
Win: Weight-Decay-Integrated Nesterov Acceleration for Adaptive Gradient Algorithms

Pan Zhou¹ Xingyu Xie^{1,2} Shuicheng Yan¹
¹Sea AI Lab ²Peking University
{zhoupan,xyxie,yansc}@sea.com xyxie@pku.cn

Abstract

Training deep networks on increasingly large-scale datasets is computationally challenging. In this work, we explore the problem of “*how to accelerate the convergence of adaptive gradient algorithms in a general manner*”, and aim at providing practical insights to boost the training efficiency. To this end, we propose an effective Weight-decay-Integrated Nesterov acceleration (Win) for adaptive algorithms to enhance their convergence speed. Taking AdamW and Adam as examples, we minimize a dynamical loss per iteration which combines the vanilla training loss and a dynamic regularizer inspired by proximal point method (PPM) to improve the convexity of the problem. Then we respectively use the first- and second-order Taylor approximations of vanilla loss to update the variable twice while fixing the above dynamic regularization brought by PPM. In this way, we arrive at our Win acceleration (like Nesterov acceleration) for AdamW and Adam that uses a conservative step and a reckless step to update twice and then linearly combines these two updates for acceleration. Next, we extend this Win acceleration to LAMB and SGD. Our transparent acceleration derivation could provide insights for other accelerated methods and their integration into adaptive algorithms. Besides, we prove the convergence of Win-accelerated adaptive algorithms by taking AdamW and Adam as examples. Experimental results testify the faster convergence speed and superior performance of our Win-accelerated AdamW, Adam, LAMB and SGD over their vanilla counterparts on vision classification tasks and language modeling tasks with CNN and Transformer backbones.

1 Introduction

Deep neural networks (DNNs) are effective to model realistic data and have been successfully applied to various applications, *e.g.* image classification [1–10] and speech recognition [11–14]. Typically, their training models can be formulated as the following nonconvex optimization problem:

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

where z is the model parameters; sample ζ is drawn from a data distribution \mathcal{D} ; the loss f is differentiable; λ is a constant. To solve problem (1), SGD [15, 16] uses its compositional structure to efficiently estimate gradient via minibatch data, and has become a dominant algorithm to train DNNs. However, on sparse data or ill-conditioned problems, SGD suffers from slow convergence speed [17, 18], as it scales the gradient uniformly in all parameter coordinate and ignores the data or problem properties on each coordinate. To resolve this issue, recent work has proposed a variety of adaptive methods, *e.g.* Adam [17] and AdamW [19], that scale each gradient coordinate according to the current geometry curvature of the loss $F(z)$. This coordinate-wise scaling greatly accelerates the optimization convergence and helps them, *e.g.* Adam and AdamW, become much more popular in DNN training.

Unfortunately, along with the increasing scale of both datasets and models, efficient DNN training even with SGD or adaptive algorithms has become very 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 dominant popularity across many DNNs. Heavy ball acceleration [20] and Nesterov acceleration [21] are widely used in SGD but are rarely studied in adaptive algorithms.

Contributions: In this work, based on a recent Nesterov-type acceleration formulation [22] and proximal point method (PPM) [23], we propose a new *Weight-decay-Integrated Nesterov acceleration* (Win) to accelerate adaptive algorithms. By taking AdamW and Adam as examples, we follow PPM spirit and minimize a dynamically regularized loss which combines vanillas loss and a dynamical regularization, and independently approximate the vanilla loss by its first- and second-order Taylor expansions to update the variable twice while fixing the above dynamic regularization. As a result, we achieve at our Win acceleration, a Nesterov-alike acceleration, for AdamW and Adam that uses a conservative step and a reckless step to update twice and then linearly combines these two updates for acceleration. Then we extend Win acceleration to LAMB [24] and SGD. This transparent acceleration derivation may motivate other accelerations and provide examples to introduce other accelerations into adaptive algorithms. Moreover, we analyze the convergence of Win-accelerated adaptive algorithms to justify their convergence superiority by using AdamW & Adam as examples.

Finally, experimental results on both vision classification tasks and language modeling tasks show that our Win-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.

2 Weight-decay-Integrated Nesterov Acceleration

In deterministic optimization, one widely used optimization-stabilizing and acceleration approach is proximal point method (PPM) [23, 25]. At the k -th iteration, PPM optimizes an ℓ_2 -regularized loss $F(\mathbf{z}) + \frac{1}{2\eta_k} \|\mathbf{z} - \mathbf{z}_{k-1}\|_2^2$ instead of the vanilla loss $F(\mathbf{z})$. This change enhances the convexity of the problem, accelerating and also stabilizing optimization [26, 27]. To make the ℓ_2 -regularized problem solvable iteratively, PPM approximates $F(\mathbf{z})$ by its first- or second-order Taylor expansion to get a close-form solution. At below, we borrow the idea in PPM to induce a *Weight-decay-Integrated Nesterov acceleration* (Win) for adaptive algorithms by using AdamW and Adam as examples.

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

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

where $\mathbf{m}_0 = \mathbf{g}_0$, $\mathbf{v}_0 = \mathbf{g}_0^2$, $\beta_1 \in [0, 1]$ and $\beta_2 \in [0, 1]$. For brevity, with a small scaler $\nu > 0$, we define

$$\mathbf{s}_k = \sqrt{\mathbf{v}_k + \nu}, \quad \mathbf{u}_k = \mathbf{m}_k / \sqrt{\mathbf{v}_k + \nu}. \quad (3)$$

Then following PPM spirit, at the k -th iteration, we minimize a regularized loss $F(\mathbf{x}) + \frac{1}{2\eta} \|\mathbf{x} - \mathbf{x}_k\|_{\mathbf{s}_k}^2$. 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 allows us to handle adaptive algorithms as shown below Eqn. (4), and 2) it also helps increase the problem convexity to speed up the convergence. To make problem solvable iteratively, we approximate $F(\mathbf{z})$ by its first-order Taylor expansion at the point \mathbf{z}_k and update \mathbf{x}_{k+1} as

$$\mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}} F(\mathbf{z}_k) + \langle \mathbf{m}_k, \mathbf{x} - \mathbf{z}_k \rangle + \frac{1}{2\eta_k} \|\mathbf{x} - \mathbf{x}_k\|_{\mathbf{s}_k}^2 + \frac{\lambda}{2} \|\mathbf{x}\|_{\mathbf{s}_k}^2 = \frac{1}{1 + \lambda\eta_k} (\mathbf{x}_k - \eta_k \mathbf{u}_k), \quad (4)$$

where $\|\mathbf{x}\|_{\mathbf{s}_k} = \sqrt{\langle \mathbf{x}, \mathbf{s}_k * \mathbf{x} \rangle}$ with element-wise product $*$, \mathbf{m}_k is used to approximate the full gradient $\nabla F(\mathbf{z}_k)$ for Taylor expansion. We add a small regularization $\frac{\lambda}{2} \|\mathbf{x}\|_{\mathbf{s}_k}^2$, as 1) it can improve the generalization in practice [19, 28]; 2) it allows us to derive Adam ($\lambda = 0$) and AdamW ($\lambda > 0$). If $\lambda = 0$, the updating (4) becomes the exact Adam algorithm. If $\lambda > 0$, the updating (4) can approximate the updating rule $\mathbf{x}_{k+1} = (1 - \lambda\eta_k)\mathbf{x}_k - \eta_k\mathbf{u}_k$ of AdamW. This is because consider $\lambda\eta_k$ is small in practice, we approximate $(1 + \lambda\eta_k)^{-1} = 1 - \lambda\eta_k + \mathcal{O}(\lambda^2\eta_k^2)$ and thus $\frac{1}{1 + \lambda\eta_k} (\mathbf{x}_k - \eta_k\mathbf{u}_k) = [1 - \lambda\eta_k + \mathcal{O}(\lambda^2\eta_k^2)]\mathbf{x}_k - [\eta_k - \mathcal{O}(\lambda\eta_k^2) + \mathcal{O}(\lambda^3\eta_k^3)]\mathbf{u}_k$ which becomes AdamW

Algorithm 1: Win-Accelerated AdamW, Adam and LAMB

Input: initialization $\mathbf{x}_0 = \mathbf{z}_0 = \mathbf{0}$, step size $\{(\eta_k, \bar{\eta}_k)\}_{k=0}^T$, moment parameters $\{\beta_1, \beta_2\}$.

```

1 while  $k < T$  do
2    $\mathbf{g}_k = \frac{1}{b} \sum_{i=1}^b \nabla f(\mathbf{z}_k; \zeta_i)$ 
3    $\mathbf{m}_k = (1 - \beta_1)\mathbf{m}_{k-1} + \beta_1\mathbf{g}_k$  /*  $\mathbf{m}_0 = \mathbf{g}_0$  */
4    $\mathbf{v}_k = (1 - \beta_2)\mathbf{v}_{k-1} + \beta_2\mathbf{g}_k^2$  /*  $\mathbf{v}_0 = \mathbf{g}_0^2$  */
5    $\mathbf{u}_k = \frac{\mathbf{m}_k}{\sqrt{\mathbf{v}_k + \nu}}$  for AdamW and Adam,  $\mathbf{u}_k = \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{m}_k/\sqrt{\mathbf{v}_k + \nu}\|_2} \frac{\mathbf{m}_k}{\sqrt{\mathbf{v}_k + \nu}}$  for LAMB
6    $\mathbf{x}_{k+1} = \frac{1}{1 + \lambda\eta_k} (\mathbf{x}_k - \eta_k\mathbf{u}_k)$ 
7    $\mathbf{z}_{k+1} = \bar{\eta}_k\tau_k\mathbf{x}_{k+1} + \eta_k\tau_k(\mathbf{z}_k - \bar{\eta}_k\mathbf{u}_k)$  with  $\tau_k = \frac{1}{\eta_k + \bar{\eta}_k + \lambda\eta_k\bar{\eta}_k}$ 
8 end while

```

by ignoring $\mathcal{O}(\eta_k^2)$ and $\mathcal{O}(\eta_k^3)$. This is one reason that we adopt the regularizer $\|\mathbf{x} - \mathbf{x}_k\|_{\mathbf{s}_k}^2$ in (4) instead of the ℓ_2 -regularization in PPM, as we can flexibly derive Adam and AdamW.

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

$$\begin{aligned} \mathbf{z}_{k+1} &= \operatorname{argmin}_{\mathbf{z}} F(\mathbf{z}_k) + \langle \mathbf{m}_k, \mathbf{z} - \mathbf{z}_k \rangle + \frac{1}{2\bar{\eta}_k}\|\mathbf{z} - \mathbf{z}_k\|_{\mathbf{s}_k}^2 + \frac{1}{2\eta_k}\|\mathbf{z} - \mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2 + \frac{\lambda}{2}\|\mathbf{z}\|_{\mathbf{s}_k}^2 \\ &= \bar{\eta}_k\tau_k\mathbf{x}_{k+1} + \eta_k\tau_k(\mathbf{z}_k - \bar{\eta}_k\mathbf{u}_k), \end{aligned} \quad (5)$$

where $\tau_k = \frac{1}{\eta_k + \bar{\eta}_k + \lambda\eta_k\bar{\eta}_k}$, \mathbf{m}_k can approximate $\nabla F(\mathbf{x}_k)$ as shown in Theorem 1 in Appendix B.

For more clear, we introduce a virtual sequence $\mathbf{y}_{k+1} = \mathbf{z}_k - \bar{\eta}_k\mathbf{u}_k$ in Win, and rewrite (5) as

$$\mathbf{x}_{k+1} = (1 + \lambda\eta_k)^{-1}(\mathbf{x}_k - \eta_k\mathbf{u}_k), \quad \mathbf{y}_{k+1} = \mathbf{z}_k - \bar{\eta}_k\mathbf{u}_k, \quad \mathbf{z}_{k+1} = \bar{\eta}_k\tau_k\mathbf{x}_{k+1} + \eta_k\tau_k\mathbf{y}_{k+1}. \quad (6)$$

See detailed steps in Algorithm 1. Interestingly, Win acceleration is similar to Nesterov-type acceleration, since they both use a conservative step η_k and a reckless step $\bar{\eta}_k$ 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{z}_{k+1} .

Our Win-acceleration is quit simple and efficient, as our accelerated AdamW/Adam only adds an extra simple algorithmic step, i.e. the 7th step in Algorithm 1, on vanilla AdamW/Adam. Moreover, for the only extra hyper-parameter, the reckless step $\bar{\eta}_k$, in Algorithm 1 over AdamW/Adam, we always set it as $\bar{\eta}_k = 2\eta_k$, which works well in our all experiments.

Extension to LAMB and SGD. Here we generalize Win acceleration to LAMB [24] and SGD [15]. For LAMB, it scales the update \mathbf{u}_k of AdamW in Eqn. (3) so that \mathbf{u}_k is at the same magnitude of the network weight \mathbf{x}_k . That is, it changes the update rule $\mathbf{x}_{k+1} = (1 - \lambda\eta_k)\mathbf{x}_k - \eta_k\mathbf{m}_k/\mathbf{s}_k$ in AdamW to $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \frac{\|\mathbf{x}_k\|_2}{\|\mathbf{r}_k + \lambda\mathbf{x}_k\|_2}(\mathbf{r}_k + \lambda\mathbf{x}_k)$ where $\mathbf{r}_k = \mathbf{m}_k/\mathbf{s}_k$. This modification is to avoid too large or small update, improving optimization efficiency. To extend Win acceleration to LAMB, we inherit this scaling spirit, and scale the update \mathbf{u}_k in (3) to the following one:

$$\mathbf{u}_k = (\|\mathbf{x}_k\|_2/\|\mathbf{m}_k/\mathbf{s}_k\|_2) \cdot (\mathbf{m}_k/\mathbf{s}_k). \quad (7)$$

Next, we can follow Eqn. (4) and (5) to update, and summarize detailed steps in Algorithm 1.

For SGD, applying Win acceleration to it is quite direct. Specifically, the only algorithmic difference between SGD and AdamW on the ℓ_2 -regularized problems is that SGD has no second-order moment \mathbf{v}_k while AdamW has. So we can borrow the acceleration framework of AdamW to accelerate SGD by setting $\mathbf{s}_k = \mathbf{1} \in \mathbb{R}^d$ in Eqn. (3), (4) and (5), and obtain WIN-accelerated SGD:

$$\mathbf{m}_k = \beta_1\mathbf{m}_{k-1} + \beta_1'\mathbf{g}_k, \quad \mathbf{x}_{k+1} = \frac{1}{1 + \lambda\eta_k}(\mathbf{x}_k - \eta_k\mathbf{m}_k), \quad \mathbf{z}_{k+1} = \bar{\eta}_k\tau_k\mathbf{x}_{k+1} + \eta_k\tau_k(\mathbf{z}_k - \bar{\eta}_k\mathbf{m}_k), \quad (8)$$

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-M in Pytorch).

Convergence Analysis. Theorem 1 in Appendix B analyzes the convergence of Win-accelerated adaptive algorithms to justify their convergence superiority by using AdamW & Adam as examples.

Table 1: ImageNet top-1 accuracy (%) of ResNet50&101 whose official optimizer is LAMB due to the stronger data augmentation for better performance. * is reported in [29].

Epoch	ResNet50				ResNet101			
	100	200	300	avg.	100	200	300	avg.
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-M	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}
Adam	76.9	78.4	78.8	78.1	78.4	80.2	80.6	79.7
Adam-Win	77.8	78.8	79.3	78.7 _{+0.6}	79.2	80.6	81.0	80.3 _{+0.6}
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}
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}

Table 2: ImageNet top-1 accuracy (%) of ViT and PoolFormer whose default optimizers are both AdamW. * and \diamond are respectively reported in [28] and [30].

Epoch	ViT-S			ViT-B			PoolFormer-S12		
	150	300	avg.	150	300	avg.	150	300	avg.
SGD-M	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}
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	80.5	80.3 _{+0.9}	75.6	77.1	76.4 _{+1.1}
AdamW	78.3	79.8*	79.1	79.5	81.8*	80.7	75.2	77.1*	76.2
AdamW-Win	79.3	80.8	80.1 _{+1.0}	81.0	82.2	81.6 _{+0.9}	76.7	77.6	77.2 _{+1.0}
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}

3 Experiments

For clarity, we call our accelerated algorithm “X-Win”, where “X” denotes vanilla optimizers. In all experiments, our accelerated algorithms, *e.g.* AdamW-Win, always use the default hyper-parameters of vanilla optimizers, *e.g.* moment parameters β_1 and β_2 in AdamW; and set $\bar{\eta}_k = 2\eta_k$.

Results on ResNets and ViTs. Table 1 reports accuracy of ResNets under the setting in [29], and Table 2 gives the performance of ViT [2] and PoolFormer [30]. Our accelerated algorithms always outperform their corresponding non-accelerated version. On ResNet, LAMB-Win achieves remarkable improvement over the official optimizer LAMB for this setting; SGD-Win also surpasses heavy-ball accelerated SGD (SGD-H) and Nesterov accelerated SGD (SGD-M). On ViTs, our accelerated algorithms consistently outperform the corresponding non-accelerated counterparts. Fig. 1 shows the faster convergence behaviors of our accelerated algorithms over non-accelerated counterparts which could benefit their better performance under the same computational cost.

Results on Transformer-XL. Table 3 shows that under different training steps on WikiText-103 dataset, our accelerated Adam-Win always achieves lower test PPL than the official Adam optimizer of Transformer-XL-base, and improves 1.5 average test PPL over Adam.

Table 3: Test PPL of Transformer-XL-B. * is officially reported.

Transformer-XL	Training Steps			
	50k	100k	200k	avg.
Adam	28.5	25.5	24.2*	26.7
Adam-Win	26.7	25.0	24.0	25.2 _{+1.5}

4 Conclusion

In this work, we adopt 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.

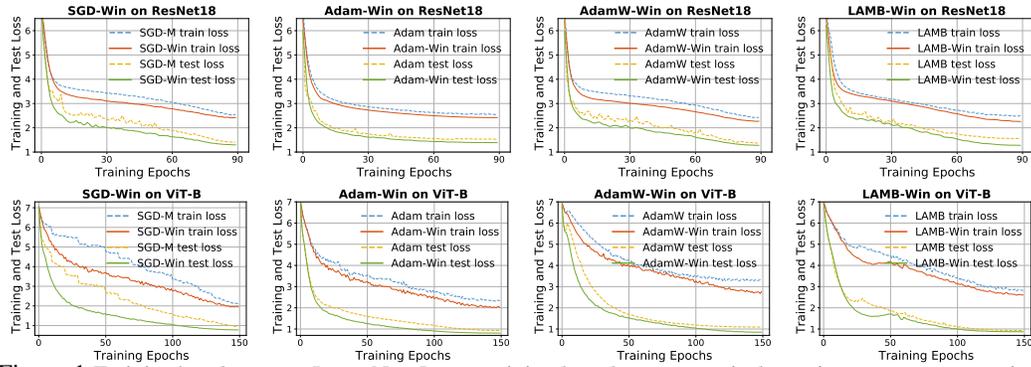


Figure 1: Training/test losses on ImageNet. Larger training loss than test one is due to its strong augmentation.

References

- [1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [3] Pan Zhou, Xiaotong Yuan, Huan Xu, Shuicheng Yan, and Jiashi Feng. Efficient meta learning via mini-batch proximal update. In *Neural Information Processing Systems*, 2019.
- [4] Pan Zhou, Caiming Xiong, Richard Socher, and Steven Hoi. Theory-inspired path-regularized differential network architecture search. In *Neural Information Processing Systems*, 2020.
- [5] Pan Zhou, Caiming Xiong, Xiaotong Yuan, and Steven Hoi. A theory-driven self-labeling refinement method for contrastive representation learning. In *Neural Information Processing Systems*, 2021.
- [6] Yuxuan Liang, Pan Zhou, Roger Zimmermann, and Shuicheng Yan. Dualformer: Local-global stratified transformer for efficient video recognition. In *European Conference on Computer Vision*, pages 577–595. Springer, 2022.
- [7] 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.
- [8] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10819–10829, 2022.
- [9] Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng Yan. Inception transformer. *arXiv preprint arXiv:2205.12956*, 2022.
- [10] 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*, 2022.
- [11] T. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran. Deep convolutional neural networks for LVCSR. In *ICASSP*, pages 8614–8618. IEEE, 2013.
- [12] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10):1533–1545, 2014.
- [13] Yubei Xiao, Ke Gong, Pan Zhou, Guolin Zheng, Xiaodan Liang, and Liang Lin. Adversarial meta sampling for multilingual low-resource speech recognition. In *Association for the Advancement of Artificial Intelligence*, 2021.
- [14] Guolin Zheng, Yubei Xiao, Ke Gong, Pan Zhou, Xiaodan Liang, and Liang Lin. Wav-bert: Cooperative acoustic and linguistic representation learning for low-resource speech recognition. *arXiv preprint arXiv:2109.09161*, 2021.
- [15] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

- [16] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234. PMLR, 2016.
- [17] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *Int’l Conf. Learning Representations*, 2014.
- [18] Robin M Schmidt, Frank Schneider, and Philipp Hennig. Descending through a crowded valley—benchmarking deep learning optimizers. In *International Conference on Machine Learning*, pages 9367–9376. PMLR, 2021.
- [19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.
- [20] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.
- [21] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.
- [22] Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.
- [23] Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.
- [24] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations*, 2019.
- [25] R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.
- [26] 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.
- [27] Pan Zhou, Hanshu Yan, Xiaotong Yuan, Jiashi Feng, and Shuicheng Yan. Towards understanding why lookahead generalizes better than sgd and beyond. *Advances in Neural Information Processing Systems*, 34:27290–27304, 2021.
- [28] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.
- [29] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021.
- [30] 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. *arXiv preprint arXiv:2111.11418*, 2021.
- [31] Jinghui Chen, Dongruo Zhou, Yiqi Tang, Ziyang Yang, Yuan Cao, and Quanquan Gu. Closing the generalization gap of adaptive gradient methods in training deep neural networks. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3267–3275, 2021.
- [32] Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *Advances in Neural Information Processing Systems*, 33:18795–18806, 2020.
- [33] 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 *International Conference on Learning Representations*, 2019.
- [34] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 248–255, 2009.
- [35] Jürgen Schmidhuber, Sepp Hochreiter, et al. Long short-term memory. *Neural Comput*, 9(8):1735–1780, 1997.
- [36] Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Using Large Corpora*, 273, 1994.

- [37] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.
- [38] Sangdoon Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032, 2019.
- [39] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.
- [40] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.
- [41] Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. *arXiv preprint arXiv:1902.09843*, 2019.
- [42] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.
- [43] Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, and Shuicheng Yan. Adan: Adaptive nesterov momentum algorithm for faster optimizing both cnns and vits. *Axriv*, 2022.

Appendix

The appendix contains the technical proofs of convergence results and some additional experimental details of the paper entitled “Win: Weight-Decay-Integrated Nesterov Acceleration for Adaptive Gradient Algorithms”. It is structured as follows. Appendix A provides more experimental results and details. In Appendix C, we define some necessary notations for our analysis. Then Appendix D provides some auxiliary lemmas throughout this document. Then Appendix E presents the proof of the convergence results in Sec. 1, i.e., the proof of Theorem 1. Finally, Appendix F provides the proofs of some auxiliary lemmas in Appendix D.

A More Experimental Results and Details

A.1 More Experimental Results

Results on ResNet18. Here we follow the conventional supervised training setting used in ResNets [1] and evaluate our accelerated algorithms on ImageNet [34]. Due to limited space, we defer the hyper-parameter settings of the four accelerated algorithms in Table 4 into Appendix A.2.

Table 4 shows that our 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%, 2.2% and 2.6% improvement over their corresponding non-accelerated counterparts, AdamW, Adam and LAMB. Moreover, SGD-Win improves SGD-H (i.e. SGD + heavy ball) by 3.4%, and also surpasses SGD-M by 0.5%, where SGD-M is the Nesterov-accelerated SGD, also validating the superiority of our Win acceleration. Besides, our accelerated algorithms, i.e. SGD-Win, AdamW-Win and LAMB-Win, beat several other optimizers, e.g. AdaBound, Radam [33], Nadam, Padam [31] and AdaBelief, in which Nadam uses Nesterov acceleration to estimate its first-order gradient moment. Actually, LAMB-Win sets a new SoTA top-1 accuracy on ResNet18. All these results show the strong compatibility and superiority of our Win-acceleration in adaptive algorithms.

Robust Analysis. For the only extra hyper-parameter $\bar{\eta}_k$ in our accelerated algorithms over their non-accelerated counterparts, in experiments, we always set $\bar{\eta}_k = \gamma \eta_k$, where $\gamma = 2$ determines the relation between the reckless step $\bar{\eta}_k$ and the conservative step η_k . Here we investigate the effects of γ to the accelerated algorithms on ResNet50 by taking AdamW-Win and LAMB-Win as examples because of their superior performance. Table 5 shows the stable performance of AdamW-Win and LAMB-Win when tuning γ in a relatively large range, thus testifying the robustness of AdamW-Win and LAMB-Win to the hyper-parameter γ .

Results on LSTM. We follow AdaBelief to test our accelerated algorithms via training three-layered LSTM [35] on the Penn TreeBank dataset [36] for 200 epochs. See optimization and training details in Appendix A.2.

From Table 6, one can observe that our Win-accelerated algorithms consistently surpass the corresponding non-accelerated counterparts, and actually bring 1.2 overall average perplexity improvement over the four non-accelerated counterparts.

Table 4: ImageNet top-1 accuracy (%) of ResNet18. *, † and ‡ are respectively reported in [31], [32] and [33].

AdaBound	68.1*	Radam	67.7*
Nadam	68.8	Padam	70.1*
SGD-H	67.3	AdaBelief	70.1†
SGD-M	70.2*	Adam	66.5‡
SGD-Win	70.7 _{+0.5}	Adam-Win	68.7 _{+2.2}
AdamW	67.9*	LAMB	68.5
AdamW-Win	71.0 _{+3.1}	LAMB-Win	71.1 _{+2.6}

Table 5: Effects of γ to top-1 accuracy (%) of AdamW-Win and LAMB-Win on ResNet50.

γ	1.5	2	3	4	6	8
AdamW-Win	77.9	78.0	78.0	77.9	78.1	78.0
LAMB-Win	78.3	78.4	78.4	78.4	78.5	78.3

Table 6: Test perplexity of LSTM on Penn Treebank. * is reported by AdaBelief [32].

AdaBound	63.6*	Radam	70.0*
Yogi	67.5*	AdaBelief	61.2*
SGD-H	67.4	Padam	63.2*
SGD-M	63.8*	Adam	64.3*
SGD-Win	61.6 _{+2.2}	Adam-Win	62.7 _{+1.6}
AdamW	67.0*	LAMB	66.8
AdamW-Win	66.5 _{+0.5}	LAMB-Win	66.2 _{+0.6}

A.2 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 [1] and [29], to this section.

For 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 reckless step $\bar{\eta}_k$ is always $2\times$ larger than its conservative step η_k for all iterations, i.e. $\bar{\eta}_k = 2\eta_k$. For AdamW-Win, LAMB-Win, Adam-Win, 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, its other key parameters, such as “grad averaging” and “trust clip”, also adopt the default ones in vanilla LAMB. For SGD-Win, it uses 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 [1] and evaluate our accelerated algorithms on ImageNet [34]. For data augmentation in [1], 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.

Settings on ResNet50&101. For these two networks, we use “A2 training recipe” in [29] to train them, since this training setting uses stronger data augmentation and largely improves CNNs’ performance. Specifically, the data augmentation in [29] uses random crop, horizontal flipping with probability, Mixup with parameter 0.1 [37], CutMix with parameter 1.0 and probability 0.5 [38], and RandAugment [39] with $M = 7$, $N = 2$ and $MSTD = 0.5$. Moreover, it often use binary cross-entropy (BCE) loss for training.

On both ResNet50 and ResNet101, for base learning rate, we respectively set it as 2×10^{-3} , 8×10^{-3} , 1×10^{-3} , and 0.8 for AdamW-Win, LAMB-Win, Adam-Win and SGD-Win. Moreover, we follow the default setting and use cosine learning rate decay. On both ResNet50 and ResNet101, for weight decay, we respectively set it as 5×10^{-2} , 2×10^{-2} , 10^{-5} , and 5×10^{-4} for AdamW-Win, LAMB-Win, Adam-Win and SGD-Win. On both ResNet50 and ResNet101, for warm-up epoch number, we respectively set it as 5, 5, 10, 5 for AdamW-Win, LAMB-Win, Adam-Win and SGD-Win.

Settings on ViT and PoolFormer. We follow the widely used official training setting of ViTs [28, 30]. For this setting, data augmentation includes random crop, horizontal flipping with probability, Mixup with parameter 0.8 [37], CutMix with parameter 1.0 and probability 0.5 [38], RandAugment [39] 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.

On both ViT-S and ViT-B, for base learning rate, we respectively set it as 2×10^{-3} , 5×10^{-3} , 1×10^{-4} , and 0.8 for AdamW-Win, LAMB-Win, Adam-Win and SGD-Win. Moreover, we follow the default setting and use cosine learning rate decay. On both ResNet50 and ResNet101, for weight decay, we respectively set it as 5×10^{-2} , 2×10^{-2} , 10^{-5} , and 5×10^{-4} for AdamW-Win, LAMB-Win, Adam-Win and SGD-Win. On both ResNet50 and ResNet101, for warm-up epoch number, we respectively set it as 5, 60, 30, 5 for AdamW-Win, LAMB-Win, Adam-Win and SGD-Win. For AdamW-Win, following the default setting in AdamW, its minibatch size is 1024 for ViT-S and 512 for ViT-B. For all other accelerated optimizer, their minibatch sizes are always 1024.

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. Moreover, we follow the default setting and divide the learning rate by 10 at epoch 100 and 145. For weight decay, we respectively 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. 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 base learning rate, we set it as 4×10^{-4} for Adam-Win. Moreover, we follow the default setting and use cosine learning rate decay. For weight

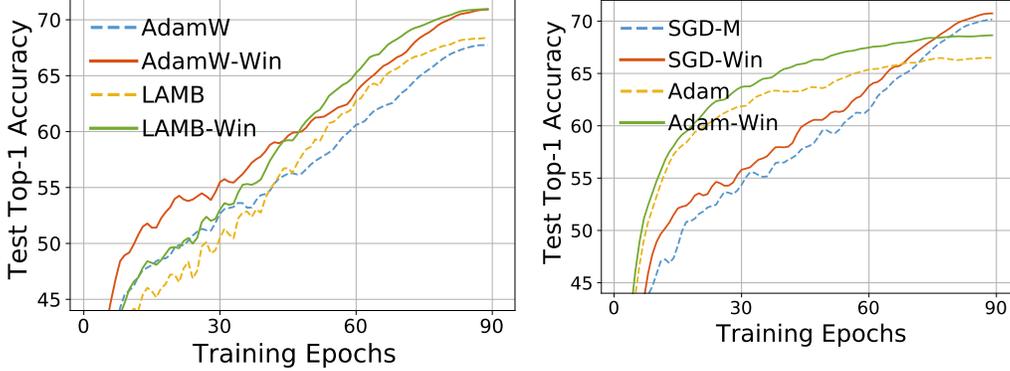


Figure 2: Test accuracy curve of SGD-Win and Adam-Win on ResNet18. See the curves of AdamW-Win and LAMB-Win in manuscript.

decay, we set it as 10^{-6} for Adam-Win. For warm-up steps, we set it as 2000. Following the default setting, we set minibatch size as 60×4 .

Test accuracy curves of SGD-Win and Adam-Win on ResNet18. Here we investigate the convergence behaviors of our accelerated algorithms and hope to explain their better test performance over their non-accelerated counterparts. In each sub-figure pair of 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, *e.g.* AdamW-Win, show much faster convergence behaviors than their non-accelerated counterparts, *e.g.* AdamW. Moreover, SGD-Win also converges faster than Nesterove-accelerated SGD, *i.e.* SGD-M. We also plot the curves of test accuracy in Fig. 2, showing the superior convergence speed of AdamW-Win and LAMB-Win over their corresponding non-accelerated versions. Fig. 2 also reveals SGD-Win and Adam-Win enjoy faster convergence than their non-accelerated counterparts in terms of test accuracy. So these faster convergence behaviors could contribute to our accelerated algorithms for their higher performance over non-accelerated algorithms under the same computational cost.

B Convergence Analysis

Here we investigate the convergence performance of Win-accelerated algorithms by taking accelerated AdamW, Adam and SGD as examples, as these algorithms are more preferably used in deep learning field. Moreover, since we aim to accelerate deep network training which is highly nonconvex problems, we focus on analyzing nonconvex problems to accord with the practical setting.

For analysis, we follow previous works on optimizer, *e.g.* [17, 40–43], and introduce necessary assumptions.

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

Assumption 2 (*Unbiased and bounded gradient estimation*). The gradient estimation \mathbf{g}_k is unbiased, *i.e.* for $\forall k$, $\mathbb{E}[\mathbf{g}_k] = \nabla F(\mathbf{z}_k)$, and its magnitude and variance are bounded, namely, for $\forall k$, $\|\mathbf{g}_k\|_\infty \leq c_\infty$ and $\mathbb{E}[\|\nabla F(\mathbf{z}_k) - \mathbf{g}_k\|_2] \leq \sigma$ with two universal constants c_∞ and σ .

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

$$F_k(\mathbf{z}) = F(\mathbf{z}) + \frac{\lambda_k}{2} \|\mathbf{z}\|_{\mathbf{s}_k}^2 = \mathbb{E}_\zeta[f(\mathbf{z}; \zeta)] + \frac{\lambda_k}{2} \|\mathbf{z}\|_{\mathbf{s}_k}^2, \quad (9)$$

where \mathbf{s}_k is given in (3). To obtain (9), following PPM spirit and Eqn. (4), one can approximate $F(\mathbf{z})$ by its first-order Taylor expansion, and obtain Eqn. (4) with \mathbf{x} replaced by \mathbf{z} to update $\mathbf{z}_{k+1} = \frac{1}{1+\lambda_k\eta_k}(\mathbf{z}_k - \eta_k\mathbf{m}_k/\mathbf{s}_k)$. Since $\lambda_k\eta_k$ is very small, one can follow the discussion below Eqn. (4) and approximate \mathbf{z}_{k+1} as $\mathbf{z}_{k+1} = (1 - \lambda_k\eta_k)\mathbf{z}_k - \eta_k\mathbf{m}_k/\mathbf{s}_k$ which becomes the update rule of AdamW. This is the reason that our analysis on Win-accelerated AdamW involves a dynamic loss $F_k(\mathbf{z})$ in (9). Note, for Win-accelerated Adam ($\lambda_k=0$), $F_k(\mathbf{z})$ degenerates to the vanilla loss $F(\mathbf{z})$.

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 proof in Appendix E.

Theorem 1. Suppose Assumptions 1 and 2 hold, and $\mathbf{x}_* \in \operatorname{argmin}_{\mathbf{x}} F(\mathbf{x})$. Let $\bar{\eta}_k = \gamma \eta_k$, $\gamma > 1$, $\eta_k = \eta \leq \mathcal{O}\left(\frac{\nu^{1.25} b \epsilon^2}{c^{1.5} \gamma^{2.5} \sigma^2 L}\right)$, $\beta_1 \leq \mathcal{O}\left(\frac{\nu^{0.5} b \epsilon^2}{c \sigma^2}\right)$, $\beta_2 \in (0, 1)$, $c = (c_\infty^2 + \nu)^{0.5}$, $\lambda_k = \lambda \left(1 - \frac{\beta_2 c_\infty^2}{\nu}\right)^k$ ($k > 0$) and $\lambda_0 = 0$ with a constant $\lambda > 0$. Then after $T = \mathcal{O}\left(\frac{c_\infty^{2.5} \gamma^{2.5} \sigma^2 L \Delta}{\nu^{1.25} b \epsilon^4}\right)$ iterations with minibatch size b and $\Delta = F(\mathbf{x}_0) - F(\mathbf{x}_*)$, the sequence $\{(\mathbf{x}_k, \mathbf{z}_k)\}_{k=0}^T$ 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

$$\frac{1}{T} \sum_{k=0}^{T-1} \mathbb{E} \left[\|\nabla F_k(\mathbf{x}_k)\|_2^2 + \frac{1}{4} \|\mathbf{m}_k + \lambda_k \mathbf{x}_k * \mathbf{s}_k\|_2^2 \right] \leq \epsilon^2.$$

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

$$\frac{1}{T} \sum_{k=0}^{T-1} \max \left\{ \mathbb{E} \|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\|_2^2, \mathbb{E} \|\mathbf{m}_k - \nabla F(\mathbf{z}_k)\|_2^2 \right\} \leq \left(16 + \frac{1}{2c} \nu^{0.5} L\right) \epsilon^2.$$

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 iterations, the average gradient $\frac{1}{T} \sum_{k=0}^{T-1} \mathbb{E} [\|\nabla F_k(\mathbf{x}_k)\|_2^2]$ is upper bounded by ϵ^2 , guaranteeing the algorithmic convergence. Theorem 1 b) indicates the gradient moment \mathbf{m}_k can well estimate the full gradient $\nabla F(\mathbf{z}_k)$ and also $\nabla F(\mathbf{x}_k)$ because of their small distances, guaranteeing the good Taylor approximation used in Eqn. (4) and (5).

C 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 $\nu \leq \|\mathbf{v}_k + \nu\|_\infty \leq c_\infty^2 + \nu$ in Lemma 2 (see Appendix D), for brevity, let

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

Also we define

$$\mathbf{w}_k := \mathbf{m}_k + \lambda \mathbf{x}_k * \mathbf{s}_k, \quad \mathbf{x}_{k+1} - \mathbf{x}_k = -\frac{\eta_k}{1 + \lambda \eta_k} \frac{\mathbf{m}_k + \lambda \mathbf{x}_k * \mathbf{s}_k}{\mathbf{s}_k} = -\frac{\eta_k}{1 + \lambda \eta_k} \frac{\mathbf{w}_k}{\mathbf{s}_k}.$$

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 (10):

$$\begin{cases} \mathbf{g}_k = \frac{1}{b} \sum_{i=1}^b \nabla f(\mathbf{z}_k; \zeta_i); \\ \mathbf{m}_k = (1 - \beta_1) \mathbf{m}_k + \beta_1 \mathbf{g}_k; \\ \mathbf{v}_k = (1 - \beta_2) \mathbf{v}_k + \beta_2 \mathbf{g}_k^2; \\ \mathbf{x}_{k+1} = \frac{1}{1 + \lambda \eta_k} \left(\mathbf{x}_k - \eta_k \frac{\mathbf{m}_k}{\mathbf{s}_k} \right) \\ \mathbf{y}_{k+1} = \mathbf{z}_k - \bar{\eta}_k \frac{\mathbf{m}_k}{\mathbf{s}_k} \\ \mathbf{z}_{k+1} = \bar{\eta}_k \tau_k \mathbf{x}_{k+1} + \eta_k \tau_k \mathbf{y}_{k+1} \end{cases} \quad (10)$$

where $\mathbf{m}_0 = \mathbf{g}_0$ and $\mathbf{v}_0 = \mathbf{g}_0^2$.

For analysis, we further define

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

where $\lambda_k = \frac{\lambda}{2} \sum_{i=1}^k \left(\frac{1-\mu}{2}\right)^i$ ($k > 0$) and $\lambda_0 = 0$ in which $\mu = \frac{\beta_2 c_\infty^2}{\nu}$. In the following, we mainly use these notations to finish our proofs.

D Auxiliary Lemmas

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

Lemma 2. Suppose the sequence $\{\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k\}$ are updated by Eqn. (10). That is, $\mathbf{x}_{k+1} = \frac{1}{1+\lambda\eta_k} \left(\mathbf{x}_k - \eta_k \frac{\mathbf{m}_k}{\mathbf{s}_k} \right)$, $\mathbf{y}_{k+1} = \mathbf{z}_k - \bar{\eta}_k \frac{\mathbf{m}_k}{\mathbf{s}_k}$, $\mathbf{z}_{k+1} = \bar{\eta}_k \tau_k \mathbf{x}_{k+1} + \eta_k \tau_k \mathbf{y}_{k+1}$, $\mathbf{s}_k = \sqrt{\mathbf{v}_k + \nu}$. Then $\{\mathbf{m}_k, \mathbf{s}_k\}$ satisfies Assume $c_{s,\infty} \leq \|g_k\|_\infty \leq c_\infty$, then we have

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

See its proof in Appendix F.1.

Lemma 3. [43] Suppose the sequence $\{\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k\}$ are updated by Eqn. (10). That is, $\mathbf{x}_{k+1} = \frac{1}{1+\lambda\eta_k} \left(\mathbf{x}_k - \eta_k \frac{\mathbf{m}_k}{\mathbf{s}_k} \right)$, $\mathbf{y}_{k+1} = \mathbf{z}_k - \bar{\eta}_k \frac{\mathbf{m}_k}{\mathbf{s}_k}$, $\mathbf{z}_{k+1} = \bar{\eta}_k \tau_k \mathbf{x}_{k+1} + \eta_k \tau_k \mathbf{y}_{k+1}$, $\mathbf{s}_k = \sqrt{\mathbf{v}_k + \bar{\nu}}$. Then $\{\mathbf{x}_k\}$ satisfies

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

Lemma 4. [43] Suppose the sequence $\{\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k\}$ are updated by Eqn. (10). That is, $\mathbf{x}_{k+1} = \frac{1}{1+\lambda\eta_k} \left(\mathbf{x}_k - \eta_k \frac{\mathbf{m}_k}{\mathbf{s}_k} \right)$, $\mathbf{y}_{k+1} = \mathbf{z}_k - \bar{\eta}_k \frac{\mathbf{m}_k}{\mathbf{s}_k}$, $\mathbf{z}_{k+1} = \bar{\eta}_k \tau_k \mathbf{x}_{k+1} + \eta_k \tau_k \mathbf{y}_{k+1}$, $\mathbf{s}_k = \sqrt{\mathbf{v}_k + \bar{\nu}}$. Then $\{\mathbf{z}_k\}$ satisfies

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

Lemma 5. Suppose the sequence $\{\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k\}$ are updated by Eqn. (10). That is, $\mathbf{x}_{k+1} = \frac{1}{1+\lambda\eta_k} \left(\mathbf{x}_k - \eta_k \frac{\mathbf{m}_k}{\mathbf{s}_k} \right)$, $\mathbf{y}_{k+1} = \mathbf{z}_k - \bar{\eta}_k \frac{\mathbf{m}_k}{\mathbf{s}_k}$, $\mathbf{z}_{k+1} = \bar{\eta}_k \tau_k \mathbf{x}_{k+1} + \eta_k \tau_k \mathbf{y}_{k+1}$, $\mathbf{s}_k = \sqrt{\mathbf{v}_k + \bar{\nu}}$. By setting $\eta_k = \eta$, $\bar{\eta}_k = \bar{\eta}$, $\beta_{1,k} = \beta_1$ and $\beta_{2,k} = \beta_2$, then we have

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

where $\rho_{k+1} = \eta\tau\rho_k$ and $\rho_1 = 1$.

See its proof in Appendix F.2.

Lemma 6. Suppose the sequence $\{\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k\}$ are updated by Eqn. (10). That is, $\mathbf{x}_{k+1} = \frac{1}{1+\lambda\eta_k} \left(\mathbf{x}_k - \eta_k \frac{\mathbf{m}_k}{\mathbf{s}_k} \right)$, $\mathbf{y}_{k+1} = \mathbf{z}_k - \bar{\eta}_k \frac{\mathbf{m}_k}{\mathbf{s}_k}$, $\mathbf{z}_{k+1} = \bar{\eta}_k \tau_k \mathbf{x}_{k+1} + \eta_k \tau_k \mathbf{y}_{k+1}$, $\mathbf{s}_k = \sqrt{\mathbf{v}_k + \bar{\nu}}$. By setting $\eta_k = \eta$, $\bar{\eta}_k = \bar{\eta}$, $\beta_{1,k} = \beta_1$ and $\beta_{2,k} = \beta_2$, then we have

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

where

$$\begin{aligned} \Pi_k &:= \frac{2\bar{\eta}^2}{(1+\lambda\eta)^2} \left\| \frac{\mathbf{w}_{k-1}}{\mathbf{s}_{k-1}} \right\|^2 + \frac{2\rho_k \bar{\eta}^2 \tau^2 (\bar{\eta} - \eta)^2}{(1-\eta\tau)} \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}} \left\| \frac{\mathbf{w}_i}{\mathbf{s}_i} \right\|^2 \\ \Pi'_k &:= \frac{\rho_k \eta \tau (\bar{\eta} - \eta)^2}{(1-\eta\tau)(1+\lambda\eta)^2} \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}} \left\| \frac{\mathbf{w}_i}{\mathbf{s}_i} \right\|^2, \end{aligned}$$

where $\rho_{k+1} = \eta\tau\rho_k$ and $\rho_1 = 1$.

see its proof in Appendix F.3.

E Proof of Theorem 1

Proof. Recall our definition $\tilde{F}_k(\mathbf{z}_k) = F(\mathbf{z}) + \frac{\lambda_k}{2} \|\mathbf{z}\|_{\mathbf{s}_k}^2 = \mathbb{E}_\zeta[f(\mathbf{z}; \zeta)] + \frac{\lambda_k}{2} \|\mathbf{z}\|_{\mathbf{s}_k}^2$, in the (11). By using the smoothness of $f(\boldsymbol{\theta}; \zeta)$, we can obtain

$$\begin{aligned}
& \tilde{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 + \lambda_{k+1} \|\mathbf{x}_{k+1}\|_{\mathbf{s}_{k+1}}^2 \\
& \stackrel{\textcircled{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}}{1-\mu} \|\mathbf{x}_{k+1}\|_{\mathbf{s}_k}^2 \\
& \stackrel{\textcircled{2}}{\leq} F(\mathbf{x}_k) + \lambda_k \|\mathbf{x}_k\|_{\mathbf{s}_k}^2 + \langle \nabla F(\mathbf{x}_k) + \lambda \mathbf{x}_k * \mathbf{s}_k, \mathbf{x}_{k+1} - \mathbf{x}_k \rangle + \frac{L}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + \frac{\lambda}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_{\mathbf{s}_k}^2 \\
& = \tilde{F}_k(\mathbf{x}_k) - \frac{\eta_k}{1+\lambda\eta_k} \left\langle \nabla F(\mathbf{x}_k) + \lambda \mathbf{x}_k * \mathbf{s}_k, \frac{\mathbf{w}_k}{\mathbf{s}_k} \right\rangle + \frac{L\eta_k^2}{2(1+\lambda\eta_k)^2} \left\| \frac{\mathbf{w}_k}{\mathbf{s}_k} \right\|^2 + \frac{\lambda\eta_k^2}{2(1+\lambda\eta_k)^2} \left\| \frac{\mathbf{w}_k}{\mathbf{s}_k} \right\|_{\mathbf{s}_k}^2 \\
& = \tilde{F}_k(\mathbf{x}_k) + \frac{1}{2} \left\| \sqrt{\frac{\eta_k}{(1+\lambda\eta_k)\mathbf{s}_k}} (\nabla F(\mathbf{x}_k) + \lambda \mathbf{x}_k * \mathbf{s}_k - \mathbf{w}_k) \right\|^2 - \frac{1}{2} \left\| \sqrt{\frac{\eta_k}{(1+\lambda\eta_k)\mathbf{s}_k}} (\nabla F(\mathbf{x}_k) + \lambda \mathbf{x}_k * \mathbf{s}_k) \right\|^2 \\
& \quad - \frac{1}{2} \left\| \sqrt{\frac{\eta_k}{(1+\lambda\eta_k)\mathbf{s}_k}} \mathbf{w}_k \right\|^2 + \frac{L\eta_k^2}{2(1+\lambda\eta_k)^2} \left\| \frac{\mathbf{w}_k}{\mathbf{s}_k} \right\|^2 + \frac{\lambda\eta_k^2}{2(1+\lambda\eta_k)^2} \left\| \frac{\mathbf{w}_k}{\mathbf{s}_k} \right\|_{\mathbf{s}_k}^2 \\
& \leq \tilde{F}_k(\mathbf{x}_k) + \frac{\eta_k}{2c_1(1+\lambda\eta_k)} \|\nabla F(\mathbf{x}_k) - \mathbf{m}_k\|^2 - \frac{\eta_k}{2c_2(1+\lambda\eta_k)} \|\nabla F_k(\mathbf{x}_k)\|^2 \\
& \quad - \frac{\eta_k}{2c_2(1+\lambda\eta_k)} \left[1 - \frac{c_2 L \eta_k}{c_1^2(1+\lambda\eta_k)} - \frac{c_2 \lambda \eta_k}{c_1(1+\lambda\eta_k)} \right] \|\mathbf{w}_k\|^2 \\
& \stackrel{\textcircled{3}}{\leq} \tilde{F}_k(\mathbf{x}_k) + \frac{\eta_k}{2c_1(1+\lambda\eta_k)} \|\nabla F(\mathbf{x}_k) - \mathbf{m}_k\|^2 - \frac{\eta_k}{2c_2(1+\lambda\eta_k)} \|\nabla F_k(\mathbf{x}_k)\|^2 - \frac{\eta_k}{4c_2(1+\lambda\eta_k)} \|\mathbf{w}_k\|^2,
\end{aligned}$$

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

② holds because in Lemma 3, we have

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

③ holds, since we set $\eta_k \leq \frac{c_1^2(1+\lambda\eta_k)}{2c_2(L+\lambda c_1)}$ such that $\frac{c_2 L \eta_k}{c_1^2(1+\lambda\eta_k)} + \frac{c_2 \lambda \eta_k}{c_1(1+\lambda\eta_k)} \leq \frac{1}{2}$.

From Lemma 6, by setting $\eta_k = \eta$, $\bar{\eta}_k = \bar{\eta}$ and $\beta_{1,k} = \beta_1$, we have

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

where

$$\begin{aligned}
\Pi_k & := \frac{2\bar{\eta}^2}{(1+\lambda\eta)^2} \left\| \frac{\mathbf{w}_{k-1}}{\mathbf{s}_{k-1}} \right\|^2 + \frac{2\rho_k \bar{\eta}^2 \tau^2 (\bar{\eta} - \eta)^2}{(1-\eta\tau)} \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}} \left\| \frac{\mathbf{w}_i}{\mathbf{s}_i} \right\|^2 \\
\Pi'_k & := \frac{\rho_k \eta \tau (\bar{\eta} - \eta)^2}{(1-\eta\tau)(1+\lambda\eta)^2} \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}} \left\| \frac{\mathbf{w}_i}{\mathbf{s}_i} \right\|^2.
\end{aligned} \tag{12}$$

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

$$\begin{aligned}
\Pi_k & \leq \bar{\Pi}_k := \frac{2\bar{\eta}^2}{c_1(1+\lambda\eta)^2} \|\mathbf{w}_{k-1}\|^2 + \frac{2\rho_k \bar{\eta}^2 \tau^2 (\bar{\eta} - \eta)^2}{c_1(1-\eta\tau)} \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}} \|\mathbf{w}_i\|^2 \\
\Pi'_k & \leq \bar{\Pi}'_k := \frac{\rho_k \eta \tau (\bar{\eta} - \eta)^2}{c_1(1-\eta\tau)(1+\lambda\eta)^2} \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}} \|\mathbf{w}_i\|^2.
\end{aligned} \tag{13}$$

Therefore, we have

$$\begin{aligned}
& \tilde{F}_{k+1}(\mathbf{x}_{k+1}) \\
& \leq \tilde{F}_k(\mathbf{x}_k) - \frac{\eta}{2c_2(1+\lambda\eta)} \|\nabla F_k(\mathbf{x}_k)\|^2 - \frac{\eta}{4c_2(1+\lambda\eta)} \|\mathbf{w}_k\|^2 \\
& \quad + \frac{\eta(1-\beta_1)}{c_1(1+\lambda\eta)} \mathbb{E} \left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{z}_{k-1})\|^2 \right] + \frac{\eta \bar{\Pi}_k (1-\beta_1)^2 L^2}{c_1 \beta_1 (1+\lambda\eta)} + \frac{\eta \beta_1^2 \sigma^2}{c_1(1+\lambda\eta)b} + \frac{\eta L \bar{\Pi}'_k}{c_1(1+\lambda\eta)}
\end{aligned} \tag{14}$$

From Lemma 4, we have

$$\begin{aligned}
& \mathbb{E} \left[\|\mathbf{m}_k - \nabla F(\mathbf{z}_k)\|^2 \right] \\
& \leq (1-\beta_1) \mathbb{E} \left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{z}_{k-1})\|^2 \right] + \frac{(1-\beta_1)^2 L^2}{\beta_1} \mathbb{E} \left[\|\mathbf{z}_k - \mathbf{z}_{k-1}\|^2 \right] + \frac{\beta_1^2 \sigma^2}{b} \\
& \stackrel{\textcircled{1}}{\leq} (1-\beta_1) \mathbb{E} \left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{z}_{k-1})\|^2 \right] + \frac{(1-\beta_1)^2 L^2 \bar{\Pi}_k}{\beta_1} + \frac{\beta_1^2 \sigma^2}{b}
\end{aligned} \tag{15}$$

where we use the results in Lemma 5 that

$$\|\mathbf{z}_k - \mathbf{z}_{k-1}\|^2 \leq \Pi_k \leq \bar{\Pi}_k.$$

Then we add Eqn. (16) and $\alpha \times$ (15) as follows:

$$\begin{aligned}
& \tilde{F}_{k+1}(\mathbf{x}_{k+1}) + \alpha \mathbb{E} \left[\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\|^2 \right] \\
& \leq \tilde{F}_k(\mathbf{x}_k) - \frac{\eta}{2c_2(1+\lambda\eta)} \|\nabla F_k(\mathbf{x}_k)\|^2 - \frac{\eta}{4c_2(1+\lambda\eta)} \|\mathbf{w}_k\|^2 \\
& \quad + (1-\beta_1) \left(\frac{\eta}{c_1(1+\lambda\eta)} + \alpha \right) \mathbb{E} \left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{z}_{k-1})\|^2 \right] + \frac{\eta \bar{\Pi}_k (1-\beta_1)^2 L^2}{c_1 \beta_1 (1+\lambda\eta)} + \frac{\eta \beta_1^2 \sigma^2}{c_1(1+\lambda\eta)b} \\
& \quad + \frac{\eta L \bar{\Pi}'_k}{c_1(1+\lambda\eta)} + \frac{\alpha(1-\beta_1)^2 L^2 \bar{\Pi}_k}{\beta_1} + \frac{\alpha \beta_1^2 \sigma^2}{b}
\end{aligned} \tag{16}$$

Then by setting $\alpha = \frac{\eta(1-\beta_1)}{c_1 \beta_1 (1+\lambda\eta)}$ and $G(\mathbf{x}_{k+1}) = \tilde{F}_{k+1}(\mathbf{x}_{k+1}) + \frac{\eta(1-\beta_1)}{c_1 \beta_1 (1+\lambda\eta)} \mathbb{E} \left[\|\mathbf{m}_k - \nabla F(\mathbf{x}_k)\|^2 \right]$, we can obtain

$$\begin{aligned}
G(\mathbf{x}_{k+1}) & \leq G(\mathbf{x}_k) - \frac{\eta}{2c_2(1+\lambda\eta)} \|\nabla F_k(\mathbf{x}_k)\|^2 - \frac{\eta}{4c_2(1+\lambda\eta)} \|\mathbf{w}_k\|^2 \\
& \quad + \frac{\eta \bar{\Pi}_k (1-\beta_1)^2 L^2}{c_1 \beta_1 (1+\lambda\eta)} + \frac{\eta \beta_1^2 \sigma^2}{c_1(1+\lambda\eta)b} + \frac{\eta L \bar{\Pi}'_k}{c_1(1+\lambda\eta)} \\
& \quad + \frac{\eta(1-\beta_1)^3 L^2 \bar{\Pi}_k}{c_1 \beta_1^2 (1+\lambda\eta)} + \frac{\eta(1-\beta_1) \beta_1 \sigma^2}{c_1(1+\lambda\eta)b} \\
& \leq G(\mathbf{x}_k) - \frac{\eta}{2c_2(1+\lambda\eta)} \left\| \nabla \tilde{F}(\mathbf{x}_k) \right\|^2 - \frac{\eta}{4c_2(1+\lambda\eta)} \|\mathbf{w}_k\|^2 \\
& \quad + \frac{\eta(1-\beta_1)^2 L^2 \bar{\Pi}_k}{c_1 \beta_1^2 (1+\lambda\eta)} + \frac{\eta L \bar{\Pi}'_k}{c_1(1+\lambda\eta)} + \frac{\eta \beta_1 \sigma^2}{c_1(1+\lambda\eta)b}.
\end{aligned}$$

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

$$\begin{aligned}
& \frac{1}{T} \sum_{k=0}^{T-1} \mathbb{E} \left[\|\nabla F_k(\mathbf{x}_k)\|^2 + \frac{1}{2} \|\mathbf{w}_k\|^2 \right] \\
& \leq \frac{2c_2(1+\lambda\eta)}{\eta T} [G(\mathbf{x}_0) - G(\mathbf{x}_T)] + \frac{2c_2 \beta_1 \sigma^2}{c_1 b T} + \frac{2c_2(1-\beta_1)^2 L^2}{c_1 \beta_1^2 T} \sum_{k=0}^{T-1} \bar{\Pi}_k + \frac{2c_2 L}{c_1 T} \sum_{k=0}^{T-1} \bar{\Pi}'_k \\
& \leq \frac{2c_2(1+\lambda\eta) \Delta}{\eta T} + \frac{2c_2 \beta_1 \sigma^2}{c_1 b T} + \frac{2c_2(1-\beta_1)^2 L^2}{c_1 \beta_1^2 T} \sum_{k=0}^{T-1} \bar{\Pi}_k + \frac{2c_2 L}{c_1 T} \sum_{k=0}^{T-1} \bar{\Pi}'_k
\end{aligned}$$

where

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

where $\Delta = F(\mathbf{x}_0) - F(\mathbf{x}_*)$; \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 $\sum_{k=0}^{T-1} \bar{\Pi}_k$ and $\sum_{k=0}^{T-1} \bar{\Pi}'_k$. Firstly, we have

$$\begin{aligned}
\sum_{k=0}^{T-1} \bar{\Pi}_k &= \sum_{k=0}^{T-1} \left[\frac{2\bar{\eta}^2}{c_1(1+\lambda\eta)^2} \|\mathbf{w}_{k-1}\|^2 + \frac{2\rho_k\bar{\eta}^2\tau^2(\bar{\eta}-\eta)^2}{c_1(1-\eta\tau)} \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}} \|\mathbf{w}_i\|^2 \right] \\
&= \frac{2\bar{\eta}^2}{c_1(1+\lambda\eta)^2} \sum_{k=0}^{T-1} \left[\|\mathbf{w}_{k-1}\|^2 \right] + \frac{2\bar{\eta}^2\tau^2(\bar{\eta}-\eta)^2}{c_1(1-\eta\tau)} \sum_{k=0}^{T-1} \rho_k \left[\sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}} \|\mathbf{w}_i\|^2 \right] \\
&= \frac{2\bar{\eta}^2}{c_1(1+\lambda\eta)^2} \sum_{k=0}^{T-1} \left[\|\mathbf{w}_{k-1}\|^2 \right] + \frac{2\bar{\eta}^2\tau^2(\bar{\eta}-\eta)^2}{c_1(1-\eta\tau)} \sum_{k=0}^{T-1} \frac{1}{\rho_{k+1}} \left[\sum_{i=k}^{T-1} \rho_i \|\mathbf{w}_k\|^2 \right] \\
&\stackrel{\textcircled{1}}{\leq} \frac{2\bar{\eta}^2}{c_1(1+\lambda\eta)^2} \sum_{k=0}^{T-1} \left[\|\mathbf{w}_{k-1}\|^2 \right] + \frac{2\bar{\eta}^2\tau(\bar{\eta}-\eta)^2}{c_1\eta(1-\eta\tau)^2} \sum_{k=0}^{T-1} \left[\|\mathbf{w}_k\|^2 \right] \\
&\stackrel{\textcircled{2}}{\leq} \frac{2\gamma^2\eta^2}{c_1} \left(\frac{1}{(1+\lambda\eta)^2} + \frac{\eta\tau(\gamma-1)^2}{(1-\eta\tau)^2} \right) \sum_{k=0}^{T-1} \left[\|\mathbf{w}_k\|^2 \right]
\end{aligned}$$

where $\textcircled{1}$ holds since 1) $\sum_{i=k}^{T-1} \rho_i = \frac{\rho_k(1-\eta^{T-k}\tau^{T-k})}{1-\eta\tau} \leq \frac{\rho_k}{1-\eta\tau}$ and 2) $\rho_{k+1} = \eta\tau\rho_k$ and $\rho_1 = 1$; $\textcircled{2}$ holds by setting $\bar{\eta} = \gamma\eta$. Similarly, we can bound

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

Therefore, we have

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

where $\textcircled{1}$ holds since we choose proper η and β_1 such that

$$\begin{aligned}
\frac{4c_2\gamma^2\eta^2(1-\beta_1)^2 L^2}{c_1^2\beta_1^2} \left(\frac{1}{(1+\lambda\eta)^2} + \frac{\eta\tau(\gamma-1)^2}{(1-\eta\tau)^2} \right) &\leq \frac{1}{8} \\
\frac{2c_2 L\eta^2(\gamma-1)^2}{c_1^2(1-\eta\tau)^2(1+\lambda\eta)^2} &\leq \frac{1}{8}
\end{aligned} \tag{17}$$

Now we discuss how to select η and β_1 such that (17) holds. To begin with, we have $\frac{1}{(1+\lambda\eta)^2} \leq 1$ and $\frac{\eta\tau(\gamma-1)^2}{(1-\eta\tau)^2} = \frac{1+\gamma+\lambda\gamma\eta}{\gamma(1+\lambda\eta)}(\gamma-1)^2 \leq \frac{1+\gamma}{\gamma}(\gamma-1)^2 < \gamma^2$ in which $\eta\tau = \frac{\eta}{\eta+\gamma\eta+\lambda\gamma\eta^2}$. Then we only need

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

So we arrive at

$$\frac{1}{T} \sum_{k=0}^{T-1} \mathbb{E} \left[\|\nabla F_k(\mathbf{x}_k)\|^2 + \frac{1}{4} \|\mathbf{w}_k\|^2 \right] \leq \frac{2c_2(1+\lambda\eta)\Delta}{\eta T} + \frac{2c_2\beta_1\sigma^2}{c_1b} \stackrel{\textcircled{1}}{\leq} \epsilon^2,$$

where we set $T \geq \frac{6c_2(1+\lambda\eta)\Delta}{\eta\epsilon^2}$ and $\beta_1 \leq \frac{c_1b\epsilon^2}{6c_2\sigma^2}$. This result directly bounds

$$\begin{aligned} \frac{1}{T} \sum_{k=0}^{T-1} \|\mathbf{s}_k * (\mathbf{x}_k - \mathbf{x}_{k+1})\|^2 &= \frac{\eta^2}{(1+\lambda\eta)^2 T} \sum_{k=0}^{T-1} \|\mathbf{m}_k + \lambda\mathbf{x}_k * \mathbf{s}_k\|^2 \\ &\leq \frac{\eta^2}{(1+\lambda\eta)^2 T} \sum_{k=0}^{T-1} \|\mathbf{w}_k\|^2 \leq \frac{4\eta^2\epsilon^2}{(1+\lambda\eta)^2}. \end{aligned}$$

Moreover, from Lemma 5, we have

$$\begin{aligned} \frac{1}{T} \sum_{k=0}^{T-1} \mathbb{E} \|\mathbf{y}_k - (1+\lambda\bar{\eta})\mathbf{x}_k\|^2 &\stackrel{\textcircled{1}}{\leq} \frac{1}{T} \sum_{k=0}^{T-1} \frac{\rho_k(\bar{\eta}-\eta)^2}{(1-\eta\tau)(1+\lambda\eta)^2} \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}} \left\| \frac{\mathbf{w}_i}{\mathbf{s}_i} \right\|^2 \stackrel{\textcircled{2}}{=} \frac{1}{\eta\tau T} \sum_{k=0}^{T-1} \Pi'_k \\ \frac{1}{T} \sum_{k=0}^{T-1} \mathbb{E} \|\mathbf{z}_k - \mathbf{x}_k\|^2 &\stackrel{\textcircled{1}}{\leq} \frac{1}{T} \sum_{k=0}^{T-1} \frac{\rho_k\eta\tau(\bar{\eta}-\eta)^2}{(1-\eta\tau)(1+\lambda\eta)^2} \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}} \left\| \frac{\mathbf{w}_i}{\mathbf{s}_i} \right\|^2 \stackrel{\textcircled{2}}{=} \frac{1}{T} \sum_{k=0}^{T-1} \Pi'_k \\ \frac{1}{T} \sum_{k=0}^{T-1} \mathbb{E} \|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 &\stackrel{\textcircled{1}}{\leq} \frac{1}{T} \sum_{k=0}^{T-1} \left[\frac{2\bar{\eta}^2}{(1+\lambda\eta)^2} \left\| \frac{\mathbf{w}_k}{\mathbf{s}_k} \right\|^2 + \frac{2\rho_{k+1}\bar{\eta}^2\tau^2(\bar{\eta}-\eta)^2}{(1-\eta\tau)} \sum_{i=0}^k \frac{1}{\rho_{i+1}} \left\| \frac{\mathbf{w}_i}{\mathbf{s}_i} \right\|^2 \right] \\ &\stackrel{\textcircled{2}}{=} \frac{1}{T} \sum_{k=0}^{T-1} \Pi_k \end{aligned}$$

where $\textcircled{1}$ holds by using Lemma 5; $\textcircled{2}$ holds by using the definition in Eqn. (12). Now remaining task is to upper bound $\frac{1}{T} \sum_{k=0}^{T-1} \Pi_k$ and $\frac{1}{T} \sum_{k=0}^{T-1} \Pi'_k$. From the bound in Eqn. (13) 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

$$\begin{aligned} \frac{1}{T} \sum_{k=0}^{T-1} \Pi_k &\leq \frac{1}{T} \sum_{k=0}^{T-1} \bar{\Pi}_k \leq \frac{2\gamma^2\eta^2}{c_1T} \left(\frac{1}{(1+\lambda\eta)^2} + \frac{\eta\tau(\gamma-1)^2}{(1-\eta\tau)^2} \right) \sum_{k=0}^{T-1} \mathbb{E} \left[\|\mathbf{w}_k\|^2 \right] \stackrel{\textcircled{1}}{\leq} \frac{c_1^3\beta_1^2\epsilon^2}{4c_2(1-\beta_1)^3L^2} \\ \frac{1}{T} \sum_{k=0}^{T-1} \Pi'_k &\leq \frac{1}{T} \sum_{k=0}^{T-1} \bar{\Pi}'_k \leq \frac{\eta^2(\gamma-1)^2}{c_1(1-\eta\tau)^2(1+\lambda\eta)^2T} \sum_{k=0}^{T-1} \mathbb{E} \left[\|\mathbf{w}_k\|^2 \right] \stackrel{\textcircled{1}}{\leq} \frac{c_1\epsilon^2}{4c_2L} \end{aligned}$$

where $\textcircled{1}$ holds, since 1) $\frac{1}{T} \sum_{k=0}^{T-1} \mathbb{E} \|\mathbf{w}_k\|^2 \leq 4\epsilon^2$; 2) we use the results in Eqn. (17) to obtain

$$\begin{aligned} \frac{2\gamma^2\eta^2}{c_1} \left(\frac{1}{(1+\lambda\eta)^2} + \frac{\eta\tau(\gamma-1)^2}{(1-\eta\tau)^2} \right) &\leq \frac{c_1^3\beta_1^2}{16c_2(c_1^2\beta_1+1-\beta_1)(1-\beta_1)^2L^2} \leq \frac{c_1^3\beta_1^2}{16c_2(1-\beta_1)^3L^2} \\ \frac{\eta^2(\gamma-1)^2}{c_1(1-\eta\tau)^2(1+\lambda\eta)^2} &\leq \frac{c_1}{16c_2L} \end{aligned}$$

Therefore, we have

$$\begin{aligned} \frac{1}{T} \sum_{k=0}^{T-1} \mathbb{E} \|\mathbf{y}_k - (1+\lambda\bar{\eta})\mathbf{x}_k\|^2 &\leq \frac{c_1\epsilon^2}{4c_2L\eta\tau} = \frac{c_1(1+\gamma+\lambda\gamma\eta)\epsilon^2}{4c_2L} \\ \frac{1}{T} \sum_{k=0}^{T-1} \mathbb{E} \|\mathbf{z}_k - \mathbf{x}_k\|^2 &\leq \frac{c_1\epsilon^2}{4c_2L} \\ \frac{1}{T} \sum_{k=0}^{T-1} \mathbb{E} \|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 &\leq \frac{c_1^3\beta_1^2\epsilon^2}{4c_2(1-\beta_1)^3L^2}. \end{aligned}$$

Besides, we have

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

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

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

From Eqn. (15), we have

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

By setting $\rho_{k+1} = (1 - \beta_1)\rho_k$ and $\rho_0 = 1$, we have

$$\begin{aligned}
\frac{1}{\rho_k} \mathbb{E} \left[\|\mathbf{m}_k - \nabla F(\mathbf{z}_k)\|^2 \right] &\leq \frac{1}{\rho_{k-1}} \mathbb{E} \left[\|\mathbf{m}_{k-1} - \nabla F(\mathbf{z}_{k-1})\|^2 \right] + \frac{(1 - \beta_1)^2 L^2 \bar{\Pi}_k}{\beta_1 \rho_k} + \frac{\beta_1^2 \sigma^2}{b \rho_k} \\
&\leq \frac{1}{\rho_0} \mathbb{E} \left[\|\mathbf{m}_0 - \nabla F(\mathbf{z}_0)\|^2 \right] + \sum_{i=1}^k \left[\frac{(1 - \beta_1)^2 L^2 \bar{\Pi}_i}{\beta_1 \rho_i} + \frac{\beta_1^2 \sigma^2}{b \rho_i} \right]
\end{aligned}$$

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

$$\beta_1 \leq \frac{c_1 b \epsilon^2}{6c_2 \sigma^2},$$

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

For η , it should satisfy

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

Considering $(1 - \eta\tau)(1 + \lambda\eta) = \frac{\gamma(1+\lambda\eta)^2}{1+\gamma+\lambda\gamma\eta} \geq \frac{\gamma}{\gamma+1}$, $\frac{\gamma}{(\gamma-1)(\gamma+1)} \geq \frac{1}{\gamma}$ and $c_1 = \nu^{0.5} \ll 1$, then we have

$$\begin{aligned}
\eta &\leq \min \left(\frac{c_1 \beta_1}{36c_2^{0.5}(\gamma+1)^2L}, \frac{c_1}{4c_2^{0.5}\gamma L^{0.5}}, \frac{c_1^2}{2c_2(L+\lambda c_1)} \right) \\
&= \min \left(\frac{c_1 b \epsilon^2}{36c_2^{1.5}(\gamma+1)^2L\sigma^2}, \frac{c_1}{4c_2^{0.5}\gamma L^{0.5}}, \frac{c_1^2(1+\lambda\eta)}{2c_2(L+\lambda c_1)} \right) = \frac{c_1 b \epsilon^2}{36c_2^{1.5}(\gamma+1)^2L\sigma^2}
\end{aligned}$$

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

$$T \geq \frac{6c_2(1+\lambda\eta)\Delta}{\eta\epsilon^2} = \mathcal{O} \left(\frac{c_2\Delta}{\epsilon^2} \frac{36c_2^{1.5}(\gamma+1)^2L\sigma^2}{c_1 b \epsilon^2} \right)$$

The proof is completed. \square

F Proofs of Auxiliary Lemmas

F.1 Proof of Lemma 2

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

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

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

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

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. \square

F.2 Proof of Lemma 5

Proof. To begin with, we have

$$\begin{aligned} & \mathbf{y}_{k+1} - (1 + \lambda\bar{\eta}_k)\mathbf{x}_{k+1} \\ &= \mathbf{z}_k - \bar{\eta}_k \frac{\mathbf{m}_k}{\mathbf{s}_k} - \frac{1 + \lambda\bar{\eta}_k}{1 + \lambda\eta_k} \left(\mathbf{x}_k - \eta_k \frac{\mathbf{m}_k}{\mathbf{s}_k} \right) \\ &= \bar{\eta}_{k-1}\tau_{k-1}\mathbf{x}_k + \eta_{k-1}\tau_{k-1}\mathbf{y}_k - \bar{\eta}_k \frac{\mathbf{m}_k}{\mathbf{s}_k} - \frac{1 + \lambda\bar{\eta}_k}{1 + \lambda\eta_k} \left(\mathbf{x}_k - \eta_k \frac{\mathbf{m}_k}{\mathbf{s}_k} \right) \\ &= \eta_{k-1}\tau_{k-1}(\mathbf{y}_k - (1 + \lambda\bar{\eta}_{k-1})\mathbf{x}_k) - \left(\bar{\eta}_k - \frac{1 + \lambda\bar{\eta}_{k-1}}{1 + \lambda\eta_{k-1}}\eta_k \right) \frac{\mathbf{m}_k}{\mathbf{s}_k} + \frac{\lambda(\eta_k - \bar{\eta}_k)}{1 + \lambda\eta_k}\mathbf{x}_k \\ &\stackrel{\textcircled{1}}{=} \eta_{k-1}\tau_{k-1}(\mathbf{y}_k - (1 + \lambda\bar{\eta}_{k-1})\mathbf{x}_k) - \left(\bar{\eta}_k - \frac{1 + \lambda\bar{\eta}_{k-1}}{1 + \lambda\eta_{k-1}}\eta_k \right) \frac{\mathbf{w}_k - \lambda\sqrt{\mathbf{v}_k}}{\mathbf{s}_k} + \frac{\lambda(\eta_k - \bar{\eta}_k)}{1 + \lambda\eta_k}\mathbf{x}_k \\ &= \eta_{k-1}\tau_{k-1}(\mathbf{y}_k - (1 + \lambda\bar{\eta}_{k-1})\mathbf{x}_k) - \left(\bar{\eta}_k - \frac{1 + \lambda\bar{\eta}_{k-1}}{1 + \lambda\eta_{k-1}}\eta_k \right) \frac{\mathbf{w}_k}{\mathbf{s}_k} \\ &\quad + \left(\lambda\bar{\eta}_k - \frac{1 + \lambda\bar{\eta}_{k-1}}{1 + \lambda\eta_{k-1}}\lambda\eta_k + \frac{\lambda(\eta_k - \bar{\eta}_k)}{1 + \lambda\eta_k} \right) \mathbf{x}_k \\ &\stackrel{\textcircled{2}}{=} \eta\tau(\mathbf{y}_k - (1 + \lambda\bar{\eta})\mathbf{x}_k) - \frac{\bar{\eta} - \eta}{1 + \lambda\eta} \frac{\mathbf{w}_k}{\mathbf{s}_k} \end{aligned}$$

where $\textcircled{1}$ holds since $\mathbf{w}_k := \mathbf{m}_k + \lambda\mathbf{x}_k * \mathbf{s}_k$; $\textcircled{2}$ holds since we set all $\eta_k = \eta$ and $\bar{\eta}_k = \bar{\eta}$ which gives $\tau_k = \tau = \frac{1}{\eta + \bar{\eta} + \lambda\eta\bar{\eta}}$. Therefore, by defining $\rho_{k+1} = \eta\tau\rho_k$ and $\rho_1 = 1$, then we have

$$\frac{\mathbf{y}_{k+1} - (1 + \lambda\bar{\eta})\mathbf{x}_{k+1}}{\rho_{k+1}} = \frac{\mathbf{y}_k - (1 + \lambda\bar{\eta})\mathbf{x}_k}{\rho_k} - \frac{1}{\rho_{k+1}} \frac{\bar{\eta} - \eta}{1 + \lambda\eta} \frac{\mathbf{w}_k}{\mathbf{s}_k} \quad (k \geq 1)$$

For $k = 0$, we have

$$\begin{aligned} \mathbf{y}_1 - (1 + \lambda\bar{\eta})\mathbf{x}_1 &= \mathbf{z}_0 - \bar{\eta} \frac{\mathbf{m}_0}{\mathbf{s}_0} - \frac{1 + \lambda\bar{\eta}}{1 + \lambda\eta} \left(\mathbf{x}_0 - \eta \frac{\mathbf{m}_0}{\mathbf{s}_0} \right) \\ &= \mathbf{z}_0 - \bar{\eta} \frac{\mathbf{w}_0 - \lambda\mathbf{s}_0 * \mathbf{x}_0}{\mathbf{s}_0} - \frac{1 + \lambda\bar{\eta}}{1 + \lambda\eta} \left(\mathbf{x}_0 - \eta \frac{\mathbf{w}_0 - \lambda\mathbf{s}_0 * \mathbf{x}_0}{\mathbf{s}_0} \right) \\ &= \mathbf{z}_0 - \mathbf{x}_0 - \frac{\bar{\eta} - \eta}{1 + \lambda\eta} \frac{\mathbf{w}_0}{\mathbf{s}_0}. \end{aligned}$$

In this way, one can obtain

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

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

$$\begin{aligned}\left\| \frac{\mathbf{y}_{k+1} - (1 + \lambda\bar{\eta})\mathbf{x}_{k+1}}{\rho_{k+1}} \right\|^2 &= \left\| \sum_{i=0}^k \frac{\rho_{k+1}(1 - \eta\tau)}{\rho_{i+1}} \frac{\bar{\eta} - \eta}{\rho_{k+1}(1 - \eta\tau)(1 + \lambda\eta)} \frac{\mathbf{w}_i}{\mathbf{s}_i} \right\|^2 \\ &\stackrel{\textcircled{1}}{\leq} \sum_{i=0}^k \frac{\rho_{k+1}(1 - \eta\tau)}{\rho_{i+1}} \frac{(\bar{\eta} - \eta)^2}{\rho_{k+1}^2(1 - \eta\tau)^2(1 + \lambda\eta)^2} \left\| \frac{\mathbf{w}_i}{\mathbf{s}_i} \right\|^2\end{aligned}$$

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

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

Moreover, we can also bound

$$\begin{aligned}\left\| \mathbf{z}_{k+1} - \mathbf{x}_{k+1} \right\|^2 &= \left\| \bar{\eta}\tau\mathbf{x}_{k+1} + \eta\tau\mathbf{y}_{k+1} - \mathbf{x}_{k+1} \right\|^2 \\ &= \eta\tau \left\| \mathbf{y}_{k+1} - (1 + \lambda\bar{\eta})\mathbf{x}_{k+1} \right\|^2 \\ &\leq \frac{\rho_{k+1}\eta\tau(\bar{\eta} - \eta)^2}{(1 - \eta\tau)(1 + \lambda\eta)^2} \sum_{i=0}^k \frac{1}{\rho_{i+1}} \left\| \frac{\mathbf{w}_i}{\mathbf{s}_i} \right\|^2.\end{aligned}$$

On the other hand, we have

$$\begin{aligned}\left\| \mathbf{z}_{k+1} - \mathbf{z}_k \right\| &= \left\| \bar{\eta}\tau\mathbf{x}_{k+1} + \eta\tau\mathbf{y}_{k+1} - \mathbf{z}_k \right\| \\ &\stackrel{\textcircled{1}}{=} \left\| \bar{\eta}\tau\mathbf{x}_{k+1} + \eta\tau\mathbf{y}_{k+1} - \mathbf{y}_{k+1} - \bar{\eta} \frac{\mathbf{m}_k}{\mathbf{s}_k} \right\| \\ &= \left\| \bar{\eta}\tau\mathbf{x}_{k+1} + \eta\tau\mathbf{y}_{k+1} - \mathbf{y}_{k+1} - \bar{\eta} \frac{\mathbf{w}_k - \lambda\mathbf{x}_k * \mathbf{s}_k}{\mathbf{s}_k} \right\| \\ &= \left\| \bar{\eta}\tau(1 + \lambda\eta) \left((1 + \lambda\bar{\eta})\mathbf{x}_{k+1} - \mathbf{y}_{k+1} \right) - \frac{\bar{\eta}}{1 + \lambda\eta} \frac{\mathbf{w}_k}{\mathbf{s}_k} \right\| \\ &\leq \bar{\eta}\tau(1 + \lambda\eta) \left\| (1 + \lambda\bar{\eta})\mathbf{x}_{k+1} - \mathbf{y}_{k+1} \right\| + \frac{\bar{\eta}}{1 + \lambda\eta} \left\| \frac{\mathbf{w}_k}{\mathbf{s}_k} \right\|\end{aligned}$$

where ① we plug in $\mathbf{y}_{k+1} = \mathbf{z}_k - \bar{\eta}_k \frac{\mathbf{m}_k}{\mathbf{s}_k}$. Then we can upper bound

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

The proof is completed. \square

E.3 Proof of Lemma 6

Proof. From Lemma 4, we have

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

where in $\textcircled{1}$, we use the results in Lemma 5 that

$$\|\mathbf{z}_k - \mathbf{z}_{k-1}\|^2 \leq \Pi_k := \frac{2\bar{\eta}^2}{(1 + \lambda\eta)^2} \left\| \frac{\mathbf{w}_{k-1}}{\mathbf{s}_{k-1}} \right\|^2 + \frac{2\rho_k \bar{\eta}^2 \tau^2 (\bar{\eta} - \eta)^2}{(1 - \eta\tau)} \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}} \left\| \frac{\mathbf{w}_i}{\mathbf{s}_i} \right\|^2.$$

Then we have

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

where in $\textcircled{1}$, we use the results in Lemma 5 that

$$\|\mathbf{z}_k - \mathbf{x}_k\|^2 \leq \Pi'_k := \frac{\rho_k \eta \tau (\bar{\eta} - \eta)^2}{(1 - \eta\tau)(1 + \lambda\eta)^2} \sum_{i=0}^{k-1} \frac{1}{\rho_{i+1}} \left\| \frac{\mathbf{w}_i}{\mathbf{s}_i} \right\|^2.$$

The proof is completed. \square