GAME: A GLOBALLY ADAPTIVE METHOD FOR TRAINING DEEP NEURAL NETWORKS

Dong Wang[1], Huatian Zhang[2], Tao Xu[1], Fanhua Shang[1], Hongying Liu[1], Yuanyuan Liu[1], Shengmei Shen[3]*

[1]School of Artificial Intelligence, Xidian University [2]University of Science and Technology of China
[3]Pensees AI institute of Singapore

ABSTRACT

In recent years, optimization for deep learning has attracted considerable attention. In terms of highly non-convex loss surface in deep learning, some adaptive stochastic gradient descent (SGD) methods such as ADAM and AMSGRAD have been developed to accelerate the training of deep neural networks. AMSGRAD indicates that the adaptive methods may be hard to converge to the optimums for some convex problems due to the divergence of the adaptive learning rate as in ADAM. Besides, we can find that AMSGRAD sometimes fails to beat ADAM on unseen data, and stops updating the second-order estimate too early to explore an optimal area. ADAM is easily affected by current gradient, which may lead to some bad cases. To address these issues, we propose a novel globally adaptive algorithm, called GAME, which enjoys a more smooth and global second-order estimate. GAME inherits the advantages of both AMSGRAD and ADAM (i.e., invariance and adaptivity), and thus it does not stop updating parameters prematurely and can remain in an optimal area to achieve better generalization than ADAM. We also give some theoretical analysis for our algorithm. Various experimental results show our algorithm is effective for various deep learning tasks.

Index Terms— deep learning, optimization, computer vision, image classification, second-order estimate

1. INTRODUCTION

In the past several years, deep neural networks (DNNs) have made great progress in many fields, such as speech recognition [1], computer vision [2, 3] and natural language processing [4, 5]. The key of deep learning is to extract features automatically [6]. Backpropagation (BP) based on error feedback can attribute DNNs this wonderful ability. To further improve the performance of BP, researchers have developed many optimization methods including stochastic gradient descent (SGD) [7], momentum [8, 9], ADAGRAD [10], RMSPROP [11], ADAM [12] and so on[13, 14].

Different optimizers are used to train various deep learning models. Researchers mainly focus on faster training speed and better generalization. Some adaptive methods have a fast training speed. One of the most popular methods is ADAM, which uses the first and second-order moment estimates to update model parameters. Although it has achieved great success in training many DNNs, [15] proved that ADAM even diverges for solving some convex optimization problems. Therefore, [15] also employed the maximal second-order moment estimate in AMSGRAD to fix the error in ADAM. However, we find that AMSGRAD does not outperform ADAM for solving many highly non-convex problems, especially training DNNs. We hold this view that it is mainly because AMSGRAD usually stops updating the second-order estimate prematurely, which is closely related to the step-size. This leads to the fact that AMSGRAD does not implement an enough long search in the weight space (see Fig. 2). A proper long search usually stands for a fine generalization [16]. Furthermore, thanks to the exponential moving average, ADAM updates the second-order estimate too frequently and it is easily affected by the current sampling gradient. This results in two bad phenomena: (1) For relatively large gradients, ADAM immediately has a large second-order estimate and a small step-size, hindering itself to search the further space. (2) For relatively small gradients, it gains a large step-size and goes beyond the optimum in a flat area, which has a great generalization [17, 18]. We find that the second-order estimate should be updated in a mild way – not too slow or too fast. It should reflect the changing tendency of gradients but

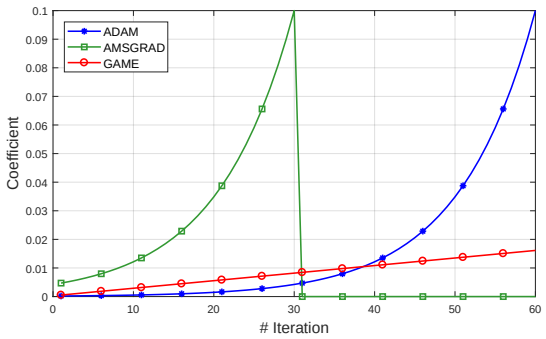
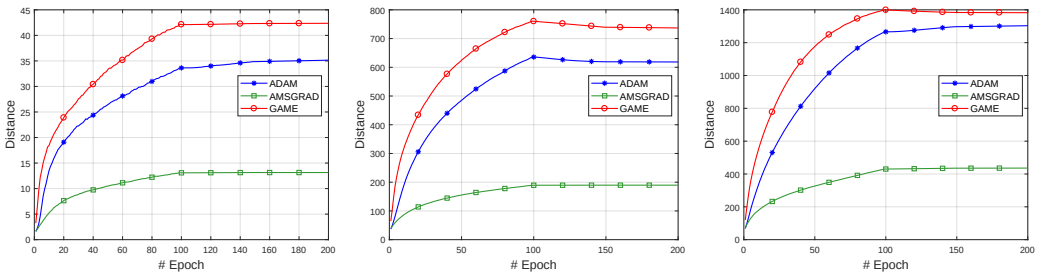


Fig. 1. The weighted coefficient for the past square of the gradient as the number of epochs increases. It is clear that ADAM [12] has a larger coefficient for the recent gradient and AMSGRAD [15] has a larger coefficient for the early gradient. In contrast, GAME has a relatively stable weighted coefficient, which does not increase or decrease explosively.

*Fanhua Shang is the corresponding author.



(a) Layer 1 (b) Layer 2 (c) Layer 3

Fig. 2. Comparison of ADAM, AMSGRAD and GAME for training ResNet on CIFAR100. We record the L_1 norm (distance) between current and initial parameter vectors of three random layers at different depths. Our GAME reaches the furthest.

not be affected by some extreme values for a long time (like AMSGRAD). To address these issues, we develop a **G**lobally **A**daptive **ME**thod (GAME) with some new updating rules, which leads to a more stable second-order moment estimate.

2. ALGORITHM

In this section, we propose a globally adaptive algorithm with a slowly increasing weighted coefficient for past gradients in second-order estimates. Algorithm 1 gives the pseudo-code of our GAME algorithm. All the operations between vectors are element-wise. Let $f(\theta)$ be a stochastic objective function that can cause the stochasticity.

2.1. Notations

For two vectors a, b with the same dimensions, $c = a/b$ is defined as the element-wise division and c has the same dimension. a^2 means the element-wise square operation and $\sqrt{a}$ stands for the element-wise square root. $\|a\|_1$ is the L_1 norm of the vector a and $\|a\|$ is the L_2 norm of a. a_i means the i-th coordinate of vector a. We use $[d]$ to denote the sequence $[1, 2, 3, ..., d]$. $\mathcal{F}$ is the parameter domain.The projection $\Pi_{\mathcal{F},A}(y)$ for $A \in \mathcal{S}_+^d$ ($\mathcal{S}_+^d$ is a set of all positive definite matrices of size $d \times d$) is defined as: $\arg\min_{x\in\mathcal{F}} \left\|A^{1/2}(x-y)\right\|$ for $y \in \mathbb{R}^d$. $\mathcal{F}$ has bounded diameter D_∞ if $\|x-y\|_\infty \le D_\infty$ for all $x, y \in \mathcal{F}$. $[g_t]$ is the gradient sequence generated by the algorithm. g_t denotes the t-th item in $[g_t]$.

2.2. Non-convergence of ADAM

Note that α_t is the learning rate in iteration t, and V_t denotes the second-order moment estimate. Following the work in [15], we give the variable Γ, which stands for the difference between two adjacent true learning rates. Only Γ_t keeping non-negative means convergence(More details can be found in [15]).

$$\Gamma_t = \left(\frac{\sqrt{V_t}}{\alpha_t} - \frac{\sqrt{V_{t-1}}}{\alpha_{t-1}}\right). \tag{1}$$

Lemma 1 *With the setting $\alpha_t = \alpha/(t+1)$ (an initial value $\alpha > 0$) and the gradient g_t generated by ADAM in iteration t, we find $\exists\ T_1 \le T_2$, $\forall\ \beta \in (0,1)$, $\forall\ t \in [T_1, T_2]$, $g_t = 0$, such that the positive semi-definiteness of Γ_t can not be satisfied.*

We attribute this non-convergence issue to the exponential moving average (EMA) in ADAM. When the gradient becomes smaller (e.g., 0), the true learning rate $\frac{\alpha_t}{\sqrt{V_t}}$ becomes larger in spite of a decreasing learning rate α_t and the initial learning rate α. This causes the non-convergence issue.

2.3. Our GAME Method

By rewriting the formula in ADAM (β_2 denotes its momentum coefficient in the second-order moment estimation), we model the second-order estimations as follows:

$$\text{ADAM}: \quad V_t = (1-\beta_2)\beta_2^t \sum_{i=1}^{t} \beta_2^{-i} g_i^2, \tag{2}$$

$$\text{GAME (ours)}: \quad V_t = A(t) \sum_{i=1}^{t} B(i) g_i^2, \tag{3}$$

where $A(t)$ and $B(t)$ are two coefficients. The update rule can be rewritten as the following equivalent recurrence formula

$$\text{GAME (ours)}: V_t = \frac{A(t)}{A(t-1)} V_{t-1} + B(i)|_{i=t} A(t) g_i^2. \tag{4}$$

Using our new model, $A(t) = (1-\beta_2)\beta_2^t$ and $B(i) = \beta_2^{-i}$ for ADAM. So GAME contains ADAM. We can find that $A(t)$ in ADAM is decreasing as t increases, and $B(i) = \beta_2^{-i}$ weights the past gradient in an exponential way. This means that ADAM relies on the current gradient too much. To address this issue, we propose to replace the exponential moving average with linear weighting. We make $B(i) = O(i)$ to improve ADAM. And we find that there exists a relationship in ADAM, that is,

$$\lim_{t\to+\infty} A(t) \sum_{i=1}^{t} B(i) = O(1). \tag{5}$$

Algorithm 1 GAME

Input: α_t: Learning rate at iteration t, $\beta \in [0, 1)$: Exponential decay rate for 1^{st} order estimate, θ_0: Initial parameter vector, $m_0 \leftarrow 0$ (Initial 1^{st} order estimate vector), $v_0 \leftarrow 0$ (Initial 2^{nd} order estimate vector), $t \leftarrow 0$ (Initial global step), and $\epsilon \leftarrow$ 1e-8.
$f(\theta)$: Objective function with $\theta \in \mathcal{F}$.
// Current mini-batch I_t, current 1^{st} order estimate m_t and current 2^{nd} order estimate v_t.
while $t \neq$max iteration **do**
$t = t + 1$
$\hat{g}_t \leftarrow$ Stochastic Gradient $\nabla_\theta f_{I_t}(\theta_t)$;
$\hat{m}_t = \beta \cdot m_{t-1} + (1 - \beta) \cdot \hat{g}_t$; *// 1^{st} order estimate*
$m_t = \hat{m}_t/(1 - \beta^t)$; *// fix the estimate bias*
$v_t = (t \cdot v_{t-1} + \hat{g}_t^2)/(t + 2)$ and $\hat{V}_t = \text{diag}\,(v_t)$; *// 2^{nd} order estimate*
$\theta_t = \Pi_{\mathcal{F}, \sqrt{\hat{V}_t}}(\theta_{t-1} - \alpha_t \cdot m_t/(\sqrt{v_t} + \epsilon))$; *// update parameters*
end while
Output: θ_T.

If $B(i) = O(i)$, we have $A(t) = O(1/t^2)$. Suppose $A(t) = 1/(t^2 + 3t + 2)$ and $B(i) = i + 1$, we get our GAME algorithm by rearranging the recurrence formula, which can address the non-convergence issue of ADAM.

Lemma 2 *Our GAME algorithm can ensure that the sequence Γ_t in (1) is non-negative, which addresses the non-convergence issue in ADAM.*

3. NON-CONVEX CONVERGENCE ANALYSIS

Considering that the loss surface in the training of a deep neural network is highly non-convex, it is necessary to investigate non-convex convergence rate of our algorithm GAME. Given a fixed length of input and target, we have the following objective function to minimize at step t:

$$f(\theta_t) := \frac{1}{n} \sum_{i=1}^{n} f_i(\theta_t),$$

where n is the number of training samples.

Assumption 1 ***(L-smooth)*** *$f(\theta)$ is first-order differentiable and has L-Lipschitz gradient. $\forall \theta_1, \theta_2 \in R^d$,*

$$\|\nabla f(\theta_1) - \nabla f(\theta_2)\| \leq L\|\theta_1 - \theta_2\| \;\; and \;\; f(\theta^*) > -\infty.$$

Assumption 2 ***(Gradient Bound)*** *At the t-th iteration, the sampling gradient $\hat{g}_t$ and the real gradient $\nabla f(\theta_t)$ are both bounded, meaning $\forall t > 1$, there is a constant M such that*

$$\|\nabla f(\theta_t)\| \leq M \quad and \quad \|\hat{g}_t\| \leq M.$$

Assumption 3 ***(Unbiased Estimate)*** *We assume the sampling gradient is unbiased and the noise is independent,*

$$\hat{g}_t = \nabla f(\theta_t) + \tau_t \quad and \quad E[\tau_t] = 0.$$

Note that τ_i is independent of τ_j if $i \neq j$.

Theorem 1 *Suppose that Assumptions 1, 2 and 3 are satisfied, and for some constant $C > 0$, $\|\alpha_t m_t/(\sqrt{v_t} + \epsilon)\| \leq C$, $\forall\, t > 0$, then we can get the following conclusion:*

$$\begin{aligned}
&E\big[\sum_{i=1}^{t} \alpha_t \langle \nabla f(\theta_i), \nabla f(\theta_i)/(\sqrt{v_t} + \epsilon)\rangle\big] \\
\leq &K_1 E\big[\sum_{i=1}^{t} \|\frac{\alpha_i \hat{g}_i}{\sqrt{v_i} + \epsilon}\|^2\big] \\
&+ K_2 E\big[\sum_{i=2}^{t} \|\frac{\alpha_i}{\sqrt{v_i} + \epsilon} - \frac{\alpha_{i-1}}{\sqrt{v_{i-1}} + \epsilon}\|_1\big] \\
&+ K_3 E\big[\sum_{i=2}^{t} \|\frac{\alpha_i}{\sqrt{v_i} + \epsilon} - \frac{\alpha_{i-1}}{\sqrt{v_{i-1}} + \epsilon}\|^2\big] + K_4,
\end{aligned}$$

where K_1, K_2, K_3 and K_4 are constants independent of T.

Corollary 1 *Assume $\exists\, c > 0$ such that $|(\hat{g}_1)_i| \geq c$, $\forall\, i \in [d]$, we have for any T,*

$$\min_{t \in [T]} E[\|f(\theta_t)\|^2] \leq \frac{1}{\log(T+2) - \log 2}\Big(P_1 + P_2 \frac{T-1}{T+1}\Big), \quad (6)$$

where P_1 and P_2 are constants independent of T. We can find that as iteration T increases, the minimum of training loss $f(\theta_t)$ is close to zero. It means our algorithm GAME can converge under the non-convex condition.

4. EXPERIMENTS

In the section, we compare our GAME algorithm with its counterparts, ADAM and AMSGRAD. All the codes are implemented in PyTorch, and will be made publicly available.

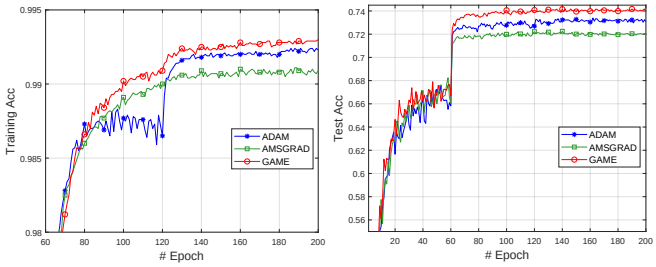
4.1. Hyperparameter Setting

We implement a proper search for the initial learning rate.

- ADAM: Although ADAM is an adaptive method, its initial learning rate is still carefully chosen from {1e-1, 1e-2, 1e-3, 1e-4, 1e-5}, as suggested in [12]. And the other three hyperparameters ($\beta_1, \beta_2, \epsilon$) are set to the default values, i.e., (0.9, 0.999, 1e-8).
- AMSGRAD: Following the setting in [15], AMSGRAD has the same hyperparameters as ADAM.
- GAME: To express the adaptivity in our algorithm, we use the same learning rate as ADAM. All the other hyperparameters are set to the same as ADAM.

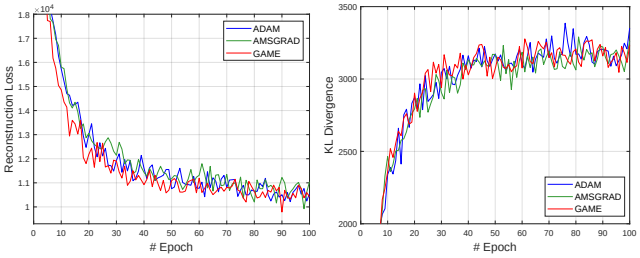
	ResNet-18	ResNet-50	ResNet-101	DenseNet-121	Transformer
ADAM	0.2688±0.0020	0.2558±0.0011	0.2338±0.0003	0.5434±0.0035	232.14±0.40
AMSGRAD	0.2774±0.0007	0.2734±0.0031	0.2693±0.0053	0.5427±0.0029	229.71±1.20
GAME	**0.2608±0.0020**	**0.2382±0.0003**	**0.2295±0.0013**	**0.5211±0.0038**	**221.80±0.20**

Table 1. The lowest error (e.g., for ResNet-101) or perplexity (for Transformer) of ADAM, AMSGRAD and GAME on the testing sets. We find that GAME achieves the best performance in all the experiments.



(a) Training accuracy for ResNet-18 (b) Training accuracy for ResNet-18

Fig. 3. Comparison of ADAM, AMSGRAD and GAME for training ResNet-18 on image recognition tasks.



(a) Reconstruction loss for VAE (b) KL divergence for VAE

Fig. 4. Comparison of ADAM, AMSGRAD and GAME on image generation tasks.

4.2. Image Classification

ResNet. We evaluate the performance of our GAME algorithm for training ResNet-18, ResNet-50 and ResNet-101 [2] on CIFAR-100 [19], as shown in Table 1 and Fig. 3. We find that our GAME has a competitive convergence rate on the training set and gets a training accuracy close to 100%. In particular, GAME achieves the highest test accuracy. Concretely, in terms of test accuracy, GAME is 0.6% better than ADAM and 1.6% better than AMSGRAD on ResNet-18, 1.7% better than ADAM and 3.5% better than AMSGRAD on ResNet-50, and 0.4% and 4.0% better than them on ResNet-101. Considering the speed of ADAM, AMSGRAD and GAME to achieve their highest accuracies, GAME is 2.4 times faster than ADAM, 2.3 times faster than AMSGRAD on ResNet-18, 2.2 times and 1.9 times faster on ResNet-50, and 1.6 times and 1.3 times faster than them on ResNet-101.

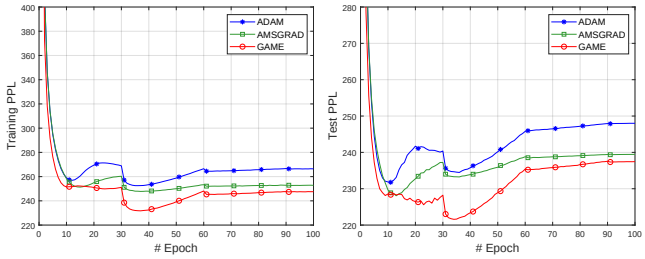
DenseNet. We apply GAME to train DenseNet-121 [20] on Tiny-ImageNet [21], as shown in Table 1. GAME gets an extremely similar training loss to ADAM and AMSGRAD, while GAME achieves 2.2% better test accuracy than ADAM and 2.1% better than AMSGRAD. In terms of the convergence speed, GAME is the same as ADAM and slightly faster than AMSGRAD by 1.1 times.

4.3. Generative Models

Variational Auto Encoder (VAE) [22] is a popular generative model. We also apply our GAME algorithm to train it on MNIST [23] and record the reconstruction loss and KL divergence, as shown in Fig. 4. We find that the three algorithms have a similar performance and GAME has a faster speed to minimize the reconstruction loss at the first 20 epochs.

4.4. Language Modeling

Finally, we implement a language modeling task on the Wikitext-2 dataset [24] with the Transformer [25], as shown in Table 1 and Fig. 5. We also find that GAME has the fastest convergence speed and lowest perplexity on both the training set and test set. The perplexity of language model produced by GAME is lower than that of ADAM by 10.3, and lower than that of AMSGRAD by 7.9, as shown in Table 1.



(a) Training perplexity for Transformer (b) Test perplexity for Transformer

Fig. 5. Comparison of ADAM, AMSGRAD and GAME for language modeling tasks.

5. CONCLUSIONS

In this paper, we proposed a new globally adaptive method (called GAME) for training various deep neural networks. GAME has a relatively smooth and global second-order moment estimate, enjoying a further exploration. We also provided some theoretical analysis, which shows that our GAME algorithm has a sublinear convergence rate in the non-convex setting. We conducted many experiments on image classification, generative models and language modeling tasks. All the results indicated that the global information based on gradients is rather valid for adaptive methods.

6. REFERENCES

[1] Shubham Toshniwal, Anjuli Kannan, Chung-Cheng Chiu, Yonghui Wu, Tara N Sainath, and Karen Livescu, "A comparison of techniques for language model integration in encoder-decoder speech recognition," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 369–375.

[2] 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*, 2016, pp. 770–778.

[3] Joseph Redmon and Ali Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[5] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems 32*, pp. 5754–5764. 2019.

[6] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436, 2015.

[7] Herbert Robbins and Sutton Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[8] Boris T Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[9] Yurii Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence o (1/kˆ 2)," in *Doklady an ussr*, 1983, vol. 269, pp. 543–547.

[10] John Duchi, Elad Hazan, and Yoram Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[11] Geoffrey Hinton, "Lecture 6d: a separate, adaptive learning rate for each connection. slides of lecture neural networks for machine learning," Tech. Rep., Technical report, Slides of Lecture Neural Networks for Machine Learning, 2012.

[12] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *In Proceedings of 3rd International Conference on Learning Representations*, 2015.

[13] Aryan Mokhtari and Alec Koppel, "High-dimensional nonconvex stochastic optimization by doubly stochastic successive convex approximation," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6287–6302, 2020.

[14] Aryan Mokhtari, Alec Koppel, and Alejandro Ribeiro, "A class of parallel doubly stochastic algorithms for large-scale learning," *arXiv preprint arXiv:1606.04991*, 2016.

[15] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar, "On the convergence of adam and beyond," in *6th International Conference on Learning Representations*, 2018.

[16] Simon Carbonnelle and Christophe De Vleeschouwer, "Layer rotation: a surprisingly simple indicator of generalization in deep networks?," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, vol. 1, p. 8.

[17] Sepp Hochreiter and Jürgen Schmidhuber, "Flat minima," *Neural Computation*, vol. 9, no. 1, pp. 1–42, 1997.

[18] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," in *5th International Conference on Learning Representations*, 2017.

[19] Alex Krizhevsky, Geoffrey Hinton, et al., "Learning multiple layers of features from tiny images," Tech. Rep., Citeseer, 2009.

[20] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[21] Feifei Li et al., "Tiny imagenet challenge," `https://tiny-imagenet.herokuapp.com/`, online.

[22] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *stat*, vol. 1050, pp. 1, 2014.

[23] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges, "The mnist database of handwritten digits," `http://yann.lecun.com/exdb/mnist/`, online.

[24] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher, "Pointer sentinel mixture models," *arXiv preprint arXiv:1609.07843*, 2016.

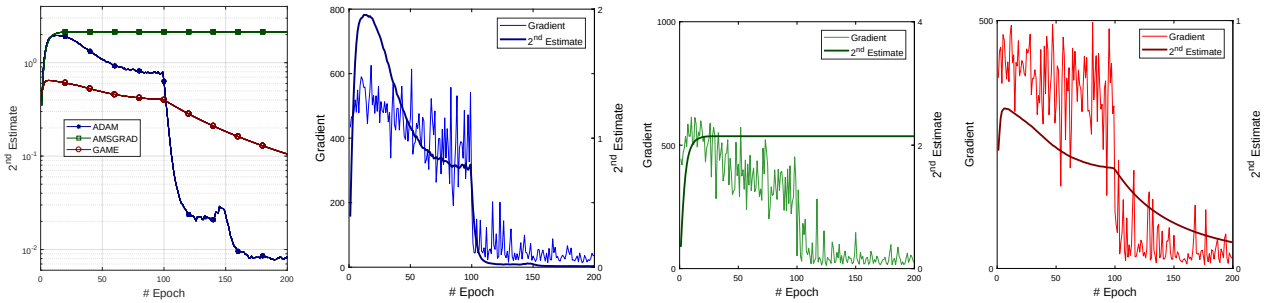
[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

Supplementary Materials for "GAME: A Globally Adaptive Method for Training Deep Neural Networks"

Paper ID: 1494

1. Gradient Visualization

In this section, we visualize gradient and the second-order estimate of ADAM, AMSGRAD and GAME. We can find each gradient has a bound, which empirically support assumption 2(gradient bound).



(a) Summary (b) ADAM (c) AMSGRAD (d) GAME

Figure 1: Comparison of ADAM, AMSGRAD and GAME for training ResNet on CIFAR 100. The L_1 norms of the second-order estimate and gradients in ADAM, AMSGRAD and GAME. As the gradient varies, ADAM also changes quickly because of its adaptivity, AMSGRAD stops updating at an early epoch, and our GAME changes relatively stably with more global information.

2. ALGORITHM

In this section, we introduce our algorithm with a slowly increasing weighted coefficient for past gradients in the second-order estimate. Algorithm 1 gives the pseudo-code of our proposed GAME. All the operations between vectors are element-wise. Let $f(\theta)$ be a stochastic objective the function can cause the stochasticity. We define t as the global step. Thus, we have $\hat{g}_t = \nabla_\theta f(\theta_t)$, which stands for the current sampling partial derivative of parameter θ_t in the loss function $f(\theta_t)$ at step t.

2.1. Notation

For two vectors a, b with the same dimensions, $c = a/b$ is defined as the element-wise division and c has the same dimension. a^2 means the element-wise square operation and $\sqrt{a}$ stands for the element-wise square root. $\|a\|_1$ is the L_1 norm of the vector a and $\|a\|$ is the L_2 norm of a. a_i means the ith

Algorithm 1 GAME

Input: δ_t: learning rate at iteration t.
$\beta \in [0, 1)$: Exponential decay rate for 1^{st} order estimate.
θ_0: initial parameter vector, $m_0 \leftarrow 0$ (Initial 1^{st} order estimate vector), $v_0 \leftarrow 0$ (Initial 2^{nd} order estimate vector), $t \leftarrow 0$ (Initial global step), $\epsilon \leftarrow 1e-8$.
$f(\theta)$: Objective function with $\theta \in \mathcal{F}$.
Current mini-batch I_t, current 1^{st} order estimate m_t and current 2^{nd} order estimate v_t.
while $t \neq$max iteration **do**
$t = t + 1$
$\hat{g}_t \leftarrow$ Stochastic Gradient $\nabla_\theta f_{I_t}(\theta_t)$;
$\hat{m}_t = \beta \cdot m_{t-1} + (1-\beta) \cdot \hat{g}_t$; ($1^{st}$ order estimate)
$m_t = \hat{m}_t / (1-\beta^t)$; (fix the estimate bias)
$v_t = (t \cdot v_{t-1} + \hat{g}_t^2)/(t+2)$ and $\hat{V}_t = \mathrm{diag}\,(v_t)$;
(2^{nd} order estimate)
$\theta_t = \Pi_{\mathcal{F}, \sqrt{\hat{V}_t}}(\theta_{t-1} - \delta_t \cdot m_t / (\sqrt{v_t} + \epsilon))$;
(update parameters)
end while

coordinate of vector a. We use $[d]$ to denote the sequence $[1, 2, 3, ..., d]$. $\mathcal{F}$ is the parameter domain. The projection $\Pi_{\mathcal{F},A}(y)$ for $A \in \mathcal{S}_+^d$ ($\mathcal{S}_+^d$ is a set of all positive definite $d \times d$ matrices) is defined as $\arg\min_{x \in \mathcal{F}} \left\| A^{1/2}(x-y) \right\|$ for y $\in \mathbb{R}^d$. $\mathcal{F}$ has bounded diameter D_∞ if $\|x-y\|_\infty \leq D_\infty$ for all $x, y \in \mathcal{F}$.

2.2. Non-convergence of ADAM

In this section, we show how exponential moving average in ADAM harms convergence in one-dimension convex optimization problem. Following the work in [2], we give the variable Γ which stands for the difference between two adjacent true learning rates. Only Γ_t keeping non-negative means convergence.

$$\Gamma_t = \left(\frac{\sqrt{V_t}}{\alpha_t} - \frac{\sqrt{V_{t-1}}}{\alpha_{t-1}} \right). \tag{1}$$

Lemma *With $\alpha_t = \alpha/(t+1)$ ($\alpha > 0$),we find $\exists\ T_1 \leq T_2$, $\forall\ \beta \in (0,1)$, $\forall\ t \in [T_1, T_2]$, $g_t = 0$, such that the positive semi-definiteness of Γ_t can not be satisfied.*

Proof

$$\frac{V_t}{\alpha_t^2} = \frac{(1-\beta_2)(t+1)^2 \beta_2^t}{\alpha^2} \sum_{i=1}^{t} \beta_2^{-i} g_i^2, \tag{2}$$

$$\frac{V_t}{\alpha_t^2} \geq \frac{V_{t-1}}{\alpha_{t-1}^2}. \tag{3}$$

If $\forall\ t > 0$, $\exists\ \beta_2 \in (0,1)$, the inequality (3) holds, ADAM will converge. However, our proof holds the opposite opinion. Under one-dimension convex optimization case, there exist some zero values in the

gradient sequence because the optimal minima may be reached and passed. But that does not mean the algorithm converges. We suppose that $\exists\, T_1 \leq T_2$, $\forall\, t \in [T_1, T_2]$, $g_t = 0$. *So* $\forall\, T_1 \leq t \leq T_2$, *we have*

$$\sum_{i=1}^{t} \beta_2^{-i} g_i^2 = \sum_{i=1}^{t-1} \beta_2^{-i} g_i^2. \tag{4}$$

To make the quantity Γ_t *in (1) positive semi-definiteness hold, we get the following limitations from the inequality (3).*

$$(t+1)^2 \beta_2^t \geq t^2 \beta_2^{t-1}, \tag{5}$$

$$(1-\beta_2)t^2 - 2\beta_2 t - \beta_2 \leq 0. \tag{6}$$

If $T_2 > \frac{\beta_2 + \sqrt{\beta_2}}{1-\beta_2}$, *the inequality (6) does not always hold* $\forall\ t > 0$. *Therefore,* β_2 *is a problem-dependent hyperparameter. Therefore, we complete the proof.*

We attribute this non-convergence issue to the exponential moving average (EMA) in ADAM. When the gradient becomes smaller (e.g., 0), the true learning rate $\frac{\alpha_t}{\sqrt{V_t}}$ becomes larger in spite of a decreasing learning rate α_t and the initial learning rate α (As shown in Figure 2, $|\theta_A^2 - \theta^*| > |\theta^0 - \theta^*| > |\theta_G^2 - \theta^*|$). This causes the non-convergence issue.

2.3. Our method

Therefore, how can we solve the aforementioned problem? We propose to make use of the global gradient to calculate the second-order estimation in an adaptive method. By rewriting the formula in ADAM, we model the second-order estimation as follows:

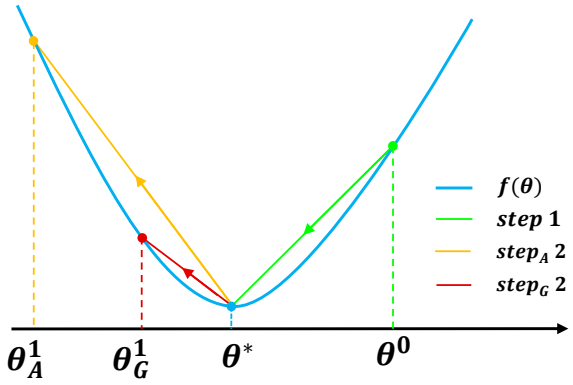


Figure 2: A schematic diagram of why ADAM diverges and GAME helps. When ADAM reaches the optimal minima θ^* alone the green line, it immediately gains a larger step-size according to the inequality (6) and moves away from θ^* alone the yellow line. But GAME still keeps true step-size decreasing and reaches a closer point to θ^* than θ^0 alone the red line. Note that "A" denotes the result of ADAM and "G" is the result of GAME.

$$ADAM: V_t = (1-\beta_2)\beta_2^t \sum_{i=1}^{t} \beta_2^{-i} g_i^2, \tag{7}$$

$$OURS: V_t = A(t)\sum_{i=1}^{t} B(i)g_i^2. \tag{8}$$

The update formula can be written as the following equivalent recurrence formula

$$OURS: V_t = \frac{A(t)}{A(t-1)}V_{t-1} + A(t)B(t)g_i^2. \tag{9}$$

Using our new model, for ADAM, $A(t) = (1-\beta_2)\beta_2^t, B(i) = \beta_2^{-i}$, we can find $A(t)$ in ADAM is decreasing as iteration t increases. And $B(i) = \beta_2^{-i}$ weights the past gradient in an exponential way. It relies the current gradient too much. To solve the problem, we propose to replace the exponential moving average with linear weight. We make $B(i) = O(i)$ to improve ADAM. Then we consider how to design $A(t)$. And we find there exists a relationship in ADAM. That is

$$\lim_{t\to+\infty} A(t)\sum_{i=1}^{t} B(i) = O(1). \tag{10}$$

In our algorithm, $B(i) = O(i)$, so we have $A(t) = O(1/t^2)$. Suppose $A(t) = 1/(t^2+3t+2)$ and $B(i) = i+1$, we get our GAME algorithm by rearranging the recurrence formula, which addresses the non-convergence issue of ADAM. We completely remove the exponential function in the second-order estimation of ADAM. By observing the non-zero weight for past gradient in ADAM and AMSGRAD, we find the closer gradients have larger weights.

Lemma *Our algorithm GAME can ensure the quantity Γ_t in (1) is non-negative, which addresses the non-convergence issue in ADAM.*

Proof

$$\begin{aligned}
\Gamma_t &= \frac{V_t}{{\alpha_t}^2} - \frac{V_{t-1}}{{\alpha_{t-1}}^2} \\
&= \frac{(t+1)^2}{\alpha^2}\frac{1}{(t+1)(t+2)}\sum_{i=1}^{t}(i+1)g_i^2 \\
&\quad - \frac{t^2}{\alpha^2}\frac{1}{t(t+1)}\sum_{i=1}^{t-1}(i+1)g_i^2 \\
&\geq \frac{1}{\alpha^2(t+1)}[\frac{(t+1)^2}{t+2} - \frac{t^2}{t+1}]\sum_{i=1}^{t-1}(i+1)g_i^2 \\
&= \frac{1}{\alpha^2(t+1)^2(t+2)}(t^2+3t+1)\sum_{i=1}^{t-1}(i+1)g_i^2 \\
&\geq 0 \quad (\forall t{>}0).
\end{aligned}$$

3. Non-convex Convergence Analysis

Considering that loss landscape for a deep neural network with multi-layer architecture and non-linear activation functions is highly non-convex, we give a theoretical guarantee for convergence. During the

training of a neural network, given a fixed length of input and target, we have the following objective function to minimize at step t:

$$f(\theta_t) := \frac{1}{n}\sum_{i=1}^{n} f_i(\theta_t),$$

where n is the number of training data. Our goal is to find the optimal solution in the following problem:

$$\theta^* = \arg\min_{\theta \in R^d} f(\theta),$$

where d is the dimension of the parameter vector. To get a clear statement, we make three standard assumptions, which are similar to [1].

Assumption 1 ***(L-smooth)*** *We attribute the smooth property to the loss function. $f(\theta)$ is first-order differentiable and has L-Lipschitz gradient. $\forall \theta_1, \theta_2 \in R^d$,*

$$\|\nabla f(\theta_1) - \nabla f(\theta_2)\| \leq L\|\theta_1 - \theta_2\|.$$

It is also lower bounded.

$$f(\theta^*) > -\infty,$$

where θ^* is an optimal solution.

Assumption 2 ***(Gradient Bound)*** *At global step t, the sampling gradient $\hat{g}_t$ and the real gradient $\nabla f(\theta_t)$ are both bounded, $\forall t > 1$,*

$$\|\nabla f(\theta_t)\| \leq M \quad and \quad \|\hat{g}_t\| \leq M.$$

Assumption 3 ***(Unbiased Estimate)*** *We assume the sampling gradient is unbiased and the noise in it is independent,*

$$\hat{g}_t = \nabla f(\theta_t) + \tau_t \quad and \quad E[\tau_t] = 0.$$

Note that τ_i is independent of τ_j if $i \neq j$.

Assumptions 1 (L-smooth) and 3 (Unbiased Estimate) are often used in convergence analysis for non-convex problems. Assumption 2 (Gradient Bound) is reasonable in many experiments (e.g., deep neural networks for image classification). In Figure 3, we find the sampling gradient has a limited bound and it oscillates within a limited value, which verifies our assumptions are rational.
In the following, we will give our theoretical results. To study how the learning rate δ_t and second-order estimate v_t make efforts to the convergence, we give and prove the following theorem 1. It has a restriction for them.

Theorem 1 *Suppose that Assumptions 1, 2 and 3 are satisfied, and for some constant $N > 0$, $\|\delta_t m_t/(\sqrt{v_t}+\epsilon)\| \leq N$, $\forall\, t > 0$, then we can get the following conclusion:*

$$E[\sum_{i=1}^{t} \delta_t \langle \nabla f(\theta_i), \nabla f(\theta_i)/(\sqrt{v_t} + \epsilon)\rangle]$$

$$
\begin{aligned}
\leq & K_1 E[\sum_{i=1}^{t} \| \frac{\delta_i \hat{g}_i}{\sqrt{v_i}+\epsilon} \|^2] \\
& + K_2 E[\sum_{i=2}^{t} \| \frac{\delta_i}{\sqrt{v_i}+\epsilon} - \frac{\delta_{i-1}}{\sqrt{v_{i-1}}+\epsilon} \|_1] \\
& + K_3 E[\sum_{i=2}^{t} \| \frac{\delta_i}{\sqrt{v_i}+\epsilon} - \frac{\delta_{i-1}}{\sqrt{v_{i-1}}+\epsilon} \|^2] + K_4,
\end{aligned}
\tag{11}
$$

where K_1, K_2, K_3 and K_4 are constants independent of T.

Corollary 1 *Assume $\exists\, c > 0$ such that $|(\hat{g}_1)_i| \geq c$, $\forall\, i \in [d]$, we have for any T,*

$$
\min E[\|f(\theta_t)\|^2] \leq \frac{1}{\log(T+2) - \log 2} \left(P_1 + P_2 \frac{T-1}{T+1} \right), \tag{12}
$$

where P_1 and P_2 are constants independent of T.

In this section, we get a convergence rate of our GAME on the training set, which is sublinear. The theoretical conclusion is based on the non-convex objective function and some standard assumptions. Indeed, the convergence rate on unseen data is rather attractive, and few works focus on this issue. The theory about the generalization rate is needed to be included in future work.

4. Learning Rates Sensitivity Analysis

In the section, we test the learning rate sensitivity of GAME, and compare its performance with those of ADAM and AMSGRAD, as shown in Figure 3.

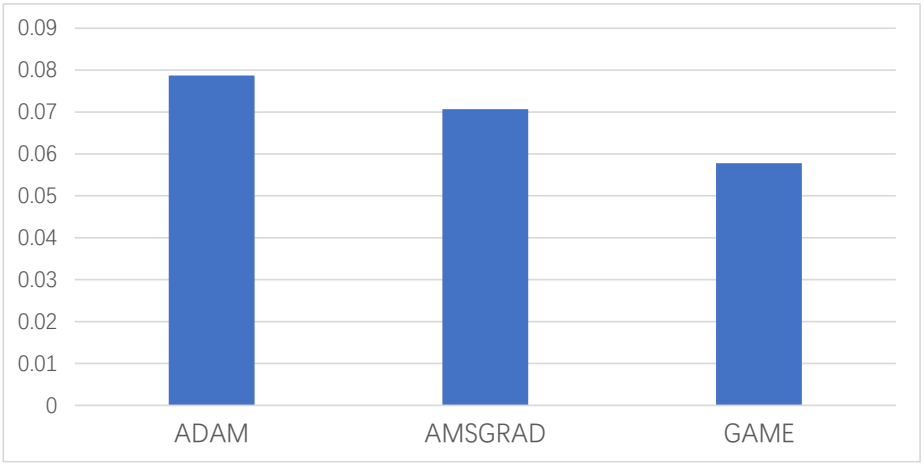


Figure 3: Variance of the three algorithms (i.e., ADAM, AMSGRAD, and GAME) with different learning rates for ResNet-18 on CIFAR-100. Learning rates are chosen from {1e-3, 1e-4, 1e-5}.

The adaptive methods are proposed to make optimization not so sensitive to their learning rate. Thus, the test for learning rate sensitivity analysis is important. We find that AMSGRAD has a slightly smaller variance than ADAM. This is because of invariance of the second-order estimate in AMSGRAD. While ADAM is easily affected by the current sampling gradient. Thus the invariance in AMSGRAD damages adaptivity. The proposed GAME method can make use of the advantages of ADAM and AMSGRAD. Therefore, GAME has the smallest variance for various learning rates.

5. β_2 Sensitivity

In this subsection, we illustrate that β_2 is a problem-dependent hyperparameter in Section 3.1 of main paper. We present the performances of ADAM with different values of β_2 in Figure 4. Note that we set β_1 to a popular value 0.9.

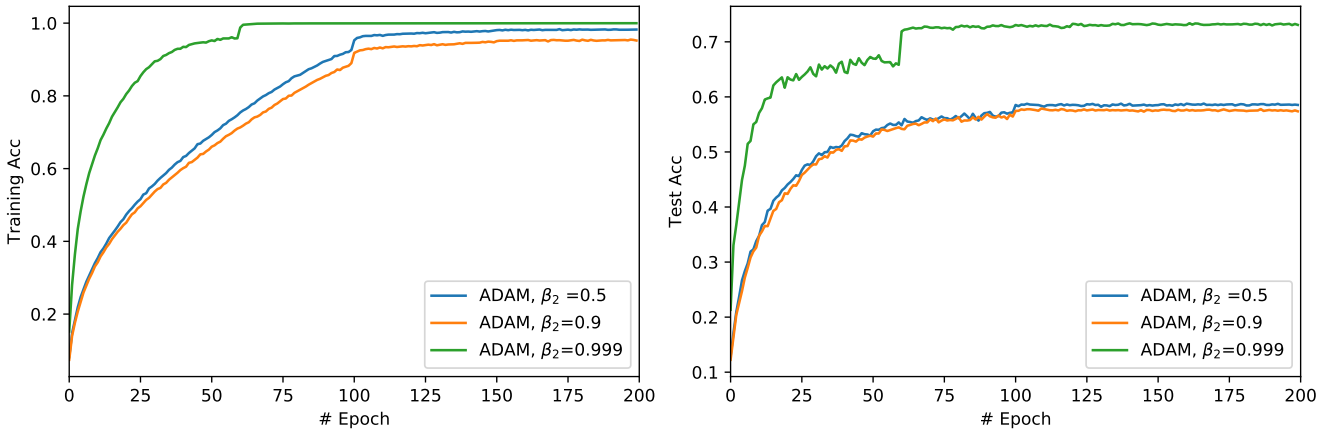


Figure 4: Training accuracy and test accuracy of ADAM with different values of β_2.

We find that different values of β_2 really harm optimization and generalization. This confirms our results that β_2 is task-dependent. Since we have removed the hyperparameter β_2 in GAME, then GAME will not suffer from this issue.

6. Hyperparameter Selection

We give all the hyperparameters (i.e., learning rate and weight decay) in our experiments, as shown in Tables 1-6.

optimizer	learning rate	β_1	β_2	weigh decay
ADAM	0.0001	0.9	0.999	0.0005
AMSGRAD	0.0001	0.9	0.999	0.0005
GAME	0.0001	0.9	N/A	0.0005

Table 1: ResNet-18 on CIFAR-100.

7. Additional Experiments

We list all evaluation on computer vision tasks in Fig.5.

optimizer	learning rate	β_1	β_2	weigh decay
ADAM	0.0001	0.9	0.999	0.0005
AMSGRAD	0.0001	0.9	0.999	0.0005
GAME	0.0001	0.9	N/A	0.0005

Table 2: ResNet-50 on CIFAR-100.

optimizer	learning rate	β_1	β_2	weigh decay
ADAM	0.0001	0.9	0.999	0.0005
AMSGRAD	0.0001	0.9	0.999	0.0005
GAME	0.0001	0.9	N/A	0.0005

Table 3: ResNet-101 on CIFAR-100.

optimizer	learning rate	β_1	β_2	weigh decay
ADAM	0.001	0.9	0.999	0.0005
AMSGRAD	0.001	0.9	0.999	0.0005
GAME	0.001	0.9	N/A	0.0005

Table 4: DenseNet-121 on CIFAR-100.

optimizer	learning rate	β_1	β_2	weigh decay
ADAM	0.001	0.9	0.999	0.0005
AMSGRAD	0.001	0.9	0.999	0.0005
GAME	0.001	0.9	N/A	0.0005

Table 5: VAE on MNIST.

optimizer	learning rate	β_1	β_2	weigh decay
ADAM	0.0001	0.9	0.999	0.0005
AMSGRAD	0.0001	0.9	0.999	0.0005
GAME	0.0001	0.9	N/A	0.0005

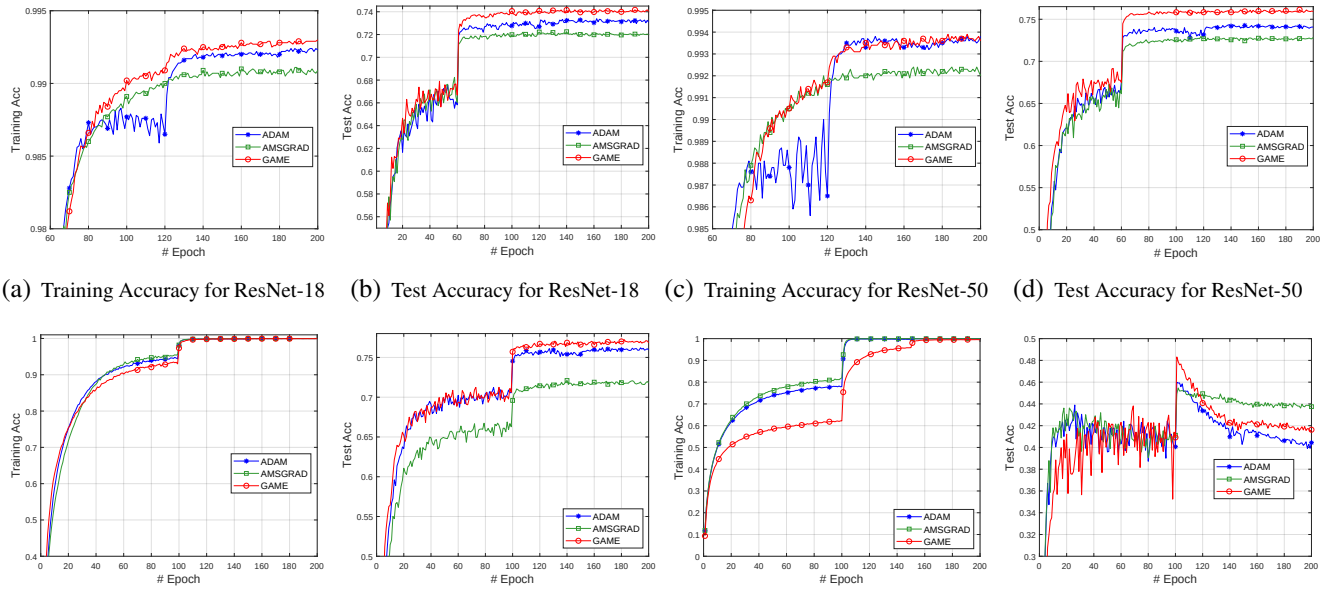
Table 6: Transformer on Wikitext-2.

8. Proof of Theorem 1 (Non-convex Convergence Analysis)

Proof *We first give a total inequality. Then we divide the inequality into several parts. And we give an upper bound for each part. Finally, we get the convergence rate.*

$$\theta_{t+1} = \theta_t - \frac{\delta}{1-\beta^t} \cdot \frac{\beta m_{t-1} + (1-\beta) g_t}{\sqrt{v_t} + \varepsilon}, \tag{13}$$

$$\theta_t = \theta_{t-1} - \delta \cdot \frac{m_{t-1}}{\sqrt{v_{t-1}} + \varepsilon}, \tag{14}$$



(a) Training Accuracy for ResNet-18 (b) Test Accuracy for ResNet-18 (c) Training Accuracy for ResNet-50 (d) Test Accuracy for ResNet-50

(e) Training Accuracy for ResNet-101 (f) Test Accuracy for ResNet-101 (g) Training Accuracy for DenseNet-121 (h) Test Accuracy for DenseNet-121

Figure 5: Comparison of ADAM, AMSGRAD and GAME for training various deep networks on image recognition tasks.

$$m_{t-1} = -\frac{\sqrt{v_{t-1}} + \varepsilon}{\delta} \left(\theta_t - \theta_{t-1}\right). \tag{15}$$

Thus, we have

$$\begin{aligned}
\theta_{t+1} - \theta_t &= \frac{\beta}{1-\beta^t} \cdot \frac{\sqrt{v_{t-1}} + \varepsilon}{\sqrt{v_t} + \varepsilon} \cdot \left(\theta_t - \theta_{t-1}\right) - \frac{\delta(1-\beta)}{1-\beta^t} \cdot \frac{g_t}{\sqrt{v_t} + \varepsilon} \\
&= \frac{\beta}{1-\beta^t} \cdot \left(\theta_t - \theta_{t-1}\right) + \frac{\beta}{1-\beta^t} \left(\frac{\sqrt{v_{t-1}} + \varepsilon}{\sqrt{v_t} + \varepsilon} - 1\right) \left(\theta_t - \theta_{t-1}\right) \\
&\quad - \frac{\delta(1-\beta)}{1-\beta^t} \frac{g_t}{\sqrt{v_t} + \varepsilon} \\
&= \frac{\beta}{1-\beta^t} \left(\theta_t - \theta_{t-1}\right) - \frac{\beta}{1-\beta^t} \left(\frac{\delta}{\sqrt{v_t} + \varepsilon} - \frac{\delta}{\sqrt{v_{t-1}} + \varepsilon}\right) m_{t-1} \\
&\quad - \frac{\delta(1-\beta)}{1-\beta^t} \frac{g_t}{\sqrt{v_t} + \varepsilon}.
\end{aligned} \tag{16}$$

Since $\theta_{t+1} - \theta_t = \left(1 - \frac{\beta}{1-\beta^t}\right) \theta_{t+1} + \frac{\beta}{1-\beta^t} \left(\theta_{t+1} - \theta_t\right) - \left(1 - \frac{\beta}{1-\beta^t}\right) \theta_t$*, so we have*

$$
\begin{aligned}
&\left(1-\frac{\beta}{1-\beta^t}\right)\theta_{t+1}+\frac{\beta}{1-\beta^t}\left(\theta_{t+1}-\theta_t\right)\\
=&\left(1-\frac{\beta}{1-\beta^t}\right)\theta_t+\theta_{t+1}-\theta_t\\
=&\left(1-\frac{\beta}{1-\beta^t}\right)\theta_t+\frac{\beta}{1-\beta^t}\left(\theta_t-\theta_{t-1}\right)-\frac{\beta}{1-\beta^t}\left(\frac{\delta}{\sqrt{v_t}+\varepsilon}-\frac{\delta}{\sqrt{v_{t-1}}+\varepsilon}\right)m_{t-1}\\
&-\frac{\delta(1-\beta)}{1-\beta^t}\frac{g_t}{\sqrt{v_t}+\varepsilon},
\end{aligned}
\tag{17}
$$

Divide both sides by $\frac{1-\beta-\beta_t}{1-\beta_t}$,

$$
\begin{aligned}
&\theta_{t+1}+\frac{\beta}{1-\beta^t-\beta}\left(\theta_{t+1}-\theta_t\right)\\
=&\theta_t+\frac{\beta}{1-\beta^t-\beta}\left(\theta_t-\theta_{t-1}\right)-\frac{\beta}{1-\beta^t-\beta}\left(\frac{\delta}{\sqrt{v_t}+\varepsilon}-\frac{\delta}{\sqrt{v_{t-1}}+\varepsilon}\right)m_{t-1}\\
&-\frac{\delta(1-\beta)}{1-\beta^t-\beta}\cdot\frac{g_t}{\sqrt{v_t}+\varepsilon}.
\end{aligned}
\tag{18}
$$

Based on the above results, we let

$$
y_t=\theta_t+\frac{\beta}{1-\beta^t-\beta}\left(\theta_t-\theta_{t-1}\right), \tag{19}
$$

Define the sequence,

$$
y_{t+1}=\theta_{t+1}+\frac{\beta}{1-\beta^{t+1}-\beta}\left(\theta_{t+1}-\theta_t\right), \tag{20}
$$

Then (6) can written as

$$
\begin{aligned}
y_{t+1}=&y_t+\left(\frac{\beta}{1-\beta^{t+1}-\beta}-\frac{\beta}{1-\beta^t-\beta}\right)\left(\theta_{t+1}-\theta_t\right)\\
&-\frac{\beta}{1-\beta^t-\beta}\left(\frac{\delta}{\sqrt{u_t}+\varepsilon}-\frac{\delta}{\sqrt{t_1}+\varepsilon}\right)m_{t-1}-\frac{\delta(1-\beta)}{1-\beta^t-\beta}\cdot\frac{g_t}{\sqrt{v_t}+\varepsilon},
\end{aligned}
\tag{21}
$$

and

$$
\begin{aligned}
y_{t+1}=&y_t-\left(\frac{\beta}{1-\beta^{t+1}-\beta}-\frac{\beta}{1-\beta^t-\beta}\right)\frac{\delta m_t}{\sqrt{v_t}+\varepsilon}\\
&-\frac{\beta}{1-\beta^t-\beta}\left(\frac{\delta}{\sqrt{v_t}+\varepsilon}-\frac{\delta}{\sqrt{v_{t-1}}+\varepsilon}\right)m_{t-1}-\frac{\delta(1-\beta)}{1-\beta^t-\beta}\cdot\frac{g_t}{\sqrt{v_t}+\varepsilon}
\end{aligned}
\tag{22}
$$

So we complete a large part of proof. Next we will divide the goal into several parts.

By the L*-smooth of* f,

$$
f\left(y_{t+1}\right)\leq f\left(y_t\right)+\left\langle\nabla f\left(y_t\right),d_t\right\rangle+\frac{L}{2}\left\|d_t\right\|^2 \tag{23}
$$

$$d_t = y_{t+1} - y_t, \tag{24}$$

$$\begin{aligned} d_t = &\left(\frac{\beta}{1-\beta^{t+1}-\beta} - \frac{\beta}{1-\beta^t-\beta}\right)\frac{\delta m_t}{\sqrt{v_t}+\varepsilon} - \frac{\beta}{1-\beta^t-\beta}\left(\frac{\delta}{\sqrt{v_t}+\varepsilon} - \frac{\delta}{\sqrt{v_{t-1}+\varepsilon}}\right)m_{t-1} \\ &- \frac{\delta(1-\beta)}{1-\beta^t-\beta}\frac{g_t}{\sqrt{v_t}+\varepsilon}. \end{aligned} \tag{25}$$

According to the above analysis, we have

$$\begin{aligned} &E\left[f\left(y_{t+1}\right) - f\left(y_1\right)\right] \\ &= E\left[\sum_{i=1}^{t}\left(f\left(y_{i+1}\right) - f\left(y_i\right)\right)\right] \\ &\leqslant E\left[\sum_{i=1}^{t}\left\langle \nabla f\left(y_i\right), d_i\right\rangle + \frac{L}{2}\left\|d_i\right\|^2\right] \\ &= -E\left[\sum_{i=1}^{t}\left\langle \nabla f\left(y_i\right), \frac{\beta}{1-\beta^i\beta}\left(\frac{\delta}{\sqrt{v_i}+\varepsilon} - \frac{\delta}{\sqrt{v_{i-1}}+\varepsilon}\right)m_{i-1}\right\rangle\right] \\ &-E\left[\sum_{i=1}^{t}\left\langle \nabla f\left(y_i\right), \frac{\delta(1-\beta)}{1-\beta^i-\beta}\frac{g_i}{\sqrt{v_i}+\varepsilon}\right\rangle\right] \\ &-E\left[\sum_{i=1}^{t}\left\langle \nabla f\left(y_i\right), \left(\frac{\beta}{1-\beta^{i+1}-\beta} - \frac{\beta}{1-\beta^i-\beta}\right)\frac{\delta m_i}{\sqrt{v_i}+\varepsilon}\right\rangle\right] \\ &+E\left[\sum_{i=1}^{t}\frac{L}{2}\left\|d_i\right\|^2\right] \\ &= T_1 + T_2 + T_3 + E\left[\sum_{i=1}^{\frac{t}{2}}\frac{L}{2}\left\|d_i\right\|^2\right]. \end{aligned} \tag{26}$$

And we also have

$$\begin{aligned} &E\left[\sum_{i=1}^{t}\frac{L}{2}\left\|d_i\right\|^2\right] \\ \leqslant &E\left[\sum_{i=1}^{t}\frac{3L}{2}\left\|\left(\frac{\beta}{1-\beta^{i+1}-\beta} - \frac{\beta}{1-\beta^i-\beta}\right)\frac{\delta m_i}{\sqrt{v_i}+\varepsilon}\right\|^2\right] \\ &+ E\left[\sum_{i=1}^{t}\frac{3L}{2}\left\|\frac{\beta}{1-\beta^i-\beta}\left(\frac{\delta}{\sqrt{v_i}+\varepsilon} - \frac{\delta}{\sqrt{v_{i-1}}+\varepsilon}\right)m_{i-1}\right\|^2\right] \\ &+ E\left[\frac{3L}{2}\sum_{i=1}^{t}\left\|\frac{\delta(1-\beta)}{1-\beta^i-\beta}\frac{g_i}{\sqrt{v_i}+\varepsilon}\right\|^2\right] \\ = &T_4 + T_5 + T_6, \end{aligned} \tag{27}$$

Therefore, we get the following result.

$$E\left[f\left(y_{t+1}\right) - f\left(y_1\right)\right] \leqslant T_1 + T_2 + T_3 + T_4 + T_5 + T_6. \tag{28}$$

In the next, we will prove each part in detail.

Given $\|g_t\| \leqslant H$, $m_t = \frac{\beta}{1-\beta^t} m_{t-1} + \frac{1-\beta}{1-\beta^t} g_t$ and assume $\|m_t\| \leqslant H, \|\nabla f(x)\| \leqslant H$, we have an upper bound for T_1.

$$\begin{aligned} T_1 =& - E\left[\sum_{i=1}^{t}\left\langle \nabla f\left(y_i\right), \frac{\beta}{1-\beta^i-\beta}\left(\frac{\delta}{\sqrt{v_i}+\varepsilon} - \frac{\delta}{\sqrt{v_{i-1}}+\varepsilon}\right) m_{i-1}\right\rangle\right] \\ \leqslant & E\left[\sum_{i=1}^{t}\left\|\nabla f\left(y_i\right)\right\|\left\|m_{i-1}\right\| \frac{\beta}{1-\beta^i-\beta}\sum_{j=1}^{d}\left|\left(\frac{\delta}{\sqrt{v_i}+\varepsilon} - \frac{\delta}{\sqrt{v_{i-1}+\varepsilon}}\right)_j\right|\right] \\ \leqslant & H^2 \frac{\beta}{1-2\beta} E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\left|\left(\frac{\delta}{\sqrt{v_i}+\varepsilon} - \frac{\delta}{\sqrt{v_{i-1}}+\varepsilon}\right)_j\right|\right] \end{aligned} \tag{29}$$

By assuming $\left\|\frac{\delta m_t}{\sqrt{v_t}+\varepsilon}\right\| \leqslant G$, we also have an upper bound for T_3.

$$T_3 = -E\left[\sum_{i=1}^{t}\left\langle \nabla f\left(y_i\right), \left(\frac{\beta}{1-\beta^{i-1}-\beta} - \frac{\beta}{1-\beta^i-\beta}\right)\frac{\delta m_i}{\sqrt{v_i}+\varepsilon}\right\rangle\right] \tag{30}$$

Therefore,

$$\begin{aligned} T_3 \leqslant & E\left[\sum_{i=1}^{t}\left|\frac{\beta}{1-\beta^{i+1}-\beta} - \frac{\beta}{1-\beta^i-\beta}\right|\frac{1}{2}\left(\left\|\nabla f\left(y_i\right)\right\|^2 + \left\|\frac{\delta m_i}{\sqrt{v_i}+\varepsilon}\right\|^2\right)\right] \\ \leqslant & E\left[\sum_{i=1}^{t}\left|\frac{\beta}{1-\beta^{i+1}-\beta} - \frac{\beta}{1-\beta^i-\beta}\right|\cdot\frac{1}{2}\left(H^2+G^2\right)\right] \\ = & \sum_{i=1}^{t}\left(\frac{\beta}{1-\beta^i-\beta} - \frac{\beta}{1-\beta^{i+1}-\beta}\right)\cdot\frac{1}{2}\left(H^2+G^2\right) \\ = & \left(\frac{\beta}{1-2\beta} - \frac{\beta}{1-\beta^{t+1}-\beta}\right)\left(H^2+G^2\right) \end{aligned} \tag{31}$$

And for T_4 and T_5, we have similar conclusions.

$$\begin{aligned} \frac{2}{3L}T_4 =& E\left[\sum_{i=1}^{t}\left\|\left(\frac{\beta}{1-\beta^{i+1}-\beta} - \frac{\beta}{1-\beta^i-\beta}\right)\frac{\delta m_i}{\sqrt{v_i}+\varepsilon}\right\|^2\right] \\ \leqslant & E\left[\sum_{i=1}^{t}\left(\frac{\beta}{1-\beta^{i+1}-\beta} - \frac{\beta}{1-\beta^i-\beta}\right)^2 G^2\right] \\ \leqslant & \left(\frac{\beta}{1-2\beta} - \frac{\beta}{1-\beta^{t+1}-\beta}\right)^2 G^2 \end{aligned} \tag{32}$$

$$\begin{aligned} \frac{2}{3L}T_5 =& E\left[\sum_{i=1}^{t}\left\|\frac{\beta}{1-\beta^i-\beta}\left(\frac{\delta}{\sqrt{v_i}+\varepsilon} - \frac{\delta}{\sqrt{v_{i-1}}+\varepsilon}\right) m_{i-1}\right\|^2\right] \\ \leqslant & \left(\frac{\beta}{1-2\beta}\right)^2 H^2 E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\left(\frac{\delta}{\sqrt{v_i}+\varepsilon} - \frac{\delta}{\sqrt{v_{i-1}}+\varepsilon}\right)_j^2\right] \end{aligned} \tag{33}$$

Then we give the upper bound of T_2. Let $\theta_1 = \theta_0$.

$$T_2 = -E\left[\sum_{i=1}^{t}\left\langle \nabla f\left(y_i\right), \frac{\delta(1-\beta)}{1-\beta^i-\beta}\frac{g_i}{\sqrt{v_i}+\varepsilon}\right\rangle\right] \tag{34}$$

$$y_i - \theta_i = \frac{\beta}{1-\beta^i-\beta}\left(\theta_i - \theta_{i-1}\right) = -\frac{\beta}{1-\beta^i-\beta}\frac{\delta m_{i-1}}{\sqrt{v_{i-1}}+\varepsilon} \tag{35}$$

$$y_1 = \theta_1 + \frac{1}{1-2\beta}\left(\theta_1 - \theta_0\right) = \theta_1 \tag{36}$$

And we have

$$\begin{aligned} T_2 &= -E\left[\sum_{i=1}^{t}\left\langle \nabla f\left(y_i\right), \frac{\delta(1-\beta)}{1-\beta^i-\beta}\frac{g_i}{\sqrt{v_i}+\varepsilon}\right\rangle\right] \\ &= -E\left[\sum_{i=1}^{t}\frac{\delta(1-\beta)}{1-\beta^i-\beta}\left\langle \nabla f\left(\theta_i\right), \frac{g_i}{\sqrt{v_i}+\varepsilon}\right\rangle\right] \\ &\quad - E\left[\sum_{i=1}^{t}\frac{\delta(1-\beta)}{1-\beta^i-\beta}\left\langle \nabla f\left(y_i\right) - \nabla f\left(\theta_i\right), \frac{g_i}{\sqrt{v_i}+\varepsilon}\right\rangle\right] \end{aligned} \tag{37}$$

And

$$\begin{aligned} &-E\left[\textstyle\sum_{i=1}^{t}\frac{\delta(1-\beta)}{1-\beta^i-\beta}\left\langle \nabla f\left(y_i\right) - \nabla f\left(\theta_i\right), \frac{g_i}{\sqrt{v_i}+\varepsilon}\right\rangle\right] \\ \leqslant\ & E\left[\textstyle\sum_{i=2}^{t}\frac{1}{2}\left\|\nabla f\left(y_i\right) - \nabla f\left(\theta_i\right)\right\|^2 + \frac{1}{2}\left\|\frac{\delta(1-\beta)}{1-\beta^i-\beta}\cdot\frac{g_i}{\sqrt{v_i}+\varepsilon}\right\|^2\right] \\ \leqslant\ & \frac{L^2}{2}T_7 + \frac{1}{2}E\left[\textstyle\sum_{i=2}^{t}\left\|\frac{\delta(1-\beta)}{1-\beta^i-\beta}\cdot\frac{g_i}{\sqrt{v_i}+\varepsilon}\right\|^2\right] \end{aligned} \tag{38}$$

Note that

$$\begin{aligned} &\left\|\nabla f\left(y_i\right) - \nabla f\left(\theta_i\right)\right\| \\ \leqslant\ & L\left\|y_i - \theta_i\right\| \\ =\ & L\left\|\frac{\beta}{1-\beta^i-\beta}\frac{\delta m_{i-1}}{\sqrt{v_i}+\varepsilon}\right\| \end{aligned} \tag{39}$$

$$T_7 = E\left[\sum_{i=2}^{t}\left\|\frac{B}{1-\beta^i-\beta}\frac{\delta m_{i-1}}{\sqrt{v_{i-1}}+\varepsilon}\right\|^2\right] \tag{40}$$

In terms of the following sequence,

$$m_i = \frac{\beta}{1-\beta^i}m_{i-1} + \frac{1-\beta}{1-\beta^i}g_i \tag{41}$$

we have

$$m_i = \sum_{k=1}^{i}\left[\left(\prod_{i=k+1}^{i}\frac{\beta}{1-\beta^l}\right)\left(\frac{1-\beta}{1-\beta^k}g_k\right)\right] \tag{42}$$

$$
\begin{aligned}
T_7 &\leq \left(\frac{\beta}{1-2\beta}\right)^2 E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\left(\frac{\delta m_{i-1}}{\sqrt{v_{i-1}}+\varepsilon}\right)_j^2\right] \\
&= \left(\frac{\beta}{1-2\beta}\right)^2 E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\left(\sum_{k=1}^{i-1}\frac{\delta\left(\prod_{l=k+1}^{i}\frac{\beta}{1-\beta^l}\right)\left(\frac{1-\beta}{1-\beta^k}g_k\right)}{\sqrt{v_{i-1}}+\varepsilon}\right)_j^2\right] \\
&\leq 2\left(\frac{\beta}{1-2\beta}\right)^2 E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\left(\sum_{k=1}^{i-1}\frac{\delta\left(\prod_{l=k+1}^{i}\frac{\beta}{1-\beta^l}\right)\left(\frac{1-\beta}{1-\beta^k}g_k\right)}{\sqrt{v_k}+\varepsilon}\right)_j^2\right] \\
&+ 2\left(\frac{\beta}{1-2\beta}\right)^2 E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\left(\sum_{k=1}^{i-1}\left(\prod_{i=k+1}^{i}\frac{\beta}{1-\beta^l}\right)\left(\frac{1-\beta}{1-\beta^k}\right)(g_k)\,j\left(\frac{\delta}{\sqrt{v_{i-1}+\varepsilon}}-\frac{\delta}{\sqrt{v}_k+\varepsilon}\right)\right)_j^2\right]
\end{aligned}
\tag{43}
$$

We get the upper bound for T_7. The following gives the upper bound of T_8 and T_9.

$$
\begin{aligned}
T_8 =& E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\left(\sum_{k=1}^{i-1}\frac{\delta\left(\prod_{i=k+1}^{j}\left(\frac{\beta}{1-\beta^l}\right)\left(\frac{1-\beta}{1-\beta^k}g_k\right)\right)}{\sqrt{v_k}+\varepsilon}\right)_j^2\right] \\
&= E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\sum_{k=1}^{i-1}\sum_{p=1}^{i-1}\left(\frac{\delta g_k}{\sqrt{v_k}+\varepsilon}\right)_j\left(\prod_{l=k+1}^{i-1}\frac{\beta}{1-\beta^l}\right)\left(\frac{1-\beta}{1-\beta^k}\right)\left(\frac{\delta g_p}{\sqrt{v_p}+\varepsilon}\right)_j\left(\prod_{q=p+1}^{i-1}\frac{\beta}{1-\beta^q}\right)\left(\frac{1-\beta}{1-\beta^p}\right)\right] \\
&\leq E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\sum_{k=1}^{j-1}\sum_{p=1}^{i-1}\left(\frac{\beta}{1-\beta}\right)^{i-k-1}\left(\frac{\beta}{1-\beta}\right)^{i-p-1}\cdot\frac{1}{2}\left(\left(\frac{\delta g_k}{\sqrt{\nu_k}+\varepsilon}\right)_j^2+\left(\frac{\delta g_p}{\sqrt{\nu_p+\varepsilon}}\right)_j^2\right)\right] \\
&= E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\sum_{k=1}^{i-1}\left(\frac{\beta}{1-\beta}\right)^{i-k-1}\left(\frac{\delta g_k}{\sqrt{v_k}+\varepsilon}\right)_j^2\sum_{p=1}^{i-1}\left(\left(\frac{\beta}{1-\beta}\right)^{i-p-1}\right)\right] \\
&\leqslant \frac{1-\beta}{1-2\beta}E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\sum_{k=1}^{i-1}\left(\frac{\beta}{1-\beta}\right)^{i-k-1}\left(\frac{\delta g_k}{\sqrt{v_k}+\varepsilon}\right)_j^2\right] \\
&= \frac{1-\beta}{1-2\beta}E\left[\sum_{k=1}^{t-1}\sum_{j=1}^{d}\sum_{i=k+1}^{t}\left(\frac{\beta}{1-\beta}\right)^{i-k-1}\left(\frac{\delta g_k}{\sqrt{V_k}+\varepsilon}\right)_j^2\right] \\
&\leqslant \left(\frac{1-\beta}{1-2\beta}\right)^2 E\left[\sum_{k=1}^{t-1}\sum_{j=1}^{d}\left(\frac{\delta g_k}{\sqrt{\nu_k}+\varepsilon}\right)_j^2\right] \\
&= \left(\frac{1-\beta}{1-2\beta}\right)^2 E\left[\sum_{i=1}^{t-1}\left\|\frac{\delta g_i}{\sqrt{v_i}+\varepsilon}\right\|^2\right]
\end{aligned}
\tag{44}
$$

$$
\begin{aligned}
T_9 =& E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\left(\sum_{k=1}^{i-1}\left(\prod_{i=k+1}^{i}\frac{\beta}{1-\beta^k}\right)\left(\frac{1-\beta}{1-\beta^k}\right)(g_k)_j\left(\frac{\delta}{\sqrt{v_{i-1}+\varepsilon}}-\frac{\delta}{\sqrt{v_k}+\varepsilon}\right)_j\right)^2\right] \\
\leqslant & H^2 E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\left(\sum_{k=1}^{i-1}\left(\prod_{l=k+1}^{i}\frac{\beta}{1-\beta^l}\right)\left|\frac{\delta}{\sqrt{v_i}+\varepsilon}-\frac{\delta}{\sqrt{v_k}+\varepsilon}\right|_j\right)^2\right] \\
\leqslant & H^2 E\left[\sum_{i=1}^{t-1}\sum_{j=1}^{d}\left(\sum_{k=1}^{i}\left(\frac{\beta}{1-\beta}\right)^{i-k}\left|\frac{\delta}{\sqrt{v_i}+\varepsilon}-\frac{\delta}{\sqrt{v_k}+\varepsilon}\right|_j\right)^2\right] \\
\leqslant & H^2 E\left[\sum_{i=1}^{t-1}\sum_{j=1}^{d}\left(\sum_{k=1}^{i}\left(\frac{\beta}{1-\beta}\right)^{i-k}\sum_{l=k+1}^{i}\left|\frac{\delta}{\sqrt{v_l}+\varepsilon}-\frac{\delta}{\sqrt{v_{l-1}}+\varepsilon}\right|_j\right)^2\right] \\
\leqslant & H^2\left(\frac{1-\beta}{1-2\beta}\right)^2\left(\frac{\beta}{1-2\beta}\right)^2 E\left[\sum_{j=1}^{d}\sum_{i=2}^{t-1}\left|\frac{\delta}{\sqrt{v_l}+\varepsilon}-\frac{\delta}{\sqrt{v_{l-1}}+\varepsilon}\right|_j^2\right]
\end{aligned}
\tag{45}
$$

Now let us try to prove Eq. (25). For the second term in (25) and based on (26), (27), (32) and (33), we have

$$
\begin{aligned}
& -E\left[\sum_{i=1}^{t}\frac{\delta(1-\beta)}{1-\beta^i-\beta}\left\langle\nabla f\left(y_i\right)-\nabla f\left(\theta_i\right),\frac{g_i}{\sqrt{v_i}+\varepsilon}\right\rangle\right] \\
& \leqslant L^2\left(\frac{\beta}{1-2\beta}\right)^2\left(\frac{1-\beta}{1-2\beta}\right)^2 E\left[\sum_{i=1}^{t-1}\left\|\frac{\delta g_i}{\sqrt{v_i}+\varepsilon}\right\|^2\right] \\
& + L^2H^2\left(\frac{1-\beta}{1-2\beta}\right)^2\left(\frac{\beta}{1-2\beta}\right)^4 E\left[\sum_{j=1}^{d}\sum_{i=2}^{t-1}\left|\frac{\delta}{\sqrt{v_l}+\varepsilon}-\frac{\delta}{\sqrt{v_{l-1}}+\varepsilon}\right|_j^2\right] \\
& + \frac{1}{2}E\left[\sum_{i=2}^{t}\left\|\frac{\delta(1-\beta)}{1-\beta^i-\beta}\cdot\frac{g_i}{\sqrt{v_i}+\varepsilon}\right\|^2\right]
\end{aligned}
\tag{46}
$$

Return to the equation (25), by assuming $g_t=\nabla f\left(\theta_t\right)+\xi_t$, $E\left[\xi_t\right]=0$, we have

$$
\begin{aligned}
& E\left[\sum_{i=1}^{t}\frac{\delta(1-\beta)}{1-\beta^i-\beta}\left\langle\nabla f\left(\theta_i\right),\frac{g_i}{\sqrt{v_i}+\varepsilon}\right\rangle\right] \\
& = E\left[\sum_{i=1}^{t}\frac{\delta(1-\beta)}{1-\beta^i-\beta}\left\langle\nabla f\left(\theta_i\right),\frac{\nabla f\left(\theta_i\right)+\xi_i}{\sqrt{v_i}+\varepsilon}\right\rangle\right] \\
& = E\left[\sum_{i=1}^{t}\frac{\delta(1-\beta)}{1-\beta^i-\beta}\left\langle\nabla f\left(\theta_i\right),\frac{\nabla f\left(\theta_i\right)}{\sqrt{v_i}+\varepsilon}\right\rangle\right] \\
& + E\left[\sum_{i=1}^{t}\frac{\delta(1-\beta)}{1-\beta^i-\beta}\left\langle\nabla f\left(\theta_i\right),\frac{\xi_i}{\sqrt{v_i}+\varepsilon}\right\rangle\right]
\end{aligned}
\tag{47}
$$

$$
\begin{aligned}
&E\left[\sum_{i=1}^{t}\frac{\delta(1-\beta)}{1-\beta^i-\beta}\left\langle\nabla f\left(\theta_i\right),\frac{\xi_i}{\sqrt{v_i}+\varepsilon}\right\rangle\right] \\
&=E\left[\sum_{i=2}^{t}\frac{(1-\beta)}{1-\beta^i-\beta}\left\langle\nabla f\left(\theta_i\right),\quad \xi_i\left(\frac{\delta}{\sqrt{v_i}+\varepsilon}-\frac{\delta}{\sqrt{v_{i-1}+\varepsilon}}\right)\right\rangle\right] \\
&+E\left[\sum_{i=2}^{t}\frac{\delta(1-\beta)}{1-\beta^i-\beta}\left\langle\nabla f\left(\theta_i\right),\frac{\xi_i}{\sqrt{v_{i-1}}+\varepsilon}\right\rangle\right]+E\left[\frac{\delta(1-\beta)}{1-2\beta}\left\langle\nabla f\left(\theta_1\right),\frac{\xi_1}{\sqrt{v_1}+\varepsilon}\right\rangle\right] \\
&\geqslant E\left[\sum_{i=2}^{t}\frac{1-\beta}{1-\beta^i-\beta}\left\langle\nabla f\left(\theta_i\right),\xi_i\left(\frac{\delta}{\sqrt{v}_i+\varepsilon}-\frac{\delta}{\sqrt{v_{i-1}}+\varepsilon}\right)\right\rangle\right] \\
&-2H^2E\left[\sum_{j=1}^{d}\frac{1-\beta}{1-2\beta}\left(\frac{\delta}{\sqrt{v}_1+\varepsilon}\right)_j\right]
\end{aligned}
\tag{48}
$$

$$
\begin{aligned}
&E\left[\sum_{i=2}^{t}\frac{1-\beta}{1-\beta^i-\beta}\left\langle\nabla f\left(\theta_i\right),\xi_i\left(\frac{\delta}{\sqrt{v}_i+\varepsilon}-\frac{\delta}{\sqrt{v_{i-1}}+\varepsilon}\right)\right\rangle\right] \\
&=E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\frac{1-\beta}{1-\beta^i-\beta}\left(\nabla f\left(\theta_i\right)\right)_j\left(\xi_i\right)_j\left(\frac{\delta}{(\sqrt{v_i})_j+\varepsilon}-\frac{\delta}{(\sqrt{v_{i-1}})_j+\varepsilon}\right)\right] \\
&\geqslant -E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\frac{1-\beta}{1-\beta^i-\beta}|\left(\nabla f\left(\theta_i\right)\right)_j||\left(\xi_i\right)_j|\frac{\delta}{(\sqrt{v_i})_j+\varepsilon}-\frac{\delta}{(\sqrt{v_{i-1}})_j+\varepsilon}|\right] \\
&\geqslant -2H^2E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\left|\frac{\delta}{(\sqrt{v_i})_j+\varepsilon}-\frac{\delta}{(\sqrt{v_{i-1}})_j+\varepsilon}\right|\cdot\frac{1-\beta}{1-\beta^i-\beta}\right]
\end{aligned}
\tag{49}
$$

Based on the descriptions in (37), we have

$$
\begin{aligned}
&-E\left[\sum_{i=1}^{t}\frac{\delta(1-\beta)}{1-\beta^i-\beta}<\nabla f\left(\theta_i\right),\frac{g_i}{\sqrt{v_i}+\varepsilon}>\right] \\
&\leqslant 2H^2E\left[\sum_{i=2}^{t}\sum_{j=1}^{d}\left|\frac{\delta}{\sqrt{v_i}+\varepsilon}-\frac{\delta}{\sqrt{v_{i-1}}+\varepsilon}\right|_j\cdot\frac{1-\beta}{1-\beta^i-\beta}\right] \\
&+2H^2E\left[\sum_{j=1}^{d}\frac{1-\beta}{1-2\beta}\left(\frac{\delta}{\sqrt{v_1}+\varepsilon}\right)_j\right] \\
&-E\left[\sum_{i=1}^{t}\frac{\delta(1-\beta)}{1-\beta^i-\beta}\left\langle\nabla f\left(\theta_i\right),\frac{\nabla f\left(\theta_i\right)}{\sqrt{v_i}+\varepsilon}\right\rangle\right]
\end{aligned}
\tag{50}
$$

At last, we return to the beginning

$$
\begin{aligned}
&E\left[f\left(y_{t+1}\right)-f\left(y_1\right)\right] \leqslant \sum_{i=1}^{6} T_i \\
&\leqslant\left(H^2 \frac{\beta}{1-2 \beta}+2 H^2 \frac{1-\beta}{1-2 \beta}\right) E\left[\sum_{i=2}^{t} \sum_{j=1}^{d}\left|\left(\frac{\delta}{\sqrt{v_i}+\varepsilon}-\frac{\delta}{\sqrt{v_{i-1}}+\varepsilon}\right)_j\right|\right] \\
&+\left(\frac{3 L}{2}\left(\frac{\beta}{1-2 \beta}\right)^2 H^2+L^2 H^2\left(\frac{1-\beta}{1-2 \beta}\right)^2\left(\frac{\beta}{1-2 \beta}\right)^4\right) E\left[\sum_{j=1}^{d} \sum_{i=2}^{t-1}\left(\frac{\delta}{\sqrt{v_i}+\varepsilon}-\frac{\delta}{\sqrt{v_{i-1}}+\varepsilon}\right)_j^2\right] \\
&+\left(\frac{3 L}{2}\left(\frac{1-\beta}{1-2 \beta}\right)^2+L^2\left(\frac{\beta}{1-2 \beta}\right)^2\left(\frac{1-\beta}{1-2 \beta}\right)^2+\frac{1}{2}\left(\frac{1-\beta}{1-2 \beta}\right)^2\right) E\left[\sum_{i=1}^{t}\left\|\frac{\delta g_i}{\sqrt{v_i}+\varepsilon}\right\|^2\right] \\
&+\left(\frac{\beta}{1-2 \beta}-\frac{\beta}{1-\beta^{t+1}-\beta}\right)\left(H^2+G^2\right)+\frac{3 L}{2}\left(\frac{\beta}{1-2 \beta}-\frac{\beta}{1-\beta^{t+1}-\beta}\right) G^2 \\
&+2 H^2 \frac{1-\beta}{1-2 \beta} E\left[\sum_{j=1}^{d}\left(\frac{\delta}{\sqrt{v_1}+\varepsilon}\right)_j\right]-\frac{1-\beta}{1-2 \beta} E\left[\sum_{i=1}^{t} \delta\left\langle \nabla f\left(\theta_i\right), \frac{\nabla f\left(\theta_i\right)}{\sqrt{v_i}+\varepsilon}\right\rangle\right]
\end{aligned}
\tag{51}
$$

$$
\begin{aligned}
&E\left[\sum_{r=1}^{t} \delta\left\langle \nabla f\left(\theta_i\right), \frac{\nabla f\left(\theta_i\right)}{\sqrt{v_i}+\varepsilon}\right\rangle\right] \\
&\leq \frac{1-2 \beta}{1-\beta}\left(\frac{3 L}{2}\left(\frac{1-\beta}{1-2 \beta}\right)^2+L^2\left(\frac{\beta}{1-2 \beta}\right)^2\left(\frac{1-\beta}{1-2 \beta}\right)^2+\frac{1}{2}\left(\frac{1-\beta}{1-2 \beta}\right)^2\right] E\left[\sum_{i=1}^{t}\left\|\frac{\delta g_i}{\sqrt{v_i}+\varepsilon}\right\|^2\right] \\
&\quad+\frac{1-2 \beta}{1-\beta}\left(H^2 \frac{\beta}{1-2 \beta}+2 H^2 \frac{1-\beta}{1-2 \beta}\right) E \sum_{i=2}^{t}\left\|\frac{\delta}{\sqrt{v_i}+\varepsilon}-\frac{\delta}{\sqrt{v_{i-1}}+\varepsilon}\right\|_1 \\
&\quad+\frac{1-2 \beta}{1-\beta} \frac{3 L}{2}\left(\frac{\beta}{1-2 \beta}\right)^2 H^2+L^2 H^2\left(\frac{1-\beta}{1-2 \beta}\right)^2\left(\frac{\beta}{1-2 \beta}\right)^4 E \sum_{i=2}^{t-1}\left\|\frac{\delta}{\sqrt{v_i}+\varepsilon}-\frac{\delta}{\sqrt{v_{i-1}}+\varepsilon}\right\|^2 \\
&\quad+\frac{1-2 \beta}{1-\beta}\left(\frac{\beta}{1-2 \beta}-\frac{\beta}{1-\beta^{t+1}-\beta}\right)\left(H^2+G^2\right)+\frac{3 L}{2}\left(\frac{\beta}{1-2 \beta}-\frac{\beta}{1-\beta^{t+1}-\beta}\right) G^2 \\
&\quad+2 H^2 \frac{1-\beta}{1-2 \beta} E\left\|\frac{\delta}{\sqrt{v_1}+\varepsilon}\right\|_1+E\left[f\left(y_1\right)-f\left(y^*\right)\right].
\end{aligned}
\tag{52}
$$

We first bound non-constant terms in RHS of (11) in main paper, which is given by

$$
E\left[K_1 \sum_{t=1}^{T}\left\|\alpha_t g_t / \sqrt{\hat{v}_t}\right\|^2+K_2 \sum_{t=2}^{T}\left\|\frac{\alpha_t}{\sqrt{\hat{v}_t}}-\frac{\alpha_{t-1}}{\sqrt{\hat{v}_{t-1}}}\right\|_1+K_3 \sum_{t=2}^{T-1}\left\|\frac{\alpha_t}{\sqrt{\hat{v}_t}}-\frac{\alpha_{t-1}}{\sqrt{\hat{v}_{t-1}}}\right\|^2\right]+K_4. \tag{53}
$$

For the term with K_1, assume $\min_{j \in[d]}\left(\sqrt{\hat{v}_1}\right)_j \geq c>0$ (this is reasonable, please see Figure 1 in our

main paper), we have

$$E\left[\sum_{t=1}^{T}\left\|\alpha_t g_t/\sqrt{\hat{v}_t}\right\|^2\right] \tag{54}$$

$$\leq E\left[\sum_{t=1}^{T}\left\|\alpha_t g_t/c\right\|^2\right] = E\left[\sum_{t=1}^{T}\left\|\frac{1}{\sqrt{t}}g_t/c\right\|^2\right] = E\left[\sum_{t=1}^{T}\left(\frac{1}{c(t+1)}\right)^2\left\|g_t\right\|^2\right] \tag{55}$$

$$\leq H^2/c^2\sum_{t=1}^{T}\frac{1}{(t+1)^2} \leq H^2/c^2(\frac{1}{2}-\frac{1}{T}) \tag{56}$$

where the first inequality is due to $\left(\hat{v}_t\right)_j \geq c^2$*, and the last inequality is due to* $\sum_{t=1}^{T} 1/t \leq 1+\log T$*.*

For the term with C_2*, we have*

$$E\left[\sum_{t=2}^{T}\left\|\frac{\alpha_t}{\sqrt{\hat{v}_t}}-\frac{\alpha_{t-1}}{\sqrt{\hat{v}_{t-1}}}\right\|_1\right] = E\left[\sum_{j=1}^{d}\sum_{t=2}^{T}\left(\frac{\alpha_{t-1}}{\left(\sqrt{\hat{v}_{t-1}}\right)_j}-\frac{\alpha_t}{\left(\sqrt{\hat{v}_t}\right)_j}\right)\right] \tag{57}$$

$$= E\left[\sum_{j=1}^{d}\left(\frac{\alpha_1}{\left(\sqrt{\hat{v}_1}\right)_j}-\frac{\alpha_T}{\left(\sqrt{\hat{v}_T}\right)_j}\right)\right] \leq E\left[\sum_{j=1}^{d}\frac{\alpha_1}{\left(\sqrt{\hat{v}_1}\right)_j}\right] \leq d/c. \tag{58}$$

For the term with C_3*, we have*

$$E\left[\sum_{t=2}^{T-1}\left\|\frac{\alpha_t}{\sqrt{\hat{v}_t}}-\frac{\alpha_{t-1}}{\sqrt{\hat{v}_{t-1}}}\right\|^2\right] \tag{59}$$

$$\leq E\left[\frac{1}{c}\sum_{t=2}^{T-1}\left\|\frac{\alpha_t}{\sqrt{\hat{v}_t}}-\frac{\alpha_{t-1}}{\sqrt{\hat{v}_{t-1}}}\right\|_1\right] \tag{60}$$

$$\leq d/c^2, \tag{61}$$

where the first inequality is due to $\left|\left(\alpha_t/\sqrt{\hat{v}_t}-\alpha_{t-1}/\sqrt{\hat{v}_{t-1}}\right)_j\right| \leq 1/c$*. So we have for GAME,*

$$E\left[K_1\sum_{t=1}^{T}\left\|\alpha_t g_t/\sqrt{\hat{v}_t}\right\|^2 + K_2\sum_{t=2}^{T}\left\|\frac{\alpha_t}{\sqrt{\hat{v}_t}}-\frac{\alpha_{t-1}}{\sqrt{\hat{v}_{t-1}}}\right\|_1 + K_3\sum_{t=2}^{T-1}\left\|\frac{\alpha_t}{\sqrt{\hat{v}_t}}-\frac{\alpha_{t-1}}{\sqrt{\hat{v}_{t-1}}}\right\|^2\right] + K_4 \tag{62}$$

$$\leq K_1H^2/c^2(1+\log T) + K_2 d/c + K_3(d/c)^2 + K_4. \tag{63}$$

In terms of that $\hat{v}_t$ *is the moving average of* $\hat{g}_t$*, we have the following results*

$$\alpha/\left(\sqrt{\hat{v}_t}\right)_j \geq \frac{1}{H\sqrt{t}}, \tag{64}$$

and

$$E\left[\sum_{t=1}^{T}\alpha_i\left\langle\nabla f\left(x_t\right),\nabla f\left(x_t\right)/\sqrt{\hat{v}_t}\right\rangle\right] \geq E\left[\sum_{t=1}^{T}\frac{1}{H\sqrt{t}}\left\|\nabla f\left(x_t\right)\right\|^2\right] \geq \frac{\log(T+2)-\log 2}{H}\min_{t\in[T]}E\left[\left\|\nabla f\left(x_t\right)\right\|^2\right] \tag{65}$$

Therefore,

$$\frac{1}{H}\sqrt{T}\min_{t\in[T]} E\left[\|\nabla f\left(x_t\right)\|^2\right] \leq K_1 H^2/c^2(1+\log T) + K_2 d/c + K_3 d/c^2 + K_4. \tag{66}$$

The above inequality is equivalent to

$$\min_{t\in[T]} E\left[\|\nabla f\left(x_t\right)\|^2\right] \tag{67}$$

$$\leq \frac{H}{\log(T+2)-\log 2}\left(C_1 H^2/c^2(\frac{1}{2}-\frac{1}{T+1}) + C_2 d/c + C_3 d/c^2 + C_4\right) \tag{68}$$

$$= \frac{1}{\log(T+2)-\log 2}\left(P_1 + P_2\frac{T-1}{T+1}\right) \tag{69}$$

So we complete the proof.

References

[1] X. Chen, S. Liu, R. Sun, and M. Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. *arXiv preprint arXiv:1808.02941*, 2018.

[2] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.