Towards Better Understanding of Adaptive Gradient Algorithms in Generative Adversarial Nets

Anonymous authors
Paper under double-blind review

ABSTRACT

Adaptive gradient algorithms perform gradient-based updates using the history of gradients and are ubiquitous in training deep neural networks. While adaptive gradient methods theory is well understood for minimization problems, the underlying factors driving their empirical success in min-max problems such as GANs remain unclear. In this paper, we aim at bridging this gap from both theoretical and empirical perspectives. Theoretically, we develop an algorithm (Optimistic Stochastic Gradient, OSG) for solving a class of non-convex non-concave min-max problem and establish $O(\epsilon^{-4})$ complexity for finding $\epsilon$-first-order stationary point, in which only one stochastic first-order oracle is invoked in each iteration. An adaptive variant of the proposed algorithm (Optimistic Adagrad, OAdaGrad) is also analyzed, revealing an improved adaptive complexity $\tilde{O}(\epsilon^{-\frac{4}{1+\alpha}})$, where $\alpha$ characterizes the growth rate of the cumulative stochastic gradient and $0 \leq \alpha \leq 1/2$. To the best of our knowledge, this is the first work for establishing adaptive complexity in non-convex non-concave min-max optimization. Empirically, our experiments show that indeed adaptive gradient algorithms outperform their non-adaptive counterparts in GAN training. Moreover, this observation can be explained by the slow growth rate of the cumulative stochastic gradient, as observed empirically.

1 INTRODUCTION

Adaptive gradient algorithms (Duchi et al., 2011; Tieleman & Hinton, 2012; Kingma & Ba, 2014; Reddi et al., 2019) are very popular in training deep neural networks due to their computational efficiency and minimal need for hyper-parameter tuning (Kingma & Ba, 2014). For example, Adagrad (Duchi et al., 2011) automatically adjusts the learning rate for each dimension of the model parameter according to the information of history gradients, while its computational cost is almost the same as Stochastic Gradient Descent (SGD). However, in supervised deep learning (for example, image classification tasks using a deep convolutional neural network), there is not enough evidence showing that adaptive gradient methods converge faster than its non-adaptive counterpart (i.e., SGD) on benchmark datasets. For example, it is argued in (Wilson et al., 2017) that adaptive gradient methods often find a solution with worse performance than SGD. Specifically, Wilson et al. (2017) observed that Adagrad has slower convergence than SGD in terms of both training and testing error, while using VGG (Simonyan & Zisserman, 2014) on CIFAR10 data.

GANs (Goodfellow et al., 2014) are a popular class of generative models. In a nutshell, they consist of a generator and a discriminator, both of which are defined by deep neural networks. The generator and the discriminator are trained under an adversarial cost, corresponding to a non-convex non-concave min-max problem. GANs are known to be notoriously difficult to train. In practice, Adam (Kingma & Ba, 2014) is the defacto optimizer used for GAN training. The common optimization strategy is to alternatively update the discriminator and the generator (Arjovsky et al., 2017; Gulrajani et al., 2017). Using Adam is important in GAN training, since replacing it with non-adaptive methods (e.g. SGD) would significantly deteriorate the performance. This paper studies and attempts to answer the following question:

1Here $\tilde{O}(\cdot)$ compresses a logarithmic factor of $\epsilon$. 
Why do adaptive gradient methods outperform their non-adaptive counterparts in GAN training?

We design two algorithms with state-of-the-art complexities for solving a class of non-convex non-concave min-max problems. Namely, our algorithms are: OSG (Optimistic Stochastic Gradient) in the non-adaptive family, and OAdagrad (Optimistic Adagrad) in the adaptive case. We further prove that the convergence rate of OAdagrad to an $\epsilon$-first-order stationary point depends on the growth rate of the cumulative stochastic gradient. In our experiments, we observed an interesting phenomenon while using adaptive gradient methods for training GANs: the cumulative stochastic gradient grows at a slow rate. This observation is in line with the prediction of our theory suggesting improved convergence rate for OAdagrad in GAN training, when the growth rate of the cumulative stochastic gradient is slow.

Since GAN is a min-max optimization problem in nature, our problem of interest is to solve the following stochastic optimization problem:

\[
\min_{\mathbf{u} \in \mathcal{U}} \max_{\mathbf{v} \in \mathcal{V}} \mathbb{E}_{\xi \sim \mathcal{D}} \left[ f(\mathbf{u}, \mathbf{v}; \xi) \right],
\]

where $\mathcal{U}$, $\mathcal{V}$ are closed and convex sets, $F(\mathbf{u}, \mathbf{v})$ is possibly non-convex in $\mathbf{u}$ and non-concave in $\mathbf{v}$. $\xi$ is a random variable following an unknown distribution $\mathcal{D}$. In GAN training, $\mathbf{u}$, $\mathbf{v}$ represent the parameters of generator and discriminator respectively.

The ideal goal for solving (1) is to find a saddle point $(\mathbf{u}_*, \mathbf{v}_*) \in \mathcal{U} \times \mathcal{V}$ such that $F(\mathbf{u}_*, \mathbf{v}) \leq F(\mathbf{u}, \mathbf{v}_*) \leq F(\mathbf{u}_*, \mathbf{v}_*)$ for $\forall \mathbf{u} \in \mathcal{U}, \forall \mathbf{v} \in \mathcal{V}$.

To achieve this goal, the typical assumption usually made is that the objective function is convex-concave. When $F(\mathbf{u}, \mathbf{v})$ is convex in $\mathbf{u}$ and concave in $\mathbf{v}$, non-asymptotic guarantee in terms of the duality gap is well established by a series of work (Nemirovski & Yudin, 1978; Nemirovski, 2004; Nesterov, 2007; Nemirovski et al., 2009; Juditsky et al., 2011). However, when $F(\mathbf{u}, \mathbf{v})$ is non-convex in $\mathbf{u}$ and non-concave in $\mathbf{v}$, finding the saddle point is NP-hard in general. Instead, we focus on finding the first-order stationary point provided that the objective function is smooth. I.e. we aim to find $(\mathbf{u}, \mathbf{v}) \in \mathcal{U} \times \mathcal{V}$ such that $\nabla_\mathbf{u} F(\mathbf{u}, \mathbf{v}) = 0, \nabla_\mathbf{v} F(\mathbf{u}, \mathbf{v}) = 0$. Note that this is a necessary condition for finding the (local) saddle point.

**Related Work.** Several works designed iterative first-order deterministic (Dang & Lan, 2015) and stochastic (Iusem et al., 2017; Lin et al., 2018) algorithms for achieving the $\epsilon$-first-order stationary point with non-asymptotic guarantee. The goal is to find $\mathbf{x}$ such that $\|T(\mathbf{x})\| \leq \epsilon$ or $\mathbb{E}[\|T(\mathbf{x})\|] \leq \epsilon$, where the first-order oracle is defined as $T(\mathbf{x}) = \left[ \nabla_\mathbf{u} F(\mathbf{u}, \mathbf{v}), -\nabla_\mathbf{v} F(\mathbf{u}, \mathbf{v}) \right]^T$ with $\mathbf{x} = (\mathbf{u}, \mathbf{v})$ and the first-order stochastic oracle is the noisy observation of $T$, i.e. $T(\mathbf{x}; \xi) = \left[ \nabla_\mathbf{u} F(\mathbf{u}, \mathbf{v}; \xi), -\nabla_\mathbf{v} F(\mathbf{u}, \mathbf{v}; \xi) \right]^T$. For instance, Dang & Lan (2015) focuses on the deterministic setting. On the other hand, Iusem et al. (2017) develops a stochastic extra-gradient algorithm that enjoys $O(\epsilon^{-4})$ iteration complexity. The extra-gradient method requires two stochastic first-order oracles in one iteration, which can be computationally expensive in deep learning applications such as GANs. The inexact proximal point method developed in Lin et al. (2018) has iteration complexity $O(\epsilon^{-6})$ for finding an $\epsilon$-first-order stationary point $^2$.

To avoid the cost of an additional oracle call in extragradient step, several studies (Chiang et al., 2012; Rakhlin & Sridharan, 2013; Daskalakis et al., 2017; Gidel et al., 2018; Xu et al., 2019) proposed single-call variants of the extragradient algorithm. Some of them focus on the convex setting (e.g. Chiang et al., 2012; Rakhlin & Sridharan, 2013), while others focus on the non-convex setting (Xu et al., 2019). The closest to our work is the work by Daskalakis et al. (2017; Gidel et al., 2018), where the min-max setting and GAN training are considered. However, the convergence of those algorithms is only shown for a class of bilinear problems in (Daskalakis et al., 2017) and for monotone variational inequalities in (Gidel et al., 2018). Hence a big gap remains between the specific settings studied in Daskalakis et al. (2017; Gidel et al., 2018) and more general non-convex non-concave min-max problems. Table 1 provides a complete overview of our results and existing results. It is hard to give justice to the large body of work on min-max optimization, so we refer the interested reader to Appendix B that gives a comprehensive survey of related previous methods that are not covered in this Table.

---

$^2$The result in Lin et al. (2018) assumes the first-order oracle $T$ is a weakly-monotone operator, which is milder than the Lipschitz-continuity assumption as assumed in Iusem et al. (2017). However, simply applying the Lipschitz-continuity condition in their proof does not change their iteration complexity.
Our main goal is to design stochastic first-order algorithms with low iteration complexity, low per-iteration cost and suitable for a general class of non-convex non-concave min-max problems. The main tool we use in our analysis is variational inequality.

Let $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ be an operator and $X \subset \mathbb{R}^d$ is a closed convex set. The Stampacchia Variational Inequality (SVI) problem (Hartman & Stampacchia, 1966) is defined by the operator $T$ and $X$ and denoted by SVI$(T, X)$. It consists of finding $x_\ast \in X$ such that $(T(x_\ast), x - x_\ast) \geq 0$ for $\forall x \in X$. A similar one is Minty Variational Inequality (MVI) problem (Minty et al., 1962) denoted by MVI$(T, X)$, which consists of finding $x_\ast$ such that $(T(x), x - x_\ast) \geq 0$ for $\forall x \in X$. Min-max optimization is closely related to variational inequalities. The corresponding SVI and MVI for the min-max problem are defined through $T(x) = [\nabla_u F(u, v), -\nabla_v F(u, v)]^\top$ with $x = (u, v)$.

Our main contributions are summarized as follows:

- By assuming the Lipschitz continuity of the operator $T$ and the existence of a solution for the variational inequality MVI$(T, X)$, we propose an algorithm named Optimistic Stochastic Gradient (OSG), and show that it enjoys better adaptive complexity $O\left(e^{-\frac{\alpha^2}{\mu}}\right)$, where $\alpha$ characterizes the growth rate of cumulative stochastic gradient and $0 \leq \alpha \leq 1/2$.
- Under the same conditions, we design an adaptive gradient algorithm named Optimistic Adagrad (OAdagrad), and show that it enjoys better adaptive complexity $O\left(e^{-\frac{\alpha^2}{\mu}}\right)$.
- We demonstrate the effectiveness of our algorithms in GAN training on CIFAR10 data. Our experiments suggest that using large minibatch helps optimization, which is consistent with the proposed theory and also is in line with the recent advances in large-scale GAN training (Brock et al., 2018).

Table 1: Summary of different algorithms with IC (Iteration Complexity), PC (Per-iteration Complexity) to find $\epsilon$-SP ($\epsilon$-first-order Stationary Point), $\epsilon$-DG ($\epsilon$-Duality Gap, i.e. a point $(\hat{u}, \hat{v})$ such that $\max_v F(\hat{u}, v) - \min_u F(u, \hat{v}) \leq \epsilon$, or $\epsilon$-optim ($\epsilon$-close to the set of optimal solution). $T_\gamma$ stands for the time complexity for invoking one stochastic first-order oracle.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Assumption</th>
<th>Setting</th>
<th>IC</th>
<th>PC</th>
<th>Guarantee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extragradient (Iusem et al., 2017)</td>
<td>pseudo-monotonicity</td>
<td>stochastic</td>
<td>$O(\epsilon^{-1})$</td>
<td>$2T_\gamma$</td>
<td>$\epsilon$-SP</td>
</tr>
<tr>
<td>OMD (Daskalakis et al., 2017)</td>
<td>bilinear</td>
<td>deterministic</td>
<td>N/A</td>
<td>$T_\gamma$</td>
<td>asymptotic</td>
</tr>
<tr>
<td>AvgPastExtraSGD (Gidel et al., 2018)</td>
<td>monotonicity</td>
<td>stochastic</td>
<td>$O(\epsilon^{-2})$</td>
<td>$T_\gamma$</td>
<td>$\epsilon$-DG</td>
</tr>
<tr>
<td>OMD (Mertikopoulos et al., 2018)</td>
<td>coherence</td>
<td>stochastic</td>
<td>N/A</td>
<td>$2T_\gamma$</td>
<td>asymptotic</td>
</tr>
<tr>
<td>IPP (Lin et al., 2018)</td>
<td>MVI has solution</td>
<td>stochastic</td>
<td>$O(\epsilon^{-6})$</td>
<td>$T_\gamma$</td>
<td>$\epsilon$-SP</td>
</tr>
<tr>
<td>Alternating Gradient (Gidel et al., 2019)</td>
<td>bilinear form</td>
<td>deterministic</td>
<td>$O(\log(1/\epsilon))$</td>
<td>$T_\gamma$</td>
<td>$\epsilon$-optim</td>
</tr>
<tr>
<td>SVRE (Chavdarova et al., 2019)</td>
<td>strong-monotonicity finite sum</td>
<td>stochastic finite sum</td>
<td>$O(\log(1/\epsilon) + n + \frac{\lambda}{\mu}T_\gamma)$</td>
<td>$T_\gamma$</td>
<td>$\epsilon$-optim</td>
</tr>
<tr>
<td>Extragradient (Azizian et al., 2019)</td>
<td>strong-monotonicity</td>
<td>deterministic</td>
<td>$O(\log(1/\epsilon))$</td>
<td>$2T_\gamma$</td>
<td>$\epsilon$-optim</td>
</tr>
<tr>
<td>OSG (this work)</td>
<td>MVI has solution</td>
<td>stochastic</td>
<td>$O(\epsilon^{-4})$</td>
<td>$T_\gamma$</td>
<td>$\epsilon$-SP</td>
</tr>
<tr>
<td>OAdagrad (this work)</td>
<td>MVI has solution</td>
<td>stochastic</td>
<td>$O(\epsilon^{-4})$</td>
<td>$T_\gamma$</td>
<td>$\epsilon$-SP</td>
</tr>
</tbody>
</table>

1Here the bilinear game is defined as $\min_u \in \mathbb{R}^p \max_v \in \mathbb{R}^q u^\top Av + u^\top a + b v^\top$, where the smallest singular value of $A \in \mathbb{R}^{p \times q}$ is positive, $a \in \mathbb{R}^{p \times 1}$, $b \in \mathbb{R}^{1 \times q}$.

4Here $n, L, \mu$ denote the number of components in the finite sum structure, Lipschitz constant and strong-monotonicity parameter of the operator of variational inequality respectively.
cumulative stochastic gradient grows in a slow rate. We also show that OAdagrad outperforms Simultaneous Adam in sample quality in ImageNet generation using self-attention GANs (Zhang et al., 2018). This confirms the superiority of OAdagrad in min-max optimization.

2 Preliminaries and Notations

In this section, we fix some notations and give formal definitions of variational inequalities, and their relationship to the min-max problem (1).

Notations. Let $\mathcal{X} \subset \mathbb{R}^d$ be a closed convex set, and $\| \cdot \|$ the euclidean norm. We note $\Pi_\mathcal{X}$ the projection operator, i.e. $\Pi_\mathcal{X}(y) = \arg \min_{x \in \mathcal{X}} \| y - x \|^2$. Define $T(x) = [\nabla_u F(u, v), -\nabla_v F(u, v)]^\top$ with $x = (u, v)$ in problem (1). At every point $x \in \mathcal{X}$, we don’t have access to $T(x)$ and have only access to a noisy observations of $T(x)$. That is, $T(x; \xi)$, where $\xi$ is a random variable with distribution $\mathcal{D}$. For the ease of presentation, we use the terms stochastic gradient and stochastic first-order oracle interchangeably to stand for $T(x; \xi)$ in the min-max setting.

Definition 1 (Monotonicity). An operator $T$ is monotone if $\langle T(x) - T(y), x - y \rangle \geq 0$ for $\forall x, y \in \mathcal{X}$. An operator $T$ is pseudo-monotone if $\langle T(x), y - x \rangle \geq 0 \Rightarrow \langle T(x), y - x \rangle \geq 0$ for $\forall x, y \in \mathcal{X}$. An operator $T$ is $\gamma$-strongly-monotone if $\langle T(x) - T(y), x - y \rangle \geq \gamma^2 \| x - y \|^2$ for $\forall x, y \in \mathcal{X}$.

We give here formal definitions of monotonic operators $T$ and the $\epsilon$-first-order stationary point.

Definition 2 ($\epsilon$-First-Order Stationary Point). A point $x \in \mathcal{X}$ is called $\epsilon$-first-order stationary point if $\| T(x) \| \leq \epsilon$.

Remark: We make the following observations:

(a). From the definition, it is evident that strong-monotonicity $\Rightarrow$ monotonicity $\Rightarrow$ pseudo-monotonicity $\Rightarrow$ MVI($T, \mathcal{X}$) has solution.

(b). For the min-max problem (1), when $F(u, v)$ is convex in $u$ and concave in $v$, $T$ is monotone. And, therefore solving SVI($T, \mathcal{X}$) is equivalent to solving (1). When $T$ is not monotone, by assuming $T$ is Lipschitz continuous, it can be shown that the solution set of (1) is a subset of the solution set of SVI($T, \mathcal{X}$). However, even solving SVI($T, \mathcal{X}$) is NP-hard in general and hence we resort to finding an $\epsilon$-first-order stationary point.

Throughout the paper, we make the following assumption:

Assumption 1. (i). $T$ is $L$-Lipschitz continuous, i.e. $\| T(x_1) - T(x_2) \|_2 \leq L \| x_1 - x_2 \|_2$ for $\forall x_1, x_2 \in \mathcal{X}$.

(ii). MVI($F, \mathcal{X}$) has a solution, i.e. there exists $x_\ast$ such that $\langle T(x), x - x_\ast \rangle \geq 0$ for $\forall x \in \mathcal{X}$.

(iii). For $\forall x \in \mathcal{X}$, $\mathbb{E}[T(x; \xi) - T(x)] = \mathbb{E}[T(x; \xi) - T(x)]^2 \leq \sigma^2$.

Remark: Assumptions (i) and (iii) are commonly used assumptions in the literature of variational inequalities and non-convex optimization (Juditsky et al., 2011; Ghadimi & Lan, 2013; Iusem et al., 2017). Assumption (ii) is used frequently in previous work focusing on analyzing algorithms that solve non-monotone variational inequalities (Iusem et al., 2017; Lin et al., 2018; Mertikopoulos et al., 2018). Assumption (ii) is weaker than other assumptions usually considered, such as pseudo-monotonicity, monotonicity, or coherence as assumed in (Mertikopoulos et al., 2018). For non-convex minimization problem, it has been shown that this assumption holds while using SGD to learn neural networks (Li & Yuan, 2017; Kleinberg et al., 2018; Zhou et al., 2019).

3 Optimistic Stochastic Gradient

This section serves as a warm-up and motivation of our main theoretical contribution presented in the next section. Inspired by (Iusem et al., 2017), we present an algorithm called Optimistic Stochastic Gradient (OSG) that saves the cost of the additional oracle call as required in (Iusem et al., 2017) and maintains the same iteration complexity. The main algorithm is described in Algorithm 1.
It is worth mentioning that Algorithm 1 becomes stochastic extragradient method if one changes $T(z_{k-1}; \xi_{k-1})$ to $T(x_{k-1}; \xi_{k-1}^t)$ in line 3. Stochastic extragradient method requires to compute stochastic gradient over both sequences $\{x_k\}$ and $\{z_k\}$. In contrast, $\{x_k\}$ is an ancillary sequence in OSG and the stochastic gradient is only computed over the sequence of $\{z_k\}$. Thus, stochastic extragradient method is twice as expensive as OSG in each iteration. In some tasks (e.g. training GANs) where the stochastic gradient computation is expensive, OSG is numerically more appealing.

Algorithm 1 Optimistic Stochastic Gradient (OSG)

1: Input: $z_0 = x_0 = 0$
2: for $k = 1, \ldots, N$ do
3: \hspace{1em} $z_k = \Pi_X \left\{ x_{k-1} - \eta \cdot \frac{1}{m_k} \sum_{i=1}^{m_k} T(z_{k-1}; \xi_{k-1}^i) \right\}$
4: \hspace{1em} $x_k = \Pi_X \left\{ x_{k-1} - \eta \cdot \frac{1}{m_k} \sum_{i=1}^{m_k} T(z_{k}; \xi_{k}^i) \right\}$
5: end for

Remark: When $X = \mathbb{R}^d$, the update in Algorithm 1 becomes the algorithm in (Daskalakis et al., 2017), i.e.
\[
z_{k+1} = z_k - 2\eta \cdot \frac{1}{m_{k-1}} \sum_{i=1}^{m_k} T(z_k; \xi_{k}^i) + \eta \cdot \frac{1}{m_k} \sum_{i=1}^{m_{k-1}} T(z_{k-1}; \xi_{k-1}^i) \tag{2}\]

Theorem 1. Suppose that Assumption 1 holds. Let $r_\alpha(z_k) = \|z_k - \Pi_X (z_k - \alpha T(z_k))\|$. Let $\eta \leq 1/9L$ and run Algorithm 1 for $N$ iterations. Then we have
\[
\frac{1}{N} \sum_{k=1}^{N} \mathbb{E} \left[ r_\eta^2(z_k) \right] \leq \frac{8\|x_0 - x_*\|^2}{N} + \frac{100\eta^2}{N} \sum_{k=0}^{N} \frac{\sigma^2}{m_k},
\]

Corollary 1. Consider the unconstrained case where $X = \mathbb{R}^d$. Let $\eta \leq 1/9L$, and we have
\[
\frac{1}{N} \sum_{k=1}^{N} \mathbb{E} \|T(z_k)\|_2^2 \leq \frac{8\|x_0 - x_*\|^2}{\eta^2 N} + \frac{100}{N} \sum_{k=0}^{N} \frac{\sigma^2}{m_k}, \tag{3}\]

Remark: There are two implications of Corollary 1.

- (Increasing Minibatch Size) Let $\eta = \frac{1}{9L}$, $m_k = k + 1$. To guarantee $\frac{1}{N} \sum_{k=1}^{N} \mathbb{E} \|T(z_k)\|_2^2 \leq \epsilon^2$, the total number of iterations is $N = \tilde{O}(\epsilon^{-2})$, and the total complexity is $\sum_{k=1}^{N} m_k = \tilde{O}(\epsilon^{-4})$, where $\tilde{O}(\cdot)$ hides a logarithmic factor of $\epsilon$.

- (Constant Minibatch Size) Let $\eta = \frac{1}{9L}$, $m_k = 1/\epsilon^2$. To guarantee $\frac{1}{N} \sum_{k=1}^{N} \mathbb{E} \|T(z_k)\|_2^2 \leq \epsilon^2$, the total number of iterations is $N = O(\epsilon^{-2})$, and the total complexity is $\sum_{k=0}^{N} m_k = O(\epsilon^{-4})$.

4 Optimistic Adagrad

4.1 Adagrad for Minimization Problems

Before introducing Optimistic Adagrad, we present here a quick overview of Adagrad (Duchi et al., 2011). The main objective in Adagrad is to solve the following minimization problem:
\[
\min_{w \in \mathbb{R}^d} F(w) = \mathbb{E}_{\zeta \sim \mathcal{P}} f(w; \zeta) \tag{4}
\]
where $w$ is the model parameter, and $\zeta$ is an random variable following distribution $\mathcal{P}$. The update rule of Adagrad is
\[
w_{t+1} = w_t - \eta H_{t}^{-1}g_t, \tag{5}\]
where $\eta > 0$, $\hat{g}_i = \nabla f(w_i; \xi_i)$, $H_t = \text{diag} \left( \left( \sum_{i=1}^t \hat{g}_i \circ \hat{g}_i \right)^{\frac{1}{2}} \right)$ with $\circ$ denoting the Hadamard product. Adagrad when taking $H_t = I$ reduces to SGD. Different from SGD, Adagrad dynamically incorporates knowledge of history gradients to perform more informative gradient-based learning. When solving a convex minimization problem and the gradient is sparse, Adagrad converges faster than SGD. There are several variants of Adagrad, including Adam (Kingma & Ba, 2014), RMSProp (Tieleman & Hinton, 2012), and AmsGrad (Reddi et al., 2019). All of them share the spirit, as they take advantage of the information provided by the history of gradients. Wilson et al. (2017) provide a complete overview of different adaptive gradient methods in a unified framework. It is worth mentioning that Adagrad can not be directly applied to solve non-convex non-concave min-max problems with provable guarantee.

4.2 Optimistic Adagrad for min-max optimization

Our second algorithm named Optimistic Adagrad (OAdagrad) is an adaptive variant of OSG, which also updates minimization variable and maximization variable simultaneously. The key difference between OSG and OAdagrad is that OAdagrad inherits ideas from Adagrad to construct variable metric based on history gradients information, while OSG only utilizes a fixed metric. This difference helps us establish faster adaptive convergence under some mild assumptions.

**Assumption 2.** (i). There exists $G > 0$ and $\delta > 0$ such that $\|T(z; \xi)\|_2 \leq G$, $\|T(z; \xi)\|_\infty \leq \delta$ for all $z$ almost surely.

(ii). There exists a universal constant $D > 0$ such that $\|x_k\|_2 \leq D$ for $k = 1, \ldots, N$, and $\|x_+\|_2 \leq D$.

Remark: Assumption 2 (i) is a standard one often made in literature (Duchi et al., 2011). Assumption 2 (ii) holds when we use normalization layers in the discriminator and generator such as spectral normalization of weights (Miyato et al., 2018; Zhang et al., 2018), that will keep the norms of the weights bounded. Regularization techniques such as weight decay also ensure that the weights of the networks remain bounded throughout the training.

Define $\hat{g}_k = \frac{1}{m} \sum_{i=1}^m T(z_k; \xi_k^i)$, $\|x\|_H = \sqrt{x^T H x}$. Denote $\hat{g}_{0:k}$ by the concatenation of $\hat{g}_0, \ldots, \hat{g}_k$, and denote $\hat{g}_{0:k,i}$ by the $i$-th row of $\hat{g}_{0:k}$.

**Algorithm 2** Optimistic AdaGrad (OAdagrad)

1: **Input:** $z_0 = x_0 = 0, H_0 = \delta I$
2: for $k = 1, \ldots, N$ do
3: $z_k = x_{k-1} - \eta H_{k-1}^{-1} \hat{g}_{k-1}$
4: $x_k = x_{k-1} - \eta H_{k-1}^{-1} \hat{g}_k$
5: Update $\hat{g}_{0:k} = [\hat{g}_{0:k-1} \hat{g}_k]$, $s_{k,i} = \|\hat{g}_{0:k,i}\|$, $i = 1, \ldots, d$ and set $H_k = \delta I + \text{diag}(s_{k-1})$
6: end for

**Theorem 2.** Suppose Assumption 1 and 2 hold. Suppose $\|\hat{g}_{1:k,i}\|_2 \leq \delta k^\alpha$ with $0 \leq \alpha \leq 1/2$ for every $i = 1, \ldots, d$ and every $k = 1, \ldots, N$. When $\eta \leq \frac{\delta}{16}$, after running Algorithm 2 for $N$ iterations, we have

$$\frac{1}{N} \sum_{k=1}^N \mathbb{E}[\|T(z_k)\|_2^2] \leq \frac{8 \log(N + 1) D^2 \delta^2 (1 + d(N - 1)^\alpha)}{\eta^2 N} + \frac{100 \log(N + 1) \left( \alpha^2 / m + d \left( 2\delta^2 N^\alpha + G^2 \right) \right)}{N}.$$  

To make sure $\frac{1}{N} \sum_{k=1}^N \mathbb{E}[\|T(z_k)\|_2^2] \leq \epsilon^2$, the number of iterations is $N = \tilde{O} \left( \epsilon^{-\frac{1}{2(1-\alpha)}} \right)$, where $\tilde{O}(\cdot)$ hides a logarithmic factor of $\epsilon$.

Remark:

- We denote $\hat{g}_{1:k,i}$ by the cumulative stochastic gradient, where $\|\hat{g}_{1:k,i}\|_2 \leq \delta k^\alpha$ characterizes the growth rate of the gradient in terms of $i$-th coordinate. In our proof, a key quantity is $\sum_{i=1}^d \|\hat{g}_{1:k,i}\|_2$ that crucially affects the computational complexity of Algorithm 2. Since
\[
\sum_{i=1}^{d} \| \tilde{g}_{1,k,i} \|_2 \leq \delta d \alpha, \text{ in the worst case, } \alpha = \frac{1}{2}. \text{ But in practice, the stochastic gradient is usually sparse, and hence } \alpha \text{ can be strictly smaller than } \frac{1}{2}.
\]

- As shown in Theorem 2, the minibatch size used in Algorithm 2 for estimating the first-order oracle can be any positive constant and independent of \(c\). This is more practical than the results established in Theorem 1, since the minibatch size in Theorem 1 does either increase in terms of number of iterations or is dependent on \(c\). When \(\alpha = \frac{1}{2}\), the complexity of Algorithm 2 is \(\tilde{O}(1/\epsilon^4)\), which matches the complexity stated in Theorem 1. When \(\alpha < \frac{1}{2}\), the complexity of OAdagrad given in Algorithm 2 is \(\tilde{O}\left(\epsilon^{-\frac{2}{3\alpha}}\right)\), i.e., strictly better than that of OSG given in Algorithm 1.

**Comparison with Alternating Adam and Optimistic Adam** Alternating Adam is very popular in GAN training (Goodfellow et al., 2014; Arjovsky et al., 2017; Gulrajani et al., 2017; Brock et al., 2018). In Alternating Adam, one alternates between multiple steps of Adam on the discriminator and a single step of Adam on the generator. The key difference between OAdagrad and Alternating Adam is that OAdagrad updates the discriminator and generator simultaneously. It is worth mentioning that OAdagrad naturally fits into the framework of Optimistic Adam proposed in (Daskalakis et al., 2017). Taking \(\beta_1 = 0, \beta_1 \rightarrow 1\) in their Algorithm 1 reduces to OAdagrad with annealing learning rate. To the best of our knowledge, there is no convergence proof for Alternating Adam for non-convex non-concave problems. Our convergence proof for OAdagrad provides a theoretical justification of a special case of Optimistic Adam.

### 5 Experiments

**WGAN-GP on CIFAR10** In the first experiment, we verify the effectiveness of the proposed algorithms in GAN training using the PyTorch framework (Paszke et al., 2017). We use Wasserstein GAN with gradient penalty (WGAN-GP) (Gulrajani et al., 2017) and CIFAR10 data in our experiments. The architectures of discriminator and generator, and the penalty parameter in WGAN-GP are set to be the same as in the original paper. We compare Alternating Adam, OSG and OAdagrad, where the Alternating Adam is to run 5 steps of Adam on the discriminator before performing 1 step of Adam on the generator. We try different batch sizes (64, 128, 256) for each algorithm. For each algorithm, we tune the learning rate in the range of \(\{1 \times 10^{-3}, 2 \times 10^{-4}, 1 \times 10^{-4}, 2 \times 10^{-5}, 1 \times 10^{-5}\}\) when using batch size 64, and use the same learning rate for batch size 128 and 256. We report Inception Score (IS) (Salimans et al., 2016) as a function of number of iterations. Figure 1 suggests that OAdagrad performs better than OSG and Alternating Adam, and OAdagrad results in higher IS. In addition, for OSG and OAdagrad, large batch size helps the training, which is consistent with our theory. For Alternating Adam, the benefits of large batch size is not significant. We also compare the generated CIFAR10 images associated with these three methods, which is included in Appendix A.

**Growth Rate of Cumulative Stochastic Gradient** In the second experiment, we employ OAdagrad to train GANs and study the growth rate of the cumulative stochastic gradient (i.e., \(\sum_{i=1}^{d} \| \tilde{g}_{1,k,i} \|_2\)). We tune the learning rate from \(\{1 \times 10^{-3}, 2 \times 10^{-4}, 1 \times 10^{-4}, 2 \times 10^{-5}, 1 \times 10^{-5}\}\) and choose batch size to be 64. In Figure 2, the blue curve and red curve stand for the growth rate for OAdagrad and its corresponding tightest polynomial growth upper bound respectively. \(N\) is the number of iterations, and \(c\) is a multiplicative constant such that the red curve and blue curve...
Figure 2: Cumulative Stochastic Gradient as a function of number of iterations, where \( \text{netD} \) and \( \text{netG} \) stand for the discriminator and generator respectively. The blue curve and red curve stand for the growth rate of the cumulative stochastic gradient for OAdagrad and its corresponding tightest polynomial growth upper bound, respectively.

![Figure 2: Cumulative Stochastic Gradient](image)

Figure 3: Self-Attention GAN on ImageNet, with evaluation using Official TensorFlow Inception Score and Official TensorFlow FID. We see that OAdagrad indeed outperforms Simultaneous Adam in terms of the (TensorFlow) Inception score (higher is better), and in terms of (TensorFlow) Fréchet Inception Distance (lower is better). We don’t report here Alternating Adam since in our run it has collapsed.

![Figure 3: Self-Attention GAN on ImageNet](image)

**Self-Attention GAN on ImageNet**  In the third experiment, we consider GAN training on large-scale dataset. We use the model from Self-Attention GAN (Zhang et al., 2018) (SA-GAN) and ImageNet as our dataset. Note that in this setting the boundedness of both generator \((G)\) and discriminator \((D)\) is ensured by spectral normalization of both \(G\) and \(D\). Three separate experiments are performed, including Alternating Adam (baseline), Simultaneous Adam (Mescheder et al., 2017), and OAdagrad. It should be mentioned that the update rule of Simultaneous Adam involves performing Adam-type update for discriminator and generator simultaneously. Training is performed with batch size 128 for all experiments.

For the baseline experiment (Alternating Adam) we use the default settings and hyper parameters reported in SA-GAN (Zhang et al., 2018) (note that we are not using the same batch size of 256 as in
(Zhang et al., 2018) due to limited computational resources). In our experience, Alternating Adam training with same learning rate as in SA-GAN (0.0001 for generator and 0.0004 for discriminator) collapsed.

For both OAdagrad and Simultaneous Adam, we use different learning rate for generator and discriminator, as suggested in (Heusel et al., 2017). Specifically, the learning rates used are $10^{-3}$ for the generator and $4 \times 10^{-5}$ for the discriminator. We report both Inception Score (IS) and Fréchet Inception Distance (Heusel et al., 2017) (FID) as a function of number of iterations.

We compare the generated ImageNet images associated with the three optimization methods in Appendix A. Since Alternating Adam collapsed we don’t report its Inception Score or FID. As it can be seen in Figure 3 and Appendix A, OAdagrad outperforms simultaneous Adam in quantitative metrics (IS and FID) and in sample quality generation. We think OAdagrad would benefit from larger batch-size, in order to achieve state-of-the-art results.

6 Conclusion

In this paper, we explain the effectiveness of adaptive gradient methods in training GANs from both theoretical and empirical perspectives. Theoretically, we provide two efficient stochastic algorithms for solving a class of min-max non-convex non-concave problems with state-of-the-art computational complexities. We also establish adaptive complexity results for an Adagrad-style algorithm by using coordinate-wise stepsize according to the geometry of the history data. The algorithm is proven to enjoy faster adaptive convergence than its non-adaptive counterpart when the gradient is sparse, which is similar to Adagrad applied to convex minimization problem. We have conducted extensive empirical studies to verify our theoretical findings. In addition, our experimental results suggest that the reason why adaptive gradient methods deliver good practical performance for GAN training is due to the slow growth rate of the cumulative stochastic gradient.

References


A More Experimental Results

Comparison of Generated CIFAR10 Images by Different Optimization Methods In this section, we report the generated CIFAR10 images during the training of WGAN-GP by three optimization methods (OSG, OAdagrad, Alternating Adam). Every method uses batch size 64, and 1 iteration represents calculating the stochastic gradient with minibatch size 64 once. Figure 4 consists of images by three optimization methods at iteration 8000. Visually we can see that OAdagrad is better than Alternating Adam, and both of them are significantly better than OSG. It is consistent with the inception score results reported in Figure 1, and it also illustrates the tremendous benefits delivered by adaptive gradient methods when training GANs.

![Generated CIFAR10 Images](image1.png)

Figure 4: WGAN-GP: Generated CIFAR10 images using different optimization methods at iteration 8000.

Comparison of Generated ImageNet Images by Different Optimization Methods In this section, we report the generated ImageNet images during the training of Self-Attention GAN by three optimization methods (OAdagrad, Simultaneous Adam, Alternating Adam). Every method uses batch size 128 and 1 iteration represents calculating the stochastic gradient with minibatch 128 once. Figure 5 consists of images by three optimization methods at iteration 135000. Visually it is apparent that OAdagrad is better than Simultaneous Adam, and both of them are significantly than Alternating Adam.

![Generated ImageNet Images](image2.png)

Figure 5: Self-Attention GAN (SA-GAN): Generated ImageNet images using different optimization methods at iteration 135000. OAdagrad produces better quality images than simultaneous Adam. For both Oadagrad and simultaneous Adam we use the same learning rates: 0.001 for generator and 0.00004 for the discriminator. Alternating Adam in our experience with same learning rate as in SA-GAN 0.0001 for generator and 0.0004 for discriminator collapsed. Note that our setting is different from SA-GAN since our batchsize is 128 while it is 256 in SA-GAN. It was also noted in SA-GAN that alternating Adam is hard to train.
Unofficial PyTorch Inception Score and FID results for SA-GAN on ImageNet

![Graphs showing Inception Score and FID](image)

Figure 6: Self-Attention GAN on ImageNet, with evaluation using Unofficial PyTorch Inception Score and Unofficial PyTorch FID. We see that OAdagrad indeed outperforms Simultaneous Adam in terms of the (PyTorch) Inception score (higher is better), and in terms of (PyTorch) Fréchet Inception Distance (lower is better). We don’t report here Alternating Adam since in our run it has collapsed.

## B RELATED WORK

### Min-max Optimization and GAN Training

For convex-concave min-max optimization, the extragradient method was first proposed by (Korpelevich, 1976). Later on, under gradient Lipschitz condition, Nemirovski (2004) extended the idea of extragradient to mirror-prox and obtained the $O(1/N)$ convergence rate in terms of the duality gap (see also (Nesterov, 2007)), where $N$ is the number of iterations. When only the stochastic first-order oracle is available, the stochastic mirror-prox was analyzed by (Juditsky et al., 2011). The convergence rates for both deterministic and stochastic mirror-prox are optimal (Nemirovsky & Yudin, 1983). Recently, Bach & Levy (2019) proposed a universal algorithm that is adaptive to smoothness and noise, and simultaneously achieves optimal convergence rate.

There is a plethora of work analyzing one-sided nonconvex min-max problem, where the objective function is nonconvex in the minimization variable but concave in maximization variable. When the function is weakly-convex in terms of the minimization variable, Rafique et al. (2018) propose a stage-wise stochastic algorithm that approximately solves a convex-concave subproblem by adding a quadratic regularizer and show the first-order convergence of the equivalent minimization problem. Under the same setting, Lu et al. (2019) utilize block-based optimization strategy and show the convergence of the stationarity gap. By further assuming that the function is smooth in the minimization variable, Lin et al. (2019) show that (stochastic) gradient descent ascent is able to converge to the first-order stationary point of the equivalent minimization problem.

A more challenging problem is the non-convex non-concave min-max problem. Dang & Lan (2015) demonstrate that the deterministic extragradient method is able to converge to $\epsilon$-first-order stationary point with non-asymptotic guarantee. Under the condition that the objective function is weakly-convex and weakly-concave, Lin et al. (2018) designs a stage-wise algorithm, where in each stage a strongly-convex strongly-concave subproblem is constructed by adding quadratic terms and appropriate stochastic algorithms can be employed to approximately solve it. They also show the convergence to the stationary point. Sanjabi et al. (2018) design an alternating deterministic optimization algorithm, in which multiple steps of gradient ascent for dual variable are conducted before one step of gradient descent for primal variable is performed. They show the convergence to stationary point based on the assumption that the inner maximization problem satisfies PL condition (Polyak, 1969). Our work is different from these previous methods in many aspects. In comparison to (Lin et al., 2018), our result does not need the bounded domain assumption. Furthermore, our iteration complexity is $O(1/\epsilon^4)$ to achieve $\epsilon$-first-order stationary point while the corresponding complexity in (Lin et al., 2018) is $O(1/\epsilon^6)$. When comparing to (Sanjabi et al., 2018), we do not assume that the PL condition (Polyak-Lojasiewicz) holds. Additionally, our algorithm is stochastic and not restricted to the deterministic case. Apparently the most related work to the present one is (Iusem et al., 2017). The stochastic extragradient method analyzed in (Iusem et al., 2017) requires calculation of two
stochastic gradients per iteration, while the present algorithm only needs one since it memorizes the stochastic gradient in the previous iteration to guide the update in the current iteration. Nevertheless, we achieve the same iteration complexity as in (Iusem et al., 2017).

There are a body of work analyzing the convergence behavior of min-max optimization algorithms and its application in training GANs (Heusel et al., 2017; Daskalakis & Panageas, 2018; Nagarajan & Kolter, 2017; Grnarova et al., 2017; Yadav et al., 2017; Gidel et al., 2018; Mertikopoulos et al., 2018; Mazumdar et al., 2019). A few of them (Heusel et al., 2017; Daskalakis & Panageas, 2018; Mazumdar et al., 2019) only have asymptotic convergence. Others (Nagarajan & Kolter, 2017; Grnarova et al., 2017; Daskalakis et al., 2017; Yadav et al., 2017; Gidel et al., 2018; Mertikopoulos et al., 2018) focus on more restricted settings. For example, Nagarajan & Kolter (2017); Grnarova et al. (2017) require the concavity of the objective function in terms of dual variable. Yadav et al. (2017); Gidel et al. (2018) assume the objective to be convex-concave. Mertikopoulos et al. (2018) impose the so-called coherence condition which is stronger than our assumption. Daskalakis et al. (2017) analyze the last-iteration convergence for bilinear problem. Recently, Gidel et al. (2019) analyze the benefits of using negative momentum in alternating gradient descent to improve the training of a bilinear game. Chavdarova et al. (2019) develop a variance-reduced extragradient method and shows its linear convergence under strong monotonicity and finite-sum structure assumptions. Azizian et al. (2019) provide a unified analysis of extragradient for bilinear game, strongly monotone case, and their intermediate cases. However, none of them give non-asymptotic convergence results for the class of non-convex non-concave min-max problem considered in our paper.

C PROOF OF THEOREM 1

C.1 FACTS

Suppose \( \mathcal{X} \subset \mathbb{R}^d \) is closed and convex set, then we have

**Fact 1.** For all \( x \in \mathbb{R}^d \) and \( y \in \mathcal{X} \), \( \| \Pi_{\mathcal{X}}(x) - y \|^2 + \| \Pi_{\mathcal{X}}(x) - x \|^2 \leq \| x - y \|^2 \).

**Fact 2.** For all \( x \in \mathbb{R}^d \) and \( y \in \mathcal{X} \), \( \langle x - \Pi_{\mathcal{X}}(x), y - \Pi_{\mathcal{X}}(x) \rangle \leq 0 \).

C.2 LEMMAS

**Lemma 1.** For \( \eta \leq \frac{1}{9L} \), we have

\[
\frac{1}{2} \sum_{k=1}^{N} \| x_{k-1} - z_k \|^2 + \frac{1}{2} \sum_{k=1}^{N} \| x_k - z_k \|^2 \leq \| x_0 - x^* \|^2 - \| x_N - x^* \|^2 + 12\eta^2 \sum_{k=0}^{N} \| \epsilon_k \|^2 + \sum_{k=1}^{N} \Lambda_k \tag{7}
\]

**Proof.** Let \( x^* \in \mathcal{X}^* \), where \( \mathcal{X}^* \) is the set of optimal solutions of MVI\((T, \mathcal{X})\), i.e. \( \langle T(x), x - x^* \rangle \geq 0 \) holds for \( \forall x \in \mathcal{X} \). Define \( \epsilon_k = \frac{1}{m_k} \sum_{i=1}^{m_k} T(z_k, \xi^i_k) - T(z_k), \) and \( \tilde{T}(\epsilon_k, z_k) = T(z_k) + \epsilon_k \). For
any $x \in X$, we have

$$
\|x_k - x\|^2 = \|\Pi_X (x_{k-1} - \eta \hat{T}(\epsilon_k, z_k)) - x\|^2
$$

(a) $\|x_{k-1} - \eta \hat{T}(\epsilon_k, z_k) - x\|^2 - \|x_{k-1} - \eta \hat{T}(\epsilon_k, z_k) - \Pi_X (x_{k-1} - \eta \hat{T}(\epsilon_k, z_k))\|^2$

$$
= \|x_{k-1} - x\|^2 - \|x_{k-1} - x_k\|^2 + 2 \langle x - x_k, \eta \hat{T}(\epsilon_k, z_k) \rangle
$$

(b) $\|x_{k-1} - x\|^2 - \|x_{k-1} - z_k + z_k - x_k\|^2 + 2 \langle x - z_k, \eta \hat{T}(\epsilon_k, z_k) \rangle + 2 \langle z_k - x_k, \eta \hat{T}(\epsilon_k, z_k) \rangle$

(c) $\|x_{k-1} - x\|^2 - \|x_{k-1} - z_k\|^2 - \|z_k - x_k\|^2 - 2 \langle x_{k-1} - z_k, z_k - x_k \rangle + 2 \langle x - z_k, \eta \hat{T}(\epsilon_k, z_k) \rangle + 2 \langle z_k - x_k, \eta \hat{T}(\epsilon_k, z_k) \rangle$

\hspace{1cm} (8)

where (a) holds by using Fact 1. Note that

$$
2 \langle x_* - z_k, \eta \hat{T}(\epsilon_k, z_k) \rangle = 2 \langle x_* - z_k, \eta (T(z_k) + \epsilon_k) \rangle \leq 2 \langle x_* - z_k, \eta \epsilon_k \rangle,
$$

where the last inequality holds by the fact that $\langle x_* - z_k, T(z_k) \rangle \leq 0$ since $x_*$ is a solution of MVI($T, X$). Note that

$$
2 \langle x_k - z_k, x_{k-1} - \eta \hat{T}(\epsilon_k, z_k) - z_k \rangle
$$

$$
= 2 \langle x_k - z_k, x_{k-1} - \eta \hat{T}(\epsilon_{k-1}, z_{k-1}) - z_k \rangle + 2 \langle x_k - z_k, \eta \hat{T}(\epsilon_{k-1}, z_{k-1}) - \hat{T}(\epsilon_k, z_k) \rangle
$$

(a) $2 \eta \|x_k - z_k\| \cdot \|\hat{T}(\epsilon_{k-1}, z_{k-1}) - \hat{T}(\epsilon_k, z_k)\|

(b) $\|\Pi_X (x_{k-1} - x_k - \eta \hat{T}(\epsilon_k, z_k)) - \Pi_X (x_{k-1} - \eta \hat{T}(\epsilon_{k-1}, z_{k-1}))\| \cdot \|\hat{T}(\epsilon_{k-1}, z_{k-1}) - \hat{T}(\epsilon_k, z_k)\|

(c) $2 \eta^2 \|\hat{T}(\epsilon_{k-1}, z_{k-1}) - \hat{T}(\epsilon_k, z_k)\|^2 = 2 \eta^2 \|T(z_{k-1}) + \epsilon_{k-1} - (T(z_k) + \epsilon_k)\|^2

\hspace{1cm} (9)

$\leq 2 \eta^2 (\|T(z_{k-1}) - T(z_k)\| + \|\epsilon_{k-1}\| + \|\epsilon_k\|)^2 \leq 2 \eta^2 (L \|z_{k-1} - z_k\| + \|\epsilon_{k-1}\| + \|\epsilon_k\|)^2$

\hspace{1cm} (10)

where (a) holds by $\langle x_k - z_k, x_{k-1} - \eta \hat{T}(\epsilon_{k-1}, z_{k-1}) - z_k \rangle \leq 0$ and Cauchy-Schwartz inequality, where the former inequality comes from Fact 2 and the update rules of the algorithm, (b) holds by the update rule of $z_k$ and $x_k$, (c) holds by the nonexpansion property of the projection operator, (d) holds since $T$ is $L$-Lipschitz continuous, (e) holds since $(a + b + c)^2 \leq 3a^2 + 3b^2 + 3c^2$.

Define $\Lambda_k = 2 \langle x_* - z_k, \eta \epsilon_k \rangle$. Taking $x = x_*$ in (8) and combining (9) and (10), we have

$$
\|x_k - x_*\|^2
$$

$$
\leq \|x_{k-1} - x_*\|^2 - \|x_{k-1} - z_k\|^2 - \|z_k - x_k\|^2 + 6 \eta^2 L^2 \|z_{k-1} - z_k\|^2 + 6 \eta^2 \|\epsilon_{k-1}\|^2 + 2 \|\epsilon_k\|^2 + \Lambda_k
$$

\hspace{1cm} (11)

Noting that

$$
\|z_{k-1} - z_k\|^2 = \|z_{k-1} - x_{k-1} + x_{k-1} - x_k + x_k - z_k\|^2 \leq 3 \|z_{k-1} - x_{k-1}\|^2 + 3 \|x_{k-1} - x_k\|^2 + 3 \|x_k - z_k\|^2,
$$

we rearrange terms in (11), which yields

$$
\|x_{k-1} - x_*\|^2 - \|x_{k-1} - z_k\|^2 - \|z_k - x_k\|^2 + 6 \eta^2 L^2 (3 \|z_{k-1} - x_{k-1}\|^2 + 3 \|x_{k-1} - z_k\|^2 + 3 \|z_k - x_k\|^2)
$$

$$
\leq \|x_{k-1} - x_*\|^2 - \|x_k - x_*\|^2 + 6 \eta^2 \|\epsilon_{k-1}\|^2 + 2 \|\epsilon_k\|^2 + \Lambda_k
$$

\hspace{1cm} (12)
Under review as a conference paper at ICLR 2020

Take summation over \( k = 1, \ldots, N \) in (12) and note that \( x_0 = z_0 \), which yields

\[
\left(1 - 18\eta^2 L^2\right) \sum_{k=1}^N \|x_{k-1} - z_k\|^2 + \left(1 - 36\eta^2 L^2\right) \sum_{k=1}^N \|x_k - z_k\|^2
\]

\[
\leq \|x_0 - x_*\|^2 - \|x_N - x_*\|^2 + 12\eta^2 \sum_{k=0}^N \|\varepsilon_k\|^2 + \sum_{k=1}^N \Lambda_k
\]

(13)

By taking \( \eta \leq \frac{1}{9T} \), we have \( 1 - 36\eta^2 L^2 \geq \frac{1}{2} \), and we have the result. \( \square \)

C.3 Main Proof of Theorem 1

**Proof.** Define \( r_\eta(z_k) = \|z_k - \Pi_{\mathcal{X}}(z_k - \eta T(z_k))\| \). Our goal is to get a bound on \( r_\eta(z_k) \). We have:

\[
r_\eta^2(z_k) = \|z_k - \Pi_{\mathcal{X}}(z_k - \eta T(z_k))\|^2 = \|z_k - x_k + x_k - \Pi_{\mathcal{X}}(z_k - \eta T(z_k))\|^2
\]

\[
\leq 2 \|z_k - x_k\|^2 + 2 \|x_k - \Pi_{\mathcal{X}}(z_k - \eta T(z_k))\|^2
\]

\[
= 2 \|z_k - x_k\|^2 + 2 \Pi_{\mathcal{X}}(x_k - \eta T(\varepsilon_k, z_k)) - \Pi_{\mathcal{X}}(z_k - \eta T(z_k))\|^2
\]

(14)

\[
\leq 2 \|z_k - x_k\|^2 + 4 \|x_k - z_k\|^2 + 4\eta^2 \|T(z_k) - \hat{T}(\varepsilon_k, z_k)\|^2
\]

\[
\leq 4 \|z_k - x_k\|^2 + 4 \|x_k - z_k\|^2 + 4\eta^2 \|\varepsilon_k\|^2
\]

where (a) holds since \( (a + b)^2 \leq 2a^2 + 2b^2 \), (b) holds by the non-expansion property of the projection operator and \( (a + b)^2 \leq 2a^2 + 2b^2 \).

Let \( x_* \in \mathcal{X}^* \), where \( \mathcal{X}^* \) is the set of optimal solutions of MVI(\( T, \mathcal{X} \)), i.e. \( \langle T(x), x - x_* \rangle \geq 0 \) holds for \( \forall x \in \mathcal{X} \). Define \( \varepsilon_k = \frac{1}{m_k} \sum_{i=1}^{m_k} T(z_k, \xi^{(i)}_k) - T(z_k) \), and \( \hat{T}(\varepsilon_k, z_k) = T(z_k) + \varepsilon_k \). Define \( \Lambda_k = 2 \langle x_* - z_k, \eta \varepsilon_k \rangle \).

By summing over \( k \) in Equation (14) and using Equation (7) in Lemma 1, we have

\[
\sum_{k=1}^N r_\eta^2(z_k) \leq 4 \sum_{k=1}^N \|z_k - x_k\|^2 + 4 \sum_{k=1}^N \|x_{k-1} - z_k\|^2 + 4\eta^2 \sum_{k=1}^N \|\varepsilon_k\|^2
\]

\[
= 8 \left( \frac{1}{2} \sum_{k=1}^N \|z_k - x_k\|^2 + \frac{1}{2} \sum_{k=1}^N \|x_{k-1} - z_k\|^2 \right) + 4\eta^2 \sum_{k=0}^N \|\varepsilon_k\|^2
\]

(15)

By (7)

\[
\leq 8 \left( \|x_0 - x_*\|^2 + 12\eta^2 \sum_{k=0}^N \|\varepsilon_k\|^2 + \sum_{k=1}^N \Lambda_k \right) + 4\eta^2 \sum_{k=0}^N \|\varepsilon_k\|^2
\]

Taking expectation and divided by \( N \) on both sides, we have

\[
\frac{1}{N} \sum_{k=1}^N \mathbb{E} [r_\eta^2(z_k)] \leq 8 \left( \frac{1}{N} \|x_0 - x_*\|^2 + 12\eta^2 \sum_{k=0}^N \mathbb{E}\|\varepsilon_k\|^2 + \sum_{k=1}^N \mathbb{E}(\Lambda_k) \right) + 4\eta^2 \sum_{k=1}^N \mathbb{E}\|\varepsilon_k\|^2
\]

\[
\leq 8 \left( \|x_0 - x_*\|^2 + 12\eta^2 \sum_{k=0}^N \frac{\sigma^2}{m_k} \right) + 4\eta^2 \sum_{k=0}^N \frac{\sigma^2}{m_k}
\]

\[
= 8 \frac{\|x_0 - x_*\|^2}{N} + \frac{100\eta^2}{N} \sum_{k=0}^N \frac{\sigma^2}{m_k}.
\]

(16)
D PROOF OF THEOREM 2

In this section, we define \( g_k = T(z_k), \epsilon_k = \hat{g}_k - g_k \).

D.1 LEMMAS

**Lemma 2.** For any positive definite diagonal matrix \( H \) satisfying \( H \succeq \delta I \) with \( \delta > 0 \), if \( \|T(x_1) - T(x_2)\|_2 \leq L \|x_1 - x_2\|_2 \) for \( x_1, x_2 \in X \), then

\[
\|T(x_1) - T(x_2)\|_{H^{-1}} \leq \frac{L}{\delta} \|x_1 - x_2\|_H.
\]

**Proof.** Note that \( H \succeq \delta I \), we have \( 0 < H^{-1} \preceq \frac{1}{\delta} I \). Noting that \( \|x\|_H = \sqrt{x^T H x} \), we have

\[
\|T(x_1) - T(x_2)\|_{H^{-1}} \leq \frac{1}{\sqrt{\delta}} \|T(x_1) - T(x_2)\|_2 \leq \frac{L}{\sqrt{\delta}} \|x_1 - x_2\|_2 \leq \frac{L}{\delta} \|x_1 - x_2\|_H.
\]

\( \square \)

**Lemma 3.** When \( \eta \leq \frac{\delta}{5L} \), we have

\[
\frac{1}{2} \sum_{k=1}^{N} \|x_{k-1} - z_k\|_{H_k^{-1}}^2 + \frac{1}{2} \sum_{k=1}^{N} \|x_k - z_k\|_{H_k^{-1}}^2
\]

\[
\leq \sum_{k=1}^{N} \left( \|x_{k-1} - x_*\|_{H_k^{-1}}^2 - \|x_k - x_*\|_{H_k^{-1}}^2 \right) + 12 \eta^2 \left( \|\epsilon_0\|_{H_0^{-1}}^2 + \sum_{k=1}^{N} \|\epsilon_k\|_{H_k^{-1}}^2 \right) + \sum_{k=1}^{N} \Lambda_k
\]

(17)

**Proof.** Define \( \epsilon_k = \hat{g}_k - g_k \). For any \( x \in X \), we have

\[
\|x_k - x\|_{H_k^{-1}} = \|x_k - \eta H_k^{-1} \hat{g}_k - x\|_{H_k^{-1}} = \|x_k - \eta H_k^{-1} \hat{g}_k - x\|_{H_k^{-1}} - \|x_k - \eta H_k^{-1} g_k - x\|_{H_k^{-1}}
\]

\[
= \|x_k - x\|_{H_k^{-1}} - \|x_k - x - x\|_{H_k^{-1}} + 2 \langle x - x, \eta \hat{g}_k \rangle
\]

\[
= \|x_k - x\|_{H_k^{-1}} - \|x_k - x - x\|_{H_k^{-1}} + 2 \langle x - x, \eta \hat{g}_k \rangle + 2 \langle x - x, \eta \hat{g}_k \rangle
\]

\[
= \|x_k - x\|_{H_k^{-1}} - \|x_k - x - x\|_{H_k^{-1}} + 2 \langle x - x, \eta \hat{g}_k \rangle + 2 \langle x - x, \eta \hat{g}_k \rangle
\]

\[
= \|x_k - x\|_{H_k^{-1}} - \|x_k - x - x\|_{H_k^{-1}} + 2 \langle x - x, \eta \hat{g}_k \rangle + 2 \langle x - x, \eta \hat{g}_k \rangle
\]

\[
= \|x_k - x\|_{H_k^{-1}} - \|x_k - x - x\|_{H_k^{-1}} + 2 \langle x - x, \eta \hat{g}_k \rangle + 2 \langle x - x, \eta \hat{g}_k \rangle
\]

Note that

\[
2 \langle x_* - z_k, \eta \hat{g}_k \rangle = 2 \langle x_* - z_k, \eta (g_k + \epsilon_k) \rangle \leq 2 \langle x_* - z_k, \eta \epsilon_k \rangle,
\]

(19)

where the last inequality holds by the fact that \( \langle x_* - z_k, g_k \rangle \leq 0 \) since \( x_* \) is a solution of MVI(\( T, X \)). Note that

\[
2 \langle x_k - z_k, H_k^{-1} (x_{k-1} - z_k) - \eta \hat{g}_k \rangle
\]

\[
= 2 \langle x_k - z_k, H_k^{-1} (x_{k-1} - z_k) - \eta (g_k + \epsilon_k) \rangle + 2 \langle x_k - z_k, \eta (\hat{g}_k - g_k) \rangle
\]

\[
\leq (a) \|x_k - z_k, H_k^{-1} (x_{k-1} - z_k) - \eta (g_k + \epsilon_k) \|_{H_k^{-1}}^2 + 2 \langle x_k - z_k, \eta (\hat{g}_k - g_k) \rangle
\]

\[
\leq (b) \|x_k - z_k, H_k^{-1} (x_{k-1} - z_k) - \eta (g_k + \epsilon_k) \|_{H_k^{-1}}^2 + 2 \|g_k - \epsilon_k\|_{H_k^{-1}}^2 + \|\epsilon_k\|_{H_k^{-1}}^2
\]

\[
\leq (c) 2 \eta^2 \left( \|g_k - \epsilon_k\|_{H_k^{-1}}^2 + \|\epsilon_k\|_{H_k^{-1}}^2 \right)
\]

\[
\leq (d) 6 \eta^2 \left( \frac{L^2}{\delta^2} \|z_k - z_k\|_{H_k^{-1}}^2 + \|\epsilon_k\|_{H_k^{-1}}^2 \right)
\]

\[
\leq 6 \eta^2 \left( \frac{L^2}{\delta^2} \|z_k - z_k\|_{H_k^{-1}}^2 + \|\epsilon_k\|_{H_k^{-1}}^2 \right)
\]

\[
\leq 6 \eta^2 \left( \frac{L^2}{\delta^2} \|z_k - z_k\|_{H_k^{-1}}^2 + \|\epsilon_k\|_{H_k^{-1}}^2 \right)
\]

(20)
where (a) holds by the update rule of $z_k$ and $x_k$ in Algorithm 2, (b) holds by the triangle inequality, (c) holds by utilizing the Lipschitz continuity of $T$, Lemma 2 and the fact that $H_{k-1} \succeq \delta I$ for any $k$, (d) holds since $(a + b + c)^2 \leq 3a^2 + 3b^2 + 3c^2$.

Define $\Lambda_k = 2(x_* - z_k, \eta \epsilon_k)$. Taking $x = x_*$ in (18) and combining (19) and (20), we have

$$\|x_k - x_*\|^2_{H_{k-1}} \leq \|x_{k-1} - x_*\|^2_{H_{k-1}} - \|z_{k-1} - z_k\|^2_{H_{k-1}} - \|z_k - x_k\|^2_{H_{k-1}} + \frac{6\eta^2 L^2}{\delta^2} \|z_{k-1} - z_k\|^2_{H_{k-1}} + 6\eta^2 \epsilon_k^2_{H_{k-1}} + \Lambda_k$$

(21)

Noting that

$$\|z_{k-1} - z_k\|^2_{H_{k-1}} = \|z_{k-1} - x_{k-1} + x_{k-1} - x_k + x_k - z_k\|^2_{H_{k-1}}$$

$$\leq \|z_{k-1} - x_{k-1}\|^2_{H_{k-1}} + \|x_{k-1} - x_k\|^2_{H_{k-1}} + \|x_k - z_k\|^2_{H_{k-1}}$$

$$\leq \frac{3}{2}\|z_{k-1} - x_{k-1}\|^2_{H_{k-1}} + 3\|x_k - z_k\|^2_{H_{k-1}} + \frac{3}{2}\|z_k - x_k\|^2_{H_{k-1}},$$

we rearrange terms in (21), which yields

$$\|x_{k-1} - z_k\|^2_{H_{k-1}} + \|z_k - x_*\|^2_{H_{k-1}} - \frac{6\eta^2 L^2}{\delta^2} \left( \frac{3}{2}\|z_{k-1} - x_{k-1}\|^2_{H_{k-1}} + 3\|x_k - z_k\|^2_{H_{k-1}} + \frac{3}{2}\|z_k - x_k\|^2_{H_{k-1}} \right)$$

$$\leq \|x_{k-1} - x_*\|^2_{H_{k-1}} - \|z_{k-1} - x_*\|^2_{H_{k-1}} + \|x_k - x_*\|^2_{H_{k-1}} + 6\eta^2 \epsilon_k^2_{H_{k-1}} + 2\eta^2 \epsilon_k^2_{H_{k-1}} + \Lambda_k$$

(22)

Taking summation over $k = 1, \ldots, N$ in (22), and noting that $x_0 = z_0$, $\|x\|^2_{H_{t-1}} \geq \|x\|^2_{H_{t-1}}$ for all $x$ and $t \geq 1$, we have

$$\left(1 - \frac{18\eta^2 L^2}{\delta^2}\right) \sum_{k=1}^N \|x_{k-1} - z_k\|^2_{H_{k-1}} + \left(1 - \frac{36\eta^2 L^2}{\delta^2}\right) \sum_{k=1}^N \|x_k - z_k\|^2_{H_{k-1}}$$

$$\leq \sum_{k=1}^N \left( \|x_{k-1} - x_*\|^2_{H_{k-1}} - \|z_{k-1} - x_*\|^2_{H_{k-1}} + 12\eta^2 \left( \|\epsilon_0\|^2_{H_0} + \sum_{k=1}^N \|\epsilon_k\|^2_{H_{k-1}} \right) \right) + \sum_{k=1}^N \Lambda_k$$

(23)

By taking $\eta \leq \frac{\delta}{9L}$, we have $1 - \frac{36\eta^2 L^2}{\delta^2} \geq \frac{1}{2}$, and we have the result. □

**Lemma 4.** When $\|\hat{g}_{1,N,i}\|_2 \leq \delta N^\alpha$ with $0 \leq \alpha \leq 1/2$ for every $i$, we have

$$\sum_{k=1}^N \left( \|x_{k-1} - x_*\|^2_{H_{k-1}} - \|x_k - x_*\|^2_{H_{k-1}} \right) \leq D^2 \delta + D^2 \cdot d\delta(N-1)^\alpha$$

(24)

**Proof.**

$$\sum_{k=1}^N \left( \|x_{k-1} - x_*\|^2_{H_{k-1}} - \|x_k - x_*\|^2_{H_{k-1}} \right)$$

$$= \|x_0 - x_*\|^2_{H_0} - \|x_1 - x_*\|^2_{H_1} - \|x_2 - x_*\|^2_{H_2} - \ldots - \|x_{N-1} - x_*\|^2_{H_{N-1}} - \|x_N - x_*\|^2_{H_{N-1}}$$

$$\leq \|x_0 - x_*\|^2_{H_0} + \|x_1 - x_*\|^2_{H_1} + \|x_2 - x_*\|^2_{H_2} + \ldots + \|x_{N-1} - x_*\|^2_{H_{N-1}}$$

$$\leq \|x_0 - x_*\|^2_{H_0} + D^2 \left( \text{tr}(H_1 - H_0) + \text{tr}(H_2 - H_1) + \ldots + \text{tr}(H_{N-1} - H_{N-2}) \right)$$

$$\leq \|x_0 - x_*\|^2_{H_0} + D^2 \left( \text{tr}(H_N - H_0) \right) \leq \|x_0 - x_*\|^2_{H_0} + D^2 \text{tr}(H_N - H_0) \leq D^2 \delta + D^2 \cdot d\delta(N-1)^\alpha$$

(25)

□

**Lemma 5.** When $\|\hat{g}_{1,N,i}\|_2 \leq \delta N^\alpha$ with $0 \leq \alpha \leq 1/2$ for every $i$, we have

$$\mathbb{E} \left[ 96\eta^2 \epsilon_0^2_{H_0} + 100\eta^2 \sum_{k=1}^N \|\epsilon_k\|^2_{H_{k-1}} \right] \leq \frac{96\eta^2 \sigma^2}{m\delta} + 100\eta^2 \left( 2dN^\alpha + \frac{G^2d}{\delta} \right)$$

(26)
Proof. Note that

$$
\mathbb{E} \left[ \sum_{k=1}^{N} \| \epsilon_k^2_{H_{k-1}} \right] = \mathbb{E} \left[ \sum_{k=1}^{N} \| \hat{g}_k - g_k^2_{H_{k-1}} \right] = \sum_{k=1}^{N} \mathbb{E} \| \hat{g}_k - g_k^2_{H_{k-1}} - \sum_{k=1}^{N} \left( \mathbb{E} \| \hat{g}_k^2_{H_{k-1}} - \| g_k^2_{H_{k-1}} \right)
$$

$$
\leq \sum_{k=1}^{N} \mathbb{E} \| \hat{g}_k^2_{H_{k-1}} - \sum_{k=1}^{N} \mathbb{E} \| \hat{g}_k^2_{H_{k-1}} - \sum_{k=1}^{N} \left( \mathbb{E} \| \hat{g}_k^2_{H_{k-1}} - \| g_k^2_{H_{k-1}} \right)
$$

$$
= \sum_{k=1}^{N} \mathbb{E} \| \hat{g}_k^2_{H_{k-1}} + \sum_{k=1}^{N} \mathbb{E} \langle \hat{g}_k, (H_{k-1} - H_k^2) \rangle \leq \sum_{k=1}^{N} \mathbb{E} \| \hat{g}_k^2_{H_{k-1}} + \sum_{k=1}^{N} \mathbb{E} \left[ \text{tr}(H_{k-1}^2 - H_k^2)^2 \right]
$$

$$
\leq \frac{2 d}{\delta} \left( \sum_{i=1}^{d} \| g_{i:N}^2 \|^2 + \text{tr}(H_0)^2 \right) \leq 2 \delta d N^2 + G d \frac{2}{\delta}
$$

where (a) holds since we have \( \sum_{k=1}^{N} \mathbb{E} \| \hat{g}_k^2_{H_{k-1}} \leq 2 \sum_{k=1}^{N} \| g_{i:N}^2 \|^2 \) by the setting of \( H_k \) and utilizing Lemma 4 of (Duchi et al., 2011), (b) holds because of \( \| g_{i:N}^2 \|^2 \leq \delta N^2 \).

In addition, we have \( \mathbb{E} \| \epsilon^2_{0} \|^2_{H_{0}^{-1}} \leq \frac{2 a}{m \delta} \), and hence

$$
\mathbb{E} \left[ 96 \eta^2 \| \epsilon^2_{0} \|^2_{H_{0}^{-1}} + 100 \eta^2 \sum_{k=1}^{N} \| \epsilon_k^2_{H_{k-1}} \right] \leq 96 \eta^2 \frac{\sigma^2}{m \delta} + 100 \eta^2 \left( 2 \delta d N^2 + G^2 \frac{d}{\delta} \right)
$$

(27)

Lemma 6. When \( \| \hat{g}_{i:k} \|^2 \leq \delta k^\alpha \) with \( 0 \leq \alpha \leq 1/2 \) for every \( i \), we have

$$
\frac{1}{N} \sum_{k=1}^{N} \mathbb{E} \| T(z_k) \|^2 \leq \delta \log(N + 1) \cdot \frac{1}{N} \sum_{k=1}^{N} \mathbb{E} \| T(z_k) \|^2_{H_{k-1}}
$$

(29)

Proof. Note that

$$
\frac{1}{N} \sum_{k=1}^{N} \mathbb{E} \| T(z_k) \|^2_{H_{k-1}} = \frac{1}{N} \sum_{k=1}^{N} \mathbb{E} \langle T(z_k), H_{k-1}^2 T(z_k) \rangle \geq \frac{1}{N} \sum_{k=1}^{N} \mathbb{E} \left[ \lambda_{\min}(H_{k-1}) \cdot \| T(z_k) \|^2 \right]
$$

$$
= \frac{1}{N} \sum_{k=1}^{N} \mathbb{E} \left[ \frac{\| T(z_k) \|^2}{\lambda_{\max}(H_{k-1})} \right] \geq \frac{1}{N} \sum_{k=1}^{N} \mathbb{E} \left[ \frac{\| T(z_k) \|^2}{\delta^2 k^\alpha} \right] \geq \frac{1}{N} \sum_{k=1}^{N} \mathbb{E} \left[ \frac{\| T(z_k) \|^2}{\delta k} \right],
$$

where \( \lambda_{\min}(\cdot) \) and \( \lambda_{\max}(\cdot) \) denote minimum and maximum eigenvalue respectively. Hence, we have

$$
\frac{1}{N} \sum_{k=1}^{N} \mathbb{E} \| T(z_k) \|^2 \leq \frac{1}{N} \sum_{k=1}^{N} \sum_{k=1}^{N} \frac{1}{k} \mathbb{E} \| T(z_k) \|^2 \leq \frac{\log(N + 1)}{N} \sum_{k=1}^{N} \mathbb{E} \| T(z_k) \|^2
$$

by (30)

$$
\leq \delta \log(N + 1) \cdot \frac{1}{N} \sum_{k=1}^{N} \mathbb{E} \| T(z_k) \|^2_{H_{k-1}}
$$

(31)

\( \square \)
D.2 Main Proof of Theorem 2

Proof. Our goal is to bound $\frac{1}{N} \sum_{k=1}^N \mathbb{E} \|T(z_k)\|_2^2$. Note that

$$\|\eta T(z_k)\|_{H_{k-1}^{-1}}^2 = \left\| H_{k-1}^{1/2} \left( z_k - \eta H_{k-1}^{-1} T(z_k) \right) \right\|^2 = \left\| H_{k-1}^{1/2} \left( z_k - x_k + x_k - (z_k - \eta H_{k-1}^{-1} T(z_k)) \right) \right\|^2$$

\(\leq\) (a) $\left\| H_{k-1}^{1/2} (z_k - x_k) \right\|^2 + \left\| H_{k-1}^{1/2} \left[ x_k - (z_k - \eta H_{k-1}^{-1} T(z_k)) \right] \right\|^2$

\(\leq\) (b) $2 \left\| H_{k-1}^{1/2} (z_k - x_k) \right\|^2 + 2 \left\| H_{k-1}^{1/2} \left[ x_k - (z_k - \eta H_{k-1}^{-1} T(z_k)) \right] \right\|^2$

\(\leq\) (c) $2 \| z_k - x_k \|_{H_{k-1}}^2 + 2 \left\| H_{k-1}^{1/2} \left[ x_k - \eta H_{k-1}^{-1} \hat{g}_k - (z_k - \eta H_{k-1}^{-1} T(z_k)) \right] \right\|^2$

\(\leq\) (d) $2 \| z_k - x_k \|_{H_{k-1}}^2 + 4 \left\| x_k - z_k \right\|_{H_{k-1}}^2 + 4 \| \hat{g}_k - T(z_k) \|_{H_{k-1}}^2$

$$= 2 \| z_k - x_k \|_{H_{k-1}}^2 + 4 \left\| x_k - z_k \right\|_{H_{k-1}}^2 + 4 \| \hat{g}_k - T(z_k) \|_{H_{k-1}}^2 + 4 \eta^2 \| \epsilon_k \|_{H_{k-1}}^2$$

(32)

where (a) holds by the triangle inequality, (b) is due to $(a+b)^2 \leq 2a^2 + 2b^2$, (c) holds by the update rule of $x_k$ of Algorithm 2, and (d) comes from the triangle inequality and $(a+b)^2 \leq 2a^2 + 2b^2$.

Taking summation over $k = 1, \ldots, N$ over (32) and invoking Lemma 3, we have

$$\sum_{k=1}^N \|\eta T(z_k)\|_{H_{k-1}^{-1}}^2 \leq \sum_{k=1}^N \left( 2 \| z_k - x_k \|_{H_{k-1}}^2 + 4 \left\| x_k - z_k \right\|_{H_{k-1}}^2 + 4 \eta^2 \| \epsilon_k \|_{H_{k-1}}^2 \right)$$

$$\leq 8 \left( \frac{1}{2} \sum_{k=1}^N \| z_k - x_k \|_{H_{k-1}}^2 + \frac{1}{2} \sum_{k=1}^N \left\| x_k - z_k \right\|_{H_{k-1}}^2 \right) + 4 \eta^2 \sum_{k=1}^N \| \epsilon_k \|_{H_{k-1}}^2$$

By (17) \(\leq\)

$$\leq 8 \left( \sum_{k=1}^N \left( \| x_{k-1} - x_a \|_{H_{k-1}}^2 - \| x_k - x_a \|_{H_{k-1}}^2 \right) + 12 \eta^2 \left( \| \epsilon_0 \|_{H_{0-1}}^2 + \sum_{k=1}^N \| \epsilon_k \|_{H_{k-1}}^2 \right) + \sum_{k=1}^N \Lambda_k \right)$$

$$+ 4 \eta^2 \sum_{k=1}^N \| \epsilon_k \|_{H_{k-1}}^2$$

$$= 8 \sum_{k=1}^N \left( \| x_{k-1} - x_a \|_{H_{k-1}}^2 - \| x_k - x_a \|_{H_{k-1}}^2 \right) + 96 \eta^2 \| \epsilon_0 \|_{H_{0-1}}^2 + 100 \eta^2 \sum_{k=1}^N \| \epsilon_k \|_{H_{k-1}}^2 + 8 \sum_{k=1}^N \Lambda_k$$

(33)

Taking expectation on both sides, and invoking Lemma 4 and Lemma 5, and noting that

$$\mathbb{E} \left[ \sum_{k=1}^N \Lambda_k \right] = 0,$$

we have

$$\sum_{k=1}^N \mathbb{E} \|\eta T(z_k)\|_{H_{k-1}^{-1}}^2 \leq 8 \left( D^2 \delta + D^2 \cdot d \delta (N - 1)^{\alpha} \right) + \frac{96 \eta^2 \sigma^2}{m \delta} + 100 \eta^2 \left( 2 \delta d N^\alpha + \frac{G^2 d}{\delta} \right)$$

(34)

Dividing $\eta^2 N$ on both sides, we have

$$\frac{1}{N} \sum_{k=1}^N \mathbb{E} \|T(z_k)\|_{H_{k-1}}^2 \leq \frac{8}{\eta^2 N} \left( D^2 \delta + D^2 \cdot d \delta (N - 1)^{\alpha} \right) + \frac{96 \sigma^2}{N m \delta} + \frac{100}{N} \left( 2 \delta d N^\alpha + \frac{G^2 d}{\delta} \right)$$

(35)

By combining (35) and Lemma 6, we have

$$\frac{1}{N} \sum_{k=1}^N \mathbb{E} \|T(z_k)\|_2^2 \leq \frac{8 \log(N + 1)}{\eta^2 N} \left( D^2 \delta + D^2 \cdot d \delta (N - 1)^{\alpha} \right) + \frac{96 \sigma^2 \log(N + 1)}{m N} + \frac{100 \log(N + 1)}{N} \left( 2 \delta d N^\alpha + G^2 d \right)$$

$$\leq \frac{8 \log(N + 1) D^2 \delta^2 (1 + d(N - 1)^{\alpha})}{\eta^2 N} + \frac{100 \log(N + 1) \left( \sigma^2 / m + d \left( 2 \delta^2 N^\alpha + G^2 \right) \right)}{N}$$

(36)