STOCHASTIC GRADIENT METHODS WITH BLOCK DIAGONAL MATRIX ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Adaptive gradient approaches that automatically adjust the learning rate on a per-feature basis have been very popular for training deep networks. This rich class of algorithms includes ADAGRAD, RMSPROP, ADAM, and recent extensions. All these algorithms have adopted diagonal matrix adaptation, due to the prohibitive computational burden of manipulating full matrices in high-dimensions. In this paper, we show that block-diagonal matrix adaptation can be a practical and powerful solution that can effectively utilize structural characteristics of deep learning architectures to significantly improve convergence and out-of-sample generalization. We present ADABLOCK, a general framework for block-diagonal matrix adaption via coordinate grouping, which includes counterparts of the aforementioned algorithms. We prove its convergence in non-convex optimization and provide generalization error bounds, highlighting benefits compared to diagonal versions. In addition, we propose two techniques enriching the ADABLOCK family: i) an efficient spectrum-clipping scheme that benefits from superior generalization performance of SGD and ii) a randomized layer-wise block diagonal adaptation scheme to further reduce computational cost. Extensive experiments show that ADABLOCK achieves state-of-the-art results on several deep learning tasks, and can outperform adaptive diagonal methods, vanilla SGD, as well as a modified version of full-matrix adaptation proposed very recently.

1 INTRODUCTION

Stochastic gradient descent (SGD, Robbins & Monro (1951)) is a dominant approach for training large-scale machine learning models such as deep networks. At each iteration of this iterative method, the model parameters are updated in the opposite direction of the gradient of the objective function typically evaluated on a mini-batch, with step size controlled by a *learning rate*. While vanilla SGD uses a common learning rate across coordinates (possibly varying across time), several *adaptive learning rate* algorithms have been developed that scale the gradient coordinates by square roots of some form of average of the squared values of past gradients coordinates. The first key approach in this class, ADAGRAD (Duchi et al., 2011; McMahan & Streeter, 2010), uses a per-coordinate learning rate based on squared past gradients, and has been found to outperform vanilla SGD on sparse data. However, in non-convex dense settings where gradients are dense, performance is degraded, since the learning rate shrinks too rapidly due to the accumulation of all past squared gradient in its denominator. To address this issue, variants of ADAGRAD have been proposed that use the exponential moving average (EMA) of past squared gradients to essentially restrict the window of accumulated gradients to only few recent ones. Examples of such methods include ADADELTA (Zeiler, 2012), RMSPROP (Tieleman & Hinton, 2012), ADAM (Kingma & Ba, 2015), and NADAM (Dozat, 2016).

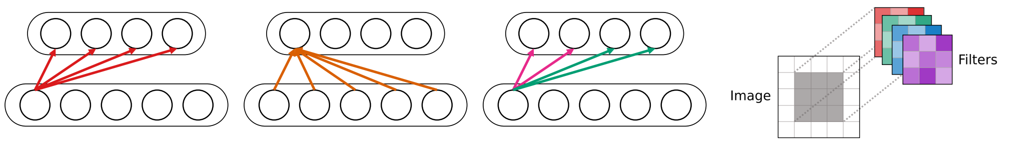
Despite their popularity and great success in some applications, the above EMA-based adaptive approaches have raised several concerns. Wilson et al. (2017) studied their out-of-sample generalization and observed that on several popular deep learning models their generalization is worse than vanilla SGD. Recently, Reddi et al. (2018) showed that they may not converge to the optimum (or critical point) even in simple convex settings with constant minibatch size, and noted that the effective learning rate of EMA methods can increase fairly quickly while for convergence it should decrease or at least have a controlled increase over iterations. AMSGRAD, proposed in Reddi et al. (2018) to fix this issue, did not yield conclusive improvements in terms of generalization ability. To simultaneously benefit from the generalization ability of vanilla SGD and the fast training of adaptive approaches, Luo et al. (2019) recently proposed ADABOUND and AMSBOUND as variants of ADAM and AMSGRAD, which employ dynamic bounds on learning rates to guard against extreme learning rates. Chen et al. (2019) introduced ADAFOM that only add momentum to the first moment estimate while using the same second moment estimate as ADAGRAD. Zaheer et al. (2018) showed that increasing minibatch sizes enables convergence of ADAM, and proposed YOGI which employs additive adaptive updates to prevent informative gradients from being forgotten too quickly. Yu et al. (2017) considers a variant of diagonal adaptation where, for each neural network layer, the gradients are normalized by the ℓ_2 norm of the layer's gradients.

We note that all the aforementioned adaptive algorithms deal with adaptation in a limited way, namely they only employ *diagonal* information of Gradient of Outer-Product ($g_t g_t^T$ where g_t is the stochastic gradient at time t, a.k.a. GOP). Though initially discussed in Duchi et al. (2011), *full* matrix adaptation has been mostly ignored due to its prohibitive computational overhead in high-dimensions. The only exception is the GGT algorithm Agarwal et al. (2019); it uses a modified version of full-matrix ADAGRAD with exponentially attenuated gradient history as in ADAM, but truncated to a small window parameter so the preconditioning matrix becomes low rank thereby computing its inverse square root effectively. Lafond et al. (2017) proposes a block diagonal structure in the context of natural gradients which always requires a probabilistic model.

Contributions. In this paper, we propose an extended form of SGD learning with *block-diagonal* matrix adaptation that can better utilize the structural characteristics of deep learning architectures. We also show that it can be a practical and powerful solution, which can actually outperform vanilla SGD and achieve state-of-the-art results on several deep learning tasks. More specifically, the main contributions of this paper are as follows:

- We provide an SGD framework with *block diagonal matrix adaptation* via *coordinate grouping*, ADABLOCK. This framework takes advantage of richer information on interactions across different gradient coordinates, while relaxing the expensive computational cost of full matrix adaptation in large-scale problems. In addition, we introduce several grouping strategies that are practically useful for deep learning problems.
- We provide the first convergence analysis of our framework in the non-convex setting, and highlight difference and benefits compared with diagonal versions. We investigate how the block sizes affect the convergence theoretically and empirically. Moreover, we provide insights on why ADABLOCK can improve generalization compared to usual diagonal approaches.
- We introduce *spectrum-clipping*, a non-trivial extension of Luo et al. (2019), to further boost the generalization ability of ADABLOCK. Spectrum-clipping allows the block diagonal matrix to become a constant multiple of the identity matrix in the latter part of training, similarly to vanilla SGD. In addition, we propose RADABLOCK, a randomized layer-wise ADABLOCK scheme, to further reduce computational cost.
- We evaluate the training and generalization ability of our approaches on popular deep learning tasks. Our experiments reveal that block diagonal methods perform better than diagonal approaches, even for small grouping sizes, and can also outperform vanilla SGD and the modified version of full-matrix adaptation GGT.

Notation. For a vector x, $\|x\|_p$ denotes the vector p-norm, and $\|x\|$ is $\|x\|_2$ if not specified. For a matrix A, $|||A|||_p$ indicates the matrix p-norm for matrix A, and $\lambda(A)$ returns a eigenvalue list (spectrum) of A. $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ denote the minimum and maximum eigenvalue of A respectively. The function $\text{Clip}(x, a, b)$ represents clipping x element-wise with the interval $I = [a, b]$. Lastly, $\log |A|$ denotes the log-determinant of a matrix A.



(a) input-neuron (b) output-neuron (c) partially group (d) filter-wise group

Figure 1: Examples of coordinate grouping. The weights with same color belong to the same group.

2 Adaptive Gradient Methods with Block Diagonal Matrix Adaptations via Coordinate Partitioning

In the context of stochastic optimization, Duchi et al. (2011) proposed a full-matrix variant of AdaGrad. This version employs a preconditioner which exploits first-order information only, via the sum of outer products of past gradients:

$$g_t = \nabla f(x_t), \quad G_t = G_{t-1} + g_t g_t^T, \quad x_{t+1} = x_t - \alpha_t (G_t^{1/2} + \delta I)^{-1} g_t \tag{1}$$

where g_t is a stochastic gradient at time t, α_t is a step-size, and δ is a small constant for numerical stability. Duchi et al. (2011) presented regret bounds for (1) in the convex setting. However, this approach is quite expensive due to $G_t^{1/2}$ term, so they proposed to only use the diagonal entries of G_t. Popular adaptive methods for training deep models such as RMSprop/Adam are based on such diagonal adaptation. Their general form and designs of the 2nd momentum are given in the appendix.

Duchi et al. (2011) also discussed the case where full-matrix adaptation can converge faster than its popular diagonal counterpart. Motivated by this, we first check through a toy MLP experiment whether preconditioning with exact GOP (1) can be more effective even in the deep learning context. Our experiment shows that one can achieve faster convergence and better objective values by considering the interaction between gradient coordinates (1). Details are provided in appendix due to space constraint. The caveat here is that using full GOP adaptation in real deep learning optimization problems is computationally intractable due to the square root operator in (1). Nevertheless, is the best choice to simply use diagonal approximation given the available computation budget? What if we can afford to pay a little bit more for our computations?

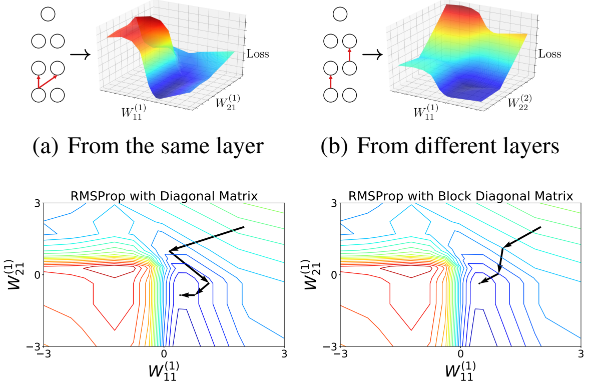
Main Algorithm: Adaptive SGD with Block Diagonal Adaptation (AdaBlock)

We address the above question and provide a family of adaptive Sgd bridging exact GOP adaptation and its diagonal approximation, via coordinate partitioning. Given a coordinate partition, we simply ignore the interactions of coordinates between different groups. For instance, given a gradient $g \in \mathbb{R}^6$, one example of constructing block diagonal matrices via coordinate partitioning is $g = (\underbrace{g_1, g_2}_{\mathcal{G}_1}, \underbrace{g_3, g_4, g_5}_{\mathcal{G}_2}, \underbrace{g_6}_{\mathcal{G}_3}) \to [g_{\mathcal{G}_1} g_{\mathcal{G}_1}^T \mid 0 \mid 0 \,;\, 0 \mid g_{\mathcal{G}_2} g_{\mathcal{G}_2}^T \mid 0 \,;\, 0 \mid 0 \mid g_{\mathcal{G}_3} g_{\mathcal{G}_3}^T]$ where $\mathcal{G}_i$ represents each group and $g_{\mathcal{G}_i}$ denotes the collection of entries corresponding to group $\mathcal{G}_i$. Both exact GOP and diagonal approximation are special cases of our family. Exploring the use of block-diagonal matrices was suggested as future work in Duchi et al. (2011), and our work therefore provides an in-depth study of this proposal. Our main algorithm, Algorithm 1, formalizes our approach for a total r groups where each group $\mathcal{G}_i$ has a size of n_i for $i \in [r]$. The Algorithm 1 can handle arbitrary coordinate grouping with appropriate reordering of entries, and groups of unequal sizes.

Algorithm 1 AdaBlock: **Ada**ptive Gradient Methods with **Block** Diagonal Matrix Adaptation

```
Input: Stepsize $\alpha_t$, initial point $x_1 \in \mathbb{R}^d$, and $\{\beta_{1,t}\}_{t=1}^T \in [0,1)$. The function $H_t$ designs $\widehat{V}_t$ with dynamic size of $r$ blocks, $\{\widehat{V}_{t,[j]}\}_{j=1}^r$.
Initialize: $m_0 = 0$, $\widehat{V}_0 = 0$.
Require: Coordinate partition $\mathcal{P}$.
for $t = 1, 2, \ldots, T$ do
    Draw a minibatch sample $\xi_t$ from $\mathbb{P}$
    $g_t \leftarrow \nabla f(x_t)$
    $m_t \leftarrow \beta_{1,t} m_{t-1} + (1 - \beta_{1,t}) g_t$
    for $j = 1, 2, \ldots, r$ do
        $\widehat{V}_{t,[j]} \leftarrow H_t(g_{1,[j]}, \cdots, g_{t,[j]};\ \mathcal{P})$
    end for
    $x_{t+1} \leftarrow x_t - \alpha_t (\widehat{V}_t^{1/2} + \delta I)^{-1} m_t$
end for
```

Effect of grouping on optimization. Figure 1 shows some grouping examples in the context of deep learning models: grouping the weights with the same color in a network can approximate the exact GOP matrix with a block diagonal matrix of several small full matrices. To see which grouping could be more effective in terms of optimization, we revisit our MLP toy example. Figure 2-(a,b) show the loss landscapes for different grouping strategies (weights other than shown are fixed as true model values). We can see that the loss landscape when grouping weights in the same layer has a much more dynamic curvature than when grouping weights in different layers. In this context, we expect that a block-diagonal preconditioner is effective in terms of optimization and illustrate this empirically by comparing the grouping version for the loss landscape with dynamic curvature (Figure 2-(a)), and its diagonal counterpart. To figure out the effect of the block-diagonal structure only, we compare both approaches using RMSPROP which does not consider the 1st-order momentum. Figure 2-(c,d) show the optimization histories. The block diagonal extension of RMSPROP converges to a stationary point in fewer steps than usual RMSPROP and shows a more stable trajectory.



(a) From the same layer (b) From different layers

(c) RMSPROP-Diag (d) RMSPROP-Group

Figure 2: Top: Loss surface for different groupings; Bottom: optimization histories on the loss surface (a).

3 ANALYSIS ON BLOCK DIAGONAL MATRIX ADAPTATIONS

3.1 CONVERGENCE IN NON-CONVEX OPTIMIZATION

In this section, we provide a theoretical analysis on the convergence of Algorithm 1. Towards this, we consider the form of non-convex optimization problem, $\min . f(x) := \mathbb{E}_{\xi\sim\mathbb{P}}\big[f(x;\xi)\big]$ where x is an optimization variable and ξ is a random variable representing randomly selected data sample from training data S. As in other works for non-convex optimization such as (Ghadimi & Lan, 2013; 2016), we study the convergence to “stationarity” and hence derive upper bounds of $\|\nabla f(x)\|^2$ by Algorithm 1, under the following mild conditions:

Assumption 1. ***(a)*** *f is differentiable, L-smooth, and lower bounded.* ***(b)*** *At time t, the algorithm can access a noisy gradient. We assume the true gradient $\nabla f(x_t)$ and noisy gradient g_t are both bounded, i.e. $\|\nabla f(x_t)\|_2, \|g_t\|_2 \le G$ for all t.* ***(c)*** *The noisy gradient g_t is unbiased and the noise is independent, i.e. $g_t = \nabla f(x_t) + \zeta_t$ where $\mathbb{E}[\zeta_t] = 0$ and $\zeta_t \perp\!\!\!\perp \zeta_s$ for $t \neq s$.* ***(d)*** *The sequence of $\beta_{1,t} \in [0,1), t \in [T]$ in Algorithm 1 is non-increasing.* ***(e)*** *$\|\alpha_t \widehat{V}_t^{-1/2} m_t\|_2 \le D$ for some $D > 0$.*

Note that the condition (a) is a key assumption in general non-convex optimization analysis, and (b)-(d) are standard ones in the line of work on stochastic gradient based solvers such as Chen et al. (2019). The last condition (e) states that the final step vector $\alpha_t \widehat{V}_t^{-1/2} m_t$ should be finite, which is also a mild condition. We are now ready to state our first theorem.

Theorem 1. *Let $Q_t := \|\alpha_{t-1}\widehat{V}_{t-1}^{-1/2} - \alpha_t \widehat{V}_t^{-1/2}\|_2 = \max_{j\in[r]}\{\|\alpha_{t-1}\widehat{V}_{t-1,[j]}^{-1/2} - \alpha_t \widehat{V}_{t,[j]}^{-1/2}\|_2\}$ for Algorithm 1 which measures the maximum difference in effective spectrums over all diagonal blocks $\widehat{V}_{t,[j]}$ and $\gamma_t := \lambda_{\min}(\alpha_t \widehat{V}_t^{-1/2})$. Then, under Assumption 1, Algorithm 1 is guaranteed to yield*

$$\min_{t\in[T]} \big[\|\nabla f(x_t)\|^2\big] \le \frac{\mathbb{E}\Bigg[C_1 \overbrace{\sum_{t=1}^{T} \Big\|\alpha_t \widehat{V}_t^{-1/2} g_t\Big\|^2}^{\textit{Term A}} + C_2 \overbrace{\sum_{t=2}^{T} Q_t}^{\textit{Term B}} + C_3 \sum_{t=2}^{T-1} Q_t^2\Bigg] + C_4}{\sum_{t=1}^{T} \gamma_t} \triangleq \frac{s_1(T)}{s_2(T)} \tag{2}$$

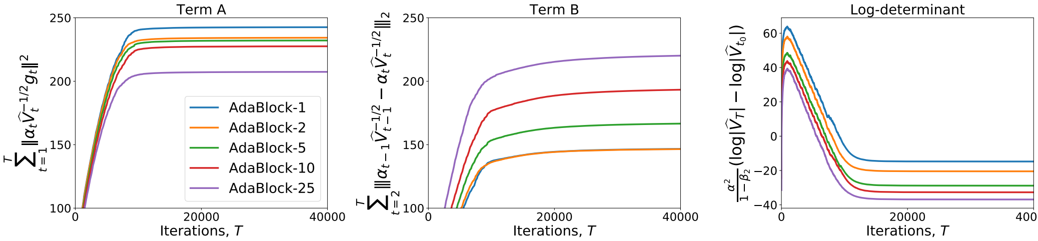


Figure 3: Empirical studies with block diagonal extension of ADAM with stepsize $\alpha_t = 10^{-3}$.

where C_1, C_2, and C_3 are constants independent of problem dimension d and the number of iterations T, and C_4 is a constant independent of T.

Note that while Theorem 1 is generally applicable to any Adam-type block diagonal matrix adaptations, the effect of block size b is implicitly represented in (2). As we will see below, the terms here can be further simplified for some cases depending on algorithmic details.

- **Sufficient Condition for Convergence.** From Theorem 1, we can see that $s_1(T) = o\big(s_2(T)\big)$ provides a sufficient condition for convergence. For example, ADAGRAD with $\alpha_t = \alpha/\sqrt{t}$ satisfies $s_1(T) = \mathcal{O}(\log|\widehat{V}_T| + \log T + 1)$ and $s_2(T) = \Omega(\sqrt{T})$, so $s_1(T) = o(s_2(T))$. In contrast, RMSPROP/ADAM with $\alpha_t = \alpha$ satisfies $s_1(T) = \mathcal{O}(\log|\widehat{V}_T| + T + 1)$ and $s_2(T) = \Omega(\sqrt{T})$, so $s_1(T) \neq o(s_2(T))$ (see Appendix for details).

- **Comparison to Prior Analysis.** The special case of (2) for a diagonal case ($b = 1$) provides the convergence bound of standard Adam-type diagonal adaptations which was previously studied in (Chen et al., 2019). However, our bound is *strictly tighter* than that of (Chen et al., 2019). Specifically, the Term B in (Chen et al., 2019) involves $\|\alpha_{t-1}/\sqrt{\widehat{v}_{t-1}} - \alpha_t/\sqrt{\widehat{v}_t}\|_1$ while ours is $\|\alpha_{t-1}/\sqrt{\widehat{v}_{t-1}} - \alpha_t/\sqrt{\widehat{v}_t}\|_\infty$.

- **Convergence of AdaGrad.** We can instantiate our theorem for block diagonal extensions of ADAGRAD/ADAFOM (Chen et al., 2019; Zou & Shen, 2018) where the benefit of using block diagonal adaptation is explicit:

Corollary 1. (ADAGRAD/ADAFOM) *Consider block diagonal version of* ADAGRAD/ADAFOM *with $\alpha_t = \alpha/\sqrt{t}$ under Assumption 1 (in case of* ADAGRAD, *$\beta_1 = 0$). Then, they achieve $s_1(T) = \mathcal{O}(\underbrace{\log|\widehat{V}_T| + \log T}_{\text{From Term A}} + \underbrace{1}_{\text{From Term B and others}})$ and $s_2(T) = \Omega(\sqrt{T})$, hence we have $\min_{t\in[T]} \mathbb{E}\big[\|\nabla f(x_t)\|^2\big] = \mathcal{O}(\log|\widehat{V}_T|/\sqrt{T} + \log T/\sqrt{T} + 1/\sqrt{T})$. Moreover, for any coordinate partition it is guaranteed that $\log|\widehat{V}_T|$ decreases as a block size b increases.*

- **Convergence of EMA-based Algorithms.** Now, we consider popular EMA-based algorithms such as RMSPROP/ADAM, i.e., the design function H_t in Algorithm 1 constructs $\widehat{V}_t$ as $\widehat{V}_t = \beta_2\widehat{V}_{t-1} + (1-\beta_2)g_t g_t^T$ with $\beta_2 \in [0, 1)$. For the convergence guarantee of this family, we need the following matrix $\Lambda_t := \alpha_{t-1}\widehat{V}_{t-1}^{-1/2} - \alpha_t\widehat{V}_t^{-1/2}$ should be *positive semidefinite* which is a *generalized version* of $\Gamma_t := \alpha_{t-1}/\sqrt{v_{t-1}} - \alpha_t/\sqrt{v_t} \geq 0$ in previous analysis (Reddi et al., 2018; Chen et al., 2019). From Proposition 1 in Appendix E, we can expect that the Term A/Term B of RMSPROP/ADAM also have similar dynamics as those of ADAGRAD. For empirical studies, we design a simple experiment with MLP 784-100-10 on MNIST dataset. We optimize the network parameters via block diagonal extension of ADAM with constant $\beta_{1,t} = 0.9$ and $\beta_2 = 0.999$. Figure 3 illustrates the Term A/Term B/Log-determinant for stepsize $\alpha_t = 10^{-3}$. In Figure 3, both the Term A and $\log|\widehat{V}_T|$ decreases as a block size b increases, which corroborates Proposition 1.

3.2 UNIFORM STABILITY AND GENERALIZATION ERROR BOUNDS OF ALGORITHM 1

The generalization error of a randomized algorithm A (e.g., SGD) on training data S is defined as $\epsilon_{\text{gen}} := \mathbb{E}_{S,A}\big[R_S(A(S)) - R(A(S))\big]$ where R_S and R denote empirical risk and population risk respectively. Hardt et al. (2015) show that an ϵ-uniformly stable algorithm satisfies $|\epsilon_{\text{gen}}| \le \epsilon$ where ϵ-uniform stability is defined as follows:

Definition 1. *(Hardt et al., 2015) Let $S, S' \in \mathcal{Z}^n$ be two datasets of size n that differ in only one example. The Algorithm A is said to be ϵ-uniformly stable if* $\sup_{z\in\mathcal{D}} \mathbb{E}_A\big[f(A(S);z) - f(A(S');z)\big] \le \epsilon$.

In order to bound the generalization error using the result of (Hardt et al., 2015), it would suffice under a Lipschitz continuity as in our Assumption 1-(b) to show that $\mathbb{E}_A\big[\|\theta - \theta'\|_2\big]$ is bounded since $\sup_{z\in\mathcal{D}} \mathbb{E}_A\big[f(A(S);z) - f(A(S');z)\big] \le G\mathbb{E}_A\big[\|\theta-\theta'\|_2\big]$. Here, we consider an EMA-based design function H_t and defer the result of AdaGrad to Appendix.

Theorem 2. (EMA-BASED) *For $\alpha_t = \alpha$ and $\beta_{1,t} = 0$, we have the following recurrence relation,*

$$\mathbb{E}\big[\Delta_{T+1}\big] \le \frac{\alpha\sqrt{T}}{n\sqrt{1-\beta_2}}\Big[\sqrt{g(\widehat{V}_T)} + \sqrt{g(\widehat{V}'_T)}\Big] + \alpha\Big(1 - \frac{1}{n}\Big)J_T$$

where t_0 denotes the time when $\widehat{V}_t$ and $\widehat{V}'_t$ becomes full-rank with the quantities

$$g(\widehat{V}_T) := \tfrac{t_0(1-\beta_2)G^2}{\delta^2} + \mathbb{E}[\underbrace{\log|\widehat{V}_T| - \log|\widehat{V}_{t_0}|}_{\textit{Term C}}] + d(T-t_0)\log\tfrac{1}{\beta_2}, \textit{ and}$$

$$J_T := G\textstyle\sum_{t=1}^T \mathbb{E}[\underbrace{\|(\widehat{V}_t^{1/2}+\delta I)^{-1}\|_2}_{\textit{Term D}} + \underbrace{\|(\widehat{V}_t'^{1/2}+\delta I)^{-1}\|_2}_{\textit{Term D}}] + L\textstyle\sum_{t=1}^T \mathbb{E}[\underbrace{\|(\widehat{V}_t'^{1/2}+\delta I)^{-1}\|_2}_{\textit{Term D}}\Delta_t].$$

In the quantities $g(\widehat{V}_T)$ and J_T, we remark the Term C/Term D since only those terms depend on the block sizes. Therefore, we investigate the dynamics of Term C/Term D as we will see below.

- **Dynamics of Term C/Term D.** In Theorem 2, the Term C is smallest when $b = d$ as in Corollary 1 while the Term D is smallest for $b = 1$ since $\max_i A_{ii} \le \lambda_{\max}(A)$ for any matrix $A \in \mathcal{S}_{++}$. By the way, the Term D can be bounded as $\|(\widehat{V}_t'^{1/2} + \delta I)^{-1}\|_2 \le 1/\delta$ which is independent of T. For empirical studies, we revisit our experiment with MLP 784-100-10 on MNIST dataset. The Figure 4 shows that $\alpha\|(\widehat{V}_t'^{1/2} + \delta I)^{-1}\|_2$ converges to α/δ (Here, $\alpha/\delta = 10^{-3}/10^{-4} = 10$) regardless of block sizes and the difference among block sizes is negligible compared to $\log|\widehat{V}_T|$ (see Figure 3). As a result, the Term C is dominant, so we can expect that $\mathbb{E}[\Delta_t]$ grows slower for $b > 1$ than $b = 1$.

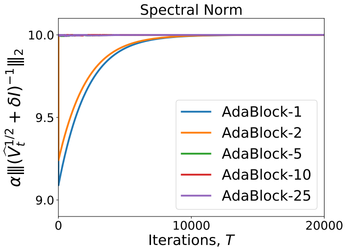


Figure 4: Term D

- **Large δ Improves Generalization.** Zaheer et al. (2018) suggest that one should use large δ such as 10^{-3} to improve generalization but with only empirical studies. In Theorem 2, it can be seen clearly that the growth rate of Δ_t is slower for large δ, which in result improves generalization.

4 Improving Generalization and Computational Cost

Interpolating SGD via Spectrum-Clipping. It has been shown in Wilson et al. (2017) that adaptive methods are better than vanilla SGD in the early stage but get worse as the learning process matures. To address this, Keskar & Socher (2017) suggest training networks with Adam at the beginning and switching to SGD later. Luo et al. (2019) propose methods AdaBound/AMSBound which clip the effective learning rate $\alpha_t/(\sqrt{v_t} + \epsilon)$ of Adam by decreasing sequence of intervals $I_t = [\eta_l(t), \eta_u(t)]$ every iteration which converges to some point, thereby resembling SGD in the end. However, this type of extension is not obvious in our framework due to the absence of *effective* learning rate in our case. Instead, we observe that the *spectral property* is important in our analysis (In Theorem 1 and 2: both convergence and generalization depend on $\log|\widehat{V}_T|$). Motivated on them, we propose a *spectrum-clipping* scheme which clips the spectrum of $\alpha_t(\widehat{V}_t^{1/2} + \delta I)^{-1}$ by decreasing sequence of intervals. For spectrum-clipping, we use the following modified update rule in Algorithm 1 after constructing $\widehat{V}_t$: **(i)** $\widehat{U}_t, \widehat{\Sigma}_t^{1/2}, \_ \leftarrow \text{SVD}(\widehat{V}_t^{1/2})$, **(ii)** $\widetilde{\Sigma}_t^{-1/2} \leftarrow \text{Clip}(\lambda(\alpha_t(\widehat{\Sigma}_t^{1/2} + \delta I)^{-1}), \lambda_l(t), \lambda_u(t))$, and **(iii)** $x_{t+1} \leftarrow x_t - \widehat{U}_t^T\widetilde{\Sigma}_t^{-1/2}\widehat{U}_t m_t$. We schedule the sizes of clipping intervals converging to a single point uniformly over the spectrum so that $\alpha_t(\widehat{V}_t^{1/2} + \delta I)^{-1}$ can be easily computed in the form of constant times identity matrix and effectively behaves like vanilla SGD.

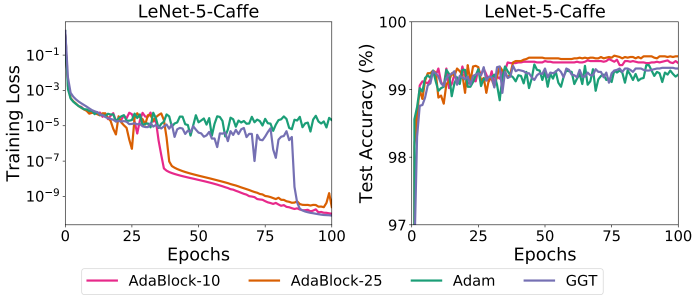
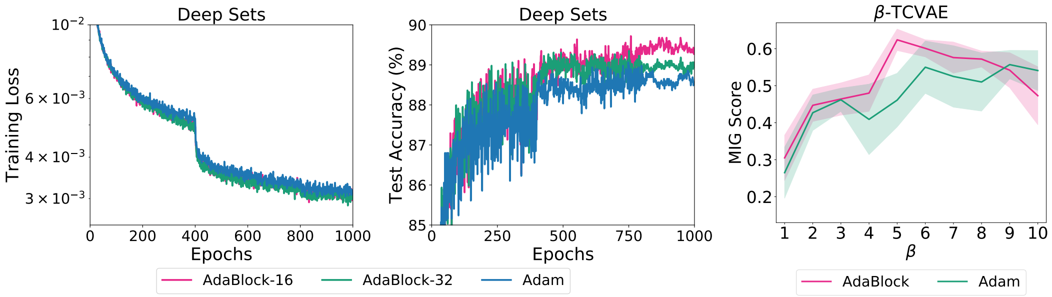


Figure 5: Results on MNIST classification.



(a) Point Cloud Classification on DeepSets

(b) MIG score on β-TCVAE

Figure 6: Results on point cloud classification and β-TCVAE.

Randomized Layer-wise AdaBlock (RAdaBlock). To further reduce the computational cost, we propose a randomized update scheme. At each iteration, we select l layers at random to be updated via AdaBlock, while the remaining layers are updated via the usual diagonal approximation. By bridging block-diagonal and diagonal adaptation, we wish to combine the advantages of both approaches: fast per-iteration time of diagonal adaptations and faster convergence/better generalization of AdaBlock. Additional considerations on computations and memory are provided in Appendix.

5 Experiments

We consider three sets of experiments. The first shows the differences between block-diagonal and diagonal versions. The second investigates whether block diagonal matrix adaptation can achieve state-of-the-art performance on benchmark architecture/dataset for various important deep learning problems. The third evaluates RadaBlock, which aims at further reducing computational cost. For the first set, we do not consider the spectrum-clipping or randomized update in Section 4 to clearly assess the effect of coordinate grouping. In Algorithm 1, coordinate grouping can be done in a number of ways. Given our insight that grouping weights in the same layer could be more effective, we consider Figure 1-(c) with grouping 10 or 25 weight parameters connected to input-neuron for dense layer and filter-wise grouping for convolutional layers in Figure 1-(d). In all our experiments, we use block diagonal version of Adam. Details on settings/hyperparameters are provided in Appendix C.

Investigating Grouping Effect. We investigate the effect of coordinate grouping on **(i)** MNIST classification, **(ii)** Point cloud classification on DeepSets, and **(iii)** Deep variational autoencoder.

MNIST Classification. We consider a simple LeNet-5 network. We use 128 mini-batch size and train networks with 100 epochs. As Figure 5-(a) illustrates the results, the learning curve looks similar in the early stage of training, but AdaBlock converges without oscillations in the latter part of training, which corroborates the effect of block sizes in Theorem 1 and 2. The generalization of AdaBlock also becomes more stable than diagonal variant and GGT, and overall superior across epochs.

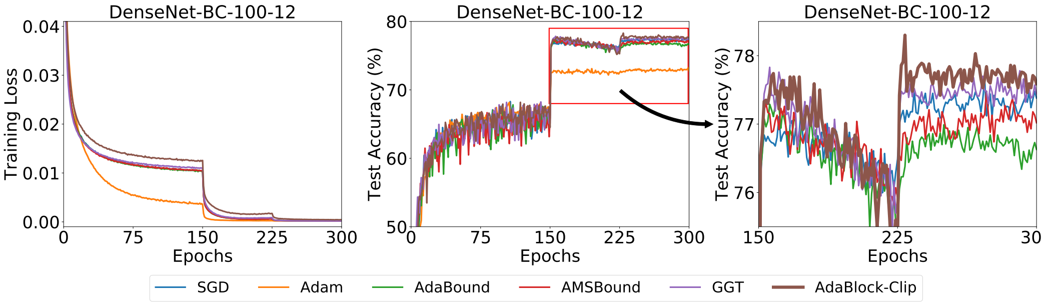
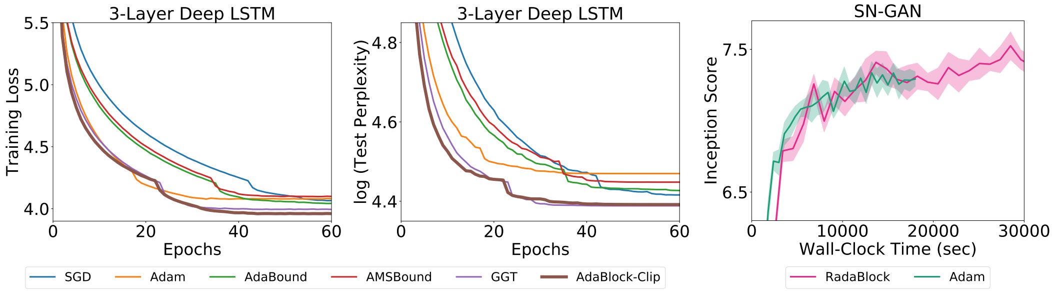


Figure 7: Results on CIFAR classification.



(a) Training loss/test perplexity on 3-Layer deep LSTM

(b) Inception score on SN-GANs

Figure 8: Results on language models and generative adversarial nets.

Point Cloud Classification. We evaluate AdaBlock on classifying point-cloud representation of a subset of ShapeNet objects (Chang et al., 2015), called ModelNet40 (Wu et al., 2015). This dataset consists of 40 classes of 3-dimensional objects. Each object is represented as a point cloud which we treat as a set of n particles in $\mathbb{R}^3$. For this task, we employ DeepSets (Zaheer et al., 2017) architecture and follow their settings with $n = 1000$. Figure 6-(a) shows that the learning curve has a similar behavior, but AdaBlock outperforms Adam for any block size in terms of generalization.

Deep Variational Autoencoder. We conduct experiments on a very recent variant of VAE called β-TCVAE (Chen et al., 2018). The goal of this model is to make the encoder $q(z|x)$ give disentangled representation z of input images x by additionally forcing $q(z) = \int q(z|x)p(x)dx$ to be factorized, which can be achieved by giving heavier penalty on total correlation. We evaluate our optimizer with the Mutual Information Gap (MIG) score they proposed, to measure disentanglement of the latent code. Following implementation in (Chen et al., 2018), we use convolutional encoder-decoder for β-TCVAE on 3D faces dataset (Paysan et al., 2009). Figure 6-(b) illustrates the results over 10 random simulations with 95% confidence intervals. AdaBlock outperforms diagonal version with a smaller variance except at $\beta \in \{9, 10\}$, and we can achieve the best performance at $\beta = 5$.

Improving Performance with Spectrum-Clipping. We demonstrate the superiority of our algorithms using more complex benchmark architecture/dataset for two popular tasks in deep learning: image classification and language modeling. For both tasks, vanilla Sgd with proper learning rate scheduling has enjoyed state-of-the-art performance. Therefore, we compare algorithms using our spectrum-clipping methods that can exploit higher generalization ability of vanilla Sgd.

CIFAR Classification. We conduct experiments using DenseNet architecture (Huang et al., 2017). Figure 7 illustrates our results on CIFAR-100 dataset, and the figure for CIFAR-10 is in appendix. In both cases, the training speed of AdaBlock at the early stage is similar or slightly slower, but we can arrive at the *state-of-the-art* generalization performance in the end among all comparison algorithms. Specifically, we can achieve great improvement in generalization about 0.5%.

Language Models. We use recurrent networks (Zaremba et al., 2014), base architectures still frequently used today for language modeling. While (Zaremba et al., 2014) uses only two layers maximum, we add one more layer to consider more complex and deeper networks. To consider similar model capacity as (Zaremba et al., 2014), we use 500 hidden units on each layer. Based on this architecture, we build a word-level language model using 3-layer LSTM (Hochreiter & Schmidhuber, 1997) on Penn TreeBank (PTB) dataset (Marcus et al., 1994). Figure 8-(a) shows the experimental results: the optimizer with spectrum-clipping of ADABLOCK outperforms all the other algorithms w.r.t. learning curve. It achieves similar perplexity as GGT and outperforms the other methods.

Reducing Computational Cost with RADABLOCK. We consider generative adversarial nets (a.k.a. GANs). Since it is well-known that training GANs generally takes a lot of time, it is reasonable to evaluate RADABLOCK on this task. We choose recently proposed spectral normalization GANs (a.k.a. SN-GANs) which control the Lipschitz constant of the discriminator (Miyato et al., 2018), thereby stabilizing the training procedure. SN-GANs are still generally used as baselines, so we use a CIFAR-10 dataset on standard CNN architecture and the inception score (Salimans et al., 2016) for quantitative assessment. For RADABLOCK, at each iteration, we update two layers via ADABLOCK chosen randomly, one for the generator and one for the discriminator. Figure 8-(b) depicts inception score vs. wall-clock time for each method. RADABLOCK achieves higher inception score in wall-clock time. We conjecture that a block diagonal approximation has a regularization effect and leave this as an open question for future work.

6 CONCLUDING REMARKS

We proposed ADABLOCK, a general adaptive gradient framework that approximates exact GOP with block diagonal matrices via coordinate grouping, to effectively utilize structural characteristics of deep learning architectures. We analyzed convergence and generalization for our approach, highlighting benefits compared to its popular diagonal counterpart, and confirmed our findings theoretically and empirically. We also proposed a spectrum-clipping extension which achieved state-of-the-art generalization performance on popular deep learning tasks and a randomized approach to further reduce computational cost. As future work, we plan to explore strategies for (i) setting the clipping parameters in spectrum-clipping, to strike the best balance between training speed and generalization ability, and (ii) selecting the layers that would benefit most from block-diagonal adaptation at each iteration in RADABLOCK.

REFERENCES

Naman Agarwal, Brian Bullins, Xinyi Chen, Elad Hazan, Karan Singh, Cyril Zhang, and Yi Zhang. Efficient full-matrix adaptive regularization. In *International Conference on Machine Learning*, pp. 102–110, 2019.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pp. 2610–2620, 2018.

Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. In *International Conference on Learning Representation (ICLR)*, 2019.

Timothy Dozat. Incorporating nesterov momentum into adam. *ICLR Workshop*, 2016.

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

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Saeed Ghadimi and Guanghui Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, 2016.

Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240*, 2015.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

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

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

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*, pp. 4700–4708, 2017.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representation (ICLR)*, 2015.

Jean Lafond, Nicolas Vasilache, and Léon Bottou. Diagonal rescaling for neural networks. *arXiv preprint arXiv:1705.09319*, 2017.

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

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The penn treebank: annotating predicate argument structure. In *Proceedings of the workshop on Human Language Technology*, pp. 114–119. Association for Computational Linguistics, 1994.

H Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. In *Conference on Computational Learning Theory (COLT)*, 2010.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=B1QRgziT-`.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 296–301. Ieee, 2009.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representation (ICLR)*, 2018.

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

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pp. 2234–2242, 2016.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

Ashia C. Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Adams Wei Yu, Lei Huang, Qihang Lin, Ruslan Salakhutdinov, and Jaime Carbonell. Block-normalized gradient method: An empirical study for training deep neural network. *arXiv preprint arXiv:1707.04822*, 2017.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pp. 3391–3401, 2017.

Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pp. 9793–9803, 2018.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Shuai Zheng and James T Kwok. Blockwise adaptivity: Faster training and better generalization in deep learning. *arXiv preprint arXiv:1905.09899*, 2019.

Fangyu Zou and Li Shen. On the convergence of adagrad with momentum for training deep neural networks. *arXiv preprint arXiv:1808.03408*, 2018.

Appendix

A Toy MLP example: Full GOP adaptation vs. Diagonal approximation

We consider a structured MLP (two nodes in two hidden layers followed by single output). For hidden units, we use ReLU activation (Nair & Hinton, 2010) and the sigmoid unit for the binary output. We generate $n = 10$ i.i.d. observations: $x_i \sim \mathcal{N}(0, I_2)$ and y_i from this two layered MLP given x_i. The results of our toy experiment are depicted in Figure 9.

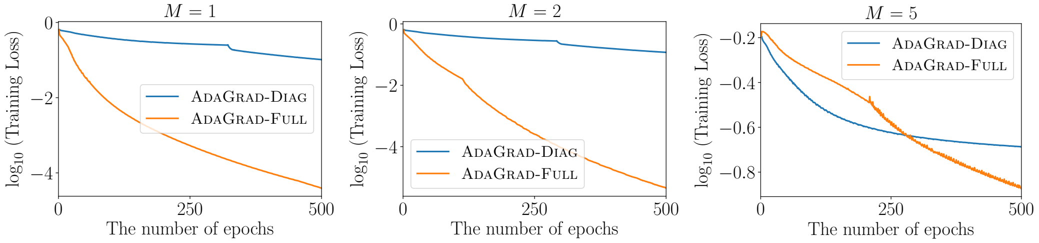


Figure 9: Comparison of AdaGrad diagonal version and full matrix version varying the minibatch size M.

B Computations and memory considerations

Table 1: Test error (%) for CIFAR dataset.

	SGD	Adam	AdaBound	AmsBound	GGT	AdaBlock-Clip
CIFAR-10	4.51	6.07	4.78	4.77	6.33	**4.34**
CIFAR-100	22.27	26.51	22.5	22.52	22.17	**21.7**

Table 2: Average time (sec) per iteration for CIFAR-100 experiments.

SGD	Adam	AdaBound	AmsBound	GGT	AdaBlock	AdaBlock-Clip
0.080	0.108	0.126	0.130	0.166	0.226	0.251

Compared with full matrix adaptation, working with a block diagonal matrix is computationally more efficient as it allows for decoupling computations with respect to each small full sub-matrix. In Algorithm 1, the procedures for constructing the block diagonal matrix and for updating parameters for each block by computing the “inverse” square root of each sub-matrix can be done in a parallel manner. As the group size increases, the block diagonal matrix becomes closer to the full matrix, resulting in greater computational cost. Therefore, we consider small group size for our numerical experiments. For instance, on CIFAR-100 dataset, the average time per iteration of AdaBlock methods is at most twice that of their diagonal counterparts, but this comes with a great advantage since AdaBlock shows significant improvement in generalization. In comparison the average time per iteration of GGT Agarwal et al. (2019) (the modified version of full-matrix AdaGrad) is 1.5 times that of Adam but the generalization of GGT is worse (see Table 1 and 2). Moreover, the wall-clock time performance of AdaBlock can be improved using RAdaBlock. As can be seen in Figure 8, RAdaBlock achieves higher inception score in wall-clock time.

In terms of memory, our method is more efficient than GGT Agarwal et al. (2019) (the modified version of full-matrix ADAGRAD). For example, consider models with a total of d parameters. For Algorithm 1, assume that $\widehat{V}_t$ is a block diagonal matrix with r sub-matrices, and each block has size $b \times b$ (so, $br = d$). Also, assume that the truncated window size for GGT is w. GGT needs a memory size of $\mathcal{O}(wd)$, and our algorithm requires $\mathcal{O}(rb^2) = \mathcal{O}(bd)$. We consider small group size $b = 10$ or 25 for our experiments while the recommended window size of GGT is 200 (Agarwal et al., 2019). Therefore, our algorithm is more memory-efficient and the benefit is more pronounced as the number of model parameters d is large, which is the case in popular deep learning models/architectures.

C Hyperparameters and Additional Experimental Results

We use the recommended step size or tune it in the range $[10^{-4}, 10^{2}]$ for all comparison algorithms. For ADAM based algorithms, we use default decay parameters $(\beta_1, \beta_2) = (0.9, 0.999)$. For a diagonal version of ADAM variant algorithm, we choose numerical stability parameter $\epsilon = 10^{-3}$ since the larger value of ϵ can improve the generalization performance as discussed in (Zaheer et al., 2018). For spectrum-clipping in Section 4, we use the same intervals $\lambda_l(t) = \alpha^*(1 - \frac{1}{(1-\beta_2)t+1})$ and $\lambda_u(t) = \alpha^*(1 + \frac{1}{(1-\beta_2)t})$ as in Luo et al. (2019). For γ and α^* in clipping bound functions, we consider $\gamma \in \{0.0001, 0.0005, 0.001\}$ and choose $\alpha^* \in \{\alpha_{\text{SGD}}, 5\alpha_{\text{SGD}}, 10\alpha_{\text{SGD}}\}$ where α_{SGD} is the best-performing initial learning rate for vanilla SGD (These hyperparameter candidates are based on the empirical studies in Luo et al. (2019)). As in Luo et al. (2019), our results are also not sensitive to choice of γ and α^*. With these hyperparameters, we consider maximum 300 epochs training time, and mini-batch size or learning rate scheduling are introduced in each experiment description. Our Algorithm 2 requires SVD procedures to compute the square root of a block diagonal matrix. We apply SVD efficiently to all small sub-matrices simultaneously through batch mode of SVD.

MNIST Classification. We consider the following LeNet-5 network architecture, 20C5 - MP2 - 50C5 - MP2 - 500FC - softmax. Designing for corroborating our theoretical results, we employ the numerical stability parameter $\delta = \epsilon = 10^{-4}$.

Deep Sets. For point cloud classification, we follow the same settings in Zaheer et al. (2017) and use the same architecture in Appendix H of Zaheer et al. (2017). Also, we use the author's implementation only replacing the optimizer with ADABLOCK. As they use $\epsilon = 10^{-3}$ for ADAM, we also employ the same value as $\delta = 10^{-3}$ in Algorithm 1 for fair comparison.

β-TCVAE. For experiments on generative models β-TCVAE, we use the author's implementation only replacing the ADAM optimizer with our BLOCK-ADAM. We use convolutional networks for encoder-decoder they consider in Chen et al. (2018) and mini-batch size 2048. For generative models, the small ϵ value for ADAM works better than large value in our experiences, so we employ $\delta = \epsilon = 10^{-8}$.

CIFAR classification. According to experiment settings in (Huang et al., 2017), we use mini-batch size 64 and consider maximum 300 epochs. Also, we use a *step-decay* learning rate scheduling in which the learning rate is divided by 10 at 50% and 75% of the total number of training epochs. With this setting, vanilla SGD with a momentum factor 0.9 performs best with initial learning rate $\alpha^* = 0.1$, so we use this value for our bound functions of spectrum-clipping, $\lambda_l(t)$ and $\lambda_u(t)$. As recommended in Zaheer et al. (2018), we employ $\delta = \epsilon = 10^{-3}$ for better generalization.

Language Models. In this experiment, we use a *dev-decay* learning rate scheduling Wilson et al. (2017) where we reduce learning rate by a constant factor if the model does not attain a new best validation performance at each epoch as in Zaremba et al. (2014). Under this setting, vanilla SGD performs best when the initial learning rate $\alpha^* = 5$. As recommended in Zaheer et al. (2018), we employ $\delta = \epsilon = 10^{-3}$ for better generalization.

SN-GANs. Since the ADAM is a classical optimizer (than vanilla SGD) for training GANs with tuned decaying parameters (β_1, β_2). As parameter configuration in Miyato et al. (2018) (Table 1 in Section 4), we set $(\beta_1, \beta_2) = (0.5, 0.999)$ with stepsize $\alpha = 0.002$. The number of updates on discriminator n_{dis} per single update on generator is set to 1. For RADABLOCK, we update two layers via ADABLOCK randomly at each iteration, one for the generator and one for the discriminator. As noted in β-TCVAE, the small ϵ works better in our experiences, so we employ $\delta = \epsilon = 10^{-8}$.

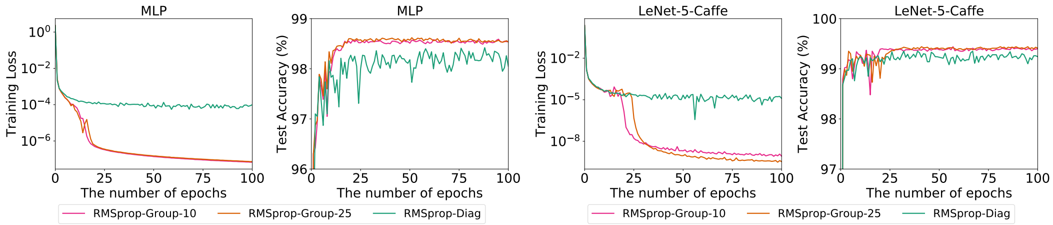


Figure 10: Training loss/test accuracy for MLP/LeNet-5-Caffe with block diagonal RMSPROP.

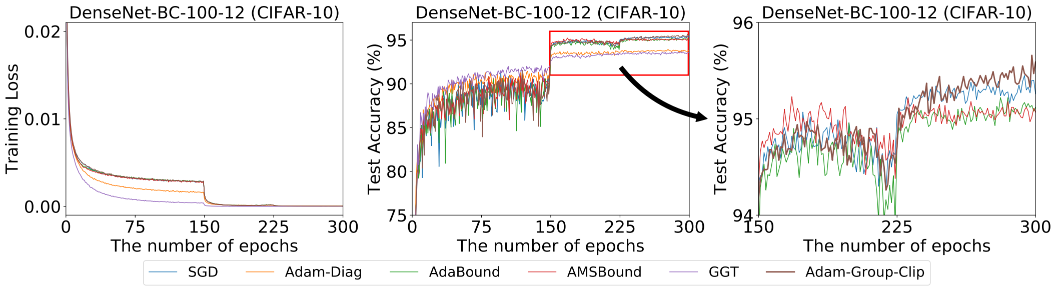


Figure 11: Training curve and test accuracy for DenseNet-BC-100-12 on CIFAR-10 dataset.

D General Frameworks

Algorithm 2 Adaptive Gradient Methods with Block Diagonal Matrix Adaptations via Grouping

```
Input: Stepsize $\alpha_t$, initial point $x_1 \in \mathbb{R}^d$, $\beta_1 \in [0,1)$, and the function $H_t$ which designs $\widehat{V}_t$.
Initialize: $m_0 = 0$, $\widehat{V}_0 = 0$, and $t = 0$.
Assumption: We have $r$ blocks with each size $n_i \times n_i$ and $n_1 + \cdots + n_r = d$, and $\beta_{1,t} \geq \beta_{1,t+1}$
for $t = 1, 2, \ldots, T$ do
    Draw a minibatch sample $\xi_t$ from $\mathbb{P}$
    offset $\leftarrow 0$
    $G_t \leftarrow 0$
    $g_t \leftarrow \nabla f(x_t)$
    $m_t \leftarrow \beta_{1,t} m_{t-1} + (1 - \beta_{1,t}) g_t$
    for each group index $j = 1, 2, \ldots, r$ do
        $g_t^{(j)} \leftarrow g_t[\text{offset} : \text{offset} + n_j]$
        $G_t[\text{offset} : \text{offset} + n_j, \text{offset} : \text{offset} + n_j] \leftarrow g_t^{(j)} \left(g_t^{(j)}\right)^T$
        offset $\leftarrow$ offset + $n_j$
    end for
    $\widehat{V}_t \leftarrow H_t(G_1, \cdots, G_t)$
    $x_{t+1} \leftarrow x_t - \alpha_t(\widehat{V}_t^{1/2} + \delta I)^{-1} m_t$
end for
```

We provide the general frameworks of adaptive gradient methods with exact full matrix adaptations. The Algorithm 3 and 4 represent the general framework for each case. We can identify algorithms according to the functions h_t (Table 3) and H_t (Table 4) which determine the dynamics of $\widehat{v}_t$ and $\widehat{V}_t$ respectively. Also, the Algorithm 2 is a detail version of the Algorithm 1.

Algorithm 3 General Adaptive Gradient Methods approximating $g_t g_t^T$ via DIAGONAL Matrix

```
Input: Initial point $x_1 \in \mathbb{R}^d$, stepsize $\{\alpha_t\}_{t=1}^T$, decay parameters $\beta_{1,t}, \beta_2 \in [0,1]$, and $\epsilon > 0$.
Initialize: $m_0 = 0$, $\widehat{v}_0 = 0$.
for $t = 1, 2, \ldots, T$ do
    Draw a minibatch sample $\xi_t$ from $\mathbb{P}$
    $g_t \leftarrow \nabla f(x_t; \xi_t)$
    $G_t \leftarrow \mathrm{diag}(g_t g_t^T)$
    $m_t \leftarrow \beta_{1,t} m_{t-1} + (1 - \beta_{1,t}) g_t$
    $\widehat{v}_t \leftarrow h_t(G_1, G_2, \ldots, G_t)$
    $x_{t+1} \leftarrow x_t - \alpha_t m_t / (\sqrt{\widehat{v}_t} + \epsilon)$
end for
Output: $\widehat{x}$.
```

Algorithm 4 General Adaptive Gradient Methods with the exact $g_t g_t^T$ (FULL Matrix)

```
Input: Initial point $x_1 \in \mathbb{R}^d$, stepsize $\{\alpha_t\}_{t=1}^T$, decay parameters $\beta_{1,t}, \beta_2 \in [0,1]$, and $\delta > 0$.
Initialize: $m_0 = 0$, $\widehat{V}_0 = 0$.
for $t = 1, 2, \ldots, T$ do
    Draw a minibatch sample $\xi_t$ from $\mathbb{P}$
    $g_t \leftarrow \nabla f(x_t; \xi_t)$
    $G_t \leftarrow g_t g_t^T$
    $m_t \leftarrow \beta_{1,t} m_{t-1} + (1 - \beta_{1,t}) g_t$
    $\widehat{V}_t \leftarrow H_t(G_1, G_2, \ldots, G_t)$
    $x_{t+1} \leftarrow x_t - \alpha_t (\widehat{V}_t^{1/2} + \delta I)^{-1} m_t$
end for
Output: $\widehat{x}$.
```

Table 3: Variants of diagonal matrix adaptations

$\beta_{1,t}$ $\widehat{v}_t$	$\beta_{1,t} = 0$	$\beta_{1,t} = \beta_1$
1	SGD	-
$(1/t) \sum_{t=1}^T g_t^2$	ADAGRAD	ADAFOM
$\beta_2 \widehat{v}_{t-1} + (1 - \beta_2) g_t^2$	RMSPROP	ADAM
$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$, $\widehat{v}_t = \max\{\widehat{v}_{t-1}, v_t\}$	-	AMSGRAD

Table 4: Variants of full matrix adaptations

$\beta_{1,t}$ $\widehat{V}_t$	$\beta_{1,t} = 0$	$\beta_{1,t} = \beta_1$
$\widehat{V}_t = I$	SGD	-
$\widehat{V}_t = \frac{1}{T} \sum_{t=1}^T g_t g_t^T$	ADAGRAD	ADAFOM
$\widehat{V}_t = \beta_2 \widehat{V}_{t-1} + (1 - \beta_2) g_t g_t^T$	RMSPROP	ADAM
$V_t = U_t \Sigma_t U_t^T$, $\widehat{V}_t = U_t \max\{\widehat{\Sigma}_{t-1}, \Sigma_t\} U_t^T$	-	AMSGRAD

E DETAILS FOR CONVERGENCE ANALYSIS IN SECTION 3.1

E.1 PROOFS FOR COROLLARY 1

Before moving onto proofs, we need the following technical lemma

Lemma 1. *(Lemma 12 in Hazan et al. (2007)) For positive definite matrices A and B, the following inequality holds*

$$\mathrm{Tr}\big(A^{-1}(A - B)\big) \le \log|A| - \log|B|$$

For ADAGRAD, we have $\alpha_t = \alpha/\sqrt{t}$ and $\widehat{V}_t = \frac{1}{t} \sum_{i=1}^t g_i g_i^T := \frac{1}{t} \widehat{G}_t$. First, we bound the Term A/Term B to find the $s_1(T)$.

The Term A is

$$\sum_{t=1}^{T} \|\alpha_t \widehat{V}_t^{-1/2} g_t\|^2 = \sum_{t=1}^{t_0} \|\alpha_t \widehat{V}_t^{-1/2} g_t\|^2 + \underbrace{\sum_{t=t_0+1}^{T} \|\alpha_t \widehat{V}_t^{-1/2} g_t\|^2}_{T_1}$$

Since the first term in RHS is independent of T, so we only need to bound T_1. The quantity T_1 can be bound as

$$\begin{aligned}
T_1 &= \sum_{t=t_0+1}^{T} \|\alpha_t \widehat{V}_t^{-1/2} g_t\|^2 \\
&= \alpha^2 \sum_{t=t_0+1}^{T} \frac{1}{t} \|V_t^{-1/2} g_t\|^2 \\
&= \alpha^2 \sum_{t=t_0+1}^{T} \operatorname{Tr}\big(\widehat{V}_t^{-1} \frac{1}{t} g_t g_t^T\big) \\
&\le \alpha^2 \sum_{t=t_0+1}^{T} \Big[\log\big|\widehat{V}_t\big| - \log\big|\frac{t-1}{t}\widehat{V}_{t-1}\big|\Big] \\
&= \alpha^2 \sum_{t=t_0+1}^{T} \Big[\log\big|\widehat{V}_t\big| - \log\big|\widehat{V}_{t-1}\big| + d \log \frac{t}{t-1}\Big] \\
&= \alpha^2 \big(\log|\widehat{V}_T| - \log|\widehat{V}_{t_0}| + d \log \frac{T}{t_0}\big) = \mathcal{O}(\log|\widehat{V}_T| + \log T)
\end{aligned}$$

Next, we bound the Term B. Similarly to the Term A, the Term B can be splitted as follows

$$\begin{aligned}
\sum_{t=2}^{T} Q_t &= \sum_{t=2}^{T} \big\|\!\big|\alpha_{t-1} \widehat{V}_{t-1}^{-1/2} - \alpha_t \widehat{V}_t^{-1/2}\big\|\!\big|_2 \\
&= \sum_{t=2}^{t_0} \big\|\!\big|\alpha_{t-1} \widehat{V}_{t-1}^{-1/2} - \alpha_t \widehat{V}_t^{-1/2}\big\|\!\big|_2 + \underbrace{\sum_{t=t_0+1}^{T} \big\|\!\big|\alpha_{t-1} \widehat{V}_{t-1}^{-1/2} - \alpha_t \widehat{V}_t^{-1/2}\big\|\!\big|_2}_{T_2}
\end{aligned}$$

Also, in this case, the first term in RHS is independent of T, so we only bound T_2. The T_2 can be bound as

$$\begin{aligned}
\sum_{t=t_0+1}^{T} \big\|\!\big|\alpha_{t-1} \widehat{V}_{t-1}^{-1/2} - \alpha_t \widehat{V}_t^{-1/2}\big\|\!\big|_2 &= \alpha \sum_{t=t_0+1}^{T} \Big\|\!\Big|\frac{1}{\sqrt{t-1}} \widehat{V}_{t-1}^{-1/2} - \frac{1}{\sqrt{t}} \widehat{V}_t^{-1/2}\Big\|\!\Big|_2 \\
&= \alpha \sum_{t=t_0+1}^{T} \big\|\!\big|\widehat{G}_{t-1}^{-1/2} - \widehat{G}_t^{-1/2}\big\|\!\big|_2 \\
&\le \alpha \sum_{t=t_0+1}^{T} \operatorname{Tr}\big(\widehat{G}_{t-1}^{-1/2} - \widehat{G}_t^{-1/2}\big) \\
&= \alpha\big(\operatorname{Tr}(\widehat{G}_{t_0}^{-1/2}) - \operatorname{Tr}(\widehat{G}_T^{-1/2})\big) \\
&\le \alpha \operatorname{Tr}(\widehat{G}_{t_0}^{-1/2}) = \mathcal{O}(1)
\end{aligned}$$

The remaining term is $\sum_{t=2}^{T-1} Q_t^2$ which can be derived from Term B with slight modifications.

$$\begin{aligned}
\sum_{t=2}^{T-1} Q_t^2 &= \sum_{t=2}^{T-1} \big\|\!\big|\alpha_{t-1} \widehat{V}_{t-1}^{-1/2} - \alpha_t \widehat{V}_t^{-1/2}\big\|\!\big|_2^2 \\
&= \sum_{t=2}^{t_0} \big\|\!\big|\alpha_{t-1} \widehat{V}_{t-1}^{-1/2} - \alpha_t \widehat{V}_t^{-1/2}\big\|\!\big|_2^2 + \underbrace{\sum_{t=t_0+1}^{T-1} \big\|\!\big|\alpha_{t-1} \widehat{V}_{t-1}^{-1/2} - \alpha_t \widehat{V}_t^{-1/2}\big\|\!\big|_2^2}_{T_3}
\end{aligned}$$

Similarly to the Term B, we can bound T_3 with a little modification as

$$
\begin{aligned}
\sum_{t=t_0+1}^{T-1} \left\|\left|\alpha_{t-1}\widehat{V}_{t-1}^{-1/2} - \alpha_t \widehat{V}_t^{-1/2}\right|\right\|_2^2 &= \alpha^2 \sum_{t=t_0+1}^{T-1} \left\|\left|\widehat{G}_{t-1}^{-1/2} - \widehat{G}_t^{-1/2}\right|\right\|_2^2 \\
&= \alpha^2 \sum_{t=t_0+1}^{T-1} \mathrm{Tr}\Big(\big(\widehat{G}_{t-1}^{-1/2} - \widehat{G}_t^{-1/2}\big)^2\Big) \\
&\le \alpha^2 \sum_{t=t_0+1}^{T-1} \mathrm{Tr}\Big(\widehat{G}_{t-1}^{-1} - \widehat{G}_t^{-1}\Big) \\
&= \alpha^2 \mathrm{Tr}\big(\widehat{G}_{t_0}^{-1} - \widehat{G}_{T-1}^{-1}\big) \\
&\le \alpha^2 \mathrm{Tr}\big(\widehat{G}_{t_0}^{-1}\big)
\end{aligned}
$$

Therefore, we have $s_1(T) = \mathcal{O}\big(\log\big|\widehat{V}_T\big| + \log T + 1\big)$. Lastly, we should bound the LHS of 2.

$$
\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \alpha_t \Big\langle \nabla f(x_t), \widehat{V}_t^{-1/2}\nabla f(x_t)\Big\rangle\right] &\ge \mathbb{E}\left[\sum_{t=1}^{T} \gamma_t \Big\|\nabla f(x_t)\Big\|^2\right] \\
&\ge \mathbb{E}\left[\min_{t\in[T]}\{\|\nabla f(x_t)\|^2\} \sum_{t=1}^{T} \gamma_t\right]
\end{aligned}
$$

Now, we bound the sum of γ_t.

$$
\begin{aligned}
\gamma_t &= \lambda_{\min}(\alpha_t \widehat{V}_t^{-1/2}) = \alpha\lambda_{\min}(G_t^{-1/2}) \\
&= \frac{\alpha}{\lambda_{\max}(G_t^{1/2})} \\
&= \frac{\alpha}{\lambda_{\max}(G_t)^{1/2}} \\
&\ge \frac{\alpha}{\Big(\sum_{\tau=1}^{t} \|g_\tau\|^2\Big)^{1/2}} \ge \frac{\alpha}{\sqrt{T}G_\infty}
\end{aligned}
$$

From above observations, we have

$$
\sum_{t=1}^{T} \gamma_t \ge \sum_{t=1}^{T} \frac{\alpha}{\sqrt{T}G_\infty} = \frac{\alpha\sqrt{T}}{G_\infty}
$$

Therefore, we have $\Omega\big(s_2(T)\big) = \Omega(\sqrt{T})$. Finally, we conclude that

$$
\min_{t\in[T]} \big\|\nabla f(x_t)\big\|^2 = \mathcal{O}\left(\frac{\log\big|\widehat{V}_T\big| + \log T + 1}{\sqrt{T}}\right)
$$

Now, we need a following lemma for relation between block sizes and convergence.

Lemma 2. *(Fischer's inequality, Theorem 7.8.5 in Horn & Johnson (2012)) For a positive definite matrix $A \in \mathbb{R}^{n\times n}$, let $B \in \mathbb{R}^{k\times k}$ and $C \in \mathbb{R}^{n-k\times n-k}$ be top left corner of A and bottom right corner of A respectively. Then,* $\det(A) \le \det(B)\det(C)$ *holds.*

This lemma says that $\log|\widehat{V}_T|$ becomes larger as a block size b decreases.

E.2 Proposition for RMSprop/Adam

Now, we can obtain *intuition* on the dynamics of Term A/Term B for EMA-based algorithms with the following proposition.

Proposition 1. *For block diagonal extensions of* RMSprop/Adam*, we set exponentially decaying stepsize $\alpha_t = \alpha(\sqrt{\beta_2})^{t-1}$. Then, the Term A depends on the* $\log|\widehat{V}_T| - \log|\widehat{V}_{t_0}|$ *and the Term B is upper bounded with constant for any block sizes. Here, t_0 is the time when $\widehat{V}_t$ becomes full-rank.*

Proof. First, we will bound the Term A.

$$\sum_{t=1}^{T} \|\alpha_t \widehat{V}_t^{-1/2} g_t\|^2 = \sum_{t=1}^{t_0} \|\alpha_t \widehat{V}_t^{-1/2} g_t\|^2 + \underbrace{\sum_{t=t_0+1}^{T} \|\alpha_t \widehat{V}_t^{-1/2} g_t\|^2}_{T_1}$$

The first term in RHS is independent of T, so we only have to bound the term T_1. The Term T_1 can be bound as follows

$$\begin{aligned}
T_1 &= \sum_{t=t_0+1}^{T} \|\alpha_t \widehat{V}_t^{-1/2} g_t\|^2 = \sum_{t=t_0+1}^{T} \|\alpha(\sqrt{\beta_2})^{t-1} \widehat{V}_t^{-1/2} g_t\|^2 \\
&\leq \sum_{t=t_0+1}^{T} \|\alpha \widehat{V}_t^{-1/2} g_t\|^2 = \alpha^2 \sum_{t=t_0+1}^{T} \|\widehat{V}_t^{-1/2} g_t\|^2 = \alpha^2 \sum_{t=t_0+1}^{T} \mathrm{Tr}(\widehat{V}_t^{-1} g_t g_t^T) \\
&\leq \frac{\alpha^2}{1-\beta_2} \sum_{t=t_0+1}^{T} \Big[\log|\widehat{V}_t| - \log|\beta_2 \widehat{V}_{t-1}|\Big] \\
&= \frac{\alpha^2}{1-\beta_2} \sum_{t=t_0+1}^{T} \Big[\log|\widehat{V}_t| - \log|\widehat{V}_{t-1}| + d\log\frac{1}{\beta_2}\Big] \\
&= \frac{\alpha^2}{1-\beta_2} \Big(\underbrace{\log|\widehat{V}_T| - \log|\widehat{V}_{t_0}|}_{\text{Dependent on } T} + d(T-t_0)\log\frac{1}{\beta_2}\Big)
\end{aligned}$$

Summing up all the terms, we have

$$\sum_{t=1}^{T} \|\alpha_t \widehat{V}_t^{-1/2} g_t\|^2 \leq \sum_{t=1}^{t_0} \|\alpha_t \widehat{V}_t^{-1/2} g_t\|^2 + \frac{\alpha^2}{1-\beta_2} \Big(\underbrace{\log|\widehat{V}_T| - \log|\widehat{V}_{t_0}|}_{\text{Dependent on } T} + d(T-t_0)\log\frac{1}{\beta_2}\Big)$$

Now, we derive the bound for the Term B.

$$\begin{aligned}
\sum_{t=2}^{T} Q_t &= \sum_{t=2}^{T} \big\|\big|\alpha_{t-1} \widehat{V}_{t-1}^{-1/2} - \alpha_t \widehat{V}_t^{-1/2}\big\|\big|_2 \\
&= \sum_{t=2}^{t_0} \big\|\big|\alpha_{t-1} \widehat{V}_{t-1}^{-1/2} - \alpha_t \widehat{V}_t^{-1/2}\big\|\big|_2 + \sum_{t=t_0+1}^{T} \big\|\big|\alpha_{t-1} \widehat{V}_{t-1}^{-1/2} - \alpha_t \widehat{V}_t^{-1/2}\big\|\big|_2
\end{aligned}$$

As in the Term A, the first term in RHS is independent of T, and we can bound the second term as

$$\begin{aligned}
\sum_{t=t_0+1}^{T} \big\|\big|\alpha_{t-1} \widehat{V}_{t-1}^{-1/2} - \alpha_t \widehat{V}_t^{-1/2}\big\|\big|_2 &= \alpha \sum_{t=2}^{T} \big\|\big|(\sqrt{\beta_2})^{t-2} \widehat{V}_{t-1}^{-1/2} - (\sqrt{\beta_2})^{t-1} \widehat{V}_t^{-1/2}\big\|\big|_2 \\
&= \alpha \sum_{t=t_0+1}^{T} (\sqrt{\beta_2})^{t-1} \Big\|\Big|\frac{1}{\sqrt{\beta_2}} \widehat{V}_{t-1}^{-1/2} - \widehat{V}_t^{-1/2}\Big\|\Big|_2 \\
&= \alpha \sum_{t=t_0+1}^{T} (\sqrt{\beta_2})^{t-1} \big\|\big|(\beta_2 \widehat{V}_{t-1})^{-1/2} - \widehat{V}_t^{-1/2}\big\|\big|_2 \\
&\leq \alpha \sum_{t=t_0+1}^{T} (\sqrt{\beta_2})^{t-1} \mathrm{Tr}\Big((\beta_2 \widehat{V}_{t-1})^{-1/2} - \widehat{V}_t^{-1/2}\Big) \\
&= \alpha \sum_{t=t_0+1}^{T} \mathrm{Tr}\Big((\sqrt{\beta_2})^{t-2} \widehat{V}_{t-1}^{-1/2} - (\sqrt{\beta_2})^{t-1} \widehat{V}_t^{-1/2}\Big) \\
&= \alpha\Big(\mathrm{Tr}\big((\sqrt{\beta_2})^{t_0-1} \widehat{V}_{t_0}^{-1/2}\big) - \mathrm{Tr}\big((\sqrt{\beta_2})^{T-1} \widehat{V}_T^{-1/2}\big)\Big) \\
&\leq \alpha(\sqrt{\beta_2})^{t_0-1} \mathrm{Tr}\big(\widehat{V}_{t_0}^{-1/2}\big)
\end{aligned}$$

Therefore, the bound for the Term B is independent of T.

$$\sum_{t=2}^{T} Q_t \leq \sum_{t=2}^{t_0} \big\|\big|\alpha_{t-1} \widehat{V}_{t-1}^{-1/2} - \alpha_t \widehat{V}_t^{-1/2}\big\|\big|_2 + \alpha(\sqrt{\beta_2})^{t_0-1} \mathrm{Tr}\big(\widehat{V}_{t_0}^{-1/2}\big) = \mathcal{O}(1)$$

As a result, we can expect that the Term A is related to the log-determinant of $\widehat{V}_T$ and the Term B is upper bounded as constant as in ADAGRAD. □

F Proofs of Main Theorems

We study the following minimization problem,

$$\min f(x) \coloneqq \mathbb{E}_\xi[f(x;\xi)]$$

under the assumption 1. The parameter x is an optimization variable, and ξ is a random variable representing randomly selected data sample from $\mathcal{D}$. We study the convergence analysis of the Algorithm 1. For analysis in stochastic convex optimization, one can refer to Duchi et al. (2011). For analysis in non-convex optimization with full matrix adaptations, we follow the arguments in the paper Chen et al. (2019). As we will show, the convergence of the adaptive full matrix adaptations depends on the changes of *effective spectrum* while the diagonal counterpart depends on the changes of *effective stepsize*. We assume that $\widehat{V}_t^{-1/2}$ means pseudo-inverse of $\widehat{V}_t^{1/2}$ if it is not full-rank. Note that, our proof can be applied to exact full matrix adaptations, Algorithm 1

F.1 Technical Lemmas for Theorem 1

Lemma 3. *Consider the sequence*

$$z_t = x_t + \frac{\beta_{1,t}}{1-\beta_{1,t}}(x_t - x_{t-1})$$

Then, the following holds true

$$z_{t+1} - z_t = -\left(\frac{\beta_{1,t+1}}{1-\beta_{1,t+1}} - \frac{\beta_{1,t}}{1-\beta_{1,t}}\right)\alpha_t \widehat{V}_t^{-1/2} m_t - \frac{\beta_{1,t}}{1-\beta_{1,t}}\left(\alpha_t \widehat{V}_t^{-1/2} - \alpha_{t-1}\widehat{V}_{t-1}^{-1/2}\right)m_{t-1} - \alpha_t \widehat{V}_t^{-1/2} g_t$$

Proof. By our update rule, we can derive

$$\begin{aligned}
x_{t+1} - x_t =& -\alpha_t \widehat{V}_t^{-1/2} m_t \\
\overset{(i)}{=}& -\alpha_t \widehat{V}_t^{-1/2}(\beta_{1,t} m_{t-1} + (1-\beta_{1,t}) g_t) \\
=& -\alpha_t \beta_{1,t} \widehat{V}_t^{-1/2} m_{t-1} - \alpha_t(1-\beta_{1,t})\widehat{V}_t^{-1/2} g_t \\
\overset{(ii)}{=}& -\alpha_t \beta_{1,t} \widehat{V}_t^{-1/2}\Big(-\frac{1}{\alpha_{t-1}}\widehat{V}_{t-1}^{1/2}(x_t - x_{t-1})\Big) - \alpha_t(1-\beta_{1,t})\widehat{V}_t^{-1/2} g_t \\
=& \frac{\alpha_t}{\alpha_{t-1}}\beta_{1,t}(\widehat{V}_t^{-1}\widehat{V}_{t-1})^{1/2}(x_t - x_{t-1}) - \alpha_t(1-\beta_{1,t})\widehat{V}_t^{-1/2} g_t \\
=& \beta_{1,t}(x_t - x_{t-1}) + \beta_{1,t}\left(\frac{\alpha_t}{\alpha_{t-1}}(\widehat{V}_t^{-1}\widehat{V}_{t-1})^{1/2} - I_d\right)(x_t - x_{t-1}) - \alpha_t(1-\beta_{1,t})\widehat{V}_t^{-1/2} g_t \\
\overset{(iii)}{=}& \beta_{1,t}(x_t - x_{t-1}) - \beta_{1,t}\left(\alpha_t \widehat{V}_t^{-1/2} - \alpha_{t-1}\widehat{V}_{t-1}^{-1/2}\right)m_{t-1} - \alpha_t(1-\beta_{1,t})\widehat{V}_t^{-1/2} g_t
\end{aligned}$$

The reasoning follows

(i) By definition of m_t.

(ii) Since $x_t = x_{t-1} - \alpha_{t-1}\widehat{V}_{t-1}^{-1/2} m_{t-1}$, we can solve as $m_{t-1} = -\frac{1}{\alpha_{t-1}}\widehat{V}_{t-1}^{1/2}(x_t - x_{t-1})$.

(iii) Similarly to (ii), we can have $\widehat{V}_{t-1}^{1/2}(x_t - x_{t-1})/\alpha_{t-1} = -m_{t-1}$.

Since $x_{t+1} - x_t = (1-\beta_{1,t})x_{t+1} + \beta_{1,t}(x_{t+1} - x_t) - (1-\beta_{1,t})x_t$, we can further derive by combining the above,

$$\begin{aligned}
& (1-\beta_{1,t})x_{t+1} + \beta_{1,t}(x_{t+1} - x_t) \\
=& (1-\beta_{1,t})x_t + \beta_{1,t}(x_t - x_{t-1}) - \beta_{1,t}\left(\alpha_t \widehat{V}_t^{-1/2} - \alpha_{t-1}\widehat{V}_{t-1}^{-1/2}\right)m_{t-1} - \alpha_t(1-\beta_{1,t})\widehat{V}_t^{-1/2} g_t
\end{aligned}$$

By dividing both sides by $1-\beta_{1,t}$,

$$\begin{aligned} & x_{t+1} + \frac{\beta_{1,t}}{1-\beta_{1,t}}(x_{t+1}-x_t) \\ = \; & x_t + \frac{\beta_{1,t}}{1-\beta_{1,t}}(x_t - x_{t-1}) - \frac{\beta_{1,t}}{1-\beta_{1,t}}\left(\alpha_t \widehat{V}_t^{-1/2} - \alpha_{t-1}\widehat{V}_{t-1}^{-1/2}\right)m_{t-1} - \alpha_t \widehat{V}_t^{-1/2} g_t \end{aligned}$$

Define the sequence

$$z_t = x_t + \frac{\beta_{1,t}}{1-\beta_{1,t}}(x_t - x_{t-1})$$

Then, we obtain

$$\begin{aligned} z_{t+1} &= z_t + \left(\frac{\beta_{1,t+1}}{1-\beta_{1,t+1}} - \frac{\beta_{1,t}}{1-\beta_{1,t}}\right)(x_{t+1}-x_t) \\ &\quad - \frac{\beta_{1,t}}{1-\beta_{1,t}}\left(\alpha_t \widehat{V}_t^{-1/2} - \alpha_{t-1}\widehat{V}_{t-1}^{-1/2}\right)m_{t-1} - \alpha_t \widehat{V}_t^{-1/2} g_t \\ &= z_t - \left(\frac{\beta_{1,t+1}}{1-\beta_{1,t+1}} - \frac{\beta_{1,t}}{1-\beta_{1,t}}\right)\alpha_t \widehat{V}_t^{-1/2} m_t \\ &\quad - \frac{\beta_{1,t}}{1-\beta_{1,t}}\left(\alpha_t \widehat{V}_t^{-1/2} - \alpha_{t-1}\widehat{V}_{t-1}^{-1/2}\right)m_{t-1} - \alpha_t \widehat{V}_t^{-1/2} g_t \end{aligned}$$

By putting z_t to the left hand side, we can get desired relations. □

Lemma 4. *Suppose that the assumptions in Theorem 1 hold, then*

$$\mathbb{E}[f(z_{t+1}) - f(z_1)] \le \sum_{i=1}^{6} T_i$$

where

$$\begin{aligned} T_1 &= -\mathbb{E}\left[\sum_{i=1}^{t}\left\langle \nabla f(z_i), \frac{\beta_{1,i}}{1-\beta_{1,i}}\left(\alpha_i \widehat{V}_i^{-1/2} - \alpha_{i-1}\widehat{V}_{i-1}^{-1/2}\right)m_{i-1}\right\rangle\right] \\ T_2 &= -\mathbb{E}\left[\sum_{i=1}^{t}\alpha_i\left\langle \nabla f(z_i), \widehat{V}_i^{-1/2} g_i\right\rangle\right] \\ T_3 &= -\mathbb{E}\left[\sum_{i=1}^{t}\left\langle \nabla f(z_i), \left(\frac{\beta_{1,i+1}}{1-\beta_{1,i+1}} - \frac{\beta_{1,i}}{1-\beta_{1,i}}\right)\alpha_i \widehat{V}_i^{-1/2} m_i\right\rangle\right] \\ T_4 &= \mathbb{E}\left[\sum_{i=1}^{t}\frac{3}{2}L\left\|\left(\frac{\beta_{1,i+1}}{1-\beta_{1,i+1}} - \frac{\beta_{1,i}}{1-\beta_{1,i}}\right)\alpha_t \widehat{V}_i^{-1/2} m_i\right\|^2\right] \\ T_5 &= \mathbb{E}\left[\sum_{i=1}^{t}\frac{3}{2}L\left\|\frac{\beta_{1,i}}{1-\beta_{1,i}}\left(\alpha_i \widehat{V}_i^{-1/2} - \alpha_{i-1}\widehat{V}_{i-1}^{-1/2}\right)m_{i-1}\right\|^2\right] \\ T_6 &= \mathbb{E}\left[\sum_{i=1}^{t}\frac{3}{2}L\left\|\alpha_i \widehat{V}_i^{-1/2} g_i\right\|^2\right] \end{aligned}$$

Proof. By L-Lipschitz continuous gradients, we get the following quadratic upper bound,

$$f(z_{t+1}) \le f(z_t) + \langle \nabla f(z_t), z_{t+1} - z_t\rangle + \frac{L}{2}\|z_{t+1} - z_t\|^2$$

Let $d_t = z_{t+1} - z_t$. The lemma 1 yields

$$d_t = -\left(\frac{\beta_{1,t+1}}{1-\beta_{1,t+1}} - \frac{\beta_{1,t}}{1-\beta_{1,t}}\right)\alpha_t V_t^{-1/2} m_t - \frac{\beta_{1,t}}{1-\beta_{1,t}}\left(\alpha_t V_t^{-1/2} - \alpha_{t-1}V_{t-1}^{-1/2}\right)m_{t-1} - \alpha_t V_t^{-1/2} g_t$$

Combining with Lipschitz continuous gradients, we have

$$\begin{aligned}
\mathbb{E}[f(z_{t+1}) - f(z_1)] &= \mathbb{E}\left[\sum_{i=1}^{t} f(z_{i+1}) - f(z_i)\right] \\
&\le \mathbb{E}\left[\sum_{i=1}^{t} \langle \nabla f(z_i), d_i\rangle + \frac{L}{2}\|d_i\|^2\right] \\
&= -\mathbb{E}\left[\sum_{i=1}^{t}\left\langle \nabla f(z_i), \frac{\beta_{1,i}}{1-\beta_{1,i}}\left(\alpha_i V_i^{-1/2} - \alpha_{i-1}V_{i-1}^{-1/2}\right)m_{i-1}\right\rangle\right] \\
&\quad - \mathbb{E}\left[\sum_{i=1}^{t}\alpha_i\left\langle \nabla f(z_i), V_i^{-1/2} g_i\right\rangle\right] \\
&\quad - \mathbb{E}\left[\sum_{i=1}^{t}\left\langle \nabla f(z_i), \left(\frac{\beta_{1,i+1}}{1-\beta_{1,i+1}} - \frac{\beta_{1,i}}{1-\beta_{1,i}}\right)\alpha_i V_i^{-1/2} m_i\right\rangle\right] \\
&\quad + \mathbb{E}\left[\sum_{i=1}^{t}\frac{L}{2}\|d_i\|^2\right] = T_1 + T_2 + T_3 + \mathbb{E}\left[\sum_{i=1}^{t}\frac{L}{2}\|d_i\|^2\right]
\end{aligned}$$

With $\|a+b+c\|^2 \le 3(\|a\|^2 + \|b\|^2 + \|c\|^2)$, we can finally bound by

$$\mathbb{E}[f(z_{t+1}) - f(z_1)] \le \sum_{i=1}^{6} T_i$$

□

Lemma 5. *Suppose that the assumptions in Theorem 1 hold, T_1 can be bound as*

$$T_1 \le G^2\frac{\beta_1}{1-\beta_1}\mathbb{E}\left[\sum_{i=1}^{t}\left\|\alpha_i V_i^{-1/2} - \alpha_{i-1}V_{i-1}^{-1/2}\right\|_2\right]$$

Proof. From the definition of quantity T_1,

$$\begin{aligned}
T_1 = &\ -\mathbb{E}\left[\sum_{i=1}^{t}\left\langle \nabla f(z_i), \frac{\beta_{1,i}}{1-\beta_{1,i}}\left(\alpha_i V_i^{-1/2} - \alpha_{i-1}V_{i-1}^{-1/2}\right)m_{i-1}\right\rangle\right] \\
\overset{(i)}{\le} &\ \mathbb{E}\left[\sum_{i=1}^{t}\|\nabla f(z_i)\|_2\left\|\frac{\beta_{1,i}}{1-\beta_{1,i}}\left(\alpha_i V_i^{-1/2} - \alpha_{i-1}V_{i-1}^{-1/2}\right)m_{i-1}\right\|_2\right] \\
\overset{(ii)}{\le} &\ \frac{\beta_1}{1-\beta_1}\mathbb{E}\left[\sum_{i=1}^{t}\|\nabla f(z_i)\|_2\left\|\alpha_i V_i^{-1/2} - \alpha_{i-1}V_{i-1}^{-1/2}\right\|_2\|m_{t-1}\|_2\right] \\
\overset{(iii)}{\le} &\ G^2\frac{\beta_1}{1-\beta_1}\mathbb{E}\left[\sum_{i=1}^{t}\left\|\alpha_i V_i^{-1/2} - \alpha_{i-1}V_{i-1}^{-1/2}\right\|_2\right]
\end{aligned}$$

The reasoning follows

(i) By Cauchy-Schwarz inequality.

(ii) For a matrix norm, we have $\|Ax\|_2 \le \|A\|_2\|x\|_2$. Also, $\frac{\beta_{1,i}}{1-\beta_{1,i}} = \frac{1}{1-\beta_{1,i}} - 1 \le \frac{1}{1-\beta_1} - 1 = \frac{\beta_1}{1-\beta_1}$.

(iii) By definition of m_t, we have $m_t = \beta_{1,t}m_{t-1} + (1-\beta_{1,t})g_t$. Therefore, we use a triangle inequality by $\|m_t\|_2 \le \beta_{1,t}\|m_{t-1}\|_2 + (1-\beta_1)\|g_t\|_2 \le (\beta_{1,t} + 1 - \beta_{1,t})\max\{\|m_{t-1}\|_2, \|g_t\|_2\}$. Since we have $m_0 = 0$ and $\|g_t\| \le G$, we also have $\|m_t\| \le G$ by the mathematical induction.

□

Lemma 6. *Suppose that the assumptions in Theorem 1 hold, then T_3 can be bound as*

$$T_3 \le \left(\frac{\beta_1}{1-\beta_1} - \frac{\beta_{1,t+1}}{1-\beta_{1,t+1}}\right)(G^2 + D^2)$$

Proof. By the definition of T_3,

$$\begin{aligned} T_3 = \; & -\mathbb{E}\Bigg[\sum_{i=1}^{t}\Bigg\langle \nabla f(z_i), \Bigg(\frac{\beta_{1,i+1}}{1-\beta_{1,i+1}} - \frac{\beta_{1,i}}{1-\beta_{1,i}}\Bigg)\alpha_i V_i^{-1/2} m_i\Bigg\rangle\Bigg] \\ \overset{(i)}{\leq} \; & \mathbb{E}\Bigg[\sum_{i=1}^{t}\Bigg|\frac{\beta_{1,i+1}}{1-\beta_{1,i+1}} - \frac{\beta_{1,i}}{1-\beta_{1,i}}\Bigg|\frac{1}{2}\Big(\|\nabla f(z_i)\|^2 + \|\alpha_i V_i^{-1/2} m_i\|^2\Big)\Bigg] \\ \overset{(ii)}{\leq} \; & \mathbb{E}\Bigg[\sum_{i=1}^{t}\Bigg|\frac{\beta_{1,i+1}}{1-\beta_{1,i+1}} - \frac{\beta_{1,i}}{1-\beta_{1,i}}\Bigg|\frac{1}{2}\Big(G^2 + D^2\Big)\Bigg] \\ = \; & \sum_{i=1}^{t}\Bigg(\frac{\beta_{1,i}}{1-\beta_{1,i}} - \frac{\beta_{1,i+1}}{1-\beta_{1,i+1}}\Bigg)\frac{1}{2}\Big(G^2 + D^2\Big) \\ \overset{(iii)}{\leq} \; & \Bigg(\frac{\beta_1}{1-\beta_1} - \frac{\beta_{1,t+1}}{1-\beta_{1,t+1}}\Bigg)(G^2 + D^2) \end{aligned}$$

The reasoning follows

(i) Use Cauchy-Schwarz inequality and $ab \leq \frac{1}{2}(a^2 + b^2)$ for $a, b \geq 0$.

(ii) By our assumptions on bounded gradients and bounded final step vectors.

(iii) The sum over $i = 1$ to T can be done by telescoping.

□

Lemma 7. *Suppose that the assumptions in Theorem 1 hold, T_4 can be bound as*

$$T_4 \leq \Big(\frac{\beta_1}{1-\beta_1} - \frac{\beta_{1,t+1}}{1-\beta_{1,t+1}}\Big)^2 D^2$$

Proof. By the definition of T_4,

$$\begin{aligned} \frac{2}{3L}T_4 = \; & \mathbb{E}\Bigg[\sum_{i=1}^{t}\Bigg\|\Bigg(\frac{\beta_{1,i+1}}{1-\beta_{1,i+1}} - \frac{\beta_{1,i}}{1-\beta_{1,i}}\Bigg)\alpha_i V_i^{-1/2} m_i\Bigg\|^2\Bigg] \\ \overset{(i)}{\leq} \; & \mathbb{E}\Bigg[\sum_{i=1}^{t}\Big(\frac{\beta_{1,i+1}}{1-\beta_{1,i+1}} - \frac{\beta_{1,i}}{1-\beta_{1,i}}\Big)^2 D^2\Bigg] \\ \overset{(ii)}{\leq} \; & \Big(\frac{\beta_1}{1-\beta_1} - \frac{\beta_{1,t+1}}{1-\beta_{1,t+1}}\Big)\sum_{i=1}^{t}\Big(\frac{\beta_{1,i+1}}{1-\beta_{1,i+1}} - \frac{\beta_{1,i}}{1-\beta_{1,i}}\Big)D^2 \\ \overset{(iii)}{\leq} \; & \Big(\frac{\beta_1}{1-\beta_1} - \frac{\beta_{1,t+1}}{1-\beta_{1,t+1}}\Big)^2 D^2 \end{aligned}$$

The reasoning follows

(i) From our assumptions on final step vector $\|\alpha_i \widehat{V}_i^{-1/2} m_i\|^2 \leq D$.

(ii) We use the relation $\beta_1 \geq \beta_{1,t} \leq \beta_{1,t+1}$.

(iii) By telescoping sum, we can get the final result.

□

Lemma 8. *Suppose that the assumptions in Theorem 1 hold, T_5 can be bound as*

$$\frac{2}{3L}T_5 \leq \Big(\frac{\beta_1}{1-\beta_1}\Big)^2 G^2 \mathbb{E}\Bigg[\sum_{i=2}^{t}\Bigg\|\alpha_i \widehat{V}_i^{-1/2} - \alpha_{i-1}\widehat{V}_{i-1}^{-1/2}\Bigg\|_2\Bigg]$$

Proof. By the definition of T_5,

$$
\begin{aligned}
\frac{2}{3L}T_5 = \ & \mathbb{E}\Bigg[\sum_{i=2}^{t}\Bigg\|\frac{\beta_{1,i}}{1-\beta_{1,i}}\Bigg(\alpha_i V_i^{-1/2} - \alpha_{i-1}V_{i-1}^{-1/2}\Bigg)m_{i-1}\Bigg\|^2\Bigg] \\
\overset{(i)}{\leq} \ & \mathbb{E}\Bigg[\sum_{i=2}^{t}\frac{\beta_{1,i}}{1-\beta_{1,i}}\Bigg\|\Bigg|\alpha_i \widehat{V}_i^{-1/2} - \alpha_{i-1}\widehat{V}_{i-1}^{-1/2}\Bigg|\Bigg\|_2^2 \|m_{i-1}\|_2^2\Bigg] \\
\overset{(ii)}{\leq} \ & \Big(\frac{\beta_1}{1-\beta_1}\Big)^2 G^2 \mathbb{E}\Bigg[\sum_{i=2}^{t}\Bigg\|\Bigg|\alpha_i \widehat{V}_i^{-1/2} - \alpha_{i-1}\widehat{V}_{i-1}^{-1/2}\Bigg|\Bigg\|_2^2\Bigg]
\end{aligned}
$$

The reasoning follows

(i) By the matrix norm inequality, we use $\|Ax\|_2 \leq \|\!|A|\!\|_2\|x\|_2$.

(ii) We can obtain the result using $\beta_1 \geq \beta_{1,t} \geq \beta_{1,t+1}$.

□

Lemma 9. *Suppose that the assumptions in Theorem 1 hold, The quantity T_2 can be bound as*

$$
\begin{aligned}
T_2 \leq \ & L^2\Big(\frac{\beta_1}{1-\beta_1}\Big)^2 T_8 + L^2\Big(\frac{\beta_1}{1-\beta_1}\Big)^2 T_9 + \frac{1}{2}\mathbb{E}\Bigg[\sum_{i=1}^{t}\|\alpha_i \widehat{V}_i^{-1/2} g_i\|^2\Bigg] \\
& + 2G^2\mathbb{E}\Bigg[\sum_{i=2}^{t}\Bigg\|\Bigg|\alpha_i V_i^{-1/2} - \alpha_{i-1}V_{i-1}^{-1/2}\Bigg|\Bigg\|_2\Bigg] + 2G^2\mathbb{E}\Bigg[\Bigg\|\Bigg|\alpha_1 V_1^{-1/2}\Bigg|\Bigg\|_2\Bigg] \\
& - \mathbb{E}\Bigg[\sum_{i=1}^{t}\alpha_i\Bigg\langle\nabla f(x_i), V_i^{-1/2}\nabla f(x_i)\Bigg\rangle\Bigg]
\end{aligned}
$$

Proof. First, note that,

$$
z_i - x_i = \frac{\beta_{1,i}}{1-\beta_{1,i}}(x_i - x_{i-1}) = -\frac{\beta_{1,i}}{1-\beta_{1,i}}\alpha_{i-1}\widehat{V}_{i-1}^{-1/2}m_{i-1}
$$

By the definition of T_2 and $z_1 = x_1$, we have

$$
\begin{aligned}
T_2 &= -\mathbb{E}\Bigg[\sum_{i=1}^{t}\alpha_i\Bigg\langle\nabla f(z_i), \widehat{V}_i^{-1/2}g_i\Bigg\rangle\Bigg] \\
&= -\mathbb{E}\Bigg[\sum_{i=1}^{t}\alpha_i\Bigg\langle\nabla f(x_i), \widehat{V}_i^{-1/2}g_i\Bigg\rangle\Bigg] - \mathbb{E}\Bigg[\sum_{i=1}^{t}\alpha_i\Bigg\langle\nabla f(z_i) - \nabla f(x_i), \widehat{V}_i^{-1/2}g_i\Bigg\rangle\Bigg]
\end{aligned}
$$

The second term can be bounded as

$$
\begin{aligned}
& - \mathbb{E}\Bigg[\sum_{i=1}^{t}\alpha_i\Bigg\langle\nabla f(z_i) - \nabla f(x_i), \widehat{V}_i^{-1/2}g_i\Bigg\rangle\Bigg] \\
\overset{(i)}{\leq} \ & \mathbb{E}\Bigg[\sum_{i=1}^{t}\frac{1}{2}\|\nabla f(z_i) - \nabla f(x_i)\|^2 + \frac{1}{2}\|\alpha_i\widehat{V}_i^{-1/2}g_i\|^2\Bigg] \\
\overset{(ii)}{\leq} \ & \frac{L^2}{2}T_7 + \frac{1}{2}\mathbb{E}\Bigg[\sum_{i=1}^{t}\|\alpha_i\widehat{V}_i^{-1/2}g_i\|^2\Bigg]
\end{aligned}
$$

(i) is due to Cauchy-Schwarz inequality and $ab \leq \frac{1}{2}a^2 + \frac{1}{2}b^2$ for $a, b \geq 0$. (ii) is as follows:

By L-Lipschitz continuous gradients, we have

$$
\|\nabla f(z_i) - \nabla f(x_i)\| \leq L\|z_i - x_i\| = L\Bigg\|\frac{\beta_{1,t}}{1-\beta_{1,t}}\alpha_{i-1}V_{i-1}^{-1/2}m_{i-1}\Bigg\|
$$

Let T_7 be

$$
T_7 = \mathbb{E}\Bigg[\sum_{i=1}^{t}\Bigg\|\frac{\beta_{1,i}}{1-\beta_{1,i}}\alpha_{i-1}V_{i-1}^{-1/2}m_{i-1}\Bigg\|^2\Bigg]
$$

We should bound the quantity T_7, by the definition of m_t, we have

$$m_i = \sum_{k=1}^{i}\big[\big(\prod_{l=k+1}^{i}\beta_{1,l}\big)(1-\beta_{1,k})g_k\big]$$

Plugging m_{i-1} into T_7 yields

$$\begin{aligned}
T_7 =\ & \mathbb{E}\Bigg[\sum_{i=1}^{t}\Big\|\frac{\beta_{1,i}}{1-\beta_{1,i}}\alpha_{i-1}V_{i-1}^{-1/2}m_{i-1}\Big\|^2\Bigg]\\
\overset{(i)}{\leq}\ & \Big(\frac{\beta_1}{1-\beta_1}\Big)^2\mathbb{E}\Bigg[\sum_{i=2}^{t}\Bigg\|\alpha_{i-1}V_{i-1}^{-1/2}\sum_{k=1}^{i-1}\Big[\Big(\prod_{l=k+1}^{i-1}\beta_{1,l}\Big)(1-\beta_{1,k})g_k\Big]\Bigg\|^2\Bigg]\\
=\ & \Big(\frac{\beta_1}{1-\beta_1}\Big)^2\mathbb{E}\Bigg[\sum_{i=2}^{t}\Bigg\|\sum_{k=1}^{i-1}\alpha_{i-1}V_{i-1}^{-1/2}\Big[\Big(\prod_{l=k+1}^{i-1}\beta_{1,l}\Big)(1-\beta_{1,k})g_k\Big]\Bigg\|^2\Bigg]\\
\overset{(ii)}{\leq}\ & 2\Big(\frac{\beta_1}{1-\beta_1}\Big)^2\underbrace{\mathbb{E}\Bigg[\sum_{i=2}^{t}\Bigg\|\sum_{k=1}^{i-1}\alpha_k V_k^{-1/2}\Big[\Big(\prod_{l=k+1}^{i-1}\beta_{1,l}\Big)(1-\beta_{1,k})g_k\Big]\Bigg\|^2\Bigg]}_{T_8}\\
&+2\Big(\frac{\beta_1}{1-\beta_1}\Big)^2\underbrace{\mathbb{E}\Bigg[\sum_{i=2}^{t}\Bigg\|\sum_{k=1}^{i-1}\Big(\alpha_i V_i^{-1/2}-\alpha_k V_k^{-1/2}\Big)\Big[\Big(\prod_{l=k+1}^{i-1}\beta_{1,l}\Big)(1-\beta_{1,k})g_k\Big]\Bigg\|^2\Bigg]}_{T_9}
\end{aligned}$$

(i) is by $\beta_1 \geq \beta_{1,t}$ and (ii) is by We use the fact $(a+b) \leq 2(\|a\|^2+\|b\|^2)$ in (i). We first bound T_8 as below

$$\begin{aligned}
T_8 &= \mathbb{E}\Bigg[\sum_{i=2}^{t}\Bigg\|\sum_{k=1}^{i-1}\alpha_k V_k^{-1/2}\Big[\Big(\prod_{l=k+1}^{i-1}\beta_{1,l}\Big)(1-\beta_{1,k})g_k\Big]\Bigg\|^2\Bigg]\\
&= \mathbb{E}\Bigg[\sum_{i=2}^{t}\sum_{j=1}^{d}\Bigg(\sum_{k=1}^{i-1}\alpha_k V_k^{-1/2}\Big[\Big(\prod_{l=k+1}^{i-1}\beta_{1,l}\Big)(1-\beta_{1,k})g_k\Big]\Bigg)_j^2\Bigg]\\
&= \mathbb{E}\Bigg[\sum_{i=2}^{t}\sum_{j=1}^{d}\Bigg(\sum_{k=1}^{i-1}\sum_{p=1}^{i-1}\Big(\alpha_k V_k^{-1/2}g_k\Big)_j\Big(\prod_{l=k+1}^{i-1}\beta_{1,l}\Big)(1-\beta_{1,k})\Big(\alpha_p V_p^{-1/2}g_p\Big)_j\Big(\prod_{q=p+1}^{i-1}\beta_{1,q}\Big)(1-\beta_{1,p})\Bigg)\Bigg]\\
&\leq \mathbb{E}\Bigg[\sum_{i=2}^{t}\sum_{j=1}^{d}\Bigg(\sum_{k=1}^{i-1}\sum_{p=1}^{i-1}(\beta_1^{i-1-k})(\beta_1^{i-1-p})\frac{1}{2}\Big\{\Big(\alpha_k V_k^{-1/2}g_k\Big)_j^2+\Big(\alpha_p V_p^{-1/2}g_p\Big)_j^2\Big\}\Bigg)\Bigg]\\
&= \mathbb{E}\Bigg[\sum_{i=2}^{t}\sum_{j=1}^{d}\Bigg(\sum_{k=1}^{i-1}(\beta_1^{i-1-k})\Big(\alpha_k V_k^{-1/2}g_k\Big)_j^2\sum_{p=1}^{i-1}(\beta_1^{i-1-p})\Bigg)\Bigg]\\
&\leq \frac{1}{1-\beta_1}\mathbb{E}\Bigg[\sum_{i=2}^{t}\sum_{j=1}^{d}\sum_{k=1}^{i-1}(\beta_1^{i-1-k})\Big(\alpha_k V_k^{-1/2}g_k\Big)_j^2\Bigg]\\
&= \frac{1}{1-\beta_1}\mathbb{E}\Bigg[\sum_{k=1}^{t-1}\sum_{j=1}^{d}\sum_{i=k+1}^{t}(\beta_1^{i-1-k})\Big(\alpha_k V_k^{-1/2}g_k\Big)_j^2\Bigg]\\
&= \Big(\frac{1}{1-\beta_1}\Big)^2\mathbb{E}\Bigg[\sum_{k=1}^{t-1}\sum_{j=1}^{d}\Big(\alpha_k V_k^{-1/2}g_k\Big)_j^2\Bigg] = \Big(\frac{1}{1-\beta_1}\Big)^2\mathbb{E}\Bigg[\sum_{i=1}^{t-1}\Big\|\alpha_i V_i^{-1/2}g_i\Big\|^2\Bigg]
\end{aligned}$$

For the T_9 bound, we have

$$\begin{aligned}
T_9 &= \mathbb{E}\Bigg[\sum_{i=2}^{t}\Bigg\|\sum_{k=1}^{i-1}\Big[\Big(\prod_{l=k+1}^{i-1}\beta_{1,l}\Big)(1-\beta_{1,k})\Big]\Big(\alpha_i V_i^{-1/2} - \alpha_k V_k^{-1/2}\Big)g_k\Bigg\|^2\Bigg] \\
&\leq \mathbb{E}\Bigg[\sum_{i=2}^{t}\Bigg(\sum_{k=1}^{i-1}\Big[\Big(\prod_{l=k+1}^{i-1}\beta_{1,l}\Big)(1-\beta_{1,k})\Big]\Big\|\alpha_i V_i^{-1/2} - \alpha_k V_k^{-1/2}\Big\|_2 \|g_k\|_2\Bigg)^2\Bigg] \\
&\leq \mathbb{E}\Bigg[\sum_{i=1}^{t-1}\Bigg(\sum_{k=1}^{i}\Big[\Big(\prod_{l=k+1}^{i}\beta_{1,l}\Big)\Big]\Big\|\alpha_i V_i^{-1/2} - \alpha_k V_k^{-1/2}\Big\|_2 \|g_k\|_2\Bigg)^2\Bigg] \\
&\leq G^2\mathbb{E}\Bigg[\sum_{i=1}^{t-1}\Bigg(\sum_{k=1}^{i}\beta_1^{i-k}\Big\|\alpha_i V_i^{-1/2} - \alpha_k V_k^{-1/2}\Big\|_2\Bigg)^2\Bigg] \\
&\leq G^2\mathbb{E}\Bigg[\sum_{i=1}^{t-1}\Bigg(\sum_{k=1}^{i}\beta_1^{i-k}\sum_{l=k+1}^{i}\Big\|\alpha_l V_l^{-1/2} - \alpha_{l-1} V_{l-1}^{-1/2}\Big\|_2\Bigg)^2\Bigg] \\
&\leq G^2\Big(\frac{1}{1-\beta_1}\Big)^2\Big(\frac{\beta_1}{1-\beta_1}\Big)^2\mathbb{E}\Bigg[\sum_{i=2}^{t-1}\Big\|\alpha_i V_i^{-1/2} - \alpha_{i-1} V_{i-1}^{-1/2}\Big\|_2^2\Bigg]
\end{aligned}$$

Then, the remaining term is

$$\mathbb{E}\Bigg[\sum_{i=1}^{t}\alpha_i\Big\langle\nabla f(x_i), V_i^{-1/2}g_i\Big\rangle\Bigg]$$

To find the upper bound for this term, we reparameterize $g_t = \nabla f(x_t) + \delta_t$ with $\mathbb{E}[\delta_t] = 0$, and we have

$$\begin{aligned}
&\mathbb{E}\Bigg[\sum_{i=1}^{t}\alpha_i\Big\langle\nabla f(x_i), V_i^{-1/2}g_i\Big\rangle\Bigg] \\
=\ &\mathbb{E}\Bigg[\sum_{i=1}^{t}\alpha_i\Big\langle\nabla f(x_i), V_i^{-1/2}(\nabla f(x_i) + \delta_i)\Big\rangle\Bigg] \\
=\ &\mathbb{E}\Bigg[\sum_{i=1}^{t}\alpha_i\Big\langle\nabla f(x_i), V_i^{-1/2}\nabla f(x_i)\Big\rangle\Bigg] + \Bigg[\sum_{i=1}^{t}\alpha_i\Big\langle\nabla f(x_i), V_i^{-1/2}\delta_i\Big\rangle\Bigg]
\end{aligned}$$

For the second term of last equation,

$$\begin{aligned}
&\mathbb{E}\Bigg[\sum_{i=1}^{t}\alpha_i\Big\langle\nabla f(x_i), V_i^{-1/2}\delta_i\Big\rangle\Bigg] \\
=\ &\mathbb{E}\Bigg[\sum_{i=2}^{t}\Big\langle\nabla f(x_i), \Big(\alpha_i V_i^{-1/2} - \alpha_{i-1}V_{i-1}^{-1/2}\Big)\delta_i\Big\rangle\Bigg] + \mathbb{E}\Bigg[\sum_{i=2}^{t}\alpha_{i-1}\Big\langle\nabla f(x_i), V_{i-1}^{-1/2}\delta_i\Big\rangle\Bigg] + \mathbb{E}\Bigg[\alpha_1\Big\langle\nabla f(x_1), V_1^{-1/2}\delta_1\Big\rangle\Bigg] \\
=\ &\mathbb{E}\Bigg[\sum_{i=2}^{t}\Big\langle\nabla f(x_i), \Big(\alpha_i V_i^{-1/2} - \alpha_{i-1}V_{i-1}^{-1/2}\Big)\delta_i\Big\rangle\Bigg] + \mathbb{E}\Bigg[\alpha_1\nabla f(x_1)^T V_1^{-1/2}\delta_1\Bigg] \\
\overset{(i)}{\geq}\ &\mathbb{E}\Bigg[\sum_{i=2}^{t}\Big\langle\nabla f(x_i), \Big(\alpha_i V_i^{-1/2} - \alpha_{i-1}V_{i-1}^{-1/2}\Big)\delta_i\Big\rangle\Bigg] - 2G^2\mathbb{E}\Bigg[\Big\|\alpha_1 V_1^{-1/2}\Big\|_2\Bigg]
\end{aligned}$$

The reasoning is as follows:

(i) The conditional expectation $\mathbb{E}\Big[V_{i-1}^{-1/2}\delta_i\Big|x_i, \widehat{V}_{i-1}\Big] = 0$ since the $\widehat{V}_{i-1}$ only depends on the noise variables $\xi_1, \cdots, \xi_{i-1}$ and δ_i depends on ξ_i with $\mathbb{E}[\xi_k] = 0$ for all $k \in \{1, 2, ..., i\}$. Therefore, they are independent.

Further, we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=2}^{t}\left\langle \nabla f(x_i), \left(\alpha_i V_i^{-1/2} - \alpha_{i-1} V_{i-1}^{-1/2}\right)\delta_i \right\rangle\right] &\geq -\mathbb{E}\left[\sum_{i=2}^{t}\left|\left\langle \nabla f(x_i), \left(\alpha_i V_i^{-1/2} - \alpha_{i-1} V_{i-1}^{-1/2}\right)\delta_i \right\rangle\right|\right] \\
&\overset{(ii)}{\geq} -\mathbb{E}\left[\sum_{i=2}^{t}\left\|\nabla f(x_i)\right\|_2 \left\|\left(\alpha_i V_i^{-1/2} - \alpha_{i-1}^{-1/2}\right)\delta_i\right\|_2\right] \\
&\overset{(iii)}{\geq} -2G^2 \mathbb{E}\left[\sum_{i=2}^{t}\left|\left|\left|\alpha_i V_i^{-1/2} - \alpha_{i-1} V_{i-1}^{-1/2}\right|\right|\right|_2\right]
\end{aligned}$$

Therefore, we can bound the first term

$$\begin{aligned}
&-\mathbb{E}\left[\sum_{i=1}^{t}\alpha_i\left\langle \nabla f(x_i), V_i^{-1/2} g_i \right\rangle\right] \\
\leq\ & 2G^2\mathbb{E}\left[\sum_{i=2}^{t}\left|\left|\left|\alpha_i V_i^{-1/2} - \alpha_{i-1} V_{i-1}^{-1/2}\right|\right|\right|_2\right] + 2G^2\mathbb{E}\left[\left|\left|\left|\alpha_1 V_1^{-1/2}\right|\right|\right|_2\right] - \mathbb{E}\left[\sum_{i=1}^{t}\alpha_i\left\langle \nabla f(x_i), V_i^{-1/2}\nabla f(x_i)\right\rangle\right]
\end{aligned}$$

□

Lemma 10. *(Lemma 6.8 in Chen et al. (2019)) For $a_i \leq 0$, $\beta \in [0, 1)$, and $b_i = \sum_{k=1}^{i} \beta^{i-k} \sum_{l=k+1}^{i} a_l$, we have*

$$\sum_{i=1}^{t} b_i^2 \leq \left(\frac{1}{1-\beta}\right)^2 \left(\frac{\beta}{1-\beta}\right)^2 \sum_{i=2}^{t} a_i^2$$

F.2 PROOF OF THEOREM 1

Proof. We combine the above lemmas to bound

$$\begin{aligned}
\mathbb{E}[f(z_{t+1}) - f(z_1)] &\leq \sum_{i=1}^{6} T_i \\
&\leq \underbrace{G^2 \frac{\beta_1}{1-\beta_1} \mathbb{E}\left[\sum_{i=2}^{t} \left\| \alpha_i V_i^{-1/2} - \alpha_{i-1} V_{i-1}^{-1/2} \right\|_2\right]}_{T_1} \\
&\quad + \underbrace{\left(\frac{\beta_1}{1-\beta_1} - \frac{\beta_{1,t+1}}{1-\beta_{1,t+1}}\right)(G^2 + D^2)}_{T_3} \\
&\quad + \underbrace{\Big(\frac{\beta_1}{1-\beta_1} - \frac{\beta_{1,t+1}}{1-\beta_{1,t+1}}\Big)^2 D^2}_{T_4} \\
&\quad + \underbrace{\Big(\frac{\beta_1}{1-\beta_1}\Big)^2 G^2 \mathbb{E}\left[\sum_{i=2}^{t} \left\| \alpha_i V_i^{-1/2} - \alpha_{i-1} V_{i-1}^{-1/2} \right\|_2^2\right]}_{T_5} \\
&\quad + \underbrace{\mathbb{E}\left[\sum_{i=1}^{t} \frac{3}{2} L \left\| \alpha_i V_i^{-1/2} g_i \right\|^2\right]}_{T_6} \\
&\quad + \underbrace{2G^2 \mathbb{E}\left[\sum_{i=2}^{t} \left\| \alpha_i V_i^{-1/2} - \alpha_{i-1} V_{i-1}^{-1/2} \right\|_2\right] + 2G^2 \mathbb{E}\left[\left\| \alpha_1 V_1^{-1/2} \right\|_2\right]}_{T_2} \\
&\quad \underbrace{- \mathbb{E}\left[\sum_{i=1}^{t} \alpha_i \left\langle \nabla f(x_i), V_i^{-1/2} \nabla f(x_i) \right\rangle\right]}_{T_2} \\
&\quad + \underbrace{L^2 \Big(\frac{\beta_1}{1-\beta_1}\Big)^2 \left(\Big(\frac{1}{1-\beta_1}\Big)^2 \mathbb{E}\left[\sum_{i=1}^{t-1} \left\| \alpha_i V_i^{-1/2} g_i \right\|^2\right]\right.}_{T_2} \\
&\quad \underbrace{\left. + G^2 \Big(\frac{1}{1-\beta_1}\Big)^2 \Big(\frac{\beta_1}{1-\beta_1}\Big)^2 \mathbb{E}\left[\sum_{i=2}^{t-1} \left\| \alpha_i V_i^{-1/2} - \alpha_{i-1} V_{i-1}^{-1/2} \right\|_2^2\right]\right)}_{T_2} \\
&\quad \underbrace{+ \mathbb{E}\left[\frac{1}{2} \sum_{i=1}^{t} \|\alpha_i V_i^{-1/2} g_i\|^2\right]}_{T_2}
\end{aligned}$$

By merging similar terms, we can have

$$
\begin{aligned}
\mathbb{E}[f(z_{t+1}) - f(z_1)] \le& \left(G^2 \frac{\beta_1}{1-\beta_1} + 2G^2\right) \mathbb{E}\left[\sum_{i=2}^{t} \left\| \alpha_i \widehat{V}_i^{-1/2} - \alpha_{i-1}\widehat{V}_{i-1}^{-1/2} \right\|_2\right] \\
&+ \left(\frac{3}{2}L + \frac{1}{2} + L^2 (\frac{\beta_1}{1-\beta_1})^2 (\frac{1}{1-\beta_1})^2\right) \mathbb{E}\left[\sum_{i=1}^{t} \left\| \alpha_i \widehat{V}_i^{-1/2} g_i \right\|^2\right] \\
&+ \left(1 + L^2 (\frac{1}{1-\beta_1})^2 (\frac{\beta_1}{1-\beta_1})^2\right) (\frac{\beta_1}{1-\beta_1})^2 G^2 \mathbb{E}\left[\sum_{i=2}^{t-1} \left\| \alpha_i \widehat{V}_i^{-1/2} - \alpha_{i-1}\widehat{V}_{i-1}^{-1/2} \right\|_2^2\right] \\
&+ \left(\frac{\beta_1}{1-\beta_1} - \frac{\beta_{1,t+1}}{1-\beta_{1,t+1}}\right)(G^2 + D^2) + \left(\frac{\beta_1}{1-\beta_1} - \frac{\beta_{1,t+1}}{1-\beta_{1,t+1}}\right)^2 D^2 + 2G^2 \mathbb{E}\left[\left\| \alpha_1 V_1^{-1/2} \right\|_2\right] \\
&- \mathbb{E}\left[\sum_{i=1}^{t} \alpha_i \left\langle \nabla f(x_i), V_i^{-1/2} \nabla f(x_i) \right\rangle\right]
\end{aligned}
$$

We define constants C_1, C_2, and C_3 as

$$
\begin{aligned}
C_1 &= \frac{3}{2}L + \frac{1}{2} + L^2 (\frac{\beta_1}{1-\beta_1})^2 (\frac{1}{1-\beta_1})^2 \\
C_2 &= G^2 \frac{\beta_1}{1-\beta_1} + 2G^2 \\
C_3 &= \left(1 + L^2 (\frac{1}{1-\beta_1})^2 (\frac{\beta_1}{1-\beta_1})^2\right) (\frac{\beta_1}{1-\beta_1})^2 G^2
\end{aligned}
$$

By rearranging terms, we obtain

$$
\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{t} \alpha_i \left\langle \nabla f(x_i), V_i^{-1/2} \nabla f(x_i) \right\rangle\right] \le& \mathbb{E}\left[\sum_{i=1}^{t} C_1 \left\| \alpha_i \widehat{V}_i^{-1/2} g_i \right\|^2 + C_2 \sum_{i=2}^{t} \left\| \alpha_i \widehat{V}_i^{-1/2} - \alpha_{i-1}\widehat{V}_{i-1}^{-1/2} \right\|_2\right. \\
&\left. + C_3 \sum_{i=2}^{t-1} \left\| \alpha_i \widehat{V}_i^{-1/2} - \alpha_{i-1}\widehat{V}_{i-1}^{-1/2} \right\|_2^2\right] \\
&+ \left(\frac{\beta_1}{1-\beta_1} - \frac{\beta_{1,t+1}}{1-\beta_{1,t+1}}\right)(G^2 + D^2) + \left(\frac{\beta_1}{1-\beta_1} - \frac{\beta_{1,t+1}}{1-\beta_{1,t+1}}\right)^2 D^2 \\
&+ 2G^2 \mathbb{E}\left[\left\| \alpha_1 V_1^{-1/2} \right\|_2\right] \\
\le& \mathbb{E}\left[\sum_{i=1}^{t} C_1 \left\| \alpha_i \widehat{V}_i^{-1/2} g_i \right\|^2 + C_2 \sum_{i=2}^{t} \left\| \alpha_i \widehat{V}_i^{-1/2} - \alpha_{i-1}\widehat{V}_{i-1}^{-1/2} \right\|_2\right. \\
&\left. + C_3 \sum_{i=2}^{t-1} \left\| \alpha_i \widehat{V}_i^{-1/2} - \alpha_{i-1}\widehat{V}_{i-1}^{-1/2} \right\|_2^2\right] \\
&+ \left(\frac{\beta_1}{1-\beta_1}\right)(G^2 + D^2) + \left(\frac{\beta_1}{1-\beta_1}\right)^2 D^2 + 2G^2 \mathbb{E}\left[\left\| \alpha_1 V_1^{-1/2} \right\|_2\right]
\end{aligned}
$$

Finally, we can get

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} \alpha_i \left\langle \nabla f(x_t), \widehat{V}_t^{-1/2} \nabla f(x_t) \right\rangle\right] \\
\le\ & \mathbb{E}\left[C_1 \underbrace{\sum_{t=1}^{T} \left\| \alpha_t \widehat{V}_t^{-1/2} g_i \right\|^2}_{\text{Term A}} + C_2 \underbrace{\sum_{t=2}^{T} \left\| \alpha_t \widehat{V}_t^{-1/2} - \alpha_{t-1}\widehat{V}_{t-1}^{-1/2} \right\|_2}_{\text{Term B}} + C_3 \sum_{t=2}^{T-1} \left\| \alpha_t \widehat{V}_t^{-1/2} - \alpha_{t-1}\widehat{V}_{t-1}^{-1/2} \right\|_2^2\right] + C_4
\end{aligned}
$$

with constants

$$
C_4 = \left(\frac{\beta_1}{1-\beta_1}\right)(G^2 + D^2) + \left(\frac{\beta_1}{1-\beta_1}\right)^2 D^2 + 2G^2 \mathbb{E}\left[\left\| \alpha_1 V_1^{-1/2} \right\|_2\right]
$$

with almost same constant for the diagonal version. Lastly, we have $\gamma_t = \lambda_{\min}(\alpha_t \widehat{V}_t^{-1/2})$, so

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T} \alpha_t \left\langle \nabla f(x_t), \widehat{V}_t^{-1/2} \nabla f(x_t) \right\rangle\right] &\geq \mathbb{E}\left[\sum_{t=1}^{T} \gamma_t \Big\| \nabla f(x_t) \Big\|^2\right] \\ &\geq \min_{t\in[T]} \mathbb{E}\big[\|\nabla f(x_t)\|^2\big] \sum_{t=1}^{T} \gamma_t\end{aligned}$$

Therefore, we finally have

$$\min_{t\in[T]} \big[\|\nabla f(x_t)\|^2\big] \leq \frac{\mathbb{E}\left[C_1 \overbrace{\sum_{t=1}^{T} \Big\|\alpha_t \widehat{V}_t^{-1/2} g_t\Big\|^2}^{\text{Term A}} + C_2 \overbrace{\sum_{t=2}^{T} Q_t}^{\text{Term B}} + C_3 \sum_{t=2}^{T-1} Q_t^2\right] + C_4}{\sum_{t=1}^{T} \gamma_t} \triangleq \frac{s_1(T)}{s_2(T)}$$

□

F.3 Proof of Theorem 2

For generalization error bounds, we refer the following references (Hardt et al., 2015; Zheng & Kwok, 2019). Since we have bounded gradient $\|g_t\|_2 \leq G$ and $\|\nabla f(x)\|_2 \leq G$, we also have G-Lipschitz condition. Therefore, we obtain the following relation

$$\begin{aligned}\sup_z \mathbb{E}_A\big[f(A(S);z) - f(A(S');z)\big] &\leq G\mathbb{E}_A\big[\|A(S) - A(S')\|_2\big] \\ &= G\mathbb{E}_A\big[\|\theta - \theta'\|_2\big]\end{aligned}$$

Therefore, we only have to bound the term $\Delta_t := \|\theta - \theta'\|_2$. From now, we denote $\theta := A(S)$ and $\theta' := A(S')$. We assume $\alpha_t = \alpha$ and $\beta_{1,t} = 0$ for all $t \in [T]$.

$$\begin{aligned}\theta_{T+1} &= \theta_T - \alpha_T(\widehat{V}_T^{1/2} + \delta I)^{-1} m_T \\ &= \cdots \\ &= \theta_1 - \sum_{t=1}^{T} \alpha_t (\widehat{V}_t^{1/2} + \delta I)^{-1} g_t \\ &= \theta_1 - \sum_{t=1}^{T} \alpha_t (\widehat{V}_t^{1/2} + \delta I)^{-1} \nabla f(\theta_t; z_{i_t})\end{aligned}$$

where z_{i_k} is the selected example at iteration k. Then, we can bound

$$\begin{aligned}\mathbb{E}[\Delta_{T+1}] &= \mathbb{E}\big[\|\theta_{T+1} - \theta'_{T+1}\|_2\big] \\ &= \mathbb{E}\left[\left\|\theta_1 - \sum_{t=1}^{T} \alpha_t (\widehat{V}_t^{1/2} + \delta I)^{-1} \nabla f(\theta_t; z_{i_t}) - \theta'_1 + \sum_{t=1}^{T} \alpha_t (\widehat{V}'^{1/2}_t + \delta I)^{-1} \nabla f(\theta'_t; z'_{i_t})\right\|_2\right] \\ &\leq \mathbb{E}\big[\|\theta_1 - \theta'_1\|_2\big] + \sum_{t=1}^{T} \alpha_t \mathbb{E}\left[\left\|(\widehat{V}_t^{1/2} + \delta I)^{-1} \nabla f(\theta_t; z_{i_t}) - (\widehat{V}'^{1/2}_t + \delta I)^{-1} \nabla f(\theta'_t; z'_{i_t})\right\|_2\right] \\ &= \sum_{t=1}^{T} \alpha_t \mathbb{E}\left[\left\|(\widehat{V}_t^{1/2} + \delta I)^{-1} \nabla f(\theta_t; z_{i_t}) - (\widehat{V}'^{1/2}_t + \delta I)^{-1} \nabla f(\theta'_t; z'_{i_t})\right\|_2\right]\end{aligned}$$

The probability of $z_{i_k} = z'_{i_k}$ is $1 - 1/n$. Then,

$$\begin{aligned}
&\mathbb{E}\left[\left\|(\widehat{V}_t^{1/2} + \delta I)^{-1}\nabla f(\theta_t; z_{i_t}) - (\widehat{V}_t'^{1/2} + \delta I)^{-1}\nabla f(\theta_t'; z_{i_t}')\right\|_2\right] \\
\leq\ & \frac{1}{n}\mathbb{E}\big[\|(\widehat{V}_t^{1/2} + \delta I)^{-1}\nabla f(\theta_t; z_{i_t})\|_2\big] + \frac{1}{n}\mathbb{E}\big[\|(\widehat{V}_t'^{1/2} + \delta I)^{-1}\nabla f(\theta_t'; z_{i_t}')\|_2\big] \\
&+ \Big(1 - \frac{1}{n}\Big)\mathbb{E}\big[\|(\widehat{V}_t^{1/2} + \delta)^{-1}\nabla f(\theta_t; z_{i_t}) - (\widehat{V}_t'^{1/2} + \delta I)^{-1}\nabla f(\theta_t'; z_{i_t})\|_2\big] \\
\leq\ & \frac{1}{n}\underbrace{\mathbb{E}\big[\|(\widehat{V}_t^{1/2} + \delta I)^{-1}\nabla f(\theta_t; z_{i_t})\|_2\big]}_{T_1} + \frac{1}{n}\underbrace{\mathbb{E}\big[\|(\widehat{V}_t'^{1/2} + \delta I)^{-1}\nabla f(\theta_t'; z_{i_t}')\|_2\big]}_{T_2} \\
&\quad + \Big(1 - \frac{1}{n}\Big)\underbrace{\mathbb{E}\big[\|(\widehat{V}_t^{1/2} + \delta I)^{-1}\nabla f(\theta_t; z_{i_t}) - (\widehat{V}_t'^{1/2} + \delta I)^{-1}\nabla f(\theta_t; z_{i_t})\|_2\big]}_{T_3} \\
&\quad + \Big(1 - \frac{1}{n}\Big)\underbrace{\mathbb{E}\big[\|(\widehat{V}_t'^{1/2} + \delta I)^{-1}\nabla f(\theta_t; z_{i_t}) - (\widehat{V}_t'^{1/2} + \delta I)^{-1}\nabla f(\theta_t'; z_{i_t})\|_2\big]}_{T_4}
\end{aligned}$$

Let t_0 denote the time when $\widehat{V}_t$ and $\widehat{V}_t'$ becomes full-rank. Then, we can bound T_1 as

$$\begin{aligned}
&\sum_{t=1}^{T}\alpha_t\mathbb{E}\big[\|(\widehat{V}_t^{1/2} + \delta I)^{-1}\nabla f(\theta_t; z_{i_t})\|_2\big] \\
&\leq \sqrt{T}\sqrt{\sum_{t=1}^{T}\alpha_t^2\mathbb{E}\big[\|(\widehat{V}_t^{1/2} + \delta I)^{-1}\nabla f(\theta_t; z_{i_t})\|_2^2\big]} \\
&= \sqrt{T}\sqrt{\sum_{t=1}^{t_0}\alpha_t^2\mathbb{E}\big[\|(\widehat{V}_t^{1/2} + \delta I)^{-1}\nabla f(\theta_t; z_{i_t})\|_2^2\big] + \sum_{t=t_0+1}^{T}\alpha_t^2\mathbb{E}\big[\mathrm{Tr}\big((\widehat{V}_t^{1/2} + \delta I)^{-2}\nabla f(\theta_t; z_{i_t})\nabla f(\theta_t; z_{i_t})^T\big)\big]} \\
&\leq \sqrt{T}\sqrt{\sum_{t=1}^{t_0}\alpha_t^2\mathbb{E}\big[\|(\widehat{V}_t^{1/2} + \delta I)^{-1}\nabla f(\theta_t; z_{i_t})\|_2^2\big] + \sum_{t=t_0+1}^{T}\alpha_t^2\mathbb{E}\big[\mathrm{Tr}\big(\widehat{V}_t^{-1}\nabla f(\theta_t; z_{i_t})\nabla f(\theta_t; z_{i_t})\big)\big]} \\
&\leq \sqrt{T}\sqrt{\sum_{t=1}^{t_0}\alpha^2\mathbb{E}\big[\|(\widehat{V}_t^{1/2} + \delta I)^{-1}\nabla f(\theta_t; z_{i_t})\|_2^2\big] + \frac{\alpha^2}{1-\beta_2}\mathbb{E}\big[\log\det(\widehat{V}_T) - \log\det(\widehat{V}_{t_0}) + d(T - t_0)\log\frac{1}{\beta_2}\big]} \\
&= \frac{\alpha\sqrt{T}}{\sqrt{1-\beta_2}}\sqrt{(1-\beta_2)\sum_{t=1}^{t_0}\mathbb{E}\big[\|(\widehat{V}_t^{1/2} + \delta I)^{-1}\nabla f(\theta_t; z_{i_t})\|_2^2\big] + \mathbb{E}\big[\log\det(\widehat{V}_T) - \log\det(\widehat{V}_{t_0})\big] + d(T - t_0)\log\frac{1}{\beta_2}} \\
&\leq \frac{\alpha\sqrt{T}}{\sqrt{1-\beta_2}}\sqrt{(1-\beta_2)\sum_{t=1}^{t_0}\mathbb{E}\big[\|(\delta I)^{-1}\nabla f(\theta_t; z_{i_t})\|_2^2\big] + \mathbb{E}\big[\log\det(\widehat{V}_T) - \log\det(\widehat{V}_{t_0})\big] + d(T - t_0)\log\frac{1}{\beta_2}} \\
&\leq \frac{\alpha\sqrt{T}}{\sqrt{1-\beta_2}}\sqrt{\frac{t_0(1-\beta_2)G^2}{\delta^2} + \mathbb{E}\left[\log\frac{|\widehat{V}_T|}{|\widehat{V}_{t_0}|}\right] + d(T - t_0)\log\frac{1}{\beta_2}}
\end{aligned}$$

In the same way, we can bound T_2 as

$$\sum_{t=1}^{T}\alpha_t\mathbb{E}\big[\|(\widehat{V}_t'^{1/2} + \delta I)^{-1}\nabla f(\theta_t'; z_{i_t}')\|_2\big] \leq \frac{\alpha\sqrt{T}}{\sqrt{1-\beta_2}}\sqrt{\frac{t_0(1-\beta_2)G^2}{\delta^2} + \mathbb{E}\left[\log\frac{|\widehat{V}_T'|}{|\widehat{V}_{t_0}'|}\right] + d(T - t_0)\log\frac{1}{\beta_2}}$$

For notational convenience, we set the function g as

$$g(\widehat{V}_T) = \frac{\alpha\sqrt{T}}{\sqrt{1-\beta_2}}\sqrt{\frac{t_0(1-\beta_2)G^2}{\delta^2} + \mathbb{E}\left[\log\frac{|\widehat{V}_T|}{|\widehat{V}_{t_0}|}\right] + d(T - t_0)\log\frac{1}{\beta_2}}$$

Now, we can easily bound T_3 and T_4 as

$$\begin{aligned}
\mathbb{E}\big[\|(\widehat{V}_t^{1/2} + \delta I)^{-1}\nabla f(\theta_t; z_{i_t}) - (\widehat{V}_t'^{1/2} + \delta I)^{-1}\nabla f(\theta_t; z_{i_t})\|_2\big] &\leq G\mathbb{E}\big[\big\|(\widehat{V}_t^{1/2} + \delta I)^{-1} - (\widehat{V}_t'^{1/2} + \delta I)^{-1}\big\|_2\big] \\
&\leq G\mathbb{E}\big[\big\|(\widehat{V}_t^{1/2} + \delta I)^{-1}\big\|_2 + \big\|(\widehat{V}_t'^{1/2} + \delta I)^{-1}\big\|_2\big]
\end{aligned}$$

and

$$\mathbb{E}\big[\|(\widehat{V}_t'^{1/2}+\delta I)^{-1}\nabla f(\theta_t;z_{i_t})-(\widehat{V}_t'^{1/2}+\delta I)^{-1}\nabla f(\theta_t';z_{i_t})\|_2\big]\le L\mathbb{E}\big[|\!|\!|(\widehat{V}_t'^{1/2}+\delta I)^{-1}|\!|\!|_2\Delta_t\big]$$

from $\|Ax\|_2 \le |\!|\!|A|\!|\!|_2\|x\|_2$ and Lipschitz continuous gradients. Combining all the terms, we finally have

$$\mathbb{E}\big[\Delta_{T+1}\big]\le\frac{\alpha\sqrt{T}}{n\sqrt{1-\beta_2}}\left[\sqrt{g(\widehat{V}_T)}+\sqrt{g(\widehat{V}_T')}\right]+\alpha\Big(1-\frac{1}{n}\Big)J_T$$

where

$$g(\widehat{V}_T)=\frac{t_0(1-\beta_2)G^2}{\delta^2}+\mathbb{E}\big[\underbrace{\log|\widehat{V}_T|-\log|\widehat{V}_{t_0}|}_{\text{Term C}}\big]+d(T-t_0)\log\frac{1}{\beta_2}$$

and

$$J_T=G\sum_{t=1}^{T}\mathbb{E}\big[\underbrace{|\!|\!|(\widehat{V}_t^{1/2}+\delta I)^{-1}|\!|\!|_2}_{\text{Term D}}+\underbrace{|\!|\!|(\widehat{V}_t'^{1/2}+\delta I)^{-1}|\!|\!|_2}_{\text{Term D}}\big]+L\sum_{t=1}^{T}\mathbb{E}\big[\underbrace{|\!|\!|(\widehat{V}_t'^{1/2}+\delta I)^{-1}|\!|\!|_2}_{\text{Term D}}\Delta_t\big]$$

F.4 GENERALIZATION BOUNDS FOR ADAGRAD

The main difference with EMA-BASED algorithms is the way of bounding T_1. For ADAGRAD, we set $\alpha_t=\alpha/\sqrt{t}$ and $\widehat{V}_t=\frac{1}{t}\sum_{i=1}^{t}g_ig_i^T=:\frac{1}{t}G_t$ as in Corollary 1.

Let t_0 denote the time when $\widehat{V}_t$ and $\widehat{V}_t'$ becomes full-rank. Then, we can bound T_1 for ADAGRAD as

$$\begin{aligned}
&\sum_{t=1}^{T}\alpha_t\mathbb{E}\big[\|(\widehat{V}_t^{1/2}+\delta I)^{-1}\nabla f(\theta_t;z_{i_t})\|_2\big]\\
&\le\sqrt{T}\sqrt{\sum_{t=1}^{T}\alpha_t^2\mathbb{E}\big[\|(\widehat{V}_t^{1/2}+\delta I)^{-1}\nabla f(\theta_t;z_{i_t})\|_2^2\big]}\\
&=\sqrt{T}\sqrt{\sum_{t=1}^{t_0}\alpha_t^2\mathbb{E}\big[\|(\widehat{V}_t^{1/2}+\delta I)^{-1}\nabla f(\theta_t;z_{i_t})\|_2^2\big]+\sum_{t=t_0+1}^{T}\alpha_t^2\mathbb{E}\big[\mathrm{Tr}\big((\widehat{V}_t^{1/2}+\delta I)^{-2}\nabla f(\theta_t;z_{i_t})\nabla f(\theta_t;z_{i_t})^T\big)\big]}\\
&\le\sqrt{T}\sqrt{\sum_{t=1}^{t_0}\alpha_t^2\mathbb{E}\big[\|(\widehat{V}_t^{1/2}+\delta I)^{-1}\nabla f(\theta_t;z_{i_t})\|_2^2\big]+\sum_{t=t_0+1}^{T}\alpha_t^2\mathbb{E}\big[\mathrm{Tr}\big(\widehat{V}_t^{-1}\nabla f(\theta_t;z_{i_t})\nabla f(\theta_t;z_{i_t})\big)\big]}\\
&\le\sqrt{T}\sqrt{\sum_{t=1}^{t_0}\alpha^2\mathbb{E}\big[\|(\widehat{V}_t^{1/2}+\delta I)^{-1}\nabla f(\theta_t;z_{i_t})\|_2^2\big]+\alpha^2\mathbb{E}\big[\log|\widehat{V}_T|-\log|\widehat{V}_{t_0}|+d\log\frac{T}{t_0}\big]}\\
&\le\alpha\sqrt{T}\sqrt{\sum_{t=1}^{t_0}\mathbb{E}\big[\|(\delta I)^{-1}\nabla f(\theta_t;z_{i_t})\|_2^2\big]+\mathbb{E}\big[\log|\widehat{V}_T|-\log|\widehat{V}_{t_0}|+d\log\frac{T}{t_0}\big]}\\
&\le\alpha\sqrt{T}\sqrt{\frac{t_0G^2}{\delta^2}+\mathbb{E}\big[\log|\widehat{V}_T|-\log|\widehat{V}_{t_0}|+d\log\frac{T}{t_0}\big]}
\end{aligned}$$

We can similarly bound the term T_2 only replacing $\widehat{V}_t$ with $\widehat{V}_t'$. Also, the remaining terms T_3 and T_4 can be bound in the same way as in section F.3. Therefore, we have

$$\mathbb{E}\big[\Delta_{T+1}\big]\le\frac{\alpha\sqrt{T}}{n}\left[\sqrt{g(\widehat{V}_T)}+\sqrt{g(\widehat{V}_T')}\right]+\alpha\Big(1-\frac{1}{n}\Big)J_T$$

where

$$g(\widehat{V}_T)=\frac{t_0G^2}{\delta^2}+\mathbb{E}\big[\underbrace{\log|\widehat{V}_T|-\log|\widehat{V}_{t_0}|}_{\text{Term C}}\big]+d\log\frac{T}{t_0}$$

and

$$J_T=G\sum_{t=1}^{T}\mathbb{E}\big[\underbrace{|\!|\!|(\widehat{V}_t^{1/2}+\delta I)^{-1}|\!|\!|_2}_{\text{Term D}}+\underbrace{|\!|\!|(\widehat{V}_t'^{1/2}+\delta I)^{-1}|\!|\!|_2}_{\text{Term D}}\big]+L\sum_{t=1}^{T}\mathbb{E}\big[\underbrace{|\!|\!|(\widehat{V}_t'^{1/2}+\delta I)^{-1}|\!|\!|_2}_{\text{Term D}}\Delta_t\big]$$