0,1])$ in which $\mu = \frac{\beta_2 c_\infty^2}{\nu}$;
② holds because $\lambda_k = \frac{\lambda_{k+1}}{1-\mu}$ and

$$\|\mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2 = \|\mathbf{x}_k\|_{\mathbf{s}_k}^2 + \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_{\mathbf{s}_k}^2 + 2\langle \mathbf{x}_{k+1} - \mathbf{x}_k, \mathbf{x}_k\rangle_{\mathbf{s}_k}.$$

③ holds, since we have 1) $\mathbf{w}_k := \mathbf{m}_k + \lambda_k \mathbf{x}_k \odot \mathbf{s}_k$ and $\mathbf{u}_k = \frac{\mathbf{m}_k}{\mathbf{s}_k} = \frac{\mathbf{w}_k - \lambda_k \mathbf{x}_k \odot \mathbf{s}_k}{\mathbf{s}_k}$ in Win-accelerated AdamW and Adam; 2) $\mathbf{x}_{k+1} = \frac{1}{1+\lambda_k\eta_k^x}(\mathbf{x}_k - \eta_k^x \mathbf{u}_k) = \frac{1}{1+\lambda_k\eta_k^x}\left(\mathbf{x}_k - \eta_k^x \frac{\mathbf{w}_k - \lambda_k \mathbf{x}_k \odot \mathbf{s}_k}{\mathbf{s}_k}\right) = \mathbf{x}_k - \frac{\eta_k^x}{1+\lambda_k\eta_k^x}\frac{\mathbf{w}_k}{\mathbf{s}_k}$. In this way, we can obtain

$$
\begin{aligned}
&F_{k+1}(\mathbf{x}_{k+1})\\
=& F_k(\mathbf{x}_k) + \frac{1}{2}\left\|\sqrt{\frac{\eta_k^x}{(1+\lambda_k\eta_k^x)\mathbf{s}_k}}\left(\nabla F(\mathbf{x}_k) + \lambda_k \mathbf{x}_k \odot \mathbf{s}_k - \mathbf{w}_k\right)\right\|^2 - \frac{1}{2}\left\|\sqrt{\frac{\eta_k^x}{(1+\lambda_k\eta_k^x)\mathbf{s}_k}}\mathbf{w}_k\right\|^2\\
&- \frac{1}{2}\left\|\sqrt{\frac{\eta_k^x}{(1+\lambda_k\eta_k^x)\mathbf{s}_k}}\left(\nabla F(\mathbf{x}_k) + \lambda_k \mathbf{x}_k \odot \mathbf{s}_k\right)\right\|^2 + \frac{L(\eta_k^x)^2}{2(1+\lambda_k\eta_k^x)^2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|^2 + \frac{\lambda_k(\eta_k^x)^2}{2(1+\lambda_k\eta_k^x)^2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|_{\mathbf{s}_k}^2\\
\overset{④}{\leq}& F_k(\mathbf{x}_k) + \frac{\eta_k^x}{2c_1(1+\lambda_k\eta_k^x)}\|\nabla F(\mathbf{x}_k) - \mathbf{m}_k\|^2 - \frac{\eta_k^x}{2c_2(1+\lambda_k\eta_k^x)}\|\nabla F_k(\mathbf{x}_k)\|^2\\
&- \frac{\eta_k^x}{2c_2(1+\lambda_k\eta_k^x)}\left[1 - \frac{c_2 L\eta_k^x}{c_1^2(1+\lambda_k\eta_k^x)} - \frac{c_2\lambda_k\eta_k^x}{c_1(1+\lambda_k\eta_k^x)}\right]\|\mathbf{w}_k\|^2\\
\overset{⑤}{\leq}& F_k(\mathbf{x}_k) + \frac{\eta_k^x}{2c_1(1+\lambda_k\eta_k^x)}\|\nabla F(\mathbf{x}_k) - \mathbf{m}_k\|^2 - \frac{\eta_k^x}{2c_2(1+\lambda_k\eta_k^x)}\|\nabla F_k(\mathbf{x}_k)\|^2 - \frac{\eta_k^x}{4c_2(1+\lambda_k\eta_k^x)}\|\mathbf{w}_k\|^2,
\end{aligned}
\tag{21}
$$

④ holds, because

$$
\begin{aligned}
&\mathbf{w}_k := \mathbf{m}_k + \lambda_k \mathbf{x}_k \odot \mathbf{s}_k, \qquad \mathbf{x}_{k+1} - \mathbf{x}_k = -\frac{\eta_k^x}{1+\lambda_k\eta_k^x}\frac{\mathbf{m}_k + \lambda_k \mathbf{x}_k \odot \mathbf{s}_k}{\mathbf{s}_k} = -\frac{\eta_k^x}{1+\lambda_k\eta_k^x}\frac{\mathbf{w}_k}{\mathbf{s}_k},\\
&c_1 := \nu^{0.5} \leq \|\mathbf{s}_k\|_\infty \leq c_2 := (c_\infty^2 + \nu)^{0.5}.
\end{aligned}
$$

⑤ holds, since we set $\eta_k^x \leq \frac{c_1^2(1+\lambda_k\eta_k^x)}{2c_2(L+\lambda_k c_1)}$ such that $\frac{c_2 L\eta_k^x}{c_1^2(1+\lambda_k\eta_k^x)} + \frac{c_2\lambda_k\eta_k^x}{c_1(1+\lambda_k\eta_k^x)} \leq \frac{1}{2}$.

From Lemma 11, by setting $\eta_k^x = \eta$, $\eta_k^y = \eta^y = \gamma_1\eta$ and $\beta_{1,k} = \beta_1$, we have

$$\begin{aligned}\mathbb{E}\left[\left\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\right\|^2\right] \leq & 2(1-\beta_1)\mathbb{E}\left[\left\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\right\|^2\right] + \frac{2\Pi_{1,k}^y(1-\beta_1)^2L^2}{\beta_1} \\ & + \frac{2\beta_1^2\sigma^2}{b} + 2L\Pi_{2,k}^y,\end{aligned} \tag{22}$$

where

$$\begin{aligned}\Pi_{1,k}^y :=& \frac{2(\eta^y)^2}{(1+\lambda_{k-1}\eta)^2}\left\|\frac{\mathbf{w}_{k-1}}{\mathbf{s}_{k-1}}\right\|^2 + 2\rho_k^y(\eta^y)^2(\eta^y-\eta)^2(\tau_{k-1}^y)^2(1+\lambda_{k-1}\eta)^2 \\ & \cdot\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2, \\ \Pi_{2,k}^y :=& \tau_{k-1}^y\rho_k^y\eta(\eta^y-\eta)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2,\end{aligned} \tag{23}$$

Here $\rho_{k+1}^y = \eta\tau_{k-1}^y\rho_k^y$, $\rho_1^y = 1$ and $\rho_0^y = 0$. By considering $c_2 \geq \|\mathbf{s}_k\|_\infty \geq c_1$, we have

$$\begin{aligned}\Pi_{1,k}^y \leq \bar{\Pi}_{1,k}^y :=& \frac{2(\eta^y)^2}{c_1^2(1+\lambda_{k-1}\eta)^2}\left\|\mathbf{w}_{k-1}\right\|^2 \\ & + \frac{2\rho_k^y(\eta^y)^2(\eta^y-\eta)^2(\tau_{k-1}^y)^2(1+\lambda_{k-1}\eta)^2}{c_1^2}\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\mathbf{w}_i\right\|^2, \\ \Pi_{2,k}^y \leq \bar{\Pi}_{2,k}^y :=& \frac{\tau_{k-1}^y\rho_k^y\eta(\eta^y-\eta)^2}{c_1^2}\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\mathbf{w}_i\right\|^2,\end{aligned} \tag{24}$$

Therefore, by plugging the results in Eqn. (22) into the upper bound of $F_{k+1}(\mathbf{x}_{k+1})$, we have

$$\begin{aligned}& F_{k+1}(\mathbf{x}_{k+1}) \\ \leq & F_k(\mathbf{x}_k) - \frac{\eta}{2c_2(1+\lambda_k\eta)}\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 - \frac{\eta}{4c_2(1+\lambda_k\eta)}\left\|\mathbf{w}_k\right\|^2 \\ & + \frac{\eta(1-\beta_1)}{c_1(1+\lambda_k\eta)}\mathbb{E}\left[\left\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\right\|^2\right] + \frac{\eta\bar{\Pi}_{1,k}^y(1-\beta_1)^2L^2}{c_1\beta_1(1+\lambda_k\eta)} + \frac{\eta\beta_1^2\sigma^2}{c_1(1+\lambda_k\eta)b} + \frac{\eta L\bar{\Pi}_{2,k}^y}{c_1(1+\lambda_k\eta)} \\ \overset{①}{\leq} & F_k(\mathbf{x}_k) - \frac{\eta}{2c_2(1+\lambda_k\eta)}\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 - \frac{\eta}{4c_2(1+\lambda_k\eta)}\left\|\mathbf{w}_k\right\|^2 \\ & + \frac{\eta(1-\beta_1)}{c_1}\mathbb{E}\left[\left\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\right\|^2\right] + \frac{\eta\bar{\Pi}_{1,k}^y(1-\beta_1)^2L^2}{c_1\beta_1(1+\lambda_k\eta)} + \frac{\eta\beta_1^2\sigma^2}{c_1(1+\lambda_k\eta)b} + \frac{\eta L\bar{\Pi}_{2,k}^y}{c_1(1+\lambda_k\eta)},\end{aligned} \tag{25}$$

where ① uses the fact that $0 < \lambda_k \leq \lambda$. Then, by plugging $\|\mathbf{y}_k - \mathbf{y}_{k-1}\|^2 \leq \Pi_{1,k}^y \leq \bar{\Pi}_{1,k}^y$ in Lemma 10 into Lemma 8, we have

$$\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{y}_k)\|^2\right] \leq (1-\beta_1)\mathbb{E}\left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\|^2\right] + \frac{(1-\beta_1)^2 L^2 \bar{\Pi}_{1,k}^y}{\beta_1} + \frac{\beta_1^2\sigma^2}{b}. \tag{26}$$

Then we add Eqn. (25) and $\alpha\times$ (26) as follows:

$$\begin{aligned}
&F_{k+1}(\mathbf{x}_{k+1}) + \alpha\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{y}_k)\|^2\right] \\
\leq& F_k(\mathbf{x}_k) - \frac{\eta}{2c_2(1+\lambda_k\eta)}\|\nabla F_k(\mathbf{x}_k)\|^2 - \frac{\eta}{4c_2(1+\lambda_k\eta)}\|\mathbf{w}_k\|^2 \\
&+ (1-\beta_1)\left(\frac{\eta}{c_1} + \alpha\right)\mathbb{E}\left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\|^2\right] \\
&+ \frac{\eta\bar{\Pi}_{1,k}^y(1-\beta_1)^2L^2}{c_1\beta_1(1+\lambda_k\eta)} + \frac{\eta\beta_1^2\sigma^2}{c_1(1+\lambda_k\eta)b} + \frac{\eta L\bar{\Pi}_{2,k}^y}{c_1(1+\lambda_k\eta)} + \frac{\alpha(1-\beta_1)^2L^2\bar{\Pi}_{1,k}^y}{\beta_1} + \frac{\alpha\beta_1^2\sigma^2}{b}
\end{aligned} \tag{27}$$

Then by setting $\alpha = \frac{\eta(1-\beta_1)}{c_1\beta_1}$ and $G_{k+1}(\mathbf{x}_{k+1}) = F_{k+1}(\mathbf{x}_{k+1}) + \frac{\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\|^2\right] = \mathbb{E}_\zeta[f(\mathbf{z};\zeta)] + \frac{\lambda_k}{2}\|\mathbf{z}\|_{\mathbf{s}_k}^2 + \frac{\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\|^2\right]$, we can obtain

$$\begin{aligned}
&G_{k+1}(\mathbf{x}_{k+1}) \\
\leq& G_k(\mathbf{x}_k) - \frac{\eta}{2c_2(1+\lambda_k\eta)}\|\nabla F_k(\mathbf{x}_k)\|^2 - \frac{\eta}{4c_2(1+\lambda_k\eta)}\|\mathbf{w}_k\|^2 \\
&+ \frac{\eta\bar{\Pi}_{1,k}^y(1-\beta_1)^2L^2}{c_1\beta_1(1+\lambda_k\eta)} + \frac{\eta\beta_1^2\sigma^2}{c_1(1+\lambda_k\eta)b} + \frac{\eta L\bar{\Pi}_{2,k}^y}{c_1(1+\lambda_k\eta)} + \frac{\eta(1-\beta_1)^3L^2\bar{\Pi}_{1,k}^y}{c_1\beta_1^2} + \frac{\eta(1-\beta_1)\beta_1\sigma^2}{c_1 b} \\
\overset{①}{\leq}& G_k(\mathbf{x}_k) - \frac{\eta}{2c_2(1+\lambda_k\eta)}\|\nabla F_k(\mathbf{x}_k)\|^2 - \frac{\eta}{4c_2(1+\lambda_k\eta)}\|\mathbf{w}_k\|^2 + \frac{\eta(1-\beta_1)^2L^2\bar{\Pi}_{1,k}^y}{c_1\beta_1^2} \\
&+ \frac{\eta L\bar{\Pi}_{2,k}^y}{c_1(1+\lambda_k\eta)} + \frac{\eta\beta_1\sigma^2}{c_1 b},
\end{aligned}$$

where ① uses the fact that $0 < \lambda_k \leq \lambda$. Then summing the above inequality from $k=0$ to $k=T-1$ and using $0 < \lambda_k \leq \lambda$ give

$$\begin{aligned}
&\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\nabla F_k(\mathbf{x}_k)\|^2 + \frac{1}{2}\|\mathbf{w}_k\|^2\right] \\
\leq& \frac{2c_2(1+\lambda\eta)}{\eta T}\left[G(\mathbf{x}_0) - G(\mathbf{x}_T)\right] + \frac{2c_2\beta_1\sigma^2(1+\lambda\eta)}{c_1 bT} + \frac{2c_2\beta_1^2\sigma^2}{c_1 b} \\
&+ \frac{2c_2(1-\beta_1)^2L^2(1+\lambda\eta)}{c_1\beta_1^2 T}\sum_{k=0}^{T-1}\bar{\Pi}_{1,k}^y + \frac{2c_2L}{c_1 T}\sum_{k=0}^{T-1}\bar{\Pi}_{2,k}^y \\
\leq& \frac{2c_2(1+\lambda\eta)\Delta}{\eta T} + \frac{2c_2\beta_1\sigma^2(1+\lambda\eta)}{c_1 b} + \frac{2c_2(1-\beta_1)^2L^2(1+\lambda\eta)}{c_1\beta_1^2 T}\sum_{k=0}^{T-1}\bar{\Pi}_{1,k}^y + \frac{2c_2L}{c_1 T}\sum_{k=0}^{T-1}\bar{\Pi}_{2,k}^y
\end{aligned}$$

where

$$
\begin{aligned}
&G(\mathbf{x}_0) - G(\mathbf{x}_T) \\
=&F_0(\mathbf{x}_0) + \frac{\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\left\|\mathbf{m}_{-1} - \nabla F(\mathbf{x}_{-1})\right\|^2\right] - F_T(\mathbf{x}_T) - \frac{\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\left\|\mathbf{m}_{T-1} - \nabla F(\mathbf{x}_{T-1})\right\|^2\right] \\
=&F(\mathbf{x}_0) + \lambda_0\left\|\mathbf{x}_0\right\|_{\mathbf{s}_0} - F(\mathbf{x}_T) - \lambda_T\left\|\mathbf{x}_T\right\|_{\mathbf{s}_T} - \frac{\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\left\|\mathbf{m}_{T-1} - \nabla F(\mathbf{x}_{T-1})\right\|^2\right] \\
\leq&F(\mathbf{x}_0) - F(\mathbf{x}_T) \leq \Delta
\end{aligned}
$$

where $\Delta = F(\mathbf{x}_0) - F(\mathbf{x}_\star)$; $\mathbf{x}_{-1}$ and $\mathbf{m}_{-1}$ are two virtual points which satisfy $\mathbf{m}_{-1} = \nabla F(\mathbf{x}_{-1})$. Now we try to bound $\sum_{k=0}^{T-1}\bar{\Pi}_{1,k}^y$ and $\sum_{k=0}^{T-1}\bar{\Pi}_{2,k}^y$. Firstly, we have

$$
\begin{aligned}
\sum_{k=0}^{T-1}\bar{\Pi}_k =& \sum_{k=0}^{T-1}\left[\frac{2(\eta^y)^2}{c_1^2(1+\lambda_{k-1}\eta)^2}\left\|\mathbf{w}_{k-1}\right\|^2\right. \\
&\left.+\frac{2\rho_k^y(\eta^y)^2(\eta^y-\eta)^2(\tau_{k-1}^y)^2(1+\lambda_{k-1}\eta)^2}{c_1^2}\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\mathbf{w}_i\right\|^2\right] \\
\overset{①}{\leq}&\frac{2(\eta^y)^2}{c_1^2}\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_{k-1}\right\|^2\right] + \frac{2(\eta^y)^2(\eta^y-\eta)^2}{c_1^2}\sum_{k=0}^{T-1}\rho_k^y(\tau_{k-1}^y)^2(1+\lambda_{k-1}\eta)^2 \\
&\left[\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\mathbf{w}_i\right\|^2\right] \\
\overset{②}{\leq}&\frac{2(\eta^y)^2}{c_1^2}\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_{k-1}\right\|^2\right] + \frac{2a^2(\eta^y)^2(\eta^y-\eta)^2\tau}{c_1^2\eta(1-\eta\tau)^2}\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_k\right\|^2\right] \\
\overset{③}{\leq}&\frac{2\gamma^2\eta^2}{c_1^2}\left[1+a^2(\gamma-1)^2/\gamma\right]\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_{k-1}\right\|^2\right] \overset{④}{\leq} \frac{8\gamma^3\eta^2}{c_1^2}\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_{k-1}\right\|^2\right],
\end{aligned}
\tag{28}
$$

where ① holds since $0 \leq \lambda_k \leq \lambda$; ② holds, since in Lemma 9, we prove $\sum_{k=0}^{T-1}\rho_k^y(\tau_{k-1}^y)^2(1+\lambda_{k-1}\eta)^2\left[\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\mathbf{w}_i\right\|^2\right] \leq \frac{a^2\tau}{\eta(1-\eta\tau)^2}\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_k\right\|^2\right]$, where $a \leq \frac{1}{1-\mu}$; ④ holds by setting $\eta^y = \gamma\eta$; ④ holds since $1 + a^2(\gamma-1)^2/\gamma \leq a^2\gamma \leq 4\gamma$ where we set $\mu \in (0, 0.5)$ which is consistant the practical setting $\mu = 10^{-8}$.

Similarly, we can bound

$$
\begin{aligned}
\sum_{k=0}^{T-1}\bar{\Pi}_k' =& \sum_{k=0}^{T-1}\frac{\tau_{k-1}^y\rho_k^y\eta(\eta^y-\eta)^2}{c_1^2}\left[\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\mathbf{w}_i\right\|^2\right] \\
\overset{①}{\leq}&\frac{(\eta^y-\eta)^2}{c_1^2(1-\eta\tau)^2}\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_k\right\|^2\right] \leq \frac{\eta^2\gamma^2(\gamma-1)^2}{c_1^2(1+\gamma)^2}\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_k\right\|^2\right] \leq \frac{\eta^2(\gamma-1)^2}{c_1^2}\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_k\right\|^2\right]
\end{aligned}
\tag{29}
$$

where ① holds since in Lemma 9, we prove $\sum_{k=0}^{T-1}\tau_{k-1}^{y}\rho_k^{y}\left[\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^{y}(1-\eta\tau_{i-1}^{y})(1+\lambda_i\eta)^2}\left\|\mathbf{w}_i\right\|^2\right]\leq \frac{1}{\eta(1-\eta\tau)^2}\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_k\right\|^2\right]$. Therefore, we have

$$
\begin{aligned}
&\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\nabla F_k(\mathbf{x}_k)\right\|^2+\frac{1}{2}\left\|\mathbf{w}_k\right\|^2\right]\\
\leq&\frac{2c_2(1+\lambda\eta)\Delta}{\eta T}+\frac{2c_2\beta_1\sigma^2(1+\lambda\eta)}{c_1 b}+\frac{2c_2\eta^2L(\gamma-1)^2}{c_1^3T}\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_k\right\|^2\right]\\
&\qquad\qquad+\frac{16c_2\gamma^3\eta^2(1-\beta_1)^2L^2(1+\lambda\eta)}{c_1^3\beta_1^2T}\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_{k-1}\right\|^2\right]\\
\overset{①}{\leq}&\frac{2c_2(1+\lambda\eta)\Delta}{\eta T}+\frac{2c_2\beta_1\sigma^2(1+\lambda\eta)}{c_1 b}+\frac{1}{4T}\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_k\right\|^2\right]
\end{aligned}
$$

where ① holds since we choose proper η and β_1 such that

$$
\frac{16c_2\gamma^3\eta^2(1-\beta_1)^2L^2(1+\lambda\eta)}{c_1^3\beta_1^2}\leq\frac{1}{8},\qquad\qquad\frac{2c_2\eta^2L(\gamma-1)^2}{c_1^3}\leq\frac{1}{8}\tag{30}
$$

Now we select η and β_1 such that (30) holds:

$$
\eta\leq\min\left(\frac{c_1^{1.5}\beta_1}{8\sqrt{2}c_2^{0.5}\gamma^{1.5}(1-\beta_1)L(1+\lambda\eta)^{0.5}},\frac{c_1^{1.5}}{4c_2^{0.5}L^{0.5}(\gamma-1)}\right)
$$

So we arrive at

$$
\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\nabla F_k(\mathbf{x}_k)\right\|^2+\frac{1}{4}\left\|\mathbf{w}_k\right\|^2\right]\leq\frac{2c_2(1+\lambda\eta)\Delta}{\eta T}+\frac{2c_2\beta_1(1+\lambda\eta)\sigma^2}{c_1 b}\overset{①}{\leq}\epsilon^2,\tag{31}
$$

where we set $T\geq\frac{4c_2(1+\lambda\eta)\Delta}{\eta\epsilon^2}$ and $\beta_1\leq\frac{c_1 b\epsilon^2}{4c_2(1+\lambda\eta)\sigma^2}$. This result directly bounds

$$
\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{s}_k\odot(\mathbf{x}_k-\mathbf{x}_{k+1})\right\|^2=\frac{\eta^2}{T}\sum_{k=0}^{T-1}\frac{1}{(1+\lambda_k\eta)^2}\left\|\mathbf{m}_k+\lambda\mathbf{x}_k\odot\mathbf{s}_k\right\|^2\leq\frac{\eta^2}{T}\sum_{k=0}^{T-1}\left\|\mathbf{w}_k\right\|^2\leq\eta^2\epsilon^2.
$$

Moreover, from Lemma 10, we have

$$\begin{aligned}
\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{y}_k'-(1+\lambda_{k-1}\eta^y)\mathbf{x}_k\right\|^2 \overset{①}{\leq}& \frac{1}{T}\sum_{k=0}^{T-1}\rho_k^y(\eta^y-\eta)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\
\overset{③}{=}& \frac{1}{T}\sum_{k=0}^{T-1}\Pi_{3,k}^y,\\
\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{y}_k-\mathbf{x}_k\right\|^2 \overset{①}{\leq}& \frac{1}{T}\sum_{k=0}^{T-1}\tau_{k-1}^y\rho_k^y\eta(\eta^y-\eta)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\
\overset{②}{=}& \frac{1}{T}\sum_{k=0}^{T-1}\Pi_{2,k}^y,\\
\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{y}_{k+1}-\mathbf{y}_k\right\|^2 \overset{①}{\leq}& \frac{1}{T}\sum_{k=0}^{T-1}\left[\frac{2(\eta^y)^2}{(1+\lambda_k\eta)^2}+2\rho_{k+1}^y(\eta^y)^2(\eta^y-\eta)^2(\tau_k^y)^2(1+\lambda_k\eta)^2\right.\\
&\left.\cdot\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\right]\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|^2\\
\overset{②}{\leq}& \frac{1}{T}\sum_{k=0}^{T-1}\Pi_{1,k}^y
\end{aligned}$$

where $\rho_{k+1}^y=\eta\tau_{k-1}^y\rho_k^y$, $\rho_1^y=1$ and $\rho_0^y=0$. ① holds by using Lemma 10; ② holds by using the definition in Eqn. (23); ③ holds by defining:

$$\Pi_{3,k}^y:=\rho_k^y(\eta^y-\eta)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2.$$

Now remaining task is to upper bound $\frac{1}{T}\sum_{k=0}^{T-1}\Pi_{3,k}^y$, $\frac{1}{T}\sum_{k=0}^{T-1}\Pi_{2,k}^y$ and $\frac{1}{T}\sum_{k=0}^{T-1}\Pi_{1,k}^y$. Here we first bound $\frac{1}{T}\sum_{k=0}^{T-1}\Pi_{3,k}^y$ by using almost the same proof in Eqn. (29):

$$\begin{aligned}
\frac{1}{T}\sum_{k=0}^{T-1}\Pi_{3,k}^y \overset{①}{\leq}& \frac{(\eta^y-\eta)^2}{c_1^2\eta\tau(1-\eta\tau)^2T}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right]\leq\frac{\eta^2\gamma^2(\gamma-1)^2}{c_1^2(1+\gamma)T}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right]\\
\leq& \frac{\eta^2\gamma(\gamma-1)^2}{c_1^2T}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right]\overset{②}{\leq}\frac{4\eta^2\gamma^3\epsilon^2}{c_1^2}
\end{aligned}\tag{32}$$

where ① holds since in Lemma 9 we have prove $\sum_{k=0}^{T-1}\rho_k^y\left[\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\|\mathbf{w}_i\|^2\right]\leq\frac{1}{\eta\tau(1-\eta\tau)^2}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right]$; ② holds by using $\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|\mathbf{w}_k\|^2\leq 4\epsilon^2$ in Eqn. (31).

From the bound in Eqn. (24) and the following bound on $\frac{1}{T}\sum_{k=0}^{T-1}\bar{\Pi}_k$ and $\frac{1}{T}\sum_{k=0}^{T-1}\bar{\Pi}_k'$, we have

$$\begin{aligned}
\frac{1}{T}\sum_{k=0}^{T-1}\Pi_{1,k}^y\leq\frac{1}{T}\sum_{k=0}^{T-1}\bar{\Pi}_k\leq&\frac{2a^2\gamma^3\eta^2}{c_1^2T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{w}_k\|^2\right]\overset{①}{\leq}\frac{32\eta^2\gamma^3\epsilon^2}{c_1^2},\\
\frac{1}{T}\sum_{k=0}^{T-1}\Pi_{2,k}^y\leq\frac{1}{T}\sum_{k=0}^{T-1}\bar{\Pi}_k'\leq&\frac{\eta^2(\gamma-1)^2}{c_1^2T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{w}_k\|^2\right]\overset{①}{\leq}\frac{4\eta^2\gamma^2\epsilon^2}{c_1^2}
\end{aligned}$$

where ① holds, since 1) $\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|\mathbf{w}_k\|^2 \leq 4\epsilon^2$. Therefore, we have

$$\begin{aligned}
&\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left\|\mathbf{y}'_k-(1+\lambda_k\eta^y)\mathbf{x}_k\right\|^2 \leq \frac{4\eta^2\gamma^3\epsilon^2}{c_1^2}, \quad \frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left\|\mathbf{y}_k-\mathbf{x}_k\right\|^2 \leq \frac{4\eta^2\gamma^2\epsilon^2}{c_1^2},\\
&\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left\|\mathbf{y}_{k+1}-\mathbf{y}_k\right\|^2 \leq \frac{32\eta^2\gamma^3\epsilon^2}{c_1^2}.
\end{aligned} \tag{33}$$

Besides, we have

$$\begin{aligned}
\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\mathbf{m}_k-\nabla F(\mathbf{x}_k)\right\|^2\right] \leq& \frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\mathbf{m}_k+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k-\nabla F(\mathbf{x}_k)-\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\right\|^2\right]\\
\leq& \frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\mathbf{m}_k+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\right\|^2+\left\|\nabla F(\mathbf{x}_k)+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\right\|^2\right]\\
=& \frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\mathbf{m}_k+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\right\|^2+\left\|\nabla F_k(\mathbf{x}_k)\right\|^2\right]\\
\overset{①}{\leq}& 2\left[\epsilon^2+\frac{3}{4}\times 4\epsilon^2\right] \leq 8\epsilon^2.
\end{aligned}$$

where in ① we use $\mathbf{w}_k = \mathbf{m}_k + \lambda_k\mathbf{x}_k\odot\mathbf{s}_k$. In this way, we have

$$\begin{aligned}
\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\mathbf{m}_k-\nabla F(\mathbf{y}_k)\right\|^2\right] \leq& \frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\mathbf{m}_k-\nabla F(\mathbf{x}_k)\right\|^2+\left\|\nabla F(\mathbf{x}_k)-\nabla F(\mathbf{y}_k)\right\|^2\right]\\
\leq& 16\epsilon^2+\frac{2L^2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\mathbf{x}_k-\mathbf{y}_k\right\|^2\right] \leq 16\epsilon^2+\frac{8\eta^2\gamma^2L^2\epsilon^2}{c_1^2}.
\end{aligned} \tag{34}$$

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

$$\beta_1 \leq \frac{c_1 b\epsilon^2}{4c_2(1+\lambda\eta)\sigma^2} = \mathcal{O}\left(\frac{c_1 b\epsilon^2}{c_2\sigma^2}\right),$$

where $c_1 = \nu^{0.5} \leq \|\mathbf{s}_k\|_\infty \leq \left(c_\infty^2+\nu\right)^{0.5} = c_2$.

For η, it should satisfy

$$\eta \leq \min\left(\frac{c_1^{1.5}\beta_1}{8\sqrt{2}c_2^{0.5}\gamma^{1.5}(1-\beta_1)L(1+\lambda\eta)^{0.5}}, \frac{c_1^{1.5}}{4c_2^{0.5}L^{0.5}(\gamma-1)}, \frac{c_1^2(1+\lambda\eta)}{2c_2(L+\lambda c_1)}\right)$$

Considering $\lambda\eta << 1$, $\frac{1+\lambda\eta}{1+(1-\mu)\lambda\eta} = a \leq \frac{1}{1-\mu}$, μ is a constant, and $c_1 = \nu^{0.5} << 1$, then we have

$$\eta \leq \mathcal{O}\left(\min\left(\frac{c_1^{1.5}\beta_1}{c_2^{0.5}\gamma^{1.5}L}, \frac{c_1^{1.5}}{c_2^{0.5}\gamma L^{0.5}}, \frac{c_1^2}{c_2 L}\right)\right) = \mathcal{O}\left(\frac{c_1^{2.5}b\epsilon^2}{c_2^{1.5}\gamma^{1.5}\sigma^2 L}\right)$$

where ν is often much smaller than one, and β_1 is very small. For T, we have

$$T \geq \frac{4c_2(1+\lambda\eta)\Delta}{\eta\epsilon^2} = \mathcal{O}\left(\frac{c_2\Delta}{\epsilon^2}\frac{c_2^{1.5}\gamma^{1.5}\sigma^2 L}{c_1^{2.5}b\epsilon^2}\right) = \mathcal{O}\left(\frac{c_2^{2.5}\gamma^{1.5}\sigma^2 L\Delta}{c_1^{2.5}b\epsilon^4}\right) = \mathcal{O}\left(\frac{c_2^{2.5}\gamma^{1.5}\sigma^2 L\Delta}{\nu^{1.25}b\epsilon^4}\right).$$

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

$$\mathcal{O}(Tb) = \mathcal{O}\left(\frac{c_2^{2.5}\gamma^{1.5}\sigma^2 L\Delta}{\nu^{1.25}\epsilon^4}\right).$$

The proof is completed. ■

D.2 Proof of Theorem 2

Proof Recall our definition $F_k(\mathbf{y}_k) = F(\mathbf{z}) + \frac{\lambda_k}{2}\|\mathbf{z}\|_{\mathbf{s}_k}^2 = \mathbb{E}_\zeta[f(\mathbf{z};\zeta)] + \frac{\lambda_k}{2}\|\mathbf{z}\|_{\mathbf{s}_k}^2$, in the (18). By using the smoothness of $f(\theta;\zeta)$, we can obtain

$$\begin{aligned}
&F_{k+1}(\mathbf{x}_{k+1})\\
\leq& F(\mathbf{x}_k) + \langle \nabla F(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k\rangle + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + \frac{\lambda_{k+1}}{2}\|\mathbf{x}_{k+1}\|_{\mathbf{s}_{k+1}}^2\\
\overset{①}{\leq}& F(\mathbf{x}_k) + \langle \nabla F(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k\rangle + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + \frac{\lambda_{k+1}}{2(1-\mu)}\|\mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2\\
\overset{②}{\leq}& F(\mathbf{x}_k) + \frac{\lambda_k}{2}\|\mathbf{x}_k\|_{\mathbf{s}_k}^2 + \langle \nabla F(\mathbf{x}_k) + \lambda_k \mathbf{x}_k \odot \mathbf{s}_k, \mathbf{x}_{k+1} - \mathbf{x}_k\rangle + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + \frac{\lambda_k}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_{\mathbf{s}_k}^2\\
\overset{③}{\leq}& F_k(\mathbf{x}_k) - \frac{\eta_k^x}{1+\lambda_k\eta_k^x}\left\langle \nabla F(\mathbf{x}_k) + \lambda_k\mathbf{x}_k \odot \mathbf{s}_k, \frac{\mathbf{w}_k}{\mathbf{s}_k}\right\rangle + \frac{L(\eta_k^x)^2}{2(1+\lambda_k\eta_k^x)^2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|^2 + \frac{\lambda_k(\eta_k^x)^2}{2(1+\lambda_k\eta_k^x)^2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|_{\mathbf{s}_k}^2\\
\overset{④}{=}& F_k(\mathbf{x}_k) - \frac{\alpha_k\eta_k^x}{1+\lambda_k\eta_k^x}\left\langle \nabla F(\mathbf{x}_k), \frac{\mathbf{m}_k}{\mathbf{s}_k}\right\rangle + \frac{L(\eta_k^x)^2}{2(1+\lambda_k\eta_k^x)^2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|^2 + \frac{\lambda_k(\eta_k^x)^2}{2(1+\lambda_k\eta_k^x)^2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|_{\mathbf{s}_k}^2\\
\leq& F_k(\mathbf{x}_k) + \frac{1}{2}\left\|\sqrt{\frac{\alpha_k\eta_k^x}{\mathbf{s}_k}}(\nabla F(\mathbf{x}_k) - \mathbf{m}_k)\right\|^2 - \frac{1}{2}\left\|\sqrt{\frac{\alpha_k\eta_k^x}{\mathbf{s}_k}}\nabla F(\mathbf{x}_k)\right\|^2 - \frac{1}{2}\left\|\sqrt{\frac{\eta_k^x}{\alpha_k\mathbf{s}_k}}\mathbf{w}_k\right\|^2\\
&+ \frac{L(\eta_k^x)^2}{2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|^2\\
\leq& F_k(\mathbf{x}_k) + \frac{\alpha_k\eta_k^x}{2c_1}\|\nabla F(\mathbf{x}_k) - \mathbf{m}_k\|^2 - \frac{\alpha_k\eta_k^x}{2c_2}\|\nabla F_k(\mathbf{x}_k)\|^2 - \frac{\eta_k^x}{2c_2\alpha_k}\left[1 - \frac{\alpha_k c_2 L\eta_k^x}{c_1^2}\right]\|\mathbf{w}_k\|^2\\
\overset{⑤}{\leq}& F_k(\mathbf{x}_k) + \frac{\alpha_l\eta_k^x}{2c_1}\|\nabla F(\mathbf{x}_k) - \mathbf{m}_k\|^2 - \frac{\alpha_s\eta_k^x}{2c_2}\|\nabla F_k(\mathbf{x}_k)\|^2 - \frac{\eta_k^x}{4\alpha_l c_2}\|\mathbf{w}_k\|^2,
\end{aligned} \tag{35}$$

where ① holds since Lemma 7 proves $\left\|\frac{\mathbf{s}_k}{\mathbf{s}_{k+1}}\right\|_\infty \in [1-\mu, 1+\mu]$ $(\forall p \in [0,1])$ in which $\mu = \frac{\beta_2 c_\infty^2}{\nu}$;
② holds because $\lambda_k = \frac{\lambda_{k+1}}{1-\mu}$ and

$$\|\mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2 = \|\mathbf{x}_k\|_{\mathbf{s}_k}^2 + \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_{\mathbf{s}_k}^2 + 2\langle \mathbf{x}_{k+1} - \mathbf{x}_k, \mathbf{x}_k\rangle_{\mathbf{s}_k}.$$

③ holds, since we have 1) $\mathbf{w}_k := \alpha_k\mathbf{m}_k + (1+\alpha_k)\lambda_k\mathbf{x}_k \odot \mathbf{s}_k$ and $\mathbf{u}_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\mathbf{s}_k + \lambda_k\mathbf{x}_k\|_2}\left(\frac{\mathbf{m}_k}{\mathbf{s}_k} + \lambda_k\mathbf{x}_k\right) = \alpha_k\left(\frac{\mathbf{m}_k}{\mathbf{s}_k} + \lambda_k\mathbf{x}_k\right) = \frac{\mathbf{w}_k - \lambda_k\mathbf{x}_k\odot\mathbf{s}_k}{\mathbf{s}_k}$ in Win-accelerated LAMB; 2) $\mathbf{x}_{k+1} = \frac{1}{1+\lambda_k\eta_k^x}(\mathbf{x}_k - \eta_k^x\mathbf{u}_k) = \frac{1}{1+\lambda_k\eta_k^x}\left(\mathbf{x}_k - \eta_k^x\frac{\mathbf{w}_k - \lambda_k\mathbf{x}_k\odot\mathbf{s}_k}{\mathbf{s}_k}\right) = \mathbf{x}_k - \frac{\eta_k^x}{1+\lambda_k\eta_k^x}\frac{\mathbf{w}_k}{\mathbf{s}_k}$.

④ holds, because we set $\lambda = 0$ which yields $\lambda_k = 0$ and $\mathbf{w}_k = \alpha_k \mathbf{m}_k$.
⑤ holds, since we set $\eta_k^x \leq \frac{c_1^2}{2c_2}$ such that $\frac{c_2 L \eta_k^x}{c_1^2} \leq \frac{1}{2}$.

From Lemma 11, by setting $\eta_k^x = \eta$, $\eta_k^y = \eta^y$ and $\beta_{1,k} = \beta_1$, we have

$$\begin{aligned}\mathbb{E}\left[\left\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\right\|^2\right] \leq & 2(1-\beta_1)\mathbb{E}\left[\left\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\right\|^2\right] + \frac{2\Pi_{1,k}^y (1-\beta_1)^2 L^2}{\beta_1} \\ & + \frac{2\beta_1^2\sigma^2}{b} + 2L\Pi_{2,k}^y,\end{aligned} \tag{36}$$

where

$$\begin{aligned}\Pi_{1,k}^y \leq \bar{\Pi}_{1,k}^y :=& \frac{2(\eta^y)^2}{c_1^2}\left\|\mathbf{w}_{k-1}\right\|^2 + \frac{2\rho_k^y (\eta^y)^2 (\eta^y - \eta)^2 (\tau_{k-1}^y)^2}{c_1^2} \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}^y (1 - \eta\tau_{i-1}^y)} \left\|\mathbf{w}_i\right\|^2, \\ \Pi_{2,k}^y \leq \bar{\Pi}_{2,k}^y :=& \frac{\tau_{k-1}^y \rho_k^y \eta (\eta^y - \eta)^2}{c_1^2} \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}^y (1 - \eta\tau_{i-1}^y)} \left\|\mathbf{w}_i\right\|^2,\end{aligned} \tag{37}$$

in which $\rho_{k+1}^y = \eta\tau_{k-1}^y \rho_k^y$, $\rho_1^y = 1$ and $\rho_0^y = 0$. Therefore, by plugging the results in Eqn. (36) into the upper bound of $F_{k+1}(\mathbf{x}_{k+1})$, we have

$$\begin{aligned}& F_{k+1}(\mathbf{x}_{k+1}) \\ \leq & F_k(\mathbf{x}_k) - \frac{\alpha_s \eta}{2c_2}\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 - \frac{\eta}{4\alpha_l c_2}\left\|\mathbf{w}_k\right\|^2 + \frac{\alpha_l \eta (1-\beta_1)}{c_1}\mathbb{E}\left[\left\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\right\|^2\right] \\ & + \frac{\alpha_l \eta \bar{\Pi}_{1,k}^y (1-\beta_1)^2 L^2}{c_1 \beta_1} + \frac{\alpha_l \eta \beta_1^2 \sigma^2}{c_1 b} + \frac{\alpha_l \eta L \bar{\Pi}_{2,k}^y}{c_1}\end{aligned} \tag{38}$$

Then, from Lemma 8, we have

$$\begin{aligned}& \mathbb{E}\left[\left\|\mathbf{m}_k - \nabla F(\mathbf{y}_k)\right\|^2\right] \\ \leq & (1-\beta_1)\mathbb{E}\left[\left\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\right\|^2\right] + \frac{(1-\beta_1)^2 L^2}{\beta_1}\mathbb{E}\left[\left\|\mathbf{y}_k - \mathbf{y}_{k-1}\right\|^2\right] + \frac{\beta_1^2\sigma^2}{b} \\ \overset{①}{\leq} & (1-\beta_1)\mathbb{E}\left[\left\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\right\|^2\right] + \frac{(1-\beta_1)^2 L^2 \bar{\Pi}_{1,k}^y}{\beta_1} + \frac{\beta_1^2\sigma^2}{b}\end{aligned} \tag{39}$$

where we use the results in Lemma 10 that $\left\|\mathbf{y}_k - \mathbf{y}_{k-1}\right\|^2 \leq \Pi_{1,k}^y \leq \bar{\Pi}_{1,k}^y$.

Then we add Eqn. (38) and $\alpha\times$ (39) as follows:

$$\begin{aligned}& F_{k+1}(\mathbf{x}_{k+1}) + \alpha\mathbb{E}\left[\left\|\mathbf{m}_k - \nabla F(\mathbf{y}_k)\right\|^2\right] \\ \leq & F_k(\mathbf{x}_k) - \frac{\alpha_s \eta}{2c_2}\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 - \frac{\eta}{4\alpha_l c_2}\left\|\mathbf{w}_k\right\|^2 + (1-\beta_1)\left(\frac{\alpha_l \eta}{c_1} + \alpha\right)\mathbb{E}\left[\left\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\right\|^2\right] \\ & + \frac{\alpha_l \eta \bar{\Pi}_{1,k}^y (1-\beta_1)^2 L^2}{c_1 \beta_1} + \frac{\alpha_l \eta \beta_1^2 \sigma^2}{c_1 b} + \frac{\alpha_l \eta L \bar{\Pi}_{2,k}^y}{c_1} + \frac{\alpha(1-\beta_1)^2 L^2 \bar{\Pi}_{1,k}^y}{\beta_1} + \frac{\alpha\beta_1^2\sigma^2}{b}\end{aligned} \tag{40}$$

Then by setting $\alpha = \frac{\alpha_l\eta(1-\beta_1)}{c_1\beta_1}$ and $G_{k+1}(\mathbf{x}_{k+1}) = F_{k+1}(\mathbf{x}_{k+1}) + \frac{\alpha_l\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\|^2\right] = \mathbb{E}_\zeta[f(\mathbf{z};\zeta)] + \frac{\lambda_k}{2}\|\mathbf{z}\|^2_{\mathbf{s}_k} + \frac{\alpha_l\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\|^2\right]$, we can obtain

$$
\begin{aligned}
&G_{k+1}(\mathbf{x}_{k+1})\\
\leq& G_k(\mathbf{x}_k) - \frac{\alpha_s\eta}{2c_2}\|\nabla F_k(\mathbf{x}_k)\|^2 - \frac{\eta}{4\alpha_l c_2}\|\mathbf{w}_k\|^2\\
&+ \frac{\alpha_l\eta\bar{\Pi}^y_{1,k}(1-\beta_1)^2L^2}{c_1\beta_1} + \frac{\alpha_l\eta\beta_1^2\sigma^2}{c_1 b} + \frac{\alpha_l\eta L\bar{\Pi}^y_{2,k}}{c_1} + \frac{\alpha_l\eta(1-\beta_1)^3L^2\bar{\Pi}^y_{1,k}}{c_1\beta_1^2} + \frac{\alpha_l\eta(1-\beta_1)\beta_1\sigma^2}{c_1 b}\\
=& G_k(\mathbf{x}_k) - \frac{\alpha_s\eta}{2c_2}\|\nabla F_k(\mathbf{x}_k)\|^2 - \frac{\eta}{4\alpha_l c_2}\|\mathbf{w}_k\|^2 + \frac{\alpha_l\eta(1-\beta_1)^2L^2\bar{\Pi}^y_{1,k}}{c_1\beta_1^2} + \frac{\alpha_l\eta L\bar{\Pi}^y_{2,k}}{c_1} + \frac{\alpha_l\eta\beta_1\sigma^2}{c_1 b}.
\end{aligned}
$$

Then summing the above inequality from $k=0$ to $k=T-1$ and using $0<\lambda_k\leq\lambda$ give

$$
\begin{aligned}
&\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\nabla F_k(\mathbf{x}_k)\|^2 + \frac{1}{2\alpha_s\alpha_l}\|\mathbf{w}_k\|^2\right]\\
\leq& \frac{2c_2}{\alpha_s\eta T}\left[G(\mathbf{x}_0) - G(\mathbf{x}_T)\right] + \frac{2\alpha_l c_2\beta_1\sigma^2}{\alpha_s c_1 b} + \frac{2\alpha_l c_2(1-\beta_1)^2L^2}{\alpha_s c_1\beta_1^2 T}\sum_{k=0}^{T-1}\bar{\Pi}^y_{1,k} + \frac{2\alpha_l c_2 L}{\alpha_s c_1 T}\sum_{k=0}^{T-1}\bar{\Pi}^y_{2,k}\\
\leq& \frac{2c_2\Delta}{\alpha_s\eta T} + \frac{2\alpha_l c_2\beta_1\sigma^2}{\alpha_s c_1 b} + \frac{2\alpha_l c_2(1-\beta_1)^2L^2}{\alpha_s c_1\beta_1^2 T}\sum_{k=0}^{T-1}\bar{\Pi}^y_{1,k} + \frac{2\alpha_l c_2 L}{\alpha_s c_1 T}\sum_{k=0}^{T-1}\bar{\Pi}^y_{2,k}
\end{aligned}
$$

where

$$
\begin{aligned}
&G(\mathbf{x}_0) - G(\mathbf{x}_T)\\
=& F_0(\mathbf{x}_0) + \frac{\alpha_l\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\|\mathbf{m}_{-1} - \nabla F(\mathbf{x}_{-1})\|^2\right] - F_T(\mathbf{x}_T)\\
&- \frac{\alpha_l\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\|\mathbf{m}_{T-1} - \nabla F(\mathbf{x}_{T-1})\|^2\right]\\
=& F(\mathbf{x}_0) + \lambda_0\|\mathbf{x}_0\|_{\mathbf{s}_0} - F(\mathbf{x}_T) - \lambda_T\|\mathbf{x}_T\|_{\mathbf{s}_T} - \frac{\alpha_l\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\|\mathbf{m}_{T-1} - \nabla F(\mathbf{x}_{T-1})\|^2\right]\\
\leq& F(\mathbf{x}_0) - F(\mathbf{x}_T) \leq \Delta
\end{aligned}
$$

where $\Delta = F(\mathbf{x}_0) - F(\mathbf{x}_\star)$; $\mathbf{x}_{-1}$ and $\mathbf{m}_{-1}$ are two virtual points which satisfy $\mathbf{m}_{-1} = \nabla F(\mathbf{x}_{-1})$.

Next, we can follow Eqn. (28) and (29) to bound

$$
\sum_{k=0}^{T-1}\bar{\Pi}^y_{1,k} \leq \frac{8\gamma^3\eta^2}{c_1^2}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_{k-1}\|^2\right], \qquad \sum_{k=0}^{T-1}\bar{\Pi}^y_{2,k} \leq \frac{\eta^2(\gamma-1)^2}{c_1^2}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right],
$$

where $\eta^y = \gamma\eta$. Therefore, we have

$$\begin{aligned}
&\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 + \frac{1}{2\alpha_s\alpha_l}\left\|\mathbf{w}_k\right\|^2\right] \\
\leq &\frac{2c_2\Delta}{\alpha_s\eta T} + \frac{2\alpha_l c_2\beta_1\sigma^2}{\alpha_s c_1 b} + \frac{16\alpha_l c_2\gamma^3\eta^2(1-\beta_1)^2L^2}{\alpha_s c_1^3\beta_1^2 T}\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_{k-1}\right\|^2\right] + \frac{2\alpha_l c_2\eta^2 L(\gamma-1)^2}{\alpha_s c_1^3 T}\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_k\right\|^2\right] \\
\overset{①}{\leq} &\frac{2c_2\Delta}{\alpha_s\eta T} + \frac{2\alpha_l c_2\beta_1\sigma^2}{\alpha_s c_1 b} + \frac{1}{4\alpha_s\alpha_l T}\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_k\right\|^2\right]
\end{aligned}$$

where ① holds since we choose proper η and β_1 such that

$$\frac{16\alpha_l c_2\gamma^3\eta^2(1-\beta_1)^2L^2}{\alpha_s c_1^3\beta_1^2} \leq \frac{1}{8\alpha_s\alpha_l}, \qquad \frac{2\alpha_l c_2\eta^2 L(\gamma-1)^2}{\alpha_s c_1^3} \leq \frac{1}{8\alpha_s\alpha_l} \tag{41}$$

Now we select η and β_1 such that (41) holds:

$$\eta \leq \min\left(\frac{c_1^{1.5}\beta_1}{8\sqrt{2}\alpha_l c_2^{0.5}\gamma^{1.5}(1-\beta_1)L}, \frac{c_1^{1.5}}{4\alpha_l c_2^{0.5}L^{0.5}(\gamma-1)}\right)$$

So we arrive at

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 + \frac{1}{4\alpha_s\alpha_l}\left\|\mathbf{w}_k\right\|^2\right] \leq \frac{2c_2\Delta}{\alpha_s\eta T} + \frac{2\alpha_l c_2\beta_1\sigma^2}{\alpha_s c_1 b} \overset{①}{\leq} \epsilon^2, \tag{42}$$

where we set $T \geq \frac{4c_2\Delta}{\alpha_s\eta\epsilon^2}$ and $\beta_1 \leq \frac{\alpha_s c_1 b\epsilon^2}{4\alpha_l c_2\sigma^2}$. Since $\mathbf{w}_k := \alpha_k\mathbf{m}_k + (1+\alpha_k)\lambda_k\mathbf{x}_k \odot \mathbf{s}_k$ and $\mathbf{u}_k = \alpha_k\big(\frac{\mathbf{m}_k}{\mathbf{s}_k} + \lambda_k\mathbf{x}_k\big) = \frac{\mathbf{w}_k - \lambda_k\mathbf{x}_k\odot\mathbf{s}_k}{\mathbf{s}_k}$, we have

$$\mathbf{x}_{k+1} = \frac{1}{1+\lambda_k\eta_k^x}\left(\mathbf{x}_k - \eta_k^x\mathbf{u}_k\right) = \frac{1}{1+\lambda_k\eta_k^x}\left(\mathbf{x}_k - \eta_k^x\frac{\mathbf{w}_k - \lambda_k\mathbf{x}_k\odot\mathbf{s}_k}{\mathbf{s}_k}\right) = \mathbf{x}_k - \frac{\eta_k^x}{1+\lambda_k\eta_k^x}\frac{\mathbf{w}_k}{\mathbf{s}_k}.$$

This result directly bounds

$$\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{s}_k\odot(\mathbf{x}_k - \mathbf{x}_{k+1})\right\|^2 = \frac{\eta^2}{T}\sum_{k=0}^{T-1}\left\|\mathbf{w}_k\right\|^2 \leq 4\alpha_s\alpha_l\eta^2\epsilon^2.$$

Moreover, from Lemma 10, we have

$$\begin{aligned}
\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{y}'_k-(1+\lambda_{k-1}\eta^y)\mathbf{x}_k\right\|^2 \overset{①}{\leq}& \frac{1}{T}\sum_{k=0}^{T-1}\rho_k^y(\eta^y-\eta)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\
\overset{③}{=}& \frac{1}{T}\sum_{k=0}^{T-1}\Pi_{3,k}^y,\\
\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{y}_k-\mathbf{x}_k\right\|^2 \overset{①}{\leq}& \frac{1}{T}\sum_{k=0}^{T-1}\tau_{k-1}^y\rho_k^y\eta(\eta^y-\eta)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\
\overset{②}{=}& \frac{1}{T}\sum_{k=0}^{T-1}\Pi_{2,k}^y,\\
\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{y}_{k+1}-\mathbf{y}_k\right\|^2 \overset{①}{\leq}& \frac{1}{T}\sum_{k=0}^{T-1}\left[\frac{2(\eta^y)^2}{(1+\lambda_k\eta)^2}+2\rho_{k+1}^y(\eta^y)^2(\eta^y-\eta)^2(\tau_k^y)^2(1+\lambda_k\eta)^2\right.\\
&\left.\cdot\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\right]\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|^2\\
\overset{②}{\leq}& \frac{1}{T}\sum_{k=0}^{T-1}\Pi_{1,k}^y
\end{aligned}$$

where $\rho_{k+1}^y=\eta\tau_{k-1}^y\rho_k^y$, $\rho_1^y=1$ and $\rho_0^y=0$; ① holds by using Lemma 10; ② holds by using the definition in Eqn. (37); ③ holds by defining:

$$\Pi_{3,k}^y:=\rho_k^y(\eta^y-\eta)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2.$$

Now remaining task is to upper bound $\frac{1}{T}\sum_{k=0}^{T-1}\Pi_{1,k}^y$, $\frac{1}{T}\sum_{k=0}^{T-1}\Pi_{2,k}^y$ and $\frac{1}{T}\sum_{k=0}^{T-1}\Pi_{3,k}^y$. Here we first bound $\frac{1}{T}\sum_{k=0}^{T-1}\Pi_{3,k}^y$ by using almost the same proof in Eqn. (32):

$$\frac{1}{T}\sum_{k=0}^{T-1}\Pi_{3,k}^y\overset{①}{\leq}\frac{(\eta^y-\eta)^2}{c_1^2\eta\tau(1-\eta\tau)^2T}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right]\leq\frac{\eta^2\gamma^2(\gamma-1)^2}{c_1^2(1+\gamma)T}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right]\overset{②}{\leq}\frac{4\alpha_s\alpha_l\eta^2\gamma^3\epsilon^2}{c_1^2} \tag{43}$$

where ① holds since in Lemma 9 we have prove $\sum_{k=0}^{T-1}\rho_k^y\left[\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\|\mathbf{w}_i\|^2\right]\leq\frac{1}{\eta\tau(1-\eta\tau)^2}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right]$; ② holds by using $\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|\mathbf{w}_k\|^2\leq 4\alpha_s\alpha_l\epsilon^2$ in Eqn. (42).

From the bound in Eqn. (37) and the following bound on $\frac{1}{T}\sum_{k=0}^{T-1}\bar{\Pi}_k$ and $\frac{1}{T}\sum_{k=0}^{T-1}\bar{\Pi}'_k$, we have

$$\begin{aligned}
\frac{1}{T}\sum_{k=0}^{T-1}\Pi_{1,k}^y\leq\frac{1}{T}\sum_{k=0}^{T-1}\bar{\Pi}_{1,k}^y\leq&\frac{8\gamma^3\eta^2}{c_1^2T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{w}_k\|^2\right]\overset{①}{\leq}\frac{32\alpha_s\alpha_l\gamma^3\eta^2}{c_1^2}\\
\frac{1}{T}\sum_{k=0}^{T-1}\Pi_{2,k}^y\leq\frac{1}{T}\sum_{k=0}^{T-1}\bar{\Pi}_{2,k}^y\leq&\frac{\eta^2(\gamma-1)^2}{c_1^2T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{w}_k\|^2\right]\overset{①}{\leq}\frac{4\alpha_s\alpha_l\gamma^2\epsilon^2}{c_1^2}
\end{aligned}$$

where ① holds, since 1) $\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|\mathbf{w}_k\|^2 \leq 4\alpha_s\alpha_l\epsilon^2$. Therefore, we have

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left\|\mathbf{y}'_k-(1+\lambda_k\eta^y)\mathbf{x}_k\right\|^2 \leq \frac{4\alpha_s\alpha_l\eta^2\gamma^3\epsilon^2}{c_1^2}, \qquad \frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left\|\mathbf{y}_k-\mathbf{x}_k\right\|^2 \leq \frac{4\alpha_s\alpha_l\eta^2\gamma^2\epsilon^2}{c_1^2},$$
$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left\|\mathbf{y}_{k+1}-\mathbf{y}_k\right\|^2 \leq \frac{32\alpha_s\alpha_l\eta^2\gamma^3\epsilon^2}{c_1^2}.$$

Besides, we have

$$\begin{aligned}
\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\mathbf{m}_k-\nabla F(\mathbf{x}_k)\right\|^2\right] \leq& \frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\mathbf{m}_k+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k-\nabla F(\mathbf{x}_k)-\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\right\|^2\right]\\
\leq& \frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\mathbf{m}_k+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\right\|^2+\left\|\nabla F(\mathbf{x}_k)+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\right\|^2\right]\\
=& \frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\mathbf{m}_k+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\right\|^2+\left\|\nabla F_k(\mathbf{x}_k)\right\|^2\right]\\
\overset{①}{=}& \frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\frac{1}{\alpha_k}\mathbf{w}_k\right\|^2+\left\|\nabla F_k(\mathbf{x}_k)\right\|^2\right] \overset{②}{\leq} 2\left[\epsilon^2+4\frac{\alpha_l}{\alpha_s}\epsilon^2\right] \leq \frac{10\alpha_l}{\alpha_s}\epsilon^2.
\end{aligned}$$

where in ① and ②, we use $\mathbf{w}_k := \alpha_k\mathbf{m}_k+(1+\alpha_k)\lambda_k\mathbf{x}_k\odot\mathbf{s}_k = \alpha_k\mathbf{m}_k$, with $\lambda_k = 0$ and $\alpha_s \leq \alpha_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\sqrt{\mathbf{v}_k+\nu}+\lambda_k\mathbf{x}_k\|_2} \leq \alpha_l$. In this way, we have

$$\begin{aligned}
\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\mathbf{m}_k-\nabla F(\mathbf{y}_k)\right\|^2\right] \leq& \frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\mathbf{m}_k-\nabla F(\mathbf{x}_k)\right\|^2+\left\|\nabla F(\mathbf{x}_k)-\nabla F(\mathbf{y}_k)\right\|^2\right]\\
\leq& \frac{20\alpha_l}{\alpha_s}\epsilon^2+\frac{2L^2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\mathbf{x}_k-\mathbf{y}_k\right\|^2\right] \leq \frac{20\alpha_l}{\alpha_s}\epsilon^2+\frac{8\alpha_s\alpha_l\gamma^2L^2\eta^2\epsilon^2}{c_1^2}.
\end{aligned}$$

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

$$\beta_1 \leq \frac{\alpha_s c_1 b\epsilon^2}{4\alpha_l c_2(1+\lambda\eta)\sigma^2} = \mathcal{O}\left(\frac{\alpha_s c_1 b\epsilon^2}{\alpha_l c_2\sigma^2}\right),$$

where $c_1 = \nu^{0.5} \leq \|\mathbf{s}_k\|_\infty \leq \left(c_\infty^2+\nu\right)^{0.5} = c_2$.

For η, it should satisfy

$$\eta \leq \min\left(\frac{c_1^{1.5}\beta_1}{8\sqrt{2}\alpha_l c_2^{0.5}\gamma^{1.5}(1-\beta_1)L}, \frac{c_1^{1.5}}{4\alpha_l c_2^{0.5}L^{0.5}(\gamma-1)}, \frac{c_1^2}{2c_2}\right)$$

Considering $\lambda\eta << 1$, $\frac{1+\lambda\eta}{1+(1-\mu)\lambda\eta} = a = 1$ due to $\lambda = 0$, μ is a constant, and $c_1 = \nu^{0.5} << 1$, then we have

$$\begin{aligned}
\eta \leq& \mathcal{O}\left(\min\left(\frac{c_1^{1.5}\beta_1}{\alpha_l c_2^{0.5}\gamma^{1.5}L}, \frac{c_1^{1.5}}{\alpha_l c_2^{0.5}\gamma L^{0.5}}, \frac{c_1^2}{c_2 L}\right)\right)\\
=& \mathcal{O}\left(\min\left(\frac{c_1^{2.5}b\epsilon^2}{\alpha_l c_2^{1.5}\gamma^{1.5}\sigma^2 L}, \frac{c_1^{1.5}}{\alpha_l c_2^{0.5}\gamma L^{0.5}}, \frac{c_1^2}{c_2 L}\right)\right) = \mathcal{O}\left(\frac{c_1^{2.5}b\epsilon^2}{\alpha_l c_2^{1.5}\gamma^{1.5}\sigma^2 L}\right)
\end{aligned}$$

where ν is often much smaller than one, and β_1 is very small. For T, we have

$$T \geq \frac{4c_2\Delta}{\alpha_s\eta\epsilon^2} = \mathcal{O}\left(\frac{c_2\Delta}{\alpha_s\epsilon^2}\frac{\alpha_l c_2^{1.5}\gamma^{1.5}\sigma^2 L}{c_1^{2.5}b\epsilon^2}\right) = \mathcal{O}\left(\frac{\alpha_l c_2^{2.5}\gamma^{1.5}\sigma^2 L\Delta}{\alpha_s c_1^{2.5}b\epsilon^4}\right) = \mathcal{O}\left(\frac{\alpha_l c_2^{2.5}\gamma^{1.5}\sigma^2 L\Delta}{\alpha_s \nu^{1.25}b\epsilon^4}\right).$$

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

$$\mathcal{O}\left(Tb\right) = \mathcal{O}\left(\frac{\alpha_l c_2^{2.5}\gamma^{1.5}\sigma^2 L\Delta}{\alpha_s \nu^{1.25}\epsilon^4}\right).$$

The proof is completed.

■

D.3 Proofs of Theorem 3

Proof Recall our definition $F_k(\theta_k) = F(\theta) + \frac{\lambda_k}{2}\left\|\theta\right\|_2^2 = \mathbb{E}_\zeta[f(\boldsymbol{\theta};\zeta)] + \frac{\lambda_k}{2}\left\|\theta\right\|_2^2$ in the (18). By setting $\beta_1' = 1-\beta_1$, then we have $\left\|\mathbf{m}_k\right\|_\infty \leq c_\infty$ by using Lemma 7 (see Appendix C). Also we define

$$\mathbf{w}_k := \mathbf{m}_k + \lambda\mathbf{x}_k, \qquad \mathbf{x}_{k+1} - \mathbf{x}_k = -\frac{\eta_k^x}{1+\lambda\eta_k^x}(\mathbf{m}_k + \lambda\mathbf{x}_k) = -\frac{\eta_k^x}{1+\lambda\eta_k^x}\mathbf{w}_k.$$

Note in the following, we set all $\lambda_k = \lambda$. By using the smoothness of $f(\theta;\zeta)$, we can obtain

$$\begin{aligned}
&F_{k+1}(\mathbf{x}_{k+1})\\
\leq& F(\mathbf{x}_k) + \langle\nabla F(\mathbf{x}_k), \mathbf{x}_{k+1}-\mathbf{x}_k\rangle + \frac{L}{2}\left\|\mathbf{x}_{k+1}-\mathbf{x}_k\right\|^2 + \frac{\lambda}{2}\left\|\mathbf{x}_{k+1}\right\|^2\\
\overset{①}{\leq}& F(\mathbf{x}_k) + \frac{\lambda}{2}\left\|\mathbf{x}_k\right\|^2 + \langle\nabla F(\mathbf{x}_k) + \lambda\mathbf{x}_k, \mathbf{x}_{k+1}-\mathbf{x}_k\rangle + \frac{L}{2}\left\|\mathbf{x}_{k+1}-\mathbf{x}_k\right\|^2 + \frac{\lambda}{2}\left\|\mathbf{x}_{k+1}-\mathbf{x}_k\right\|^2\\
=& F_k(\mathbf{x}_k) - \frac{\eta_k^x}{1+\lambda\eta_k^x}\langle\nabla F(\mathbf{x}_k) + \lambda\mathbf{x}_k, \mathbf{w}_k\rangle + \frac{L(\eta_k^x)^2}{2(1+\lambda\eta_k^x)^2}\left\|\mathbf{w}_k\right\|^2 + \frac{\lambda(\eta_k^x)^2}{2(1+\lambda\eta_k^x)^2}\left\|\mathbf{w}_k\right\|^2\\
=& F_k(\mathbf{x}_k) + \frac{1}{2}\left\|\sqrt{\frac{\eta_k^x}{(1+\lambda\eta_k^x)}}\left(\nabla F(\mathbf{x}_k) + \lambda\mathbf{x}_k - \mathbf{w}_k\right)\right\|^2 - \frac{1}{2}\left\|\sqrt{\frac{\eta_k^x}{(1+\lambda\eta_k^x)}}\left(\nabla F(\mathbf{x}_k) + \lambda\mathbf{x}_k\right)\right\|^2\\
&- \frac{1}{2}\left\|\sqrt{\frac{\eta_k^x}{(1+\lambda\eta_k^x)}}\mathbf{w}_k\right\|^2 + \frac{L(\eta_k^x)^2}{2(1+\lambda\eta_k^x)^2}\left\|\mathbf{w}_k\right\|^2 + \frac{\lambda(\eta_k^x)^2}{2(1+\lambda\eta_k^x)^2}\left\|\mathbf{w}_k\right\|^2\\
\overset{②}{\leq}& F_k(\mathbf{x}_k) + \frac{\eta_k^x}{2(1+\lambda\eta_k^x)}\left\|\nabla F(\mathbf{x}_k) - \mathbf{m}_k\right\|^2 - \frac{\eta_k^x}{2(1+\lambda\eta_k^x)}\left\|\nabla F_k(\mathbf{x}_k)\right\|^2\\
&- \frac{\eta_k^x}{2(1+\lambda\eta_k^x)}\left[1 - \frac{L\eta_k^x}{(1+\lambda\eta_k^x)} - \frac{\lambda\eta_k^x}{(1+\lambda\eta_k^x)}\right]\left\|\mathbf{w}_k\right\|^2\\
\overset{③}{\leq}& F_k(\mathbf{x}_k) + \frac{\eta_k^x}{2(1+\lambda\eta_k^x)}\left\|\nabla F(\mathbf{x}_k) - \mathbf{m}_k\right\|^2 - \frac{\eta_k^x}{2(1+\lambda\eta_k^x)}\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 - \frac{\eta_k^x}{4(1+\lambda\eta_k^x)}\left\|\mathbf{w}_k\right\|^2,
\end{aligned}$$

where ① holds because

$$\left\|\mathbf{x}_{k+1}\right\|_{\mathbf{s}_k}^2 = \left\|\mathbf{x}_k\right\|_{\mathbf{s}_k}^2 + \left\|\mathbf{x}_{k+1}-\mathbf{x}_k\right\|_{\mathbf{s}_k}^2 + 2\langle\mathbf{x}_{k+1}-\mathbf{x}_k, \mathbf{x}_k\rangle_{\mathbf{s}_k}.$$

② holds, because

$$\mathbf{w}_k := \mathbf{m}_k + \lambda \mathbf{x}_k, \qquad \mathbf{x}_{k+1} - \mathbf{x}_k = -\frac{\eta_k^x}{1+\lambda\eta_k^x}(\mathbf{m}_k + \lambda \mathbf{x}_k) = -\frac{\eta_k^x}{1+\lambda\eta_k^x}\mathbf{w}_k.$$

④ holds, since we set $\eta_k^x \leq \frac{c_1^2(1+\lambda\eta_k^x)}{2c_2(L+\lambda c_1)}$ such that $\frac{c_2 L\eta_k^x}{c_1^2(1+\lambda\eta_k^x)} + \frac{c_2\lambda\eta_k^x}{c_1(1+\lambda\eta_k^x)} \leq \frac{1}{2}$.

Then in the following, we can directly follow the proof of Theorem 1. This is because the only difference between accelerated SGD and AdamW is that SGD has no the second-order moment $\mathbf{v}_k$, while AdamW has. By let $\mathbf{s}_k = \mathbf{1}$ in accelerated AdamW and setting $\beta_1' = 1-\beta_1$ in accelerated SGD, then they share the exact the same updating rules. So after setting $\beta_1' = 1-\beta_1$ in accelerated SGD, to follow the proofs of Theorem 1, we only need to verify whether the auxiliary lemmas and the proof process of Theorem 1 hold for $\mathbf{s}_k = \mathbf{1}$. This is the true case. Please check our auxiliary lemmas, including Lemma 7 $\sim$ 11, and the proof process of Theorem 1. Consider $\mathbf{s}_k = \mathbf{1}$ in accelerated SGD, we have $c_1 := 1 \leq \|\mathbf{s}_k\|_\infty \leq c_2 := 1$.

In this way, by setting $\eta_k^y = \gamma\eta_k^x, \gamma > 1, \eta_k^x = \eta \leq \mathcal{O}\big(\frac{b\epsilon^2}{c^{1.5}\gamma^{2.5}\sigma^2 L}\big)$, $\beta_1 \leq \mathcal{O}\big(\frac{b\epsilon^2}{c\sigma^2}\big)$, $\beta_1' = 1-\beta_1$, $\lambda_k = \lambda$, $\lambda_0 = 0$, after $T = \mathcal{O}\big(\frac{\Delta\sigma^2 L}{b\epsilon^4}\big)$ iterations with minibatch size b and $\Delta = F(\mathbf{x}_0) - F(\mathbf{x}_\star)$, the sequence $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=0}^T$ generated by accelerated SGD satisfies the following four properties.
a) The gradient $\nabla F_k(\mathbf{x}_k)$ of the sequence $\{\mathbf{x}_k\}_{k=0}^T$ can be upper bounded by

$$\frac{1}{T}\sum\nolimits_{k=0}^{T-1}\mathbb{E}\Big[\,\|\nabla F_k(\mathbf{x}_k)\|_2^2 + \frac{1}{4}\,\|\mathbf{m}_k + \lambda_k\mathbf{x}_k\|_2^2\Big] \leq \epsilon^2.$$

b) The gradient moment $\mathbf{m}_k$ can well estimate the full gradient $\nabla F(\mathbf{x}_k)$ and $\nabla F(\mathbf{y}_k)$:

$$\frac{1}{T}\sum\nolimits_{k=0}^{T-1}\max\Big\{\mathbb{E}\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\|_2^2, \mathbb{E}\|\mathbf{m}_k - \nabla F(\mathbf{y}_k)\|_2^2\Big\} \leq 16\epsilon^2 + 8\eta^2\gamma^2 L^2\epsilon^2.$$

c) The sequence $\{(\mathbf{x}_k, \mathbf{y}_k)\}$ satisfies

$$\frac{1}{T}\sum\nolimits_{k=0}^{T-1}\Big\{\mathbb{E}\|\mathbf{x}_k - \mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2, \mathbb{E}\|\mathbf{y}_k - \mathbf{x}_k\|_2^2\Big\} \leq \big\{4\eta^2\epsilon^2, 4\eta^2\gamma^2\epsilon^2\big\}.$$

d) The stochastic gradient complexity to achieve the above three properties is $\mathcal{O}\big(\frac{c_\infty^{2.5}\Delta\sigma^2 L}{\epsilon^4}\big)$, where stochastic gradient complexity is the total evaluation number of the gradient on a single sample.

The proof is completed. ■

Appendix E. Proofs of Main Results in Sec. 4

Here we provide proofs of the main results in Sec. 4, including Theorem 4, 5 and 6.

E.1 Proof of Theorem 4

Proof Recall our definition $F_k(\mathbf{y}_k) = F(\mathbf{z}) + \frac{\lambda_k}{2}\|\mathbf{z}\|_{\mathbf{s}_k}^2 = \mathbb{E}_\zeta[f(\mathbf{z};\zeta)] + \frac{\lambda_k}{2}\|\mathbf{z}\|_{\mathbf{s}_k}^2$, in the (18). By using the smoothness of $f(\theta;\zeta)$, we can extactly follow Eqn. (21) in Appendix D.1 for proving Theorem 1 to obtain

$$\begin{aligned} F_{k+1}(\mathbf{x}_{k+1}) \leq & F_k(\mathbf{x}_k) + \frac{\eta_k^x}{2c_1(1+\lambda_k\eta_k^x)}\|\nabla F(\mathbf{x}_k) - \mathbf{m}_k\|^2 - \frac{\eta_k^x}{2c_2(1+\lambda_k\eta_k^x)}\|\nabla F_k(\mathbf{x}_k)\|^2 \\ & - \frac{\eta_k^x}{4c_2(1+\lambda_k\eta_k^x)}\|\mathbf{w}_k\|^2, \end{aligned} \tag{44}$$

where we set $\eta_k^x \leq \frac{c_1^2(1+\lambda_k\eta_k^x)}{2c_2(L+\lambda_k c_1)}$ such that $\frac{c_2 L\eta_k^x}{c_1^2(1+\lambda_k\eta_k^x)} + \frac{c_2\lambda_k\eta_k^x}{c_1(1+\lambda_k\eta_k^x)} \leq \frac{1}{2}$.

From Lemma 14, by setting $\eta_k^x = \eta$, $\eta_k^y = \eta^y$, $\eta_k^z = \eta^z$ and $\beta_{1,k} = \beta_1$, we have

$$\begin{aligned}\mathbb{E}\left[\left\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\right\|^2\right] \leq & 2(1-\beta_1)\mathbb{E}\left[\left\|\mathbf{m}_{k-1} - \nabla F(\mathbf{z}_{k-1})\right\|^2\right] + \frac{2\Pi_{1,k}^z(1-\beta_1)^2L^2}{\beta_1} \\ & + \frac{2\beta_1^2\sigma^2}{b} + 2L\Pi_{2,k}^z,\end{aligned} \tag{45}$$

where

$$\begin{aligned}\Pi_{1,k}^z :=& \frac{3(\xi^x)^2}{(\xi^z)^2(\lambda_{k-1}+\xi^x)^2}\left\|\frac{\mathbf{w}_{k-1}}{\mathbf{s}_{k-1}}\right\|^2 \\ & + 3(\xi^x+\xi^y+\lambda_{k-1})^2(\delta_{k-1}^z)^2\rho_k^z(\eta^z-\eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2 \\ & + 3(\xi^y)^4(\delta_{k-1}^y)^2(\delta_{k-1}^z)^2\rho_k^y(\eta^y-\eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2, \\ \Pi_{2,k}^z :=& 2(\xi^z\delta_{k-1}^z)^2\rho_k^z(\eta^z-\eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2 \\ & + 2(\xi^y)^4(\delta_{k-1}^z\delta_{k-1}^y)^2\rho_k^y(\eta^y-\eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2.\end{aligned}$$

where $\rho_{k+1}^y = \eta^x\tau_{k-1}^y\rho_k^y$, $\rho_1^y = 1$ and $\rho_0^y = 0$; $\rho_{k+1}^z = \eta^x\tau_{k-1}^z\rho_k^z$, $\rho_1^z = 1$ and $\rho_0^z = 0$; $\tau_k^y = \frac{1}{\eta^x+\eta^y+\lambda_k\eta^x\eta^y}$, $\tau_k^z = \frac{1}{\eta^x+\eta^z+\lambda_k\eta^x\eta^z}$, $\delta_k^y = \frac{1}{\xi^x+\xi^y+\lambda_k}$, $\delta_k^z = \frac{1}{\xi^x+\xi^y+\xi^z+\lambda_k}$.

By considering $c_2 \geq \|\mathbf{s}_k\|_\infty \geq c_1$ and setting $\eta_k^x = \eta$, $\eta_k^y = \eta^y = \gamma_y\eta$, $\eta_k^z = \eta^z = \gamma_z\eta$ and $\beta_{1,k} = \beta_1$, we have

$$\begin{aligned}\Pi_{1,k}^z \leq & \bar{\Pi}_{1,k}^z = \frac{3\gamma_z^2\eta^2}{c_1^2(1+\lambda_{k-1}\eta)^2}\left\|\mathbf{w}_{k-1}\right\|^2 + c_3\rho_k^z(\eta^z-\eta^x)^2\Phi_k + c_4\rho_k^y(\eta^y-\eta^x)^2\Psi_k, \\ \Pi_{2,k}^z \leq & \bar{\Pi}_{2,k}^z = c_5\rho_k^z(\eta^z-\eta^x)^2\Phi_k + c_6\rho_k^y(\eta^y-\eta^x)^2\Psi_k. \\ \Phi_k = & \sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\mathbf{w}_{k-1}\right\|^2, \\ \Psi_k = & \sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\mathbf{w}_{k-1}\right\|^2\end{aligned}$$

in which $c_3 = \frac{3}{c_1^2}\left(1-\frac{\gamma_y}{\gamma_y+\gamma_z+(1+\lambda_{k-1}\eta)\gamma_y\gamma_z}\right)^2$, $c_4 = \frac{3}{c_1^2(1+(1+\lambda_{k-1}\eta)\gamma_y)^2(1+(1+\lambda_{k-1}\eta)\gamma_y+\gamma_y/\gamma_z)^2}$, $c_5 = \frac{2}{((1+\lambda_{k-1}\eta)\gamma_z+1+\gamma_z/\gamma_y)^2}$, and $c_6 = \frac{2}{c_1^2(1+(1+\lambda_{k-1}\eta)\gamma_y)^2(1+(1+\lambda_{k-1}\eta)\gamma_y+\gamma_y/\gamma_z)^2}$.

Therefore, by plugging the results in Eqn. (45) into the upper bound of $F_{k+1}(\mathbf{x}_{k+1})$, we have

$$\begin{aligned}
&F_{k+1}(\mathbf{x}_{k+1})\\
\leq&F_k(\mathbf{x}_k) - \frac{\eta}{2c_2(1+\lambda_k\eta)}\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 - \frac{\eta}{4c_2(1+\lambda_k\eta)}\left\|\mathbf{w}_k\right\|^2\\
&+ \frac{\eta(1-\beta_1)}{c_1(1+\lambda_k\eta)}\mathbb{E}\left[\left\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\right\|^2\right] + \frac{\eta\bar{\Pi}_{1,k}^z(1-\beta_1)^2L^2}{c_1\beta_1(1+\lambda_k\eta)} + \frac{\eta\beta_1^2\sigma^2}{c_1(1+\lambda_k\eta)b} + \frac{\eta L\bar{\Pi}_{2,k}^z}{c_1(1+\lambda_k\eta)}\\
\overset{①}{\leq}&F_k(\mathbf{x}_k) - \frac{\eta}{2c_2(1+\lambda_k\eta)}\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 - \frac{\eta}{4c_2(1+\lambda_k\eta)}\left\|\mathbf{w}_k\right\|^2\\
&+ \frac{\eta(1-\beta_1)}{c_1}\mathbb{E}\left[\left\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\right\|^2\right] + \frac{\eta\bar{\Pi}_{1,k}^z(1-\beta_1)^2L^2}{c_1\beta_1(1+\lambda_k\eta)} + \frac{\eta\beta_1^2\sigma^2}{c_1(1+\lambda_k\eta)b} + \frac{\eta L\bar{\Pi}_{2,k}^z}{c_1(1+\lambda_k\eta)},
\end{aligned} \tag{46}$$

where ① uses the fact that $0 < \lambda_k \leq \lambda$. Then, by plugging $\left\|\mathbf{z}_k - \mathbf{z}_{k-1}\right\|^2 \leq \Pi_{1,k}^z \leq \bar{\Pi}_{1,k}^z$ in Lemma 11 into Lemma 12, we have

$$\mathbb{E}\left[\left\|\mathbf{m}_k - \nabla F(\mathbf{z}_k)\right\|^2\right] \leq (1-\beta_1)\mathbb{E}\left[\left\|\mathbf{m}_{k-1} - \nabla F(\mathbf{z}_{k-1})\right\|^2\right] + \frac{(1-\beta_1)^2L^2\bar{\Pi}_{1,k}^z}{\beta_1} + \frac{\beta_1^2\sigma^2}{b}. \tag{47}$$

Then we add Eqn. (46) and $\alpha\times$ (47) as follows:

$$\begin{aligned}
&F_{k+1}(\mathbf{x}_{k+1}) + \alpha\mathbb{E}\left[\left\|\mathbf{m}_k - \nabla F(\mathbf{y}_k)\right\|^2\right]\\
\leq&F_k(\mathbf{x}_k) - \frac{\eta}{2c_2(1+\lambda_k\eta)}\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 - \frac{\eta}{4c_2(1+\lambda_k\eta)}\left\|\mathbf{w}_k\right\|^2\\
&+ (1-\beta_1)\left(\frac{\eta}{c_1} + \alpha\right)\mathbb{E}\left[\left\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\right\|^2\right]\\
&+ \frac{\eta\bar{\Pi}_{1,k}^z(1-\beta_1)^2L^2}{c_1\beta_1(1+\lambda_k\eta)} + \frac{\eta\beta_1^2\sigma^2}{c_1(1+\lambda_k\eta)b} + \frac{\eta L\bar{\Pi}_{2,k}^z}{c_1(1+\lambda_k\eta)} + \frac{\alpha(1-\beta_1)^2L^2\bar{\Pi}_{1,k}^z}{\beta_1} + \frac{\alpha\beta_1^2\sigma^2}{b}
\end{aligned} \tag{48}$$

Then by setting $\alpha = \frac{\eta(1-\beta_1)}{c_1\beta_1}$ and $G_{k+1}(\mathbf{x}_{k+1}) = F_{k+1}(\mathbf{x}_{k+1}) + \frac{\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\left\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\right\|^2\right] = \mathbb{E}_\zeta[f(\mathbf{z};\zeta)] + \frac{\lambda_k}{2}\left\|\mathbf{z}\right\|_{\mathbf{s}_k}^2 + \frac{\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\left\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\right\|^2\right]$, we can obtain

$$\begin{aligned}
&G_{k+1}(\mathbf{x}_{k+1}) \leq G_k(\mathbf{x}_k) - \frac{\eta}{2c_2(1+\lambda_k\eta)}\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 - \frac{\eta}{4c_2(1+\lambda_k\eta)}\left\|\mathbf{w}_k\right\|^2\\
&+ \frac{\eta\bar{\Pi}_{1,k}^z(1-\beta_1)^2L^2}{c_1\beta_1(1+\lambda_k\eta)} + \frac{\eta\beta_1^2\sigma^2}{c_1(1+\lambda_k\eta)b} + \frac{\eta L\bar{\Pi}_{2,k}^z}{c_1(1+\lambda_k\eta)} + \frac{\eta(1-\beta_1)^3L^2\bar{\Pi}_{1,k}^z}{c_1\beta_1^2} + \frac{\eta(1-\beta_1)\beta_1\sigma^2}{c_1b}\\
\overset{①}{\leq}&G_k(\mathbf{x}_k) - \frac{\eta}{2c_2(1+\lambda_k\eta)}\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 - \frac{\eta}{4c_2(1+\lambda_k\eta)}\left\|\mathbf{w}_k\right\|^2 + \frac{\eta(1-\beta_1)^2L^2\bar{\Pi}_{1,k}^z}{c_1\beta_1^2}\\
&+ \frac{\eta L\bar{\Pi}_{2,k}^z}{c_1(1+\lambda_k\eta)} + \frac{\eta\beta_1\sigma^2}{c_1b},
\end{aligned}$$

where ① uses the fact that $0 < \lambda_k \leq \lambda$. Then summing the above inequality from $k = 0$ to $k = T-1$ and using $0 < \lambda_k \leq \lambda$ give

$$\begin{aligned}
&\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\nabla F_k(\mathbf{x}_k)\|^2 + \frac{1}{2}\|\mathbf{w}_k\|^2\right] \\
\leq&\frac{2c_2(1+\lambda\eta)}{\eta T}\left[G(\mathbf{x}_0) - G(\mathbf{x}_T)\right] + \frac{2c_2\beta_1\sigma^2(1+\lambda\eta)}{c_1 bT} + \frac{2c_2\beta_1^2\sigma^2}{c_1 b} \\
&+ \frac{2c_2(1-\beta_1)^2L^2(1+\lambda\eta)}{c_1\beta_1^2 T}\sum_{k=0}^{T-1}\bar{\Pi}_{1,k}^z + \frac{2c_2 L}{c_1 T}\sum_{k=0}^{T-1}\bar{\Pi}_{2,k}^z \\
\leq&\frac{2c_2(1+\lambda\eta)\Delta}{\eta T} + \frac{2c_2\beta_1\sigma^2(1+\lambda\eta)}{c_1 b} + \frac{2c_2(1-\beta_1)^2L^2(1+\lambda\eta)}{c_1\beta_1^2 T}\sum_{k=0}^{T-1}\bar{\Pi}_{1,k}^z + \frac{2c_2 L}{c_1 T}\sum_{k=0}^{T-1}\bar{\Pi}_{2,k}^z
\end{aligned}$$

where

$$\begin{aligned}
&G(\mathbf{x}_0) - G(\mathbf{x}_T) \\
=&F_0(\mathbf{x}_0) + \frac{\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\|\mathbf{m}_{-1} - \nabla F(\mathbf{x}_{-1})\|^2\right] - F_T(\mathbf{x}_T) - \frac{\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\|\mathbf{m}_{T-1} - \nabla F(\mathbf{x}_{T-1})\|^2\right] \\
=&F(\mathbf{x}_0) + \lambda_0\|\mathbf{x}_0\|_{\mathbf{s}_0} - F(\mathbf{x}_T) - \lambda_T\|\mathbf{x}_T\|_{\mathbf{s}_T} - \frac{\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\|\mathbf{m}_{T-1} - \nabla F(\mathbf{x}_{T-1})\|^2\right] \\
\leq&F(\mathbf{x}_0) - F(\mathbf{x}_T) \leq \Delta
\end{aligned}$$

where $\Delta = F(\mathbf{x}_0) - F(\mathbf{x}_\star)$; $\mathbf{x}_{-1}$ and $\mathbf{m}_{-1}$ are two virtual points which satisfy $\mathbf{m}_{-1} = \nabla F(\mathbf{x}_{-1})$. Now we try to bound $\sum_{k=0}^{T-1}\bar{\Pi}_{1,k}^z$ and $\sum_{k=0}^{T-1}\bar{\Pi}_{2,k}^z$.

$$\begin{aligned}
&\Pi_{1,k}^z \leq \bar{\Pi}_{1,k}^z = \frac{3\gamma_z^2\eta^2}{c_1^2(1+\lambda_{k-1}\eta)^2}\|\mathbf{w}_{k-1}\|^2 + c_3\rho_k^z(\eta^z-\eta^x)^2\Phi_k + c_4\rho_k^y(\eta^y-\eta^x)^2\Psi_k, \\
&\Pi_{2,k}^z \leq \bar{\Pi}_{2,k}^z = c_5\rho_k^z(\eta^z-\eta^x)^2\Phi_k + c_6\rho_k^y(\eta^y-\eta^x)^2\Psi_k. \\
&\Phi_k = \sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\|\mathbf{w}_{k-1}\|^2, \quad \Psi_k = \sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\|\mathbf{w}_{k-1}\|^2
\end{aligned} \tag{49}$$

in which $c_3 = \frac{3}{c_1^2}\left(1 - \frac{\gamma_y}{\gamma_y+\gamma_z+(1+\lambda_{k-1}\eta)\gamma_y\gamma_z}\right)^2$, $c_4 = \frac{3}{c_1^2(1+(1+\lambda_{k-1}\eta)\gamma_y)^2(1+(1+\lambda_{k-1}\eta)\gamma_y+\gamma_y/\gamma_z)^2}$, $c_5 = \frac{2}{((1+\lambda_{k-1}\eta)\gamma_z+1+\gamma_z/\gamma_y)^2}$, and $c_6 = \frac{2}{c_1^2(1+(1+\lambda_{k-1}\eta)\gamma_y)^2(1+(1+\lambda_{k-1}\eta)\gamma_y+\gamma_y/\gamma_z)^2}$.

Firstly, we have

$$\begin{aligned}
\sum_{k=0}^{T-1}\bar{\Pi}_{1,k}^z =& \sum_{k=0}^{T-1}\left[\frac{3\gamma_z^2\eta^2}{c_1^2(1+\lambda_{k-1}\eta)^2}\|\mathbf{w}_{k-1}\|^2 + c_3\rho_k^z(\eta^z-\eta^x)^2\Phi_k + c_4\rho_k^y(\eta^y-\eta^x)^2\Psi_k\right] \\
\overset{①}{\leq}& \left[\frac{3\gamma_z^2\eta^2}{c_1^2(1+\lambda_{k-1}\eta)^2} + c_3(\eta^z-\eta^x)^2\frac{(1+\gamma_z)^2}{\gamma_z} + c_4(\eta^y-\eta^x)^2\frac{(1+\gamma_y)^2}{\gamma_y}\right]\sum_{k=0}^{T-1}\left[\|\mathbf{w}_{k-1}\|^2\right] \\
\overset{②}{\leq}& \frac{3\eta^2}{c_1^2}\left[1+\gamma_y^3+\gamma_z^3\right]\sum_{k=0}^{T-1}\left[\|\mathbf{w}_{k-1}\|^2\right]
\end{aligned} \tag{50}$$

where ① holds since $0 \leq \lambda_k \leq \lambda$; ② holds, since in Lemma 9, we prove

$$\begin{aligned}
\sum_{k=0}^{T-1} \rho_k^z \Phi_k &= \sum_{k=0}^{T-1} \rho_k^z \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}^z (1-\eta^x \tau_{i-1}^z)(1+\lambda_i \eta^x)^2} \left\|\mathbf{w}_i\right\|^2 \\
&\leq \frac{1}{\eta\tau_z(1-\eta\tau_z)^2} \sum_{k=0}^{T-1} \left[\left\|\mathbf{w}_k\right\|^2\right] = \frac{(1+\gamma_z)^2}{\gamma_z} \sum_{k=0}^{T-1} \left[\left\|\mathbf{w}_k\right\|^2\right] \\
\sum_{k=0}^{T-1} \rho_k^y \Psi_k &= \sum_{k=0}^{T-1} \rho_k^y \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}^y (1-\eta^x \tau_{i-1}^y)(1+\lambda_i \eta^x)^2} \left\|\mathbf{w}_i\right\|^2 \\
&\leq \frac{1}{\eta\tau_y(1-\eta\tau_y)^2} \sum_{k=0}^{T-1} \left[\left\|\mathbf{w}_k\right\|^2\right] = \frac{(1+\gamma_y)^2}{\gamma_y} \sum_{k=0}^{T-1} \left[\left\|\mathbf{w}_k\right\|^2\right]
\end{aligned} \tag{51}$$

where $\tau_y = \frac{1}{\eta+\eta^y}$ and $\tau_z = \frac{1}{\eta+\eta^z}$. ② holds since $c_3 \leq \frac{3}{c_1^2}$, $c_4 \leq \frac{3}{c_1^2}$, $(\gamma_y-1)^2(\gamma_y+1)^2 \leq \gamma_y^4$ and $(\gamma_z-1)^2(\gamma_z+1)^2 \leq \gamma_z^4$.

Similarly, we can bound

$$\begin{aligned}
\sum_{k=0}^{T-1} \bar{\Pi}_{2,k}^z &= \sum_{k=0}^{T-1} \left[c_5 \rho_k^z (\eta^z - \eta^x)^2 \Phi_k + c_6 \rho_k^y (\eta^y - \eta^x)^2 \Psi_k\right] \\
&\overset{①}{\leq} \left[c_5 (\eta^z - \eta^x)^2 \frac{(1+\gamma_z)^2}{\gamma_z} + c_6 (\eta^y - \eta^x)^2 \frac{(1+\gamma_y)^2}{\gamma_y}\right] \sum_{k=0}^{T-1} \left[\left\|\mathbf{w}_{k-1}\right\|^2\right] \\
&\overset{②}{\leq} \frac{2\eta^2}{c_1^2} \left[\gamma_y^3 + \gamma_z^3\right] \sum_{k=0}^{T-1} \left[\left\|\mathbf{w}_{k-1}\right\|^2\right]
\end{aligned} \tag{52}$$

where ① holds by using above results giving by Lemma 9. ② holds since $c_5 \leq \frac{3}{c_1^2}$, $c_6 \leq \frac{3}{c_1^2}$, $(\gamma_y-1)^2(\gamma_y+1)^2 \leq \gamma_y^4$ and $(\gamma_z-1)^2(\gamma_z+1)^2 \leq \gamma_z^4$.

Therefore, we have

$$\begin{aligned}
\frac{1}{T} \sum_{k=0}^{T-1} \mathbb{E}\left[\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 + \frac{1}{2}\left\|\mathbf{w}_k\right\|^2\right] \leq & \frac{2c_2(1+\lambda\eta)\Delta}{\eta T} + \frac{2c_2\beta_1\sigma^2(1+\lambda\eta)}{c_1 b} \\
& + \frac{6\eta^2 c_2 (1-\beta_1)^2 L^2 (1+\lambda\eta)}{c_1^3 \beta_1^2 T} \left[1 + \gamma_y^3 + \gamma_z^3\right] \sum_{k=0}^{T-1} \left[\left\|\mathbf{w}_{k-1}\right\|^2\right] \\
& + \frac{4\eta^2 c_2 L}{c_1^3 T} \sum_{k=0}^{T-1} \left[\gamma_y^3 + \gamma_z^3\right] \sum_{k=0}^{T-1} \left[\left\|\mathbf{w}_{k-1}\right\|^2\right] \\
\overset{①}{\leq} & \frac{2c_2(1+\lambda\eta)\Delta}{\eta T} + \frac{2c_2\beta_1\sigma^2(1+\lambda\eta)}{c_1 b} + \frac{1}{4T} \sum_{k=0}^{T-1} \left[\left\|\mathbf{w}_k\right\|^2\right]
\end{aligned}$$

where ① holds since we choose proper η and β_1 such that

$$\frac{6\eta^2 c_2 (1-\beta_1)^2 L^2 (1+\lambda\eta)}{c_1^3 \beta_1^2} \left[1 + \gamma_y^3 + \gamma_z^3\right] \leq \frac{1}{8}, \qquad \frac{4\eta^2 c_2 L}{c_1^3} \left[\gamma_y^3 + \gamma_z^3\right] \leq \frac{1}{8} \tag{53}$$

Now we select η and β_1 such that (53) holds:

$$\eta \leq \min\left(\frac{c_1^{1.5}\beta_1}{c_2^{0.5}(1-\beta_1)L\left(1+\gamma_y^3+\gamma_z^3\right)^{0.5}}, \frac{c_1^{1.5}}{c_2^{0.5}L^{0.5}\left(\gamma_y^3+\gamma_z^3\right)^{0.5}}\right)$$

So we arrive at

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\nabla F_k(\mathbf{x}_k)\|^2+\frac{1}{4}\|\mathbf{w}_k\|^2\right] \leq \frac{2c_2(1+\lambda\eta)\Delta}{\eta T}+\frac{2c_2\beta_1(1+\lambda\eta)\sigma^2}{c_1 b} \overset{①}{\leq} \epsilon^2, \tag{54}$$

where we set $T \geq \frac{4c_2(1+\lambda\eta)\Delta}{\eta\epsilon^2}$ and $\beta_1 \leq \frac{c_1 b\epsilon^2}{4c_2(1+\lambda\eta)\sigma^2}$. This result directly bounds

$$\frac{1}{T}\sum_{k=0}^{T-1}\|\mathbf{s}_k\odot(\mathbf{x}_k-\mathbf{x}_{k+1})\|^2=\frac{\eta^2}{T}\sum_{k=0}^{T-1}\frac{1}{(1+\lambda_k\eta)^2}\|\mathbf{m}_k+\lambda\mathbf{x}_k\odot\mathbf{s}_k\|^2 \leq \frac{\eta^2}{T}\sum_{k=0}^{T-1}\|\mathbf{w}_k\|^2 \leq \eta^2\epsilon^2.$$

which directly yields

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

Moreover, from Lemma 13, we have

$$\begin{aligned}
\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{y}_k'-(1+\lambda_{k-1}\eta^y)\mathbf{x}_k\right\|^2 &\overset{①}{\leq}\frac{1}{T}\sum_{k=0}^{T-1}\rho_k^y(\eta^y-\eta)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\
&\overset{②}{=}(\eta^y-\eta)^2\frac{1}{T}\sum_{k=0}^{T-1}\rho_k^y\Psi_k,\\
\frac{1}{T}\sum_{k=0}^{T-1}\|\mathbf{y}_k-\mathbf{x}_k\|^2 &\overset{①}{\leq}\frac{1}{T}\sum_{k=0}^{T-1}\tau_{k-1}^y\rho_k^y\eta(\eta^y-\eta)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\
&\overset{②}{=}(\eta^y-\eta)^2\frac{1}{T}\sum_{k=0}^{T-1}\tau_{k-1}^y\rho_k^y\Psi_k,\\
\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{z}_{k+1}'-(1+\lambda_k\eta^z)\mathbf{x}_{k+1}\right\|^2 &\overset{①}{\leq}\frac{1}{T}\sum_{k=0}^{T-1}\rho_{k+1}^z(\eta^z-\eta^x)^2\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\
&\overset{②}{=}(\eta^z-\eta)^2\frac{1}{T}\sum_{k=0}^{T-1}\rho_k^y\Phi_k,
\end{aligned}$$

where ① holds by using Lemma 13; ② holds by using the definitions of Ψ_k and Φ_k in Eqn. (49). Then in Eqn. (51), we have prove

$$\begin{aligned}
\sum_{k=0}^{T-1}\rho_k^z\Phi_k &\leq \frac{(1+\gamma_z)^2}{\gamma_z}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right] \leq \frac{4(1+\gamma_z)^2\epsilon^2}{\gamma_z},\\
\sum_{k=0}^{T-1}\rho_k^y\Psi_k &\leq \frac{(1+\gamma_y)^2}{\gamma_y}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right] \leq \frac{4(1+\gamma_y)^2\epsilon^2}{\gamma_y}
\end{aligned} \tag{55}$$

Besides, in Lemma 9, we also prove

$$\frac{1}{T}\sum_{k=0}^{T-1}\tau_{k-1}^{y}\rho_k^{y}\Psi_k \le \frac{1}{\eta(1-\eta/(\eta+\eta^y))^2 T}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right] \le \frac{4\epsilon^2}{\eta(1-\eta/(\eta+\eta^y))^2} = \frac{4(1+\gamma_y)^2\epsilon^2}{\eta\gamma_y^2}.$$

So we have

$$\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{y}_k' - (1+\lambda_{k-1}\eta^y)\mathbf{x}_k\right\|^2 \overset{①}{\le} 4\eta^2\gamma_y^3\epsilon^2, \quad \frac{1}{T}\sum_{k=0}^{T-1}\|\mathbf{y}_k - \mathbf{x}_k\|^2 \overset{①}{\le} 4\eta\gamma_y^2\epsilon^2,$$
$$\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{z}_{k+1}' - (1+\lambda_k\eta^z)\mathbf{x}_{k+1}\right\|^2 \overset{①}{\le} 4\eta^2\gamma_z^3\epsilon^2.$$

Then by using similar method, we can upper bound

$$\begin{aligned}
\frac{1}{T}\sum_{k=0}^{T-1}\|\mathbf{z}_{k+1}-\mathbf{x}_{k+1}\|^2 \le& \frac{1}{T}\sum_{k=0}^{T-1}\Pi_{2,k}^z \overset{①}{\le} \frac{2\eta^2}{c_1^2 T}\left[\gamma_y^3+\gamma_z^3\right]\sum_{k=0}^{T-1}\left[\|\mathbf{w}_{k-1}\|^2\right] \le \frac{8\eta^2\epsilon^2}{c_1^2}\left[\gamma_y^3+\gamma_z^3\right] \\
\frac{1}{T}\sum_{k=0}^{T-1}\|\mathbf{z}_{k+1}-\mathbf{z}_k\|^2 \le& \frac{1}{T}\sum_{k=0}^{T-1}\Pi_{1,k}^z \overset{②}{\le} \frac{3\eta^2}{c_1^2 T}\left[1+\gamma_y^3+\gamma_z^3\right]\sum_{k=0}^{T-1}\left[\|\mathbf{w}_{k-1}\|^2\right] \\
&\le \frac{12\eta^2\epsilon^2}{c_1^2}\left[1+\gamma_y^3+\gamma_z^3\right]
\end{aligned}$$

where ① uses Eqn. (52), and ② uses Eqn. (50)

On the other hand, we have

$$\begin{aligned}
\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{m}_k-\nabla F(\mathbf{x}_k)\|^2\right] \le& \frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{m}_k+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k-\nabla F(\mathbf{x}_k)-\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\|^2\right] \\
\le& \frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{m}_k+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\|^2+\|\nabla F(\mathbf{x}_k)+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\|^2\right] \\
=& \frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{m}_k+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\|^2+\|\nabla F_k(\mathbf{x}_k)\|^2\right] \\
\overset{①}{\le}& 2\left[\epsilon^2+\frac{3}{4}\times 4\epsilon^2\right] \le 8\epsilon^2.
\end{aligned}$$

where in ① we use $\mathbf{w}_k = \mathbf{m}_k + \lambda_k\mathbf{x}_k\odot\mathbf{s}_k$. In this way, we have

$$\begin{aligned}
&\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{m}_k-\nabla F(\mathbf{z}_k)\|^2\right] \\
\le& \frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{m}_k-\nabla F(\mathbf{x}_k)\|^2+\|\nabla F(\mathbf{x}_k)-\nabla F(\mathbf{z}_k)\|^2\right] \\
\le& 16\epsilon^2+\frac{2L^2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{x}_k-\mathbf{z}_k\|^2\right] \le 16\epsilon^2+\frac{8\eta^2\epsilon^2}{c_1^2}\left[\gamma_y^3+\gamma_z^3\right] = \frac{(c_1L+32c_2)}{2c_2}\epsilon^2.
\end{aligned}$$

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

$$\beta_1 \leq \frac{c_1 b\epsilon^2}{4c_2(1+\lambda\eta)\sigma^2} = \mathcal{O}\left(\frac{c_1 b\epsilon^2}{c_2\sigma^2}\right),$$

where $c_1 = \nu^{0.5} \leq \|\mathbf{s}_k\|_\infty \leq \left(c_\infty^2 + \nu\right)^{0.5} = c_2$.

For η, it should satisfy

$$\eta \leq \min\left(\frac{c_1^{1.5}\beta_1}{c_2^{0.5}(1-\beta_1)L\left(1+\gamma_y^3+\gamma_z^3\right)^{0.5}}, \frac{c_1^{1.5}}{c_2^{0.5}L^{0.5}\left(\gamma_y^3+\gamma_z^3\right)^{0.5}}, \frac{c_1^2(1+\lambda_k\eta)}{2c_2(L+\lambda_k c_1)}\right)$$

Considering $\lambda\eta << 1$, $\frac{1+\lambda\eta}{1+(1-\mu)\lambda\eta} = a \leq \frac{1}{1-\mu}$, μ is a constant, and $c_1 = \nu^{0.5} << 1$, then we have

$$\eta \leq \mathcal{O}\left(\min\left(\frac{c_1^{1.5}\beta_1}{c_2^{0.5}(\gamma_y^{1.5}+\gamma_z^{1.5})L}, \frac{c_1^{1.5}}{c_2^{0.5}(\gamma_y^{1.5}+\gamma_z^{1.5})L^{0.5}}, \frac{c_1^2}{c_2 L}\right)\right) = \mathcal{O}\left(\frac{c_1^{2.5} b\epsilon^2}{c_2^{1.5}(\gamma_y^{1.5}+\gamma_z^{1.5})\sigma^2 L}\right)$$

where ν is often much smaller than one, and β_1 is very small. For T, we have

$$\begin{aligned} T \geq & \frac{4c_2(1+\lambda\eta)\Delta}{\eta\epsilon^2} = \mathcal{O}\left(\frac{c_2\Delta}{\epsilon^2}\frac{c_2^{1.5}(\gamma_y^{1.5}+\gamma_z^{1.5})\sigma^2 L}{c_1^{2.5} b\epsilon^2}\right) \\ = & \mathcal{O}\left(\frac{c_2^{2.5}(\gamma_y^{1.5}+\gamma_z^{1.5})\sigma^2 L\Delta}{c_1^{2.5} b\epsilon^4}\right) = \mathcal{O}\left(\frac{c_2^{2.5}(\gamma_y^{1.5}+\gamma_z^{1.5})\sigma^2 L\Delta}{\nu^{1.25} b\epsilon^4}\right). \end{aligned}$$

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

$$\mathcal{O}(Tb) = \mathcal{O}\left(\frac{c_2^{2.5}(\gamma_y^{1.5}+\gamma_z^{1.5})\sigma^2 L\Delta}{\nu^{1.25}\epsilon^4}\right).$$

The proof is completed.

■

E.2 Proof of Theorem 5

Proof Recall our definition $F_k(\mathbf{y}_k) = F(\mathbf{z}) + \frac{\lambda_k}{2}\|\mathbf{z}\|_{\mathbf{s}_k}^2 = \mathbb{E}_\zeta[f(\mathbf{z};\zeta)] + \frac{\lambda_k}{2}\|\mathbf{z}\|_{\mathbf{s}_k}^2$, in the (18). By using the smoothness of $f(\theta;\zeta)$, we can extactly follow Eqn. (35) in Appendix D.2 for proving Theorem 2 to obtain

$$F_{k+1}(\mathbf{x}_{k+1}) \leq F_k(\mathbf{x}_k) + \frac{\alpha_l\eta_k^x}{2c_1}\|\nabla F(\mathbf{x}_k) - \mathbf{m}_k\|^2 - \frac{\alpha_s\eta_k^x}{2c_2}\|\nabla F_k(\mathbf{x}_k)\|^2 - \frac{\eta_k^x}{4\alpha_l c_2}\|\mathbf{w}_k\|^2,$$

where we set $\eta_k^x \leq \frac{c_1^2}{2\alpha_l c_2}$ such that $\frac{\alpha_l c_2 L\eta_k^x}{c_1^2} \leq \frac{1}{2}$.

From Lemma 14, by setting $\eta_k^x = \eta$, $\eta_k^y = \eta^y$, $\eta_k^z = \eta^z$ and $\beta_{1,k} = \beta_1$, we have

$$\begin{aligned} \mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\|^2\right] \leq & 2(1-\beta_1)\mathbb{E}\left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{z}_{k-1})\|^2\right] \\ & + \frac{2\Pi_{1,k}^z(1-\beta_1)^2L^2}{\beta_1} + \frac{2\beta_1^2\sigma^2}{b} + 2L\Pi_{2,k}^z, \end{aligned} \tag{56}$$

where

$$\begin{aligned}\Pi_{1,k}^z :=& \frac{3(\xi^x)^2}{(\xi^z)^2(\lambda_{k-1}+\xi^x)^2}\left\|\frac{\mathbf{w}_{k-1}}{\mathbf{s}_{k-1}}\right\|^2 \\ &+ 3(\xi^x+\xi^y+\lambda_{k-1})^2(\delta_{k-1}^z)^2\rho_k^z(\eta^z-\eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2 \\ &+ 3(\xi^y)^4(\delta_{k-1}^y)^2(\delta_{k-1}^z)^2\rho_k^y(\eta^y-\eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2, \\ \Pi_{2,k}^z :=& 2(\xi^z\delta_{k-1}^z)^2\rho_k^z(\eta^z-\eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2 \\ &+ 2(\xi^y)^4(\delta_{k-1}^z\delta_{k-1}^y)^2\rho_k^y(\eta^y-\eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2.\end{aligned}$$

where $\rho_{k+1}^y = \eta^x\tau_{k-1}^y\rho_k^y$, $\rho_1^y = 1$ and $\rho_0^y = 0$; $\rho_{k+1}^z = \eta^x\tau_{k-1}^z\rho_k^z$, $\rho_1^z = 1$ and $\rho_0^z = 0$; $\tau_k^y = \frac{1}{\eta^x+\eta^y+\lambda_k\eta^x\eta^y}$, $\tau_k^z = \frac{1}{\eta^x+\eta^z+\lambda_k\eta^x\eta^z}$, $\delta_k^y = \frac{1}{\xi^x+\xi^y+\lambda_k}$, $\delta_k^z = \frac{1}{\xi^x+\xi^y+\xi^z+\lambda_k}$.

By considering $c_2 \geq \|\mathbf{s}_k\|_\infty \geq c_1$ and setting $\eta_k^x = \eta$, $\eta_k^y = \eta^y = \gamma_y\eta$, $\eta_k^z = \eta^z = \gamma_z\eta$ and $\beta_{1,k} = \beta_1$, we have

$$\begin{aligned}\Pi_{1,k}^z \leq& \bar{\Pi}_{1,k}^z = \frac{3\gamma_z^2\eta^2}{c_1^2(1+\lambda_{k-1}\eta)^2}\|\mathbf{w}_{k-1}\|^2 + c_3\rho_k^z(\eta^z-\eta^x)^2\Phi_k + c_4\rho_k^y(\eta^y-\eta^x)^2\Psi_k, \\ \Pi_{2,k}^z \leq& \bar{\Pi}_{2,k}^z = c_5\rho_k^z(\eta^z-\eta^x)^2\Phi_k + c_6\rho_k^y(\eta^y-\eta^x)^2\Psi_k. \\ \Phi_k =& \sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\|\mathbf{w}_{k-1}\|^2, \quad \Psi_k = \sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\|\mathbf{w}_{k-1}\|^2\end{aligned} \tag{57}$$

in which $c_3 = \frac{3}{c_1^2}\left(1-\frac{\gamma_y}{\gamma_y+\gamma_z+(1+\lambda_{k-1}\eta)\gamma_y\gamma_z}\right)^2$, $c_4 = \frac{3}{c_1^2(1+(1+\lambda_{k-1}\eta)\gamma_y)^2(1+(1+\lambda_{k-1}\eta)\gamma_y+\gamma_y/\gamma_z)^2}$, $c_5 = \frac{2}{((1+\lambda_{k-1}\eta)\gamma_z+1+\gamma_z/\gamma_y)^2}$, and $c_6 = \frac{2}{c_1^2(1+(1+\lambda_{k-1}\eta)\gamma_y)^2(1+(1+\lambda_{k-1}\eta)\gamma_y+\gamma_y/\gamma_z)^2}$.

Therefore, by plugging the results in Eqn. (56) into the upper bound of $F_{k+1}(\mathbf{x}_{k+1})$, we have

$$\begin{aligned}F_{k+1}(\mathbf{x}_{k+1}) \leq& F_k(\mathbf{x}_k) - \frac{\alpha_s\eta}{2c_2}\|\nabla F_k(\mathbf{x}_k)\|^2 - \frac{\eta}{4\alpha_l c_2}\|\mathbf{w}_k\|^2 \\ &+ \frac{\alpha_l\eta(1-\beta_1)}{c_1}\mathbb{E}\left[\|\mathbf{m}_{k-1}-\nabla F(\mathbf{z}_{k-1})\|^2\right] + \frac{\alpha_l\eta\bar{\Pi}_{1,k}^z(1-\beta_1)^2L^2}{c_1\beta_1} \\ &+ \frac{\alpha_l\eta\beta_1^2\sigma^2}{c_1 b} + \frac{\alpha_l\eta L\bar{\Pi}_{2,k}^z}{c_1},\end{aligned} \tag{58}$$

Then, by plugging $\|\mathbf{z}_k-\mathbf{z}_{k-1}\|^2 \leq \Pi_{1,k}^z \leq \bar{\Pi}_{1,k}^z$ in Lemma 11 into Lemma 12, we have

$$\mathbb{E}\left[\|\mathbf{m}_k-\nabla F(\mathbf{z}_k)\|^2\right] \leq (1-\beta_1)\mathbb{E}\left[\|\mathbf{m}_{k-1}-\nabla F(\mathbf{z}_{k-1})\|^2\right] + \frac{(1-\beta_1)^2L^2\bar{\Pi}_{1,k}^z}{\beta_1} + \frac{\beta_1^2\sigma^2}{b}. \tag{59}$$

Then we add Eqn. (58) and $\alpha\times$ (59) as follows:

$$\begin{aligned}
&F_{k+1}(\mathbf{x}_{k+1}) + \alpha\mathbb{E}\left[\left\|\mathbf{m}_k - \nabla F(\mathbf{y}_k)\right\|^2\right] \\
\leq& F_k(\mathbf{x}_k) - \frac{\alpha_s\eta}{2c_2}\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 - \frac{\eta}{4\alpha_l c_2}\left\|\mathbf{w}_k\right\|^2 \\
&+ (1-\beta_1)\left(\frac{\alpha_l\eta}{c_1} + \alpha\right)\mathbb{E}\left[\left\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\right\|^2\right] + \frac{\alpha_l\eta\bar{\Pi}_{1,k}^z(1-\beta_1)^2L^2}{c_1\beta_1} \\
&+ \frac{\alpha_l\eta\beta_1^2\sigma^2}{c_1 b} + \frac{\alpha_l\eta L\bar{\Pi}_{2,k}^z}{c_1} + \frac{\alpha(1-\beta_1)^2L^2\bar{\Pi}_{1,k}^z}{\beta_1} + \frac{\alpha\beta_1^2\sigma^2}{b}
\end{aligned} \tag{60}$$

Then by setting $\alpha = \frac{\alpha_l\eta(1-\beta_1)}{c_1\beta_1}$ and $G_{k+1}(\mathbf{x}_{k+1}) = F_{k+1}(\mathbf{x}_{k+1}) + \frac{\alpha_l\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\left\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\right\|^2\right] = \mathbb{E}_\zeta[f(\mathbf{z};\zeta)] + \frac{\lambda_k}{2}\left\|\mathbf{z}\right\|_{\mathbf{s}_k}^2 + \frac{\alpha_l\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\left\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\right\|^2\right]$, we can obtain

$$\begin{aligned}
&G_{k+1}(\mathbf{x}_{k+1}) \\
\leq& G_k(\mathbf{x}_k) - \frac{\alpha_s\eta}{2c_2}\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 - \frac{\eta}{4\alpha_l c_2}\left\|\mathbf{w}_k\right\|^2 \\
&+ \frac{\alpha_l\eta\bar{\Pi}_{1,k}^z(1-\beta_1)^2L^2}{c_1\beta_1} + \frac{\alpha_l\eta\beta_1^2\sigma^2}{c_1 b} + \frac{\alpha_l\eta L\bar{\Pi}_{2,k}^z}{c_1} + \frac{\alpha_l\eta(1-\beta_1)^3L^2\bar{\Pi}_{1,k}^z}{c_1\beta_1^2} + \frac{\alpha_l\eta(1-\beta_1)\beta_1\sigma^2}{c_1 b} \\
\leq& G_k(\mathbf{x}_k) - \frac{\alpha_s\eta}{2c_2}\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 - \frac{\eta}{4\alpha_l c_2}\left\|\mathbf{w}_k\right\|^2 + \frac{\alpha_l\eta(1-\beta_1)^2L^2\bar{\Pi}_{1,k}^z}{c_1\beta_1^2} + \frac{\alpha_l\eta L\bar{\Pi}_{2,k}^z}{c_1} + \frac{\alpha_l\eta\beta_1\sigma^2}{c_1 b}.
\end{aligned}$$

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

$$\begin{aligned}
&\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\nabla F_k(\mathbf{x}_k)\right\|^2 + \frac{1}{2\alpha_s\alpha_l}\left\|\mathbf{w}_k\right\|^2\right] \\
\leq& \frac{2c_2}{\alpha_s\eta T}\left[G(\mathbf{x}_0) - G(\mathbf{x}_T)\right] + \frac{2\alpha_l c_2\beta_1\sigma^2}{\alpha_s c_1 bT} + \frac{2\alpha_l c_2(1-\beta_1)^2L^2}{\alpha_s c_1\beta_1^2 T}\sum_{k=0}^{T-1}\bar{\Pi}_{1,k}^z + \frac{2\alpha_l c_2 L}{\alpha_s c_1 T}\sum_{k=0}^{T-1}\bar{\Pi}_{2,k}^z \\
\leq& \frac{2c_2\Delta}{\alpha_s\eta T} + \frac{2\alpha_l c_2\beta_1\sigma^2}{\alpha_s c_1 b} + \frac{2\alpha_l c_2(1-\beta_1)^2L^2}{\alpha_s c_1\beta_1^2 T}\sum_{k=0}^{T-1}\bar{\Pi}_{1,k}^z + \frac{2\alpha_l c_2 L}{\alpha_s c_1 T}\sum_{k=0}^{T-1}\bar{\Pi}_{2,k}^z
\end{aligned}$$

where

$$\begin{aligned}
&G(\mathbf{x}_0) - G(\mathbf{x}_T) \\
=& F_0(\mathbf{x}_0) + \frac{\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\left\|\mathbf{m}_{-1} - \nabla F(\mathbf{x}_{-1})\right\|^2\right] - F_T(\mathbf{x}_T) - \frac{\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\left\|\mathbf{m}_{T-1} - \nabla F(\mathbf{x}_{T-1})\right\|^2\right] \\
=& F(\mathbf{x}_0) + \lambda_0\left\|\mathbf{x}_0\right\|_{\mathbf{s}_0} - F(\mathbf{x}_T) - \lambda_T\left\|\mathbf{x}_T\right\|_{\mathbf{s}_T} - \frac{\eta(1-\beta_1)}{c_1\beta_1}\mathbb{E}\left[\left\|\mathbf{m}_{T-1} - \nabla F(\mathbf{x}_{T-1})\right\|^2\right] \\
\leq& F(\mathbf{x}_0) - F(\mathbf{x}_T) \leq \Delta
\end{aligned}$$

where $\Delta = F(\mathbf{x}_0) - F(\mathbf{x}_\star)$; $\mathbf{x}_{-1}$ and $\mathbf{m}_{-1}$ are two virtual points which satisfy $\mathbf{m}_{-1} = \nabla F(\mathbf{x}_{-1})$. Now we can directly use Eqn. (50) and (52) to upper bound $\sum_{k=0}^{T-1}\bar{\Pi}_{1,k}^z$ and $\sum_{k=0}^{T-1}\bar{\Pi}_{2,k}^z$:

$$\sum_{k=0}^{T-1}\bar{\Pi}_{1,k}^z \leq \frac{3\eta^2}{c_1^2}\left[1+\gamma_y^3+\gamma_z^3\right]\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_{k-1}\right\|^2\right], \qquad \sum_{k=0}^{T-1}\bar{\Pi}_{2,k}^z \leq \frac{2\eta^2}{c_1^2}\left[\gamma_y^3+\gamma_z^3\right]\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_{k-1}\right\|^2\right] \tag{61}$$

Therefore, we have

$$\begin{aligned}
&\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\nabla F_k(\mathbf{x}_k)\|^2+\frac{1}{2\alpha_s\alpha_l}\|\mathbf{w}_k\|^2\right]\leq\frac{2c_2\Delta}{\alpha_s\eta T}+\frac{2\alpha_l c_2\beta_1\sigma^2}{\alpha_s c_1 bT}\\
&\quad+\frac{6\alpha_l\eta^2c_2(1-\beta_1)^2L^2}{\alpha_s c_1^3\beta_1^2T}\left[1+\gamma_y^3+\gamma_z^3\right]\sum_{k=0}^{T-1}\left[\|\mathbf{w}_{k-1}\|^2\right]+\frac{4\alpha_l\eta^2c_2L}{\alpha_s c_1^3T}\sum_{k=0}^{T-1}\left[\gamma_y^3+\gamma_z^3\right]\sum_{k=0}^{T-1}\left[\|\mathbf{w}_{k-1}\|^2\right]\\
&\overset{①}{\leq}\frac{2c_2\Delta}{\alpha_s\eta T}+\frac{2\alpha_l c_2\beta_1\sigma^2}{\alpha_s c_1 b}+\frac{1}{4\alpha_s\alpha_l T}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right]
\end{aligned}$$

where ① holds since we choose proper η and β_1 such that

$$\frac{6\alpha_l\eta^2c_2(1-\beta_1)^2L^2}{\alpha_s c_1^3\beta_1^2}\left[1+\gamma_y^3+\gamma_z^3\right]\leq\frac{1}{8\alpha_s\alpha_l},\qquad\qquad\frac{4\alpha_l\eta^2c_2L}{\alpha_s c_1^3}\left[\gamma_y^3+\gamma_z^3\right]\leq\frac{1}{8\alpha_s\alpha_l}\tag{62}$$

Now we select η and β_1 such that (62) holds:

$$\eta\leq\min\left(\frac{c_1^{1.5}\beta_1}{\alpha_l c_2^{0.5}(1-\beta_1)L\left(1+\gamma_y^3+\gamma_z^3\right)^{0.5}},\frac{c_1^{1.5}}{\alpha_l c_2^{0.5}L^{0.5}\left(\gamma_y^3+\gamma_z^3\right)^{0.5}}\right)$$

So we arrive at

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\nabla F_k(\mathbf{x}_k)\|^2+\frac{1}{4\alpha_s\alpha_l}\|\mathbf{w}_k\|^2\right]\leq\frac{2c_2\Delta}{\alpha_s\eta T}+\frac{2\alpha_l c_2\beta_1\sigma^2}{\alpha_s c_1 b}\overset{①}{\leq}\epsilon^2,\tag{63}$$

where we set $T\geq\frac{4c_2\Delta}{\alpha_s\eta\epsilon^2}$ and $\beta_1\leq\frac{\alpha_s c_1 b\epsilon^2}{4\alpha_l c_2\sigma^2}$. This result directly bounds

$$\frac{1}{T}\sum_{k=0}^{T-1}\|\mathbf{s}_k\odot(\mathbf{x}_k-\mathbf{x}_{k+1})\|^2=\frac{\eta^2}{T}\sum_{k=0}^{T-1}\|\mathbf{m}_k+\lambda\mathbf{x}_k\odot\mathbf{s}_k\|^2\leq\frac{\eta^2}{T}\sum_{k=0}^{T-1}\|\mathbf{w}_k\|^2\leq4\alpha_s\alpha_l\eta^2\epsilon^2.$$

which directly yields

$$\frac{1}{T}\sum_{k=0}^{T-1}\|\mathbf{x}_k-\mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2\leq4\alpha_s\alpha_l\eta^2\epsilon^2.$$

Moreover, from Lemma 13, we have

$$\begin{aligned}
\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{y}'_k-(1+\lambda_{k-1}\eta^y)\mathbf{x}_k\right\|^2 &\overset{①}{\leq}\frac{1}{T}\sum_{k=0}^{T-1}\rho_k^y(\eta^y-\eta)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\
&\overset{②}{=}(\eta^y-\eta)^2\frac{1}{c_1^2T}\sum_{k=0}^{T-1}\rho_k^y\Psi_k,\\
\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{y}_k-\mathbf{x}_k\right\|^2 &\overset{①}{\leq}\frac{1}{T}\sum_{k=0}^{T-1}\tau_{k-1}^y\rho_k^y\eta(\eta^y-\eta)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\
&\overset{②}{=}(\eta^y-\eta)^2\frac{1}{c_1^2T}\sum_{k=0}^{T-1}\tau_{k-1}^y\rho_k^y\Psi_k,\\
\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{z}'_{k+1}-(1+\lambda_k\eta^z)\mathbf{x}_{k+1}\right\|^2 &\overset{①}{\leq}\frac{1}{T}\sum_{k=0}^{T-1}\rho_{k+1}^z(\eta^z-\eta^x)^2\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\
&\overset{②}{=}(\eta^z-\eta)^2\frac{1}{c_1^2T}\sum_{k=0}^{T-1}\rho_k^y\Phi_k,
\end{aligned}$$

where ① holds by using Lemma 13; ② holds by using the definitions of Ψ_k and Φ_k in Eqn. (57). Then in Eqn. (51) in Appendix E.1 for proving Theorem 4, we have prove

$$\begin{aligned}
\sum_{k=0}^{T-1}\rho_k^z\Phi_k &\leq \frac{(1+\gamma_z)^2}{\gamma_z}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right]\leq\frac{4\alpha_s\alpha_l(1+\gamma_z)^2\epsilon^2}{\gamma_z},\\
\sum_{k=0}^{T-1}\rho_k^y\Psi_k &\leq \frac{(1+\gamma_y)^2}{\gamma_y}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right]\leq\frac{4\alpha_s\alpha_l(1+\gamma_y)^2\epsilon^2}{\gamma_y}
\end{aligned}\tag{64}$$

Besides, in Lemma 9, we also prove

$$\frac{1}{T}\sum_{k=0}^{T-1}\tau_{k-1}^y\rho_k^y\Psi_k\leq\frac{1}{\eta(1-\eta/(\eta+\eta^y))^2T}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right]\leq\frac{4\alpha_s\alpha_l\epsilon^2}{\eta(1-\eta/(\eta+\eta^y))^2}=\frac{4\alpha_s\alpha_l(1+\gamma_y)^2\epsilon^2}{\eta\gamma_y^2}.$$

So we have

$$\begin{aligned}
&\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{y}'_k-(1+\lambda_{k-1}\eta^y)\mathbf{x}_k\right\|^2\overset{①}{\leq}\frac{4\alpha_s\alpha_l\eta^2\gamma_y^3\epsilon^2}{c_1^2},\quad \frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{y}_k-\mathbf{x}_k\right\|^2\overset{①}{\leq}\frac{4\alpha_s\alpha_l\eta\gamma_y^2\epsilon^2}{c_1^2},\\
&\frac{1}{T}\sum_{k=0}^{T-1}\left\|\mathbf{z}'_{k+1}-(1+\lambda_k\eta^z)\mathbf{x}_{k+1}\right\|^2\overset{①}{\leq}\frac{4\alpha_s\alpha_l\eta^2\gamma_z^3\epsilon^2}{c_1^2}.
\end{aligned}$$

Then by using similar method, we can upper bound

$$\begin{aligned}
\frac{1}{T}\sum_{k=0}^{T-1}\|\mathbf{z}_{k+1}-\mathbf{x}_{k+1}\|^2 \leq & \frac{1}{T}\sum_{k=0}^{T-1}\Pi_{2,k}^z \overset{①}{\leq} \frac{2\eta^2}{c_1^2 T}\left[\gamma_y^3+\gamma_z^3\right]\sum_{k=0}^{T-1}\left[\|\mathbf{w}_{k-1}\|^2\right] \leq \frac{8\alpha_s\alpha_l\eta^2\epsilon^2}{c_1^2}\left[\gamma_y^3+\gamma_z^3\right] \\
\frac{1}{T}\sum_{k=0}^{T-1}\|\mathbf{z}_{k+1}-\mathbf{z}_{k}\|^2 \leq & \frac{1}{T}\sum_{k=0}^{T-1}\Pi_{1,k}^z \overset{②}{\leq} \frac{3\eta^2}{c_1^2 T}\left[1+\gamma_y^3+\gamma_z^3\right]\sum_{k=0}^{T-1}\left[\|\mathbf{w}_{k-1}\|^2\right] \\
\leq & \frac{12\alpha_s\alpha_l\eta^2\epsilon^2}{c_1^2}\left[1+\gamma_y^3+\gamma_z^3\right]
\end{aligned}$$

where ① uses Eqn. (52), and ② uses Eqn. (50)

On the other hand, we have

$$\begin{aligned}
\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{m}_k-\nabla F(\mathbf{x}_k)\|^2\right] \leq & \frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{m}_k+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k-\nabla F(\mathbf{x}_k)-\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\|^2\right] \\
\leq & \frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{m}_k+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\|^2+\|\nabla F(\mathbf{x}_k)+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\|^2\right] \\
= & \frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{m}_k+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k\|^2+\|\nabla F_k(\mathbf{x}_k)\|^2\right] \\
\overset{①}{=} & \frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\frac{1}{\alpha_k}\mathbf{w}_k\right\|^2+\|\nabla F_k(\mathbf{x}_k)\|^2\right] \overset{②}{\leq} 2\left[\epsilon^2+4\frac{\alpha_l}{\alpha_s}\epsilon^2\right] \leq \frac{10\alpha_l}{\alpha_s}\epsilon^2.
\end{aligned}$$

where in ① and ②, we use $\mathbf{w}_k := \alpha_k\mathbf{m}_k+(1+\alpha_k)\lambda_k\mathbf{x}_k\odot\mathbf{s}_k = \alpha_k\mathbf{m}_k$, with $\lambda_k = 0$ and $\alpha_s \leq \alpha_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\sqrt{\mathbf{v}_k+\nu}+\lambda_k\mathbf{x}_k\|_2} \leq \alpha_l$. In this way, we have

$$\begin{aligned}
& \frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{m}_k-\nabla F(\mathbf{z}_k)\|^2\right] \\
\leq & \frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{m}_k-\nabla F(\mathbf{x}_k)\|^2+\|\nabla F(\mathbf{x}_k)-\nabla F(\mathbf{z}_k)\|^2\right] \\
\leq & 4(4\alpha_s\alpha_l+1)\epsilon^2+\frac{2L^2}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\|\mathbf{x}_k-\mathbf{z}_k\|^2\right] \leq \frac{10\alpha_l}{\alpha_s}\epsilon^2+\frac{8\alpha_s\alpha_l\eta^2\epsilon^2}{c_1^2}\left[\gamma_y^3+\gamma_z^3\right]
\end{aligned}$$

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

$$\beta_1 \leq \frac{\alpha_s c_1 b\epsilon^2}{4c_2\sigma^2} = \mathcal{O}\left(\frac{\alpha_s c_1 b\epsilon^2}{c_2\sigma^2}\right),$$

where $c_1 = \nu^{0.5} \leq \|\mathbf{s}_k\|_\infty \leq \left(c_\infty^2+\nu\right)^{0.5} = c_2$.

For η, it should satisfy

$$\begin{aligned}\eta \leq & \mathcal{O}\left(\min\left(\frac{c_1^{1.5}\beta_1}{\alpha_l c_2^{0.5}(1-\beta_1)L\left(1+\gamma_y^3+\gamma_z^3\right)^{0.5}}, \frac{c_1^{1.5}}{\alpha_l c_2^{0.5}L^{0.5}\left(\gamma_y^3+\gamma_z^3\right)^{0.5}}, \frac{c_1^2}{2\alpha_l c_2 L}\right)\right)\\ = & \mathcal{O}\left(\min\left(\frac{c_1^{2.5}b\epsilon^2}{\alpha_l c_2^{1.5}(\gamma_y^{1.5}+\gamma_z^{1.5})\sigma^2 L}, \frac{c_1^{1.5}}{\alpha_l c_2^{0.5}(\gamma_y^{1.5}+\gamma_z^{1.5})L^{0.5}}, \frac{c_1^2}{\alpha_l c_2 L}\right)\right)\\ = & \mathcal{O}\left(\frac{c_1^{2.5}b\epsilon^2}{\alpha_l c_2^{1.5}(\gamma_y^{1.5}+\gamma_z^{1.5})\sigma^2 L}\right)\end{aligned}$$

For T, we have

$$\begin{aligned}T \geq & \frac{4c_2\Delta}{\alpha_s\eta\epsilon^2} = \mathcal{O}\left(\frac{c_2\Delta}{\alpha_s\epsilon^2}\frac{\alpha_l c_2^{1.5}(\gamma_y^{1.5}+\gamma_z^{1.5})\sigma^2 L}{c_1^{2.5}b\epsilon^2}\right)\\ & = \mathcal{O}\left(\frac{\alpha_l c_2^{2.5}(\gamma_y^{1.5}+\gamma_z^{1.5})\sigma^2 L\Delta}{\alpha_s c_1^{2.5}b\epsilon^4}\right) = \mathcal{O}\left(\frac{\alpha_l c_2^{2.5}(\gamma_y^{1.5}+\gamma_z^{1.5})\sigma^2 L\Delta}{\alpha_s\nu^{1.25}b\epsilon^4}\right).\end{aligned}$$

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

$$\mathcal{O}(Tb) = \mathcal{O}\left(\frac{\alpha_l c_2^{2.5}(\gamma_y^{1.5}+\gamma_z^{1.5})\sigma^2 L\Delta}{\alpha_s\nu^{1.25}\epsilon^4}\right).$$

The proof is completed. ■

E.3 Proofs of Theorem 6

Proof Recall our definition $F_k(\theta_k) = F(\theta) + \frac{\lambda_k}{2}\|\theta\|_2^2 = \mathbb{E}_\zeta[f(\boldsymbol{\theta};\zeta)] + \frac{\lambda_k}{2}\|\theta\|_2^2$ in the (18). By setting $\beta_1' = 1-\beta_1$, then we have $\|\mathbf{m}_k\|_\infty \leq c_\infty$ by using Lemma 7 (see Appendix C). Also we define

$$\mathbf{w}_k := \mathbf{m}_k + \lambda\mathbf{x}_k, \qquad \mathbf{x}_{k+1}-\mathbf{x}_k = -\frac{\eta_k^x}{1+\lambda\eta_k^x}(\mathbf{m}_k+\lambda\mathbf{x}_k) = -\frac{\eta_k^x}{1+\lambda\eta_k^x}\mathbf{w}_k.$$

Note in the following, we set all $\lambda_k = \lambda$. Following the proof of Eqn. (44) in the proofs of Theorem 5 in Appendix E.3, we can obtain

$$\begin{aligned}F_{k+1}(\mathbf{x}_{k+1}) \leq & F_k(\mathbf{x}_k) + \frac{\eta_k^x}{2c_1(1+\lambda_k\eta_k^x)}\|\nabla F(\mathbf{x}_k)-\mathbf{m}_k\|^2 - \frac{\eta_k^x}{2c_2(1+\lambda_k\eta_k^x)}\|\nabla F_k(\mathbf{x}_k)\|^2\\ & - \frac{\eta_k^x}{4c_2(1+\lambda_k\eta_k^x)}\|\mathbf{w}_k\|^2,\end{aligned}$$

where we set $\eta_k^x \leq \frac{c_1^2(1+\lambda_k\eta_k^x)}{2c_2(L+\lambda_k c_1)}$ such that $\frac{c_2 L\eta_k^x}{c_1^2(1+\lambda_k\eta_k^x)} + \frac{c_2\lambda_k\eta_k^x}{c_1(1+\lambda_k\eta_k^x)} \leq \frac{1}{2}$.

Then in the following, we can directly follow the proof of Theorem 5. This is because the only difference between accelerated SGD and AdamW is that SGD has no the second-order moment $\mathbf{v}_k$,

while AdamW has. By let $\mathbf{s}_k = \mathbf{1}$ in accelerated AdamW and setting $\beta_1' = 1 - \beta_1$ in accelerated SGD, then they share the exact the same updating rules. So after setting $\beta_1' = 1 - \beta_1$ in accelerated SGD, to follow the proofs of Theorem 5, we only need to verify whether the auxiliary lemmas and the proof process of Theorem 5 hold for $\mathbf{s}_k = \mathbf{1}$. This is the true case. Please check our auxiliary lemmas, including Lemma 7, Lemma 12 $\sim$ 14, and the proof process of Theorem 5. Consider $\mathbf{s}_k = \mathbf{1}$ in accelerated SGD, we have $c_1 := 1 \leq \|\mathbf{s}_k\|_\infty \leq c_2 := 1$. The proof is completed. ■

Appendix F. Proofs of Auxiliary Lemmas

F.1 Proof of Lemma 7

Proof To begin with, we assume that $\forall t \leq k$, it holds

$$\|\mathbf{m}_t\|_\infty \leq c_\infty, \qquad \|\mathbf{v}_t + \nu\|_\infty \leq c_\infty + \nu$$

Then we consider the case where $t = k + 1$ as follows

$$\begin{aligned}
\|\mathbf{m}_{k+1}\|_\infty &= \|(1-\beta_1)\mathbf{m}_k + \beta_1 \mathbf{g}_k\|_\infty \leq (1-\beta_1)\|\mathbf{m}_k\|_\infty + \beta_1 \|\mathbf{g}_k\|_\infty \leq c_\infty, \\
\|\mathbf{v}_{k+1}\|_\infty &= \left\|(1-\beta_2)\mathbf{v}_k + \beta_2 \mathbf{g}_k^2\right\|_\infty \leq (1-\beta_2)\|\mathbf{v}_k\|_\infty + \beta_2 \left\|\mathbf{g}_k^2\right\|_\infty \leq c_\infty^2.
\end{aligned}$$

Then we derive the second results as follows:

$$\left\|\sqrt{\frac{\mathbf{v}_k + \nu}{\mathbf{v}_{k+1} + \nu}}\right\|_\infty = \left\|\sqrt{1 + \frac{\mathbf{v}_k - \mathbf{v}_{k+1}}{\mathbf{v}_{k+1} + \nu}}\right\|_\infty = \left\|\sqrt{1 + \frac{\beta_2(\mathbf{v}_k - \mathbf{g}_k^2)}{\mathbf{v}_{k+1} + \nu}}\right\|_\infty.$$

Therefore, we have

$$1 - \frac{\beta_2 c_\infty^2}{2(c_{s,\infty}^2 + \nu)} < \sqrt{1 - \frac{\beta_2 c_\infty^2}{c_{s,\infty}^2 + \nu}} \leq \left\|\sqrt{\frac{\mathbf{v}_k + \nu}{\mathbf{v}_{k+1} + \nu}}\right\|_\infty \leq \sqrt{1 + \frac{\beta_2 c_\infty^2}{c_{s,\infty}^2 + \nu}} < 1 + \frac{\beta_2 c_\infty^2}{2(c_{s,\infty}^2 + \nu)}.$$

We complete the proof. ■

F.2 Proof of Lemma 8

Proof To begin with, for Win, we have

$$\mathbf{g}_k = \frac{1}{b}\sum_{i=1}^{b} \nabla f(\mathbf{y}_k; \zeta_i); \qquad \mathbf{m}_k = (1-\beta_1)\mathbf{m}_{k-1} + \beta_1 \mathbf{g}_k; \qquad \mathbf{v}_k = (1-\beta_2)\mathbf{v}_{k-1} + \beta_2 \mathbf{g}_k^2.$$

Based on these updating rules, we have

$$
\begin{aligned}
&\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{y}_k)\|^2\right] \\
=&\mathbb{E}\left[\|(1-\beta_1)(\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})) + (1-\beta_1)\nabla F(\mathbf{y}_{k-1}) - \nabla F(\mathbf{y}_k) + \beta_1 \mathbf{g}_k\|^2\right] \\
=&\mathbb{E}\left[\|(1-\beta_1)(\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})) + (1-\beta_1)(\nabla F(\mathbf{y}_{k-1}) - \nabla F(\mathbf{y}_k)) + \beta_1(\mathbf{g}_k - \nabla F(\mathbf{y}_k))\|^2\right] \\
\overset{①}{=}&(1-\beta_1)^2\mathbb{E}\left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\|^2\right] + (1-\beta_1)^2\mathbb{E}\left[\|\nabla F(\mathbf{y}_{k-1}) - \nabla F(\mathbf{y}_k)\|^2\right] \\
&+ \beta_1^2\mathbb{E}\left[\|\mathbf{g}_k - \nabla F(\mathbf{y}_k)\|^2\right] + 2(1-\beta_1)^2\mathbb{E}\langle \mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1}), \nabla F(\mathbf{y}_{k-1}) - \nabla F(\mathbf{y}_k)\rangle \\
\overset{②}{\leq}&(1-\beta_1)^2[1+a]\,\mathbb{E}\left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\|^2\right] + (1-\beta_1)^2\left(1+\frac{1}{a}\right)\mathbb{E}\left[\|\nabla F(\mathbf{y}_{k-1}) - \nabla F(\mathbf{y}_k)\|^2\right] \\
&+ \frac{\beta_1^2\sigma^2}{b} \\
\overset{③}{=}&(1-\beta_1)\mathbb{E}\left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\|^2\right] + \frac{(1-\beta_1)^2}{\beta_1}\mathbb{E}\left[\|\nabla F(\mathbf{y}_{k-1}) - \nabla F(\mathbf{y}_k)\|^2\right] + \frac{\beta_1^2\sigma^2}{b} \\
\leq&(1-\beta_1)\mathbb{E}\left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\|^2\right] + \frac{(1-\beta_1)^2L^2}{\beta_1}\mathbb{E}\left[\|\mathbf{y}_{k-1} - \mathbf{y}_k\|^2\right] + \frac{\beta_1^2\sigma^2}{b},
\end{aligned}
$$

where ① holds since $\mathbb{E}\langle \mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1}), \mathbf{g}_k - \nabla F(\mathbf{y}_k)\rangle = 0$ and $\mathbb{E}\langle \nabla F(\mathbf{y}_{k-1}) - \nabla F(\mathbf{y}_k), \mathbf{g}_k - \nabla F(\mathbf{y}_k)\rangle = 0$; ② holds by $\mathbb{E}\left[\|\mathbf{g}_k - \nabla F(\mathbf{y}_k)\|^2\right] \leq \frac{\sigma^2}{b}$ and ③ holds by setting $a = \frac{\beta}{1-\beta}$. The proof is completed. ■

F.3 Proof of Lemma 9

Proof To begin with, we prove

$$
\begin{aligned}
\phi(y) =& \sum_{k=0}^{T-1} \frac{1}{\rho_{k+1}^y(1-\eta\tau_{k-1}^y)(1+\lambda_k\eta)^2}\|\mathbf{w}_k\|^2\left[\sum_{i=k}^{T-1}\rho_i^y(\tau_{i-1}^y)^2(1+\lambda_{i-1}\eta)^2\right] \\
\overset{①}{\leq}& \frac{a^2}{(1-\eta\tau)}\sum_{k=0}^{T-1}\frac{1}{\rho_{k+1}^y}\|\mathbf{w}_k\|^2\left[\sum_{i=k}^{T-1}\rho_i^y(\tau_{i-1}^y)^2\right] \overset{②}{\leq} \frac{a^2\tau}{\eta(1-\eta\tau)^2}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right],
\end{aligned}
$$

where ① holds, since 1) for $i \geq k$ we have $\frac{1+\lambda_{i-1}\eta}{1+\lambda_k\eta} \leq \frac{1+\lambda_{k-1}\eta}{1+\lambda_k\eta} = \frac{1+\lambda_{k-1}\eta}{1+(1-\mu)\lambda_{k-1}\eta} \leq \frac{1+\lambda\eta}{1+(1-\mu)\lambda\eta} = a \leq \frac{1}{1-\mu}$ and 2) $\frac{1}{1-\eta\tau_{i-1}^y} = \frac{\eta+\eta^y+\lambda_{i-1}\eta^y\eta}{\eta^y+\lambda_{i-1}\eta^y\eta} = 1 + \frac{\eta}{\eta^y+\lambda_{i-1}\eta^y\eta} \leq 1 + \frac{\eta}{\eta^y} = \frac{1}{1-\eta\tau}$ whose minimum is at $\lambda_{i-1} = 0$ and $\tau = \frac{1}{\eta+\eta^y}$; ② holds, since $\sum_{i=k}^{T-1}\rho_i^y(\tau_{i-1}^y)^2 = \frac{1}{\eta}\sum_{i=k}^{T-1}\rho_{i+1}^y\tau_{i-1}^y \leq \frac{\tau}{\eta}\sum_{i=k}^{T-1}\rho_{i+1}^y \leq \frac{\tau}{\eta}\frac{\rho_{k+1}^y(1-\eta^{T-k}\tau^{T-k})}{1-\eta\tau} \leq \frac{\tau\rho_{k+1}^y}{\eta(1-\eta\tau)}$ by using $\rho_{k+1}^y = \eta\tau_{k-1}^y\rho_k^y$.

Similarly, we can bound

$$
\begin{aligned}
&\sum_{k=0}^{T-1}\tau_{k-1}^y\rho_k^y\left[\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\|\mathbf{w}_i\|^2\right] \\
=&\sum_{k=0}^{T-1}\frac{1}{\rho_{k+1}^y(1-\eta\tau_{k-1}^y)(1+\lambda_k\eta)^2}\|\mathbf{w}_k\|^2\sum_{i=k}^{T-1}\left[\tau_{i-1}^y\rho_i^y\right] \overset{①}{\leq} \frac{1}{\eta(1-\eta\tau)^2}\sum_{k=0}^{T-1}\left[\|\mathbf{w}_k\|^2\right]
\end{aligned}
$$

where ② holds, since 1) $\sum_{i=k}^{T-1}\rho_i^y\tau_{i-1}^y=\frac{1}{\eta}\sum_{i=k}^{T-1}\rho_{i+1}^y\leq\frac{1}{\eta}\frac{\rho_{k+1}^y(1-\eta^{T-k}\tau^{T-k})}{1-\eta\tau}\leq\frac{\rho_{k+1}^y}{\eta(1-\eta\tau)}$; and 2) $\frac{1}{1-\eta\tau_{i-1}^y}=\frac{\eta+\eta^y+\lambda_{i-1}\eta^y\eta}{\eta^y+\lambda_{i-1}\eta^y\eta}=1+\frac{\eta}{\eta^y+\lambda_{i-1}\eta^y\eta}\leq 1+\frac{\eta}{\eta^y}=\frac{1}{1-\eta\tau}$ whose minimum is at $\lambda_{i-1}=0$ and $\tau=\frac{1}{\eta+\eta^y}$. Then, we can bound

$$\begin{aligned}\sum_{k=0}^{T-1}\rho_k^y\left[\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\mathbf{w}_i\right\|^2\right]&=\sum_{k=0}^{T-1}\frac{1}{\rho_{k+1}^y(1-\eta\tau_{k-1}^y)(1+\lambda_k\eta)^2}\left\|\mathbf{w}_k\right\|^2\sum_{i=k}^{T-1}[\rho_i^y]\\&\overset{①}{\leq}\frac{1}{\eta\tau(1-\eta\tau)^2}\sum_{k=0}^{T-1}\left[\left\|\mathbf{w}_k\right\|^2\right]\end{aligned}$$

where ② holds since 1) $\rho_{k+1}^y=\eta\tau_{k-1}^y\rho_k^y\leq\eta\tau\rho_k^y$ and $\rho_1^y=1$ and 2) $\sum_{i=k}^{T-1}\rho_i^y\leq\frac{\rho_k^y(1-\eta^{T-k}\tau^{T-k})}{1-\eta\tau}\leq\frac{\rho_k^y}{1-\eta\tau}$ which together give $\frac{1}{\rho_{k+1}^y}\left[\sum_{i=k}^{T-1}\rho_i^y\right]\leq\frac{1}{\rho_{k+1}^y}\frac{\rho_k^y}{1-\eta\tau}\leq\frac{1}{\eta\tau}\frac{1}{1-\eta\tau}\leq\frac{1}{\eta\tau(1-\eta\tau)}$. ■

F.4 Proof of Lemma 10

Proof For Win-accelerated AdamW and Adam, we have $\mathbf{u}_k=\frac{\mathbf{m}_k}{\mathbf{s}_k}$ and $\mathbf{w}_k:=\mathbf{m}_k+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k$. For Win-accelerated LAMB, we have $\mathbf{u}_k=\frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\mathbf{s}_k+\lambda_k\mathbf{x}_k\|_2}\left(\frac{\mathbf{m}_k}{\mathbf{s}_k}+\lambda_k\mathbf{x}_k\right)$, $\mathbf{w}_k:=\alpha_k\mathbf{m}_k+(1+\alpha_k)\lambda_k\mathbf{x}_k\odot\mathbf{s}_k$ where $\alpha_k=\frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\sqrt{\mathbf{v}_k+\nu}+\lambda_k\mathbf{x}_k\|_2}$. Then, we have

$$\begin{aligned}&\mathbf{y}_{k+1}'-(1+\lambda_k\eta_k^y)\mathbf{x}_{k+1}=\mathbf{y}_k-\eta_k^y\mathbf{u}_k-\frac{1+\lambda_k\eta_k^y}{1+\lambda_k\eta_k^x}\left(\mathbf{x}_k-\eta_k^x\mathbf{u}_k\right)\\=&\eta_{k-1}^y\tau_{k-1}^y\mathbf{x}_k+\eta_{k-1}^x\tau_{k-1}^y\mathbf{y}_k'-\eta_k^y\mathbf{u}_k-\frac{1+\lambda_k\eta_k^y}{1+\lambda_k\eta_k^x}\left(\mathbf{x}_k-\eta_k^x\mathbf{u}_k\right)\\=&\eta_{k-1}^x\tau_{k-1}^y\left(\mathbf{y}_k'-(1+\lambda_k\eta_{k-1}^y)\mathbf{x}_k\right)-\left(\eta_k^y-\frac{1+\lambda_k\eta_{k-1}^y}{1+\lambda_k\eta_{k-1}^x}\eta_k^x\right)\mathbf{u}_k+\frac{\lambda_k(\eta_k^x-\eta_k^y)}{1+\lambda_k\eta_k^x}\mathbf{x}_k\\\overset{①}{=}&\eta_{k-1}^x\tau_{k-1}^y\left(\mathbf{y}_k'-(1+\lambda_k\eta_{k-1}^y)\mathbf{x}_k\right)-\left(\eta_k^y-\frac{1+\lambda_k\eta_{k-1}^y}{1+\lambda_k\eta_{k-1}^x}\eta_k^x\right)\frac{\mathbf{w}_k-\lambda_k\mathbf{x}_k\odot\mathbf{s}_k}{\mathbf{s}_k}+\frac{\lambda_k(\eta_k^x-\eta_k^y)}{1+\lambda_k\eta_k^x}\mathbf{x}_k\\=&\eta_{k-1}^x\tau_{k-1}^y\left(\mathbf{y}_k'-(1+\lambda_k\eta_{k-1}^y)\mathbf{x}_k\right)-\left(\eta_k^y-\frac{1+\lambda_k\eta_{k-1}^y}{1+\lambda_k\eta_{k-1}^x}\eta_k^x\right)\frac{\mathbf{w}_k}{\mathbf{s}_k}\\&+\left(\lambda_k\eta_k^y-\frac{1+\lambda_k\eta_{k-1}^y}{1+\lambda_k\eta_{k-1}^x}\lambda_k\eta_k^x+\frac{\lambda_k(\eta_k^x-\eta_k^y)}{1+\lambda_k\eta_k^x}\right)\mathbf{x}_k\end{aligned}$$

where ① holds since $\mathbf{w}_k:=\mathbf{m}_k+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k$ and $\mathbf{u}_k=\frac{\mathbf{m}_k}{\mathbf{s}_k}=\frac{\mathbf{w}_k-\lambda_k\mathbf{x}_k\odot\mathbf{s}_k}{\mathbf{s}_k}$ in Win-accelerated AdamW and Adam; $\mathbf{w}_k:=\alpha_k\mathbf{m}_k+(1+\alpha_k)\lambda_k\mathbf{x}_k\odot\mathbf{s}_k$ and $\mathbf{u}_k=\frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\mathbf{s}_k+\lambda_k\mathbf{x}_k\|_2}\left(\frac{\mathbf{m}_k}{\mathbf{s}_k}+\lambda_k\mathbf{x}_k\right)=\alpha_k\left(\frac{\mathbf{m}_k}{\mathbf{s}_k}+\lambda_k\mathbf{x}_k\right)=\frac{\mathbf{w}_k-\lambda_k\mathbf{x}_k\odot\mathbf{s}_k}{\mathbf{s}_k}$ in Win-accelerated LAMB. Next, we can further obtain

$$\mathbf{y}_{k+1}'-(1+\lambda_k\eta_k^y)\mathbf{x}_{k+1}\overset{①}{=}etax\tau_{k-1}^y\left(\mathbf{y}_k'-(1+\lambda_k\eta^y)\mathbf{x}_k\right)-\frac{\eta^y-\eta^x}{1+\lambda_k\eta^x}\frac{\mathbf{w}_k}{\mathbf{s}_k}\tag{65}$$

where ① holds since we set all $\eta_k^x = \eta^x$ and $\eta_k^y = \eta^y$ which gives $\tau_k^y = \tau = \frac{1}{\eta+\eta^y+\lambda_k\eta\eta^y}$. Therefore, by defining $\rho_{k+1}^y = \eta^x\tau_{k-1}^y\rho_k^y$, $\rho_1^y = 1$ and $\rho_0^y = 0$, then we have

$$\frac{\mathbf{y}'_{k+1} - (1+\lambda_k\eta^y)\mathbf{x}_{k+1}}{\rho_{k+1}^y} = \frac{\mathbf{y}'_k - (1+\lambda_k\eta^y)\mathbf{x}_k}{\rho_k^y} - \frac{1}{\rho_{k+1}^y}\frac{\eta^y-\eta^x}{1+\lambda_k\eta^x}\frac{\mathbf{w}_k}{\mathbf{s}_k}(k \geq 1)$$

For $k = 0$, we have

$$\begin{aligned}\mathbf{y}'_1 - (1+\lambda_0\eta^y)\mathbf{x}_1 =& \mathbf{y}_0 - \eta^y\mathbf{u}_0 - \frac{1+\lambda_0\eta^y}{1+\lambda_0\eta^x}(\mathbf{x}_0 - \eta^x\mathbf{u}_0)\\ =& \mathbf{y}_0 - \eta^y\frac{\mathbf{w}_0 - \lambda_0\mathbf{s}_0\odot\mathbf{x}_0}{\mathbf{s}_0} - \frac{1+\lambda_0\eta^y}{1+\lambda_0\eta^x}\left(\mathbf{x}_0 - \eta^x\frac{\mathbf{w}_0-\lambda_0\mathbf{s}_0\odot\mathbf{x}_0}{\mathbf{s}_0}\right)\\ =& \mathbf{y}_0 - \mathbf{x}_0 - \frac{\eta^y-\eta^x}{1+\lambda_0\eta^x}\frac{\mathbf{w}_0}{\mathbf{s}_0}\end{aligned}$$

In this way, one can obtain

$$\begin{aligned}\frac{\mathbf{y}'_{k+1} - (1+\lambda_k\eta^y)\mathbf{x}_{k+1}}{\rho_{k+1}^y} =& \mathbf{y}_0 - \mathbf{x}_0 - \frac{\eta^y-\eta^x}{1+\lambda_0\eta^x}\frac{\mathbf{w}_0}{\mathbf{s}_0} - \sum_{i=1}^k\frac{1}{\rho_{i+1}^y}\frac{\eta^y-\eta^x}{1+\lambda_i\eta^x}\frac{\mathbf{w}_i}{\mathbf{s}_i}\\ =& -\sum_{i=0}^k\frac{1}{\rho_{i+1}^y}\frac{\eta^y-\eta^x}{1+\lambda_i\eta^x}\frac{\mathbf{w}_i}{\mathbf{s}_i}\end{aligned}$$

where ① hold since $\mathbf{y}_0 = \mathbf{x}_0$ and $\rho_1^y = 1$. Then we can upper bound

$$\begin{aligned}\left\|\frac{\mathbf{y}'_{k+1} - (1+\lambda_k\eta^y)\mathbf{x}_{k+1}}{\rho_{k+1}^y}\right\|^2 =& \left\|\sum_{i=0}^k\frac{\rho_{k+1}^y(1-\eta^x\tau_{i-1}^y)}{\rho_{i+1}^y}\frac{\eta^y-\eta^x}{\rho_{k+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)}\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\ \overset{①}{\leq}& \sum_{i=0}^k\frac{\rho_{k+1}^y(1-\eta^x\tau_{i-1}^y)}{\rho_{i+1}^y}\frac{(\eta^y-\eta^x)^2}{(\rho_{k+1}^y)^2(1-\eta^x\tau_{i-1}^y)^2(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\ =& \frac{(\eta^y-\eta^x)^2}{\rho_{k+1}^y}\sum_{i=0}^k\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\end{aligned}$$

where ① holds since 1) $\sum_{i=0}^k\frac{1-\eta^x\tau_{i-1}^y}{\rho_{i+1}^y} = \sum_{i=0}^k(\frac{1}{\rho_{i+1}^y} - \frac{1}{\rho_i^y}) = \frac{1}{\rho_{k+1}^y}$, and 2) Jensen' inequality. Therefore, we have

$$\left\|\mathbf{y}'_{k+1} - (1+\lambda_k\eta^y)\mathbf{x}_{k+1}\right\|^2 \leq \rho_{k+1}^y(\eta^y-\eta^x)^2\sum_{i=0}^k\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2.$$

Moreover, we can also bound

$$\begin{aligned}\|\mathbf{y}_{k+1} - \mathbf{x}_{k+1}\|^2 =& \left\|\eta^y\tau_k^y\mathbf{x}_{k+1} + \eta^x\tau_k^y\mathbf{y}'_{k+1} - \mathbf{x}_{k+1}\right\|^2 = \eta^x\tau_k^y\left\|\mathbf{y}'_{k+1} - (1+\lambda_k\eta^y)\mathbf{x}_{k+1}\right\|^2\\ \leq& \tau_k^y\rho_{k+1}^y\eta^x(\eta^y-\eta^x)^2\sum_{i=0}^k\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\ =& \xi^y\delta_k^y\rho_{k+1}^y(\eta^y-\eta^x)^2\sum_{i=0}^k\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\end{aligned}$$

where $\tau_k^y\eta^x = \frac{\eta^x}{\eta^x+\eta^y+\lambda_k\eta^x\eta^y} = \xi^y\delta_k^y$ in which $\xi^x = \frac{1}{\eta^x}$, $\xi^y = \frac{1}{\eta^y}$ and $\delta_k^y = \frac{1}{\xi^x+\xi^y+\lambda_k}$.

On the other hand, we have

$$
\begin{aligned}
\|\mathbf{y}_{k+1} - \mathbf{y}_k\| &= \left\|\eta^y\tau_k^y\mathbf{x}_{k+1} + \eta^x\tau_k^y\mathbf{y}'_{k+1} - \mathbf{y}_k\right\| \overset{①}{=} \left\|\eta^y\tau_k^y\mathbf{x}_{k+1} + \eta^x\tau_k^y\mathbf{y}'_{k+1} - \mathbf{y}'_{k+1} - \eta^y\mathbf{u}_k\right\| \\
&= \left\|\eta^y\tau_k^y\mathbf{x}_{k+1} + \eta^x\tau_k^y\mathbf{y}'_{k+1} - \mathbf{y}'_{k+1} - \eta^y\frac{\mathbf{w}_k - \lambda_k\mathbf{x}_k\odot\mathbf{s}_k}{\mathbf{s}_k}\right\| \\
&= \left\|\eta^y\tau_k^y\mathbf{x}_{k+1} + \eta^x\tau_k^y\mathbf{y}'_{k+1} - \mathbf{y}'_{k+1} - \eta^y\frac{\mathbf{w}_k}{\mathbf{s}_k} + \eta^y\lambda_k\mathbf{x}_k\right\| \\
&\overset{②}{=} \left\|(\eta^y\tau_k^y + \eta^y\lambda_k)\mathbf{x}_{k+1} - (1-\eta^x\tau_k^y)\mathbf{y}'_{k+1} - \frac{\eta^y}{1+\lambda_k\eta^x}\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\| \\
&\overset{③}{=} \left\|\eta^y\tau_k^y(1+\lambda_k\eta^x)\left((1+\lambda_k\eta^y)\mathbf{x}_{k+1} - \mathbf{y}'_{k+1}\right) - \frac{\eta^y}{1+\lambda_k\eta^x}\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\| \\
&\leq \eta^y\tau_k^y(1+\lambda_k\eta^x)\left\|(1+\lambda_k\eta^y)\mathbf{x}_{k+1} - \mathbf{y}'_{k+1}\right\| + \frac{\eta^y}{1+\lambda_k\eta^x}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|
\end{aligned}
$$

where ① we plug in $\mathbf{y}'_{k+1} = \mathbf{y}_k - \eta_k^y\mathbf{u}_k$; in ② we plug in $\mathbf{x}_{k+1} = \frac{1}{1+\lambda_k\eta^x}(\mathbf{x}_k - \eta^x\mathbf{u}_k) = \frac{1}{1+\lambda_k\eta^x}\left(\mathbf{x}_k - \eta^x\frac{\mathbf{w}_k-\lambda_k\mathbf{x}_k\odot\mathbf{s}_k}{\mathbf{s}_k}\right) = \mathbf{x}_k - \frac{\eta^x}{1+\lambda_k\eta^x}\frac{\mathbf{w}_k}{\mathbf{s}_k}$ as shown in Eqn. (65); and ③ we have $\eta^y\tau_k^y + \eta^y\lambda_k = \eta^y\tau_k^y(1+\eta^y\lambda_k)(1+\eta^x\lambda_k)$ and $(1-\eta^x\tau_k^y) = \eta^y\tau_k^y(1+\eta^x\lambda_k)$. Then we can upper bound

$$
\begin{aligned}
&\|\mathbf{y}_{k+1} - \mathbf{y}_k\|^2 \\
\leq& 2(\eta^y)^2(\tau_k^y)^2(1+\lambda_k\eta^x)^2\left\|(1+\lambda_k\eta^y)\mathbf{x}_{k+1} - \mathbf{y}'_{k+1}\right\|^2 + \frac{2(\eta^y)^2}{(1+\lambda_k\eta^x)^2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|^2 \\
\leq& \frac{2(\eta^y)^2}{(1+\lambda_k\eta^x)^2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|^2 \\
&+ 2(\eta^y)^2(\tau_k^y)^2(1+\lambda_k\eta^x)^2\rho_{k+1}^y(\eta^y-\eta^x)^2\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2 \\
=& \frac{2(\xi^x)^2}{(\xi^y)^2(\xi^x+\lambda_k)^2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|^2 \\
&+ 2(\delta_k^y)^2(\xi^x+\lambda_k)^2\rho_{k+1}^y(\eta^y-\eta^x)^2\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2
\end{aligned}
$$

where $\eta^y\tau_k^y(1+\lambda_k\eta^x) = \frac{\eta^y\eta^x}{\eta^x+\eta^y+\lambda_k\eta^x\eta^y}(1/\eta^x+\lambda_k)^2 = \delta_k^y(\xi^x+\lambda_k)$ in which $\xi^x = \frac{1}{\eta^x}$, $\xi^y = \frac{1}{\eta^y}$ and $\delta_k^y = \frac{1}{\xi^x+\xi^y+\lambda_k}$. The proof is completed. ■

F.5 Proof of Lemma 11

Proof From Lemma 8, we have

$$\begin{aligned}
&\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{y}_k)\|^2\right] \\
\leq&(1-\beta_{1,k})\mathbb{E}\left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\|^2\right] + \frac{(1-\beta_{1,k})^2 L^2}{\beta_{1,k}}\mathbb{E}\left[\|\mathbf{y}_k - \mathbf{y}_{k-1}\|^2\right] + \frac{\beta_{1,k}^2\sigma^2}{b} \\
\overset{①}{\leq}&(1-\beta_{1,k})\mathbb{E}\left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\|^2\right] + \frac{\Pi_k^y(1-\beta_{1,k})^2 L^2}{\beta_{1,k}} + \frac{\beta_{1,k}^2\sigma^2}{b}
\end{aligned}$$

where in ①, we use the results in Lemma 10 that

$$\|\mathbf{y}_k - \mathbf{y}_{k-1}\|^2 \leq \Pi_{1,k}^y$$

with

$$\begin{aligned}
\Pi_{1,k}^y :=& \frac{2(\eta^y)^2}{(1+\lambda_{k-1}\eta)^2}\left\|\frac{\mathbf{w}_{k-1}}{\mathbf{s}_{k-1}}\right\|^2 \\
&+ 2\rho_k^y(\eta^y)^2(\eta^y-\eta)^2(\tau_{k-1}^y)^2(1+\lambda_{k-1}\eta)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2 \\
=&\frac{2(\xi^x)^2}{(\xi^y)^2(\xi^x+\lambda_k)^2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|^2 \\
&+ 2(\delta_k^y)^2(\xi^x+\lambda_k)^2\rho_{k+1}^y(\eta^y-\eta^x)^2\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2
\end{aligned}$$

Then we have

$$\begin{aligned}
&\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\|^2\right] \\
\leq&2\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{y}_k)\|^2\right] + 2\mathbb{E}\left[\|\nabla F(\mathbf{y}_k) - \nabla F(\mathbf{x}_k)\|^2\right] \\
\leq&2\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{y}_k)\|^2\right] + 2L\mathbb{E}\left[\|\mathbf{y}_k - \mathbf{x}_k\|^2\right] \\
\overset{①}{\leq}&2(1-\beta_{1,k})\mathbb{E}\left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{y}_{k-1})\|^2\right] + \frac{2\Pi_{1,k}^y(1-\beta_{1,k})^2L^2}{\beta_{1,k}} + \frac{2\beta_{1,k}^2\sigma^2}{b} + 2L\Pi_{2,k}^y,
\end{aligned}$$

where in ①, we use the results in Lemma 10 that

$$\begin{aligned}
\|\mathbf{y}_k - \mathbf{x}_k\|^2 \leq \Pi_{2,k}^y :=& \tau_{k-1}^y\rho_k^y\eta(\eta^y-\eta)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2 \\
=&\xi^y\delta_{k-1}^y\rho_k^y(\eta^y-\eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2
\end{aligned}$$

The proof is completed. ■

F.6 Proof of Lemma 12

Proof To begin with, for Win, we have

$$\mathbf{g}_k = \frac{1}{b}\sum_{i=1}^{b}\nabla f(\mathbf{z}_k;\zeta_i); \qquad \mathbf{m}_k = (1-\beta_1)\mathbf{m}_{k-1} + \beta_1\mathbf{g}_k; \qquad \mathbf{v}_k = (1-\beta_2)\mathbf{v}_{k-1} + \beta_2\mathbf{g}_k^2.$$

So we can directly follow the proof of Win in Appendix F.2 to get the desired resutls. The proof is completed. ■

F.7 Proof of Lemma 13

Proof For Win2-accelerated AdamW and Adam, we have $\mathbf{u}_k = \frac{\mathbf{m}_k}{\mathbf{s}_k}$ and $\mathbf{w}_k := \mathbf{m}_k + \lambda_k\mathbf{x}_k \odot \mathbf{s}_k$. For Win2-accelerated LAMB, we have $\mathbf{u}_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\mathbf{s}_k+\lambda_k\mathbf{x}_k\|_2}\big(\frac{\mathbf{m}_k}{\mathbf{s}_k} + \lambda_k\mathbf{x}_k\big)$, $\mathbf{w}_k := \alpha_k\mathbf{m}_k + (1+\alpha_k)\lambda_k\mathbf{x}_k \odot \mathbf{s}_k$ where $\alpha_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\sqrt{\mathbf{v}_k+\nu}+\lambda_k\mathbf{x}_k\|_2}$.

Since $\mathbf{y}'_{k+1} = \mathbf{y}_k - \eta_k^y\mathbf{u}_k$ for both Win and Win2, we can follow the proof of Lemma 10 in Appendix F.4 to prove

$$\begin{aligned}\left\|\mathbf{y}'_{k+1} - (1+\lambda_k\eta^y)\mathbf{x}_{k+1}\right\|^2 \le& \rho_{k+1}^y(\eta^y-\eta^x)^2\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2,\\ \|\mathbf{y}_k - \mathbf{x}_k\|^2 \le& \tau_{k-1}^y\rho_k^y\eta(\eta^y-\eta)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta\tau_{i-1}^y)(1+\lambda_i\eta)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\end{aligned}$$

where $\rho_{k+1}^y = \eta^x\tau_{k-1}^y\rho_k^y$, $\rho_1^y = 1$, $\rho_0^y = 0$, $\tau_k^y = \frac{1}{\eta_k^x+\eta_k^y+\lambda_k\eta_k^x\eta_k^y}$.

Then, since $\mathbf{z}'_{k+1} = \mathbf{z}_k - \eta_k^z\mathbf{u}_k$ which has the same updating rule of $\mathbf{y}'_{k+1} = \mathbf{y}_k - \eta_k^y\mathbf{u}_k$, we can also follow the proof of Lemma 10 in Appendix F.4 to prove

$$\left\|\mathbf{z}'_{k+1} - (1+\lambda_k\eta^z)\mathbf{x}_{k+1}\right\|^2 \le \rho_{k+1}^z(\eta^z-\eta^x)^2\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2,$$

where $\rho_{k+1}^z = \eta^x\tau_{k-1}^z\rho_k^z$, $\rho_1^z = 1$, $\rho_0^z = 0$ and $\tau_k^z = \frac{1}{\eta_k^x+\eta_k^z+\lambda_k\eta_k^x\eta_k^z}$.

On the other hand, we have

$$\begin{aligned}&\mathbf{z}_{k+1} - \mathbf{x}_{k+1}\\ =&\frac{\xi_k^z}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}\mathbf{z}'_{k+1} + \frac{\xi_k^y}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}\mathbf{y}_{k+1} + \frac{\xi_k^x}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}\mathbf{x}_{k+1} - \mathbf{x}_{k+1}\\ =&\frac{\xi_k^z}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}\mathbf{z}'_{k+1} + \frac{\xi_k^y}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}\left(\frac{\xi_k^y}{\xi_k^x+\xi_k^y+\lambda_k}\mathbf{y}'_{k+1} + \frac{\xi_k^x}{\xi_k^x+\xi_k^y+\lambda_k}\mathbf{x}_{k+1}\right)\\ &+\frac{\xi_k^x}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}\mathbf{x}_{k+1} - \mathbf{x}_{k+1}\\ =&\frac{\xi_k^z}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}\left(\mathbf{z}'_{k+1} - \left(1+\frac{\lambda_k}{\xi_k^z}\right)\mathbf{x}_{k+1}\right)\\ &+\frac{\xi_k^y}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}\frac{\xi_k^y}{\xi_k^x+\xi_k^y+\lambda_k}\left(\mathbf{y}'_{k+1} - \left(1+\frac{\lambda_k}{\xi_k^y}\right)\mathbf{x}_{k+1}\right).\end{aligned}$$

Since $\delta_i^y = \frac{1}{\xi_k^x+\xi_k^y+\lambda_k}$, $\delta_i^z = \frac{1}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}$, $\xi_k^x = \frac{1}{\eta_k^x}$, $\xi_k^y = \frac{1}{\eta_k^y}$ and $\xi_k^z = \frac{1}{\eta_k^z}$, we have

$$\begin{aligned}\|\mathbf{z}_{k+1}-\mathbf{x}_{k+1}\|^2 \leq& 2(\xi_k^z\delta_k^z)^2\left\|\mathbf{z}'_{k+1}-(1+\lambda_k\eta_k^z)\,\mathbf{x}_{k+1}\right\|^2 + 2(\xi_k^y)^4(\delta_k^z\delta_k^y)^2\left\|\mathbf{y}'_{k+1}-\left(1+\lambda_k\eta_k^y\right)\mathbf{x}_{k+1}\right\|^2\\ \leq& 2(\xi_k^z\delta_k^z)^2\rho_{k+1}^z(\eta^z-\eta^x)^2\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\ &+2(\xi_k^y)^4(\delta_k^z\delta_k^y)^2\rho_{k+1}^y(\eta^y-\eta^x)^2\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2.\end{aligned}$$

Similarly, we have

$$\begin{aligned}\mathbf{z}_k =& \mathbf{z}'_{k+1}+\eta_k^z\mathbf{u}_k \overset{①}{=} \mathbf{z}'_{k+1}+\eta_k^z\frac{\mathbf{w}_k-\lambda_k\mathbf{x}_k\odot\mathbf{s}_k}{\mathbf{s}_k} = \mathbf{z}'_{k+1}-\eta_k^z\lambda_k\mathbf{x}_k+\eta_k^z\frac{\mathbf{w}_k}{\mathbf{s}_k}\\ \overset{②}{=}& \mathbf{z}'_{k+1}-\eta_k^z\lambda_k\left(\mathbf{x}_{k+1}+\frac{\eta_k^x}{1+\lambda_k\eta_k^x}\frac{\mathbf{w}_k}{\mathbf{s}_k}\right)+\eta_k^z\frac{\mathbf{w}_k}{\mathbf{s}_k} = \mathbf{z}'_{k+1}-\eta_k^z\lambda_k\mathbf{x}_{k+1}+\frac{\eta_k^z}{1+\lambda_k\eta_k^x}\frac{\mathbf{w}_k}{\mathbf{s}_k}\\ =& \mathbf{z}'_{k+1}-\frac{\lambda_k}{\xi_k^z}\mathbf{x}_{k+1}+\frac{\xi_k^x}{\xi_i^z(\lambda_k+\xi_k^x)}\frac{\mathbf{w}_k}{\mathbf{s}_k}\end{aligned}$$

where ① holds since $\mathbf{w}_k := \mathbf{m}_k+\lambda_k\mathbf{x}_k\odot\mathbf{s}_k$ and $\mathbf{u}_k = \frac{\mathbf{m}_k}{\mathbf{s}_k} = \frac{\mathbf{w}_k-\lambda_k\mathbf{x}_k\odot\mathbf{s}_k}{\mathbf{s}_k}$ in Win-accelerated AdamW and Adam; $\mathbf{w}_k := \alpha_k\mathbf{m}_k+(1+\alpha_k)\lambda_k\mathbf{x}_k\odot\mathbf{s}_k$ and $\mathbf{u}_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\mathbf{s}_k+\lambda_k\mathbf{x}_k\|_2}\left(\frac{\mathbf{m}_k}{\mathbf{s}_k}+\lambda_k\mathbf{x}_k\right) = \alpha_k\left(\frac{\mathbf{m}_k}{\mathbf{s}_k}+\lambda_k\mathbf{x}_k\right) = \frac{\mathbf{w}_k-\lambda_k\mathbf{x}_k\odot\mathbf{s}_k}{\mathbf{s}_k}$ in Win-accelerated LAMB; ② holds since

$$\mathbf{x}_{k+1}-\mathbf{x}_k = -\frac{\eta_k^x}{1+\lambda_k\eta_k^x}\frac{\alpha_k\mathbf{m}_k+(1+\alpha_k)\lambda_k\mathbf{x}_k\odot\mathbf{s}_k}{\mathbf{s}_k} = -\frac{\eta_k^x}{1+\lambda_k\eta_k^x}\frac{\mathbf{w}_k}{\mathbf{s}_k}.$$

Next, we can further obtain

$$\begin{aligned}&\mathbf{z}_{k+1}-\mathbf{z}_k\\ =&\frac{\xi_k^z}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}\mathbf{z}'_{k+1}+\frac{\xi_k^y}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}\mathbf{y}_{k+1}+\frac{\xi_k^x}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}\mathbf{x}_{k+1}\\ &-\left(\mathbf{z}'_{k+1}-\frac{\lambda_k}{\xi_k^z}\mathbf{x}_{k+1}+\frac{\xi_k^x}{\xi_i^z(\lambda_k+\xi_k^x)}\frac{\mathbf{w}_k}{\mathbf{s}_k}\right)\\ =&-\frac{\xi_k^x+\xi_k^y+\lambda_k}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}\left(\mathbf{z}'_{k+1}-\left(1+\frac{\lambda_k}{\xi_k^z}\right)\mathbf{x}_{k+1}\right)\\ &+\frac{\xi_k^y}{\xi_k^x+\xi_k^y+\xi_k^z+\lambda_k}\frac{\xi_k^y}{\xi_k^x+\xi_k^y+\lambda_k}\left(\mathbf{y}'_{k+1}-\left(1+\frac{\lambda_k}{\xi_k^y}\right)\mathbf{x}_{k+1}\right)-\frac{\xi_k^x}{\xi_k^z(\lambda_k+\xi_k^x)}\frac{\mathbf{w}_k}{\mathbf{s}_k}\\ =&-(\xi_k^x+\xi_k^y+\lambda_k)\delta_k^z\left(\mathbf{z}'_{k+1}-\left(1+\frac{\lambda_k}{\xi_k^z}\right)\mathbf{x}_{k+1}\right)+(\xi_k^y)^2\delta_k^y\delta_k^z\left(\mathbf{y}'_{k+1}-\left(1+\frac{\lambda_k}{\xi_k^y}\right)\mathbf{x}_{k+1}\right)\\ &-\frac{\xi_k^x}{\xi_k^z(\lambda_k+\xi_k^x)}\frac{\mathbf{w}_k}{\mathbf{s}_k}.\end{aligned}$$

So we can bound $\|\mathbf{z}_{k+1}-\mathbf{z}_k\|$ as follows:

$$
\begin{aligned}
&\|\mathbf{z}_{k+1}-\mathbf{z}_k\|^2\\
\leq&3(\xi_k^x+\xi_k^y+\lambda_k)^2(\delta_k^z)^2\left\|\mathbf{z}'_{k+1}-\left(1+\frac{\lambda_k}{\xi_k^z}\right)\mathbf{x}_{k+1}\right\|^2+3(\xi_k^y)^4(\delta_k^y)^2(\delta_k^z)^2\left\|\mathbf{y}'_{k+1}-\left(1+\frac{\lambda_k}{\xi_k^y}\right)\mathbf{x}_{k+1}\right\|^2\\
&+3\frac{(\xi_k^x)^2}{(\xi_k^z)^2(\lambda_k+\xi_k^x)^2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|^2\\
\leq&3\frac{(\xi_k^x)^2}{(\xi_k^z)^2(\lambda_k+\xi_k^x)^2}\left\|\frac{\mathbf{w}_k}{\mathbf{s}_k}\right\|^2\\
&+3(\xi_k^x+\xi_k^y+\lambda_k)^2(\delta_k^z)^2\rho_{k+1}^z(\eta^z-\eta^x)^2\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\
&+3(\xi_k^y)^4(\delta_k^y)^2(\delta_k^z)^2\rho_{k+1}^y(\eta^y-\eta^x)^2\sum_{i=0}^{k}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2.
\end{aligned}
$$

The proof is completed. ■

F.8 Proof of Lemma 14

Proof From Lemma 12, we have

$$
\begin{aligned}
&\mathbb{E}\left[\|\mathbf{m}_k-\nabla F(\mathbf{z}_k)\|^2\right]\\
\leq&(1-\beta_{1,k})\mathbb{E}\left[\|\mathbf{m}_{k-1}-\nabla F(\mathbf{z}_{k-1})\|^2\right]+\frac{(1-\beta_{1,k})^2L^2}{\beta_{1,k}}\mathbb{E}\left[\|\mathbf{z}_k-\mathbf{z}_{k-1}\|^2\right]+\frac{\beta_{1,k}^2\sigma^2}{b}\\
\overset{①}{\leq}&(1-\beta_{1,k})\mathbb{E}\left[\|\mathbf{m}_{k-1}-\nabla F(\mathbf{z}_{k-1})\|^2\right]+\frac{\Pi_{1,k}^z(1-\beta_{1,k})^2L^2}{\beta_{1,k}}+\frac{\beta_{1,k}^2\sigma^2}{b}
\end{aligned}
$$

where in ①, we use the results in Lemma 13 that

$$
\|\mathbf{z}_k-\mathbf{z}_{k-1}\|^2\leq\Pi_{1,k}^z
$$

with

$$
\begin{aligned}
\Pi_{1,k}^z:=&3\frac{(\xi_{k-1}^x)^2}{(\xi_{k-1}^z)^2(\lambda_{k-1}+\xi_{k-1}^x)^2}\left\|\frac{\mathbf{w}_{k-1}}{\mathbf{s}_{k-1}}\right\|^2\\
&+3(\xi_{k-1}^x+\xi_{k-1}^y+\lambda_{k-1})^2(\delta_{k-1}^z)^2\rho_k^z(\eta^z-\eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2\\
&+3(\xi_{k-1}^y)^4(\delta_{k-1}^y)^2(\delta_{k-1}^z)^2\rho_k^y(\eta^y-\eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2.
\end{aligned}
$$

Then we have

$$\begin{aligned}
&\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\|^2\right] \\
\leq& 2\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{z}_k)\|^2\right] + 2\mathbb{E}\left[\|\nabla F(\mathbf{z}_k) - \nabla F(\mathbf{x}_k)\|^2\right] \\
\leq& 2\mathbb{E}\left[\|\mathbf{m}_k - \nabla F(\mathbf{z}_k)\|^2\right] + 2L\mathbb{E}\left[\|\mathbf{z}_k - \mathbf{x}_k\|^2\right] \\
\overset{①}{\leq}& 2(1-\beta_{1,k})\mathbb{E}\left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{z}_{k-1})\|^2\right] + \frac{2\Pi_{1,k}^2(1-\beta_{1,k})^2L^2}{\beta_{1,k}} + \frac{2\beta_{1,k}^2\sigma^2}{b} + 2L\Pi_{2,k}^z,
\end{aligned}$$

where in ①, we use the results in Lemma 13 that

$$\begin{aligned}
\|\mathbf{z}_k - \mathbf{x}_k\|^2 \leq \Pi_{2,k}^z :=& 2(\xi_{k-1}^z\delta_{k-1}^z)^2\rho_k^z(\eta^z - \eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^z(1-\eta^x\tau_{i-1}^z)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2 \\
&+ 2(\xi_{k-1}^y)^4(\delta_{k-1}^z\delta_{k-1}^y)^2\rho_k^y(\eta^y - \eta^x)^2\sum_{i=0}^{k-1}\frac{1}{\rho_{i+1}^y(1-\eta^x\tau_{i-1}^y)(1+\lambda_i\eta^x)^2}\left\|\frac{\mathbf{w}_i}{\mathbf{s}_i}\right\|^2.
\end{aligned}$$

The proof is completed. ■

References

Kwangjun Ahn and Suvrit Sra. Understanding Nesterov's acceleration via proximal point method. In *Symposium on Simplicity in Algorithms*, pages 117–130. SIAM, 2022.

Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. In *Innovations in Theoretical Computer Science*, 2017.

Rohan Anil, Vineet Gupta, Tomer Koren, Kevin Regan, and Yoram Singer. Scalable second order optimization for deep learning. *arXiv preprint arXiv:2002.09018*, 2020.

Rohan Anil, Sandra Gadanho, Da Huang, Nijith Jacob, Zhuoshu Li, Dong Lin, Todd Phillips, Cristina Pop, Kevin Regan, Gil I Shamir, et al. On the factory floor: Ml engineering for industrial-scale ads recommendation models. In *ACM Recommender Systems*, 2022.

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 *Conf. on Learning Theory*, pages 242–299, 2020.

Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *Mathematical Programming, Series A*, 2022.

Nikhil Bansal and Anupam Gupta. Potential-function proofs for gradient methods. *Theory of Computing*, 15(1):1–32, 2019.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Yoshua Bengio, Nicolas Boulanger-Lewandowski, and Razvan Pascanu. Advances in optimizing recurrent networks. In *Int'l Conf. on Acoustics, Speech and Signal Processing*, pages 8624–8628, 2013.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 9650–9660, 2021.

Augustin Cauchy et al. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

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 *Proc. Int'l Joint Conf. Artificial Intelligence*, pages 3267–3275, 2021.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. MMDetection: Open MMLab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, 2019.

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. In *Int'l Conf. Learning Representations*, 2021.

Timothy Dozat. Incorporating Nesterov momentum into Adam. In *Int'l Conf. Learning Representations Workshops*, 2016.

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

Jia Deng; Wei Dong; Richard Socher; Li-Jia Li; Kai Li; Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 248–255, 2009.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *Int'l Conf. Learning Representations*, 2021.

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*, pages arXiv–2104, 2021.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *Int'l Conf. Machine Learning*, pages 1842–1850, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *IEEE Int'l Conf. on Computer Vision*, pages 2961–2969, 2017.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Junhyung Lyle Kim, Panos Toulis, and Anastasios Kyrillidis. Convergence and stability of the stochastic proximal point algorithm with momentum. In *Learning for Dynamics and Control Conference*, pages 1034–1047. PMLR, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Int'l Conf. Learning Representations*, 2015.

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *Int'l Conf. Machine Learning*, pages 5905–5914, 2021.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. European Conf. Computer Vision*, pages 740–755, 2014.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *Int'l Conf. Learning Representations*, 2019.

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 *IEEE Int'l Conf. on Computer Vision*, pages 10012–10022, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Int'l Conf. Learning Representations*, 2018.

Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. In *Int'l Conf. Learning Representations*, 2018.

Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Using Large Corpora*, 273, 1994.

Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

Zachary Nado, Justin M Gilmer, Christopher J Shallue, Rohan Anil, and George E Dahl. A large batch optimizer reality check: Traditional, generic optimizers suffice across batch sizes. *arXiv preprint arXiv:2102.06356*, 2021.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. In *Int'l Conf. Learning Representations*, 2018.

Tong Zhang Rie Johnson. Accelerating stochastic gradient descent using predictive variance reduction. In *Proc. Conf. Neural Information Processing Systems*, pages 315–323, 2013.

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

R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on Control and Optimization*, 14(5):877–898, 1976.

Tara N. Sainath, Abdel rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. Deep convolutional neural networks for LVCSR. In *Int'l Conf. on Acoustics, Speech and Signal Processing*, pages 8614–8618. IEEE, 2013.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Int'l Conf. Machine Learning*, pages 1139–1147, 2013.

Tieleman Tijmen and Hinton Geoffrey. Lecture 6.5-rmsprop: Divide the gradient by a running 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 *Int'l Conf. Machine Learning*, pages 10347–10357, 2021.

Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021.

Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. *Proc. Conf. Neural Information Processing Systems*, 2017.

Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *Proc. Conf. Neural Information Processing Systems*, 34: 30392–30400, 2021.

Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, and Shuicheng Yan. Adan: Adaptive Nesterov momentum algorithm for faster optimizing both CNNs and ViTs. *arXiv preprint arXiv:2208.06677*, 2022.

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 *Int'l Conf. Learning Representations*, 2019.

Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 10819–10829, 2022a.

Weihao Yu, Chenyang Si, Pan Zhou, Mi Luo, Yichen Zhou, Jiashi Feng, Shuicheng Yan, and Xinchao Wang. Metaformer baselines for vision. *arXiv preprint arXiv:2210.13452*, 2022b.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *IEEE Int'l Conf. on Computer Vision*, pages 6023–6032, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *Int'l Conf. Learning Representations*, 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, Jiashi Feng, Chao Ma, Caiming Xiong, Steven Chu Hong Hoi, et al. Towards theoretically understanding why SGD generalizes better than Adam in deep learning. In *Proc. Conf. Neural Information Processing Systems*, volume 33, pages 21285–21296, 2020a.

Pan Zhou, Caiming Xiong, Richard Socher, and Steven Hoi. Theory-inspired path-regularized differential network architecture search. In *Proc. Conf. Neural Information Processing Systems*, 2020b.

Pan Zhou, Hanshu Yan, Xiaotong Yuan, Jiashi Feng, and Shuicheng Yan. Towards understanding why lookahead generalizes better than SGD and beyond. *Proc. Conf. Neural Information Processing Systems*, 34:27290–27304, 2021.

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

Pan Zhou, Xingyu Xie, and YAN Shuicheng. Win: Weight-decay-integrated nesterov acceleration for adaptive gradient algorithms. In *Int'l Conf. Learning Representations*, 2023.

Pan Zhou, Xingyu Xie, Zhoucheng Lin, and Shuicheng Yan. Towards understanding convergence and generalization of AdamW. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2024.

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. In *Proc. Conf. Neural Information Processing Systems*, volume 33, pages 18795–18806, 2020.