Abstract—We introduce biased gradient oracles to capture a setting where the function measurements have an estimation error that can be controlled through a batch size parameter. Our proposed oracles are appealing in several practical contexts, for instance, risk measurement estimation from a batch of independent and identically distributed samples, or simulation optimization, where the function measurements are “biased” due to computational constraints. In either case, increasing the batch size reduces the estimation error. We highlight the applicability of our biased gradient oracles in a risk-sensitive reinforcement learning setting. In the stochastic nonconvex optimization context, we analyze a variant of the randomized stochastic gradient algorithm with a biased gradient oracle. We quantify the convergence rate of this algorithm by deriving nonasymptotic bounds on its performance. Next, in the stochastic convex optimization setting, we derive nonasymptotic bounds for the last iterate of a stochastic gradient descent algorithm with a biased gradient oracle.

Index Terms—Biased gradient oracle, Gaussian smoothing, nonasymptotic bounds, simultaneous perturbation (SP), zeroth-order stochastic optimization.

I. INTRODUCTION

We consider the problem of minimizing a smooth objective function, when the optimization algorithm is provided with biased function measurements. This setting is motivated by practical applications, where the objective function is estimated from a batch dataset, and the estimation scheme is biased. As an example, consider the problem of estimating conditional value-at-risk (CVaR), a popular risk measure in financial applications, from a batch of independent and identically distributed (i.i.d.) samples. The classic CVaR estimator [1] requires estimation of a certain quantile of the underlying distribution, and hence, the resulting estimate is biased. As another example, one could consider a simulation optimization problem [2], where the function measurements are “biased” due to computational constraints. In both examples, increasing the batch size used for estimation decreases the estimation error.

The above-mentioned “biased stochastic optimization” setting is more general than the canonical “zeroth-order stochastic optimization” setting, because the former features an estimation error that has a positive mean, while the latter usually features an estimation error that vanishes in expectation. We extend the theory of zeroth-order stochastic optimization to our setting by formalizing two oracle models that encapsulate a biased stochastic optimization problem. In each oracle model, an algorithm obtains a noisy and biased estimate of the gradient at any chosen point. Both oracles feature a batch-size parameter that can be used to control an additive estimation error component in the gradient estimates. The difference between the two proposed biased gradient oracles is that the first oracle features a bias-variance tradeoff for the gradient estimates, while the second one does not have such a tradeoff.

The biased gradient oracles can be implemented using the simultaneous perturbation (SP) [3], [4] class of algorithms that can provide biased gradient information, using only noisy function measurements. Such an approach can be extended to cover the case of biased function measurements, which we consider in this article. The gradient estimate resulting from an SP method usually has a bias-variance tradeoff, i.e., the estimate has an additive bias of $O(\eta^2)$, where $\eta$ is a parameter to be chosen by the optimization algorithm. The variance of the gradient estimate is $O(1/\eta^2)$, and the choice of $\eta$ relates to bias-variance tradeoff [3], [5]–[7]. Under additional assumptions, one can eschew the bias-variance tradeoff, i.e., reduce the bias without adversely affecting the variance [4], [8], [9].

The focus of this article is to understand the rate of convergence of gradient-based methods with inputs from a biased gradient oracle. We derive nonasymptotic bounds on the iteration complexity of gradient-based methods for a nonconvex as well as a convex objective. In either case, we derive bounds for gradient-based methods with inputs from the following oracle models.

1) An oracle whose gradient estimates have a parameter for trading-off bias against the variance, we shall refer to this oracle as (O1).

2) An oracle where the gradient estimates have no bias-variance tradeoff, we shall refer to this oracle as (O2) further.
Note that both oracles have a batch size parameter for controlling the estimation error. Table I summarizes our bounds in the convex as well as nonconvex regimes, under two oracle models.

We now summarize our contributions in the case when the objective is nonconvex. We study the nonasymptotic performance of the randomized stochastic gradient (RSG) algorithm, proposed in [8]. The case of unbiased gradient information is addressed in the aforementioned reference, and we focus on the cases when RSG is provided inputs from (O1) or (O2). From our analysis, we observe that RSG has an iteration complexity bound of $O(1/\epsilon^2)$ with (O1) and $O(1/\epsilon^3)$ with (O2). This is not surprising, as (O1) provides a gradient estimate whose variance scales inversely with the perturbation constant $\eta$, and this is unlike the estimate from (O2), where such an inverse scaling is absent. In the case of an unbiased gradient oracle, the sample complexity of our algorithm matches the result in [8]. An advantage with our approach is that, unlike [8], we do not require knowledge of the function value at the optima for choosing the perturbation constant $\eta$. We demonstrate the applicability of our biased gradient oracles by considering a risk-sensitive optimization problem in a reinforcement learning setting.

We propose a risk-sensitive policy gradient (Risk-PG) algorithm, and show that the analysis of the RSG algorithm with (O1) applies to the Risk-PG algorithm, while the results under (O2) would apply under additional assumptions.

Next, we summarize our contributions in the case when the objective is convex. Using a proof technique, which is similar to the one employed in the nonconvex case, we provide nonasymptotic bounds for the RSG algorithm with inputs from either (O1) or (O2). A disadvantage with the RSG algorithm is that it requires knowledge of the smoothness parameter for choosing the stepsize parameter in the gradient descent update iteration. We overcome this dependence by employing a different algorithm, which is based on the stochastic gradient descent (SGD) scheme analyzed in [10]. We provide nonasymptotic bounds that hold in expectation for the final iterate of the stochastic gradient (SG) algorithm with inputs from either (O1) or (O2). For the case of unbiased gradient information, Jain et al. [10] provided an iteration complexity bound of the order $O(1/\epsilon^2)$. We also provide a similar order bound, when the gradients are obtained from (O2). On the other hand, when gradient estimates from (O1) are employed, the bound we obtain is of the order $O(1/\epsilon^3)$. The latter bound matches a minimax lower bound established in [7].

### A. Related Work

Biased stochastic optimization has been considered before in [7], [11]–[16]. Devolder et al. [11], Baes [12], and d’Aspremont [13] considered an oracle that outputs biased gradient measurements without any noise component. In contrast, we consider noisy gradient measurements with a controllable estimation error. Hu et al. [7], which is a closely related work, formalized a biased noisy gradient oracle. They derive an upper bound for a mirror descent scheme and a minimax lower bound, both for the case of a convex objective. Unlike [7], our oracle model features an additional estimation error component, which is not zero mean. In the special case of an oracle that provides biased gradient information with no estimation error, our sample complexity matches the upper bound derived in [7]. Our bounds are for a regular SGD algorithm, with the added advantage that the stepsize we employ does not require knowledge of the underlying smoothness parameter. More importantly, unlike [7], we study stochastic nonconvex optimization problems with the biased gradient oracles mentioned before.

Balasubramanian and Ghadimi [9] derived a nonasymptotic bound for a zeroth-order variant of the stochastic conditional gradient algorithm under an oracle model similar to (O2), except that the estimation error component is absent. Specializing our results to have a zero-mean estimation error would make our bounds comparable to those in[9]. Nguyen et al. [14] considered an oracle model with a batch size parameter, and proposed an algorithm that estimates the gradient on a mini-batch of sufficient size. They provided an iteration complexity bound of $O(1/\epsilon^2)$ for the case of a convex function, and this matches our bound for the oracle (O2). Our bounds for the convex case are for the “practically preferred” last iterate, while those in [14] are for an iterate chosen uniformly at random. In addition, unlike [14], our stepsize choice does not require the knowledge of the underlying smoothness parameter.
Pasupathy et al. [16], which is another closely related work, studied SG methods in a setting where the objective was estimated using batch data, and a batch size of $m$ leads to an estimation error of $O(1/m^\alpha)$. Our biased gradient oracle framework is comparable to their setting, when $\alpha = 1/2$, and for this case, our nonasymptotic bounds match their asymptotic rate.

Liu et al. [17] considered the problem of optimizing a sum of smooth functions, and propose a variant of the well-known SVRG [18] algorithm for solving this problem. Unlike their work, we do not impose a finite sum structure on the objective, and more importantly, our setting involves biased function measurements.

The rest of this article is organized as follows. Section II formulates the biased gradient oracles, along with motivating applications. Section III considers the stochastic nonconvex optimization problem, and presents nonasymptotic bounds for a randomized gradient descent algorithm with inputs from a biased gradient oracle. Section IV considers the stochastic convex optimization problem, and presents nonasymptotic bounds for a SGD algorithm. Section V highlights the applicability of our biased gradient oracles in a risk-sensitive reinforcement learning setting. Section VI provides the proofs of all the bounds, which are presented in this article. Finally, Section VII concludes this article.

**Notation:** Throughout this article we assume $\| \cdot \| = \| \cdot \|$ in $\mathbb{R}^n$, $1_n \in \mathbb{R}^{n \times n}$ is an $n \times n$ matrix with each entry as one, and $x \leq y$ when applied to vectors $x$ and $y$ means $x_i \leq y_i \forall i$.

## II. BIASED GRADIENT ORACLES

Consider the following optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x)$$

where the function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is assumed to be smooth. Gradient-based methods are very popular for solving the optimization problem formulated above, and we consider an iterative algorithm, which obtains estimate of $\nabla f(\cdot)$ through calls to a biased gradient oracle. We define two such oracles in the following, and subsequently, we provide motivating applications featuring biased function measurements.

### A. Oracle Definitions

O1) **Biased Gradient Oracle:**

2) Input: $x \in \mathbb{R}^d$, perturbation constant $\eta > 0$, and batch size $m > 0$.

3) Output: A gradient estimate $g(x, \xi, \eta, m) \in \mathbb{R}^d$ that satisfies the following.

   1) $\|E_{\xi}[g(x, \xi, \eta, m)] - \nabla f(x)\|_\infty \leq c_1 \eta^2 + \frac{c_2}{\sqrt{m}}$

   2) $E_{\xi}[\|g(x, \xi, \eta, m) - E_{\xi}[g(x, \xi, \eta, m)]\|^2] \leq \frac{c_2}{\eta}$

   for some constants $c_1, c_2, c_3 > 0$.

In the above-mentioned oracle, the parameter $\eta$ is used to tradeoff bias and variance in the gradient estimates, while the parameter $m$ is motivated by practical models, where minibatching is used for estimating the objective function. To elaborate, the function measurements are biased, however, one could choose larger values of $m$ to increase the accuracy of the function measurements. The estimation bias contributes a $\frac{1}{m}$ factor in the abovementioned oracle (see Example 2 for a concrete application with such a rate).

Next, we present an alternative to (O1), where the bias of the gradient estimates can be reduced without adversely affecting the variance.

O2) **Biased Gradient Oracle-Variant:**

2) Input: $x \in \mathbb{R}^d$, perturbation constant $\eta > 0$, and batch size $m > 0$.

3) Output: A gradient estimate $g(x, \xi, \eta, m) \in \mathbb{R}^d$ that satisfies the following.

   1) $\|E_{\xi}[g(x, \xi, \eta, m)] - \nabla f(x)\|_\infty \leq c_1 \eta + \frac{c_2}{\eta \sqrt{m}}$

   2) $E_{\xi}[\|g(x, \xi, \eta, m) - E_{\xi}[g(x, \xi, \eta, m)]\|^2] \leq c_2 \eta^2 + \frac{c_2}{\eta}$

   for some positive constants $c_1, c_2, \tilde{c}_2$, and $c_3$.

### B. Illustrative Applications

Example 1: In the regular **simulation optimization** setting [2], we are given function measurements with zero-mean noise, i.e., $f(x) = E_{\xi}[f(x, \xi)]$. In contrast, we consider a model where the function measurements have an error term with positive mean. In this model, the objective $f$ is obtained as a solution to the following subproblem over the optimization variable $y$, which belongs to a convex and compact set $\mathcal{Y}$:

$$f(x) = \min_{y \in \mathcal{Y}} E_{\xi}[H_x(y, \xi)] \quad \forall x \in \mathbb{R}^d. \quad (2)$$

Owing to computational considerations, the abovementioned subproblem cannot be solved exactly. Instead, an optimization algorithm can obtain inexact measurements $F(x, m)$, defined by

$$F(x, m) = \min_{y \in \mathcal{Y}} E_{\xi}[H_x(y, \xi)] + \epsilon(m) \quad \forall x \in \mathbb{R}^d,$$

where $m$ is the batch size parameter and $\epsilon$ is a “positive” estimation error term. Choosing a larger batch size $m$ implies the subproblem in (2) can be solved more precisely, leading to lower estimation error $\epsilon(m)$.

The oracles (O1) and (O2) are also appealing in practical applications where the objective $f$ has to be estimated from i.i.d. samples coming from a random variable (r.v.) $X$, and the estimation scheme is biased. Estimation of risk measures, such as CVaR is an example of an application where the de facto estimation scheme is biased. We shall illustrate the applicability of our oracle in the context of CVaR objective as follows.

Example 2: We begin by defining the VaR $V_\alpha(X)$ and CVaR $C_\alpha(X)$, at a prespecified level $\alpha \in (0, 1)$ as

$$V_\alpha(X) = \inf \{\xi : P[X \leq \xi \geq \alpha]\},$$

$$C_\alpha(X) = V_\alpha(X) + \frac{1}{1-\alpha} E[X - V_\alpha(X)]^+$$

where $[X]^+ = \max(0, X)$. If the distribution underlying $X$ is continuous, then $C_\alpha(X) = E[X | X \geq V_\alpha(X)]$.

We now describe a well-known estimate of CVaR using $m$ i.i.d. samples $\{X_i, i = 1, \ldots, m\}$. Notice that CVaR estimation requires an estimate of VaR. Let $\hat{V}_m, \hat{C}_m$ denote the estimates of VaR and CVaR, respectively. These quantities are
defined as follows (see [19]):
\[ \hat{V}_{m,\alpha} = X_{[\lfloor ma \rfloor]}, \hat{C}_{m,\alpha} = \frac{1}{m} \sum_{i=1}^{m} X_i \mathbb{I}\{X_i \geq \hat{V}_{m,\alpha}\} \]  
(3)

where \( X_{[i]} \) denotes the \( i \)-th order statistic \( \forall i \). Notice that \( \mathbb{E}(\hat{C}_{m,\alpha}) \neq C_0(\alpha) \), since the VaR estimate in (3) is not unbiased. However, a recent CVaR concentration result in [20] shows that if the underlying r.v. \( X \) is \( \sigma \)-sub-Gaussian\(^1\), then, for any \( \epsilon > 0 \), the following inequality holds:
\[ \mathbb{P}(\{\hat{C}_{m,\alpha} - C_0(\alpha)\} > \epsilon) \leq c_1 \exp(-c_2 m \epsilon^2 (1 - \alpha^2)) \]  
(4)

where constants \( c_1 \) and \( c_2 \) depend on \( \sigma \). Using (4), we have
\[ \mathbb{E}\left[|\hat{C}_{m,\alpha} - C_0(\alpha)|\right] = \int_0^\infty \mathbb{P}(\{\hat{C}_{m,\alpha} - C_0(\alpha)\} > \epsilon) d\epsilon \leq \frac{c_3}{\sqrt{m}} \]

where \( c_3 > 0 \) is an absolute constant.

In both the abovementioned examples, the common element is biased function measurements. Using such measurements, one could construct gradient estimates using the SP method [3]. We make this construction precise as follows.

Let \( y^+(m) = f(x + \eta \Delta) + \xi^+(m) \) and \( y^-(m) = f(x - \eta \Delta) + \xi^-(m) \), where \( \xi^+(m) \) are the estimation errors assuming a batch size of \( m \), \( \eta \) is a perturbation constant, and \( \Delta = (\Delta^1, \ldots, \Delta^d)^T \) is a \( d \)-dimensional standard Gaussian vector.

For the two abovementioned examples, it is apparent that the estimation error is \( O\left(\frac{1}{\sqrt{m}}\right) \) in expectation, if \( m \) samples are used for estimation of \( f \) at \((x \pm \eta \Delta)\) input parameters.

A gradient estimate is formed using two function evaluations (i.e., \( y^+ \) and \( y^- \)) as follows:
\[ g(x, \xi, \eta, m) = \Delta \left[ \frac{y^+(m) - y^-(m)}{2 \eta} \right] \]  
(5)

where \( \Delta \) is a \( d \)-dimensional Gaussian vector composed of standard normal r.v.s. The abovementioned estimate is referred to as Gaussian smoothed functional, as well as Gaussian smoothing. This estimate was proposed in [21], and studied later in a convex optimization setting in [4]. A related estimate is random directions stochastic approximation (RDSA) with spherical perturbations, proposed in [22].

Assuming that the underlying function \( f \) is three-times continuously differentiable, we have
\[ f(x \pm \eta \Delta) = f(x) \pm \eta \Delta^T \nabla f(x) + \frac{\eta^2}{2} \Delta^T \nabla^2 f(x) \Delta + O(\eta^3). \]

Hence,
\[ \mathbb{E}\left[\Delta^T \left(\frac{f(x + \eta \Delta) - f(x - \eta \Delta)}{2 \eta}\right)\right] = \mathbb{E}\left[\Delta \Delta^T\right] \nabla f(x) + O(\eta^2) = \nabla f(x) + O(\eta^2) \]

where we used the fact that \( \mathbb{E}[\Delta \Delta^T] = I \), since \( \Delta \) is a standard Gaussian vector. Combining the abovementioned equality with the fact that the estimation error is \( O\left(\frac{1}{\sqrt{m}}\right) \), we obtain
\[ \|\mathbb{E}_\xi[g(x, \xi, \eta, m)] - \nabla f(x)\|_\infty \leq c_1 \eta^2 + \frac{c_2}{\sqrt{m}} \]

for some constants \( c_1 \) and \( c_2 \). This satisfies the requirement a) in (O1). The requirement in b) can be shown by squaring the estimator \( g(x^*, \cdot, \cdot, \cdot) \), assuming the objective of the function is bounded, leading to an inverse scaling with the perturbation constant \( \eta^2 \).

A similar argument works for the case of a convex and smooth objective as well. In addition, a variety of distributions can be employed for the random perturbations, cf. [3], [5], [7], [23], [24].

The oracle variant defined in (O2) can also be constructed using the estimator in (5). In particular, following arguments used in [4, Lem. 3] together with the fact that estimation error is of order \( O\left(\frac{1}{\sqrt{m}}\right) \) would lead to the condition a) in (O2). As mentioned before, in a regular simulation optimization setting, one observes a sample \( F(x, \xi) \) that satisfies \( \mathbb{E}[F(x, \xi)] = f(x) \) \( \forall \xi \). If the noise \( \xi \) has bounded variance, then one can ensure condition b) in (O1), where the variance of the estimator scales inversely with \( \eta^2 \). However, under additional assumptions, such as smoothness of \( F \), one can get rid of the inverse scaling with \( \eta \), avoiding the bias-variance tradeoff through the parameter \( \eta \).

A smooth \( F \) would ensure \( \nabla f(x) = \mathbb{E}[\nabla F(x, \xi)] \), which is a condition that is common in a perturbation analysis, cf. [25], [26]. For a proof that leads to condition b) in (O2), a straightforward variation of the proof of [9, Lem. B.1], which incorporates biased function measurements can be worked out, and we omit the details.

C. Performance Metrics

We consider SG-type algorithms for solving (1), with inputs from either (O1) or (O2). An SG algorithm runs for \( N \) iterations, and outputs a point \( x_R \), that could be chosen randomly from the iterates \( x_1, \ldots, x_N \). We study SG schemes in following two contexts.

1) The case when the objective \( f \) is convex.
2) The case when the objective \( f \) is not assumed to be convex.

In case 1), we provide bounds on the optimization error, i.e., \( f(x_R) - f(x^*) \), where \( x^* \) is a minima of \( f \). On the other hand, in case 2), i.e., when the objective is nonconvex, it is difficult to bound the optimization error. A popular alternative is to establish local convergence, i.e., to a point where the gradient of the objective is small (cf. [8] and [27]). The following definition makes the optimization objective apparent in both cases.

Definition 1: Let \( x_N \in \mathbb{R}^d \) be the output of the algorithm and \( \epsilon > 0 \) be a target accuracy.

1) For a nonconvex objective \( f \), \( x_N \) is called an \( \epsilon \)-stationary point of (1) if \( \|\mathbb{E}[\nabla f(x_N)]\|^2 \leq \epsilon. \)

2) For a convex objective \( f \), \( x_N \) is called an \( \epsilon \)-optimal point of (1), if \( \|f(x_N)\| - f(x^*) \leq \epsilon \), where \( x^* \) is an optimal solution to (1).

The SG algorithms are judged using the iteration as well as sample complexities, which are defined in the following.

---

\(^1\)An r.v. \( X \) is said to be \( \sigma \)-sub-Gaussian for some \( \sigma > 0 \) if \( \mathbb{E}[\exp(\lambda X)] \leq \exp(\frac{\lambda^2}{2} \sigma^2) \), for any \( \lambda \in \mathbb{R} \).
Definition 2: The iteration complexity of an algorithm \( A \) is the number of calls \( A \) makes to a biased gradient oracle before finding an \( \epsilon \)-stationary (resp. \( \epsilon \)-optimal) point for a nonconvex (resp. convex) objective function.

Definition 3: Suppose an algorithm \( A \) makes \( N \) calls to a biased gradient oracle before finding an \( \epsilon \)-stationary (resp. \( \epsilon \)-optimal) point for a nonconvex (resp. convex) objective function. Then, the sample complexity of \( A \) is \( \sum_{i=1}^{N} m_i \), where \( m_i, i = 1, \ldots, N \), is the batch size in iteration \( i \).

### III. STOCHASTIC NONCONVEX OPTIMIZATION

In this section, we consider the problem in (1), with an objective \( f \), which is smooth, but not necessarily convex. We analyze the nonasymptotic performance of the RSG algorithm [8], with inputs from a biased gradient oracle [either (O1) or (O2)]. The pseudocode for the algorithm is given in the following. This algorithm performs an incremental update as defined in (6), and outputs a random iterate, after \( N \) iterations.

**Algorithm 1: RSG-BGO.**

**Input:** Initial point \( x_1 \in \mathbb{R}^d \), iteration limit \( N \), stepizes \( \gamma_k \), perturbation constant \( \eta_k \), batch size \( m_k \), and probability mass function \( P_R(\cdot) \) of a r.v. \( R \), with support \( \{1, \ldots, N\} \).

for \( k = 1, \ldots, R \) do

Call the oracle (O1) or (O2) with \( x_k, \eta_k \) and \( m_k \), to obtain the gradient estimate \( g_k \).

Perform the following SG update:

\[
x_{k+1} = x_k - \gamma_k g_k (x_k, \xi_k, \eta_k, m_k).
\]  

end for

Return \( x_R \).

For the nonasymptotic analysis of the RSG algorithm, we make the following assumptions.

A1) The function \( f \) has Lipschitz continuous gradient with the constant \( L > 0 \), i.e.,

\[ \| \nabla f(x) - \nabla f(y) \| \leq L \| x - y \| \quad \forall x, y \in \mathbb{R}^d. \]

A2) There exists a constant \( B > 0 \) such that \(\| \nabla f(x) \|_1 \leq B \sqrt{d} x \in \mathbb{R}^d.\)

The smoothness assumption in (A1) is standard in the analysis of gradient-based algorithms (cf. [8], [9]). The boundedness requirement in (A2) is made in the context of zeroth-order optimization in [9], and can be inferred from the assumptions common to the analysis of policy-gradient algorithms in a reinforcement learning context, cf. [28].

We provide a nonasymptotic bound for the RSG-BGO algorithm with (O1) as follows. The bounds in this section, as well as those presented in Section IV, are for a random iterate \( x_R \), where \( R \) is uniformly distributed over \( \{1, \ldots, N\} \), and the expectation is taken with respect to \( R \) and noise \( \xi_{[N]} := (\xi_1, \ldots, \xi_N) \).

**Theorem 1: RSG-BGO Under (O1):**

Assume (A1) and (A2). With the oracle (O1), suppose that the RSG-BGO algorithm is run for \( N \) iterations with the stepsize \( \gamma_k \) and perturbation constant \( \eta_k \) set as follows \( \forall k \geq 1 \):

\[
\gamma_k = \min \left\{ \frac{1}{L} \frac{\gamma_0}{N^{2/3}}, \eta_k = \frac{\eta_0}{N^{2/3}} \right\}, \quad m_k = m_0 N^2
\]

for some constant \( \gamma_0, \eta_0 > 0 \).

1) If the batch size \( m_k = m_0 N, \forall k \geq 1 \), for some constant \( m_0 > 0 \), then, for any \( N \geq 1 \), we have

\[
E \| \nabla f(x_R) \|^2 \leq 2\frac{L^2 d L G}{N} + \frac{Z_1}{N^{1/3}} + \frac{Z_2}{N^{1/3}}
\]

where \( Z_1 = \frac{2 \sqrt{d} L \gamma_0^2}{N} + \frac{L \gamma_0 \sqrt{m_0}}{N} \), \( Z_2 = \frac{4 \sqrt{d} \gamma_0^2 m_0^2}{N} \), \( c_1, c_2, c_3 \) are as defined in (O1), \( B \) is as defined in (A2), and

\[ D_f = f(x_1) - f(x^*) \]

with \( x^* \) denoting an optimal solution to (1).

2) If the batch size \( m_k = m_0 k^\beta, \forall k \geq 1 \), for some constant \( \beta \in (0,1) \) and \( m_0 > 0 \), then, for any \( N \geq 1 \), we have

\[
E \| \nabla f(x_R) \|^2 \leq 2\frac{L^2 d L G}{N} + \frac{L \gamma_0 \eta_0^2}{\eta_0^2 m_0 N^2 \beta} + \frac{2 \sqrt{d} \gamma_0 \eta_0 m_0}{N^{2/3} \beta} (1 - \beta)
\]

where constants are the same as in part 1).

**Proof:** See Section VI-A1.

**Remark 1:** The overall rate, from the abovementioned bound, is \( O(N^{-1/3}) \), and this is not surprising because the bias of the gradient cannot be made arbitrarily small by setting \( \eta \) to a low value, as the variance of the gradient estimates scales inversely with \( \eta \). The (asymptotic) convergence rate results for SP stochastic approximation (SPSA) in [5], and RDSA in [23], also exhibit the same order, under an oracle, which is a variant to (O1) (without the estimation error component).

Using the bound in case 1) of Theorem 1, it is easy to see that the iteration complexity is \( O(\frac{1}{\epsilon^2}) \), and the sample complexity is \( \Theta(N^2) \).

**Remark 2:** For understanding the dimension dependence in the iteration complexity, let us consider the special case where (O1) is implemented using either SPSA [5] or RDSA [23]. In this case, \( c_1 = \kappa_1 d^3 \) and \( c_2 = \kappa_2 d \), where \( \kappa_1, \kappa_2 > 0 \) are dimension-independent constants. Choosing \( \gamma_0 = d^{-1/3}, \eta_0 = d^{-5/6}, \) and \( m_0 = 1 \) in (7), the overall iteration complexity of RSG-BGO turns out to be \( O(\frac{1}{\epsilon^2}) \).

We provide a nonasymptotic bound for the RSG-BGO algorithm with (O2) as follows.

**Theorem 2: RSG-BGO Under (O2):**

Assume (A1) and (A2). With the oracle (O2), suppose that the RSG-BGO algorithm is run for \( N \) iterations with the stepsize \( \gamma_k \), perturbation constant \( \eta_k \), and batch size \( m_k \) set as follows \( \forall k \geq 1 \):

\[
\gamma_k = \min \left\{ \frac{1}{L} \frac{\gamma_0}{N^{2/3}}, \eta_k = \frac{\eta_0}{N^{2/3}} \right\}, \quad m_k = m_0 N^2
\]

for some constants \( \gamma_0, \eta_0, m_0 > 0 \). Let \( x_R \) be chosen uniformly at random from \( \{x_1, \ldots, x_N\} \). Then, for any \( N \geq 1 \), we have

\[
E \| \nabla f(x_R) \|^2 \leq 2\frac{L^2 d L G}{N} + \frac{Z_3}{N^{1/3}}
\]

where \( Z_3 = \frac{4 \sqrt{d} \gamma_0^2 m_0}{N} \), \( c_1, c_2, c_3 \) are as defined in (O1), \( B \) is as defined in (A2), and\( D_f = f(x_1) - f(x^*) \).
where $Z_3 = \frac{2DL}{\gamma_0} + 4BZ_4 + L\gamma_0(\frac{d^2}{N} + \frac{c_1^2\eta_0^2}{N} + c_2^2)$, $Z_4 = c_1\eta_0 + \frac{c_3}{\eta_0\sqrt{m_0}}$, constants $c_1$, $c_2$, and $c_3$ are as defined in (O2), $B$ is as defined in (A2), and $D_f$ is as defined in (9).

Proof: See Section VI-A2.

From the bound in Theorem 2, it is easy to see that the iteration complexity is $O(\frac{1}{\gamma})$, and the sample complexity is $\Theta(N^3)$. This bound is better than the corresponding bound with (O1). We believe this improvement is because the variance of the gradient estimate in (O2) does not increase when bias is reduced.

Remark 3: Ghadimi and Lan [8] derived a nonasymptotic bound for a zeroth-order variant of their RSG algorithm under an oracle, which is a variant to (O2) (without the estimation error component). Our result in Theorem 2 matches their bound. Moreover, unlike [8], we derive a nonasymptotic bound for the oracle (O2), which involves an estimation error component.

An advantage with our analysis is that it allows the iterate $x_R$ to be picked uniformly at random from $\{x_1, \ldots, x_N\}$. The net effect is that of iterating averaging, except that the averaging happens in expectation.

Remark 4: For understanding the dimension dependence in the iteration complexity, let us consider the special case where (O2) is implemented using the Gaussian smoothing approach [4], [9]. In this case, $c_1 = \frac{L(d+3)}{2}$, $c_2 = \frac{L^2(d+3)^3}{d}$, and $c_3 = 2(d+5)(B^2 + \sigma^2)$, where $\sigma^2$ is the bound on variance of the estimator of $f(x)$. Choosing the stepsize $\gamma_0 = d^{-1/2}$, $\eta_0 = d^{-1}$, and $m_0 = d$ in (11), the overall iteration complexity of RSG-BGO turns out to be $O(\frac{d}{\gamma})$.

IV. STOCHASTIC CONVEX OPTIMIZATION

In this section, we consider the problem in (1), under the assumption that $f$ is a convex function. Let $x^* \in \mathbb{R}^d$ be a minimizer of the objective $f$. We first analyze the RSG-BGO algorithm in a convex setting, and subsequently present the SGD-BGO algorithm.

A. RSG Algorithm With a Biased Gradient Oracle

The pseudocode for this algorithm is presented in Algorithm 1. We provide a nonasymptotic bound for the RSG algorithm with (O1) as follows.

Theorem 3: RSG-BGO Under (O1):

Assume (A1). With the oracle (O1), suppose that the RSG-BGO algorithm is run for $N$ iterations with the batch size $m_k = m_0 N$ $\forall k \geq 1$, for some constant $m_0 > 0$ and stepsize $\gamma_k$, perturbation constant $\eta_k$ set as defined in (7). Let $x_R$ be chosen uniformly at random from $\{x_1, \ldots, x_N\}$. Then, for any $N \geq 1$, we have

$$E[f(x_R)] - f(x^*) \leq \frac{LD^2}{N} + \frac{BL}{N^{1/3}}$$

where $K_1 = \frac{D^2}{\gamma_0} + 4\sqrt{d}DK_2 + \frac{\gamma_0K_2^2}{N} + \frac{\gamma_0^2c_3}{N}$, $K_2 = (c_1\eta_0^2 + \frac{c_3}{\eta_0\sqrt{m_0}})$, and constants $c_1$, $c_2$, and $c_3$ are as defined in (O1), and

$$D = \|x_1 - x^*\|$$

(12)

with $x^*$ denoting an optimal solution to (1).

Proof: See Section VI-B1.

Remark 5: From the aforementioned results, we have that the iteration complexity is $O(\frac{1}{\gamma})$ and the sample complexity is $\Theta(N^2)$, and these complexities match that in Theorem 1 with a nonconvex objective. However, unlike the nonconvex case, here we bound the optimization error, i.e., $E[f(x_R)] - f(x^*)$. Moreover, we believe that the overall rate of $O(\frac{1}{\gamma})$ cannot be improved as Hu et al. [7] established that the aforementioned rate is best achievable in a minimax sense for a gradient-based algorithm with inputs from a biased gradient oracle, even if there is no bias in the function measurements.

We now provide a nonasymptotic bound for the RSG algorithm with (O2) for a convex objective.

Theorem 4: RSG-BGO Under (O2):

Assume (O1). With the oracle (O2), suppose that the RSG-BGO algorithm is run for $N$ iterations with the stepsize $\gamma_k$, perturbation constant $\eta_k$, and batch size $m_k$ set as defined in (11). Let $x_R$ be chosen uniformly at random from $\{x_1, \ldots, x_N\}$. Then, for any $N \geq 1$, we have

$$E[f(x_R)] - f(x^*) \leq \frac{LD^2}{N} + \frac{K_3}{\sqrt{N}}$$

where $K_3 = \frac{D^2}{\gamma_0} + 4\sqrt{d}DK_4 + \frac{\gamma_0K_4^2}{N} + \frac{2\gamma_0^2c_3}{N} + \gamma_0c_2$, $K_4 = c_1\eta_0 + \frac{c_3}{\eta_0\sqrt{m_0}}$, constants $c_1$, $c_2$, and $c_3$ are as defined in (O2), and $D$ is as defined in (12).

Proof: See Section VI-B2.

From the bound in Theorem 4, it is easy to see that the iteration complexity is $O(\frac{1}{\gamma})$ and the sample complexity is $\Theta(N^3)$.

B. SGD Algorithm With a Biased Gradient Oracle

In this section, we study an SGD algorithm with a biased gradient oracle. Unlike the RSG-BGO algorithm whose bounds were for a random iterate, the bounds that we derive for SGD-BGO are for the last iterate, which is the preferred point in practical implementations.

We consider a constrained variant of the problem (1), i.e., $\min_{x \in W} f(x)$, under the assumption that $W$ is a bounded convex set. This assumption is made precise as follows.

A3) The set $W$ is convex and compact. Furthermore, $\|x - y\| \leq D$ $\forall x, y \in W$, for some $D > 0$.

This assumption is necessary for the analysis of the SGD-BGO algorithm presented in Algorithm 2.

Following the approach from [10], we assume the knowledge of the total number of iterations $N$, and split the horizon $\bar{N}$ into $\bar{N}$ phases. The choice of phase lengths and the stepsize decay in each phase is performed along the lines of [10]. However, unlike their work that assumed unbiased gradient information, we operate in a setting where biased gradient information is available through (O1) or (O2), and this induces significant deviations in the proof. Moreover, our setting features a perturbation constant parameter, which has to be chosen in a phase-dependent manner as well.

We make the choice of phases precise as follows:

Let $l := \inf\{i : N \cdot 2^{-i} \leq 1\}$,

$$N_i := N - \lceil N \cdot 2^{-i} \rceil, \ 0 \leq i \leq l,$

and $N_{l+1} := N$. (13)
**Algorithm 2: SGD-BGO.**

**Input:** Initial point $x_1 \in \mathcal{W}$, iteration limit $N$, step sizes $\gamma_k$, perturbation constant $\eta_k$, batch size $m_k$ and projection operator $\Pi_W$.

for $k = 1, \ldots, N$

Call the oracle (O1) or (O2) with $x_k, \eta_k$ and $m_k$, to obtain the gradient estimate $g_k$.

Perform the following SG update:

$$x_{k+1} = \Pi_W \left( x_k - \gamma_k g(x_k, \xi_k, \eta_k, m_k) \right)$$

where $\Pi_W$ is an orthogonal projection operator that projects on to the closed convex set $\mathcal{W} \subset \mathbb{R}^d$.

**end for**

Return $x_N$.

From the abovementioned phase definitions, it can be seen that $N_2$ is an increasing sequence. Furthermore, $N_1 \approx \frac{N}{2}$, $N_2 \approx \frac{N}{2} + \frac{N}{4}$, and so on.

In the following result, we provide a nonasymptotic bound on the optimization error, i.e., $\mathbb{E}[f(x_N)] - f(x^*)$ for the SGD-BGO algorithm under (O1).

**Theorem 5: SGD-BGO Under (O1):**

Assume (A2) and (A3). With the oracle (O1), suppose that the SGD-BGO algorithm is run for $N$ iterations with the stepsize $\gamma_k$, perturbation constant $\eta_k$, and batch size $m_k$ set as follows:

$$\gamma_k = \frac{\gamma_0 \cdot 2^{-i}}{\sqrt{N}}, \quad \eta_k = \frac{\eta_0 \cdot 2^{-i/4}}{N^{1/6}}, \quad m_k = 2^{2}N$$

for some constant $\gamma_0, \eta_0 > 0$, when $N_i < k \leq N_{i+1}$, $0 \leq i \leq l$, with $N_i, l$ as defined in (13). Then, for any $N \geq 4$, we have

$$\mathbb{E}[f(x_N)] - f(x^*) \leq \frac{K_5}{N^{1/3}}$$

where

$$K_5 = \frac{4D^2}{\gamma_0} + \frac{11\gamma_0 B^2}{N^{1/3}} + 67 \sqrt{D} K_6 + \frac{20\sqrt{D}K_6}{N^{1/3}} + \frac{10d \alpha D K_6}{N^{1/3}} + \frac{18\gamma_0 \alpha_0}{\eta_0}, \quad K_6 = (c_1 \eta_0^2 + c_2 \eta_0)$$

and $c_1$, $c_2$, and $c_3$ are as defined in (O1), and $D$ is as defined in (A3).

**Proof:** See Section VI-C1.

Note that, unlike [10], parameters $\eta_k$ and $m_k$ are local to our setting, and due to the inverse scaling of variance in gradient estimates with $\eta_k$, the stepizes $\gamma_k$ chosen is of $O(\frac{1}{\sqrt{N_2 \eta}})$ and not $O(\frac{1}{\sqrt{N_2}})$.

From the bound in Theorem 5, it is easy to see that the iteration complexity of SGD-BGO is $O(\frac{1}{\epsilon^2})$ and the sample complexity is $\Theta(N^2 \log_2 N)$.

**Remark 6:** The analysis used in arriving at the bounds in Theorem 5 cannot be extended to the nonconvex case. This is because the analysis takes a dual viewpoint and approaches the minima of the objective from the following, which requires convexity. Intuitively, it may be challenging to provide bounds for the last iterate sans averaging in a nonconvex optimization setting, while it is possible to provide bounds for the averaged iterate (or the random iterate of RSG-BGO, which is an average in expectation) in the nonconvex case.

**Theorem 6: SGD-BGO Under (O2):**

Assume (A2) and (A3). With the oracle (O2), suppose that the SGD-BGO algorithm is run for $N$ iterations with the stepsize $\gamma_k$, perturbation constant $\eta_k$, and batch size $m_k$ set as follows:

$$\gamma_k = \frac{\gamma_0 \cdot 2^{-i}}{\sqrt{N}}, \quad \eta_k = \frac{\eta_0 \cdot 2^{-i}}{N}, \quad m_k = 2^{3}N^3$$

for some constant $\gamma_0, \eta_0 > 0$, when $N_i < k \leq N_{i+1}$, $0 \leq i \leq l$, with $N_i, l$ as defined in (13). Then, for any $N \geq 4$, we have

$$\mathbb{E}[f(x_N)] - f(x^*) \leq \frac{K_7}{\sqrt{N}}$$

where

$$K_7 = \frac{4D^2}{\gamma_0} + \frac{11\gamma_0 B^2}{N^{1/3}} + 39 \sqrt{D} \sqrt{K_6} + \frac{20\sqrt{D} \sqrt{K_6}}{N^{1/3}} + \frac{10d \alpha D \sqrt{K_6}}{N^{1/3}} + \frac{10\gamma_0 c_2 \sqrt{K_6}}{N} + \frac{3\gamma_0}{\eta_0}$$

and $c_1$, $c_2$, and $c_3$ are as in (O2), and $D$ is as defined in (A3).

**Proof:** See Section VI-C2.

From the bound in Theorem 6, it is easy to see that the iteration complexity of SGD-BGO is $O(\frac{1}{\epsilon^2})$ and the sample complexity is $\Theta(N^4(N^2 - 1))$. Furthermore, it is interesting to note that the bound with (O2) matches up to constant factors, the bound obtained in [10] for the case when unbiased gradient information is available. Unlike [8], where the authors provide a $O(\frac{1}{\epsilon^2})$ iteration complexity bound for a random iterate using the RSG-BGO algorithm, we provide bound for the last iterate of SGD-BGO. Apart from a practical preference for using the last iterate, an advantage with our approach is that for setting the step size $\gamma_k$ and perturbation constant $\eta_k$ in (15), we do not require the knowledge of the Lipschitz constant $L$ [see (O1)] and $D_X := \|x_1 - x^*\|$. The latter quantity is typically unavailable in practice, as it relates to the initial error.

One could specialize the bounds in Theorems 3–6 for the case when the underlying oracles are implemented using SPSA or GS methods, and we omit the details due to space constraints.

**V. APPLICATION: RISK-SENSITIVE REINFORCEMENT LEARNING**

We consider a stochastic shortest path problem, with a special cost-free absorbing state, say 0. We restrict our attention to the policy considered. We define an episode as a sample path $\{a_0, \ldots, a_T\}$, where $a_T = 0$ and $\tau$ is the first passage time to state 0.

Consider a smoothly parameterized class of policies $\{\pi_x \mid x \in \mathbb{R}^d\}$. Suppose that the policy $\pi_x$ is a continuously differentiable function of the parameter $x$, a standard assumption in the policy gradient literature. Let $K_x(x^0)$ denote the total discounted cost r.v. under the policy $x$ starting in state $x^0$, i.e.,

$$K_x(x^0) = \sum_{t=0}^{\tau-1} \gamma^t k(x_t, a_t), \quad 0 < \gamma < 1$$

is the discount factor and $k(x_t, a_t)$ is the single-stage cost incurred at time instant $t$ in state $x_t$ on choosing action $a_t$. Here, actions $a_t$ are chosen according to policy $\pi_x$, which is parameterized by $x$.

The classic objective in RL is to find a policy that minimizes, in expectation, the total discounted cost. We consider a risk-sensitive RL setting, where the goal is to find a policy that optimizes a certain risk measure, i.e., the following problem:

$$\min_{x \in \mathbb{R}^d} \{\rho(K_x(x^0))\}$$

(16)
where \( x \in \mathbb{R}^d \) parameterizes the policy \( \pi_x \) and \( \rho \) is a risk measure. As examples for the risk measure, one could consider CVaR [1], a utility-based shortfall risk (UBSR) [29], and spectral risk measure (SRM) [30]. Notice that the optimization problem in (16) is nonconvex in nature. For solving the abovementioned problem using gradient-based methods, one requires the following.

1) An estimate of the risk measure for any given policy \( \pi_x \).
2) An estimate of the gradient of the risk measure w.r.t. the policy parameter \( x \).

We elaborate on these two parts as follows.

We simulate \( m \) episodes simulated using the policy \( \pi_x \), and collect samples of the total cost \( K_x(x^0) \).

Using these samples, define the empirical distribution function (EDF) \( F_m \) of \( K_x(x^0) \) as follows: \( F_m(x) = \frac{1}{m} \sum_{i=1}^{m} I_{K_x(x^0) \leq x} \) for any \( x \in \mathbb{R} \). Using the EDF, we form the estimate \( \rho_m \) of \( \rho(K_x(x^0)) \) as follows:

\[
\rho_m = \rho(F_m).
\]

Such as an estimation scheme for an abstract risk measure has been considered earlier in [31] and [32].

Next, we present a bound in expectation for the estimation error associated with (17).

**Proposition 1:** Suppose the risk measure \( \rho \) satisfies the following continuity requirement for any two distributions \( F \) and \( G \):

\[
|\rho(F) - \rho(G)| \leq LW_1(F, G)
\]

where \( W_1(F, G) \) is the Wasserstein distance\(^2\) between distributions \( F \) and \( G \) and \( L \) is as defined in (A1). Suppose the r.v. \( K_x(x^0) \) satisfies \( \mathbb{E}[K_x(x^0)^2] \leq B < \infty \), for any \( x \in \mathbb{R}^d \). Then,

\[
\mathbb{E} \left[ |\rho_m - \rho(K_x(x^0))| \right] \leq \frac{c}{\sqrt{m}}
\]

for some constant \( c \) that depends on \( B \).

**Proof:** See Section VI-D. □

The continuity requirement in (18) is satisfied by the three popular risk measures CVaR, UBSR, and SRM, and the reader is referred to [32] for details.

To construct an estimate of the gradient of the risk measure \( \rho(K_x(x^0)) \), one could employ the SP method, e.g., the estimate in (5). Using this gradient estimate and the template of the RSG-BGO algorithm, we arrive at the following update iteration for the Risk-PG algorithm:

\[
x_{k+1} = x_k - \gamma_k g_k
\]

where \( \gamma_k \) is the stepsize and \( g_k \) is an estimate of the gradient of the risk measure \( \rho(K_{x_k}(x^0)) \). To elaborate on the gradient estimation aspect of the abovementioned algorithm, we first simulate \( m_k \) trajectories of the underlying MDP with policy patters \( x_k + \eta_k \Delta_k \) and \( x_k - \eta_k \Delta_k \), respectively. Here, \( \eta_k \) is the perturbation constant and \( \Delta_k \) is a standard Gaussian vector. Using (17), we estimate the risk measures \( \rho(K_{x_k+\eta_k \Delta_k}(x^0)) \) corresponding to the aforementioned policy parameters, and then, use (5) to form \( g_k \).

For applying the nonasymptotic bounds derived earlier for RSG-BGO, we require (A1) to hold. We verify this assumption for the special case of CVaR as follows. Let \( C_\alpha(K_x(x^0)) \) denote the CVaR associated with a policy \( \pi \), which is parameterized by \( x \). Then, using the likelihood ratio method, we arrive at the following variant of the policy gradient theorem under the CVaR objective (cf. [33]):

\[
\nabla_x C_\alpha(K_x(x^0)) = \mathbb{E} \left[ (K_x(x^0) - V_\alpha(K_x(x^0))) \right]
\]

\[
\sum_{m=0}^{\tau-1} \nabla \log \pi(x_m | x_m) \left[ K_x(x^0) \geq V_\alpha(K_x(x^0)) \right].
\]

In the above, \( K_x(x^0) \) and term (I) on the RHS of the equation are Lipschitz functions due to the abovementioned policy gradient assumption. In addition, if we assume that the distribution, say \( F_x \), of \( K_x(x^0) \) is a Lipschitz function in \( x \), then we can infer that \( \nabla_x C_\alpha(K_x(x^0)) \) is the sum of product of Lipschitz functions, implying assumption (A1). One could generalize this argument to the case when \( \rho \) is a coherent risk measure, and the reader is referred to [33] for details.

Using the nonasymptotic bounds for the RSG-BGO algorithm derived earlier, we can infer that the iteration complexity for the Risk-PG algorithm is as follows.

1) \( O(\frac{1}{\gamma}) \) under the oracle (O1) (using Theorem 1).
2) \( O(\frac{1}{\gamma}) \) under the oracle (O2) (using Theorem 2).

**VI. CONVERGENCE PROOFS**

For notational convenience, we shall use \( g_k = g(x_k, \xi_k, \eta_k, m_k) \) \( \forall k \geq 1 \). Let \( \xi[k] := (\xi_1, \ldots, \xi_k) \) and \( \mathbb{E}[\xi[k]] \) denote the expectation w.r.t. \( \xi[k] \).

**A. Proofs for the RSG-BGO Algorithm With a Nonconvex Objective**

1) **Proof of Theorem 1:** In the following proposition, we state and prove a general result that holds for any choice of nonincreasing stepsize sequence, perturbation constants, and batch sizes. Subsequently, we specialize the result for the choice of parameters suggested in Theorem 1, to prove the same.

Proposition 2: Assume (A1) and (A2). With the oracle (O1), suppose that the RSG-BGO algorithm is run with a nonincreasing stepsize sequence satisfying \( 0 < \gamma_k \leq 1/L \) \( \forall k \geq 1 \) and with the probability mass function \( P_R(\cdot) \), given by

\[
P_R(k) = \frac{\gamma_k}{\sum_{i=1}^{N-1} \gamma_i}, k = 1, \ldots, N.
\]

Then, for any \( N \geq 1 \), we have

\[
\mathbb{E} \left[ \| \nabla f(x_R) \|^2 \right] \leq \frac{1}{\sum_{k=1}^{N} \gamma_k} \left[ \frac{2D_f}{(2 - L\gamma_1)} \right] + \frac{2D}{\gamma_k} \sum_{k=1}^{N} \frac{L_k^2}{(2 - L\gamma_k)} \left[ d\mathcal{E}_k^2 + \frac{\epsilon_k}{\eta_k} \right]
\]

(20)
where $\mathcal{E}_k = c_1 \eta_k^2 + \frac{c_3}{\eta_k \sqrt{m}}$, constants $c_1$, $c_2$, and $c_3$ are as defined in (O1), $B$ is as defined in (A2), and $D_f$ is given in (9).

**Proof:** We use the technique from [8]. However, our proof involves significant deviations owing to the fact that the gradient estimates in (O1) have variance that scales inversely with perturbation constant $\gamma_k$. Furthermore, unlike [8], we have the batch size $m_k$ parameter that needs to be optimized. Letting $\Delta_k = g_k - \nabla f(x_k)$ \( \forall k \geq 1 \), we have

$$
\|x_{k+1} - x_k\| = \|x_k - \gamma_k g_k - x_k\| = \gamma_k \|g_k\|
$$

and

$$
\mathbb{E}_{\xi[k]} [\Delta_k] = \mathbb{E}_{\xi_k} [\Delta_k | \xi[k-1]] = \mathbb{E}_{\xi_k} [\Delta_k | x_k]
$$

$$
= \mathbb{E}_{\xi_k} [g_k - \nabla f(x_k) | x_k] \leq \mathbb{E} \mathbf{1}_{d \times 1}
$$

(21)

where $\mathcal{E}_k = c_1 \eta_k^2 + \frac{c_3}{\eta_k \sqrt{m}}$, and

$$
\mathbb{E}_{\xi_k} [||g_k||^2] \leq \mathbb{E}_{\xi[k]} [g_k]_2^2 + c_2 / \eta_k^2.
$$

(22)

Under assumption (O1), we have

$$
f (x_{k+1})
\leq f (x_k) + \langle \nabla f (x_k), x_{k+1} - x_k \rangle + \frac{L}{2} \|x_{k+1} - x_k\|^2
$$

$$
= f (x_k) - \gamma_k (\nabla f (x_k), g_k) + \frac{L}{2} \gamma_k^2 \|g_k\|^2
$$

$$
= f (x_k) - \gamma_k \|\nabla f (x_k)\|^2 - \gamma_k \langle \nabla f (x_k), \Delta_k \rangle
$$

$$
+ \frac{L}{2} \gamma_k^2 \|g_k\|^2.
$$

(23)

Taking expectations with respect to $\xi[k]$ on both sides of (23) and using (21) and (22), we obtain

$$
\mathbb{E} \left[ f (x_{k+1}) \right]
\leq \mathbb{E} \left[ f (x_k) \right] - \gamma_k \mathbb{E} \left[ \|\nabla f (x_k)\|^2 \right]
$$

$$
- \gamma_k \mathbb{E} \left[ \langle \nabla f (x_k), \Delta_k \rangle \right] + \frac{L}{2} \gamma_k^2 \left[ \mathbb{E} \|g_k\|^2 + c_2 / \eta_k^2 \right]
$$

$$
\leq f (x_k) - \gamma_k \|\nabla f (x_k)\|^2 + \gamma_k \mathbb{E} \|\nabla f (x_k)\|_1
$$

$$
+ \frac{L}{2} \gamma_k^2 \|\nabla f (x_k)\|^2 + 2 \mathcal{E}_k \mathbb{E} \|\nabla f (x_k)\|_1
$$

$$
+ d \mathcal{E}_k^2 + c_2 / \eta_k^2
$$

$$
\leq f (x_k) - \left( \gamma_k - \frac{L}{2} \gamma_k^2 \right) \|\nabla f (x_k)\|^2
$$

$$
+ \mathcal{E}_k B \left( \gamma_k + L \gamma_k^2 \right) + \frac{L}{2} \gamma_k^2 \left[ d \mathcal{E}_k^2 + c_2 / \eta_k^2 \right]
$$

(24)

where we have used the fact that $-\|X\|_1 \leq \sum_{i=1}^d x_i$ for any vector $X$ in arriving at the inequality (24). The last inequality follows from (A2). Rearranging the terms, we obtain

$$
\gamma_k \|\nabla f (x_k)\|^2 \leq \frac{2}{2 - L \gamma_k} \left[ f (x_k) - \mathbb{E} [f (x_{k+1})] \right]
$$

$$
+ \mathcal{E}_k \left( \gamma_k + L \gamma_k^2 \right) B + \frac{L \gamma_k^2}{2 - L \gamma_k} \left[ d \mathcal{E}_k^2 + c_2 / \eta_k^2 \right].
$$

Now, summing up the above-mentioned inequality over $k = 1$ to $N$, and taking expectations with respect to the sigma-field generated by $\xi_1, \ldots, \xi_N$, we obtain

$$
\sum_{k=1}^N \gamma_k \mathbb{E}_{\xi[k]} \|\nabla f (x_k)\|^2
\leq 2 \sum_{k=1}^N \frac{\left[ \mathbb{E}_{\xi[k]} f (x_k) - \mathbb{E}_{\xi[k]} f (x_{k+1}) \right]}{(2 - L \gamma_k)}
$$

$$
+ 2 \sum_{k=1}^N \mathcal{E}_k B \left( \gamma_k + L \gamma_k^2 \right) + \frac{L \gamma_k^2}{2 - L \gamma_k} \left[ d \mathcal{E}_k^2 + c_2 / \eta_k^2 \right]
$$

$$
= 2 \left[ \frac{f (x_1)}{(2 - L \gamma_1)} - \sum_{k=2}^N \frac{\left( \mathbb{E}_{\xi[k]} f (x_k) - \mathbb{E}_{\xi[k]} f (x_{k+1}) \right)}{(2 - L \gamma_k)} \right]
$$

$$
- \sum_{k=1}^N \mathcal{E}_k B \left( \gamma_k + L \gamma_k^2 \right)
$$

$$
+ \frac{L \gamma_k^2}{2 - L \gamma_k} \left[ d \mathcal{E}_k^2 + c_2 / \eta_k^2 \right].
$$

(25)

Noting that $\mathbb{E}_{\xi[k]} [f (x_k)] \geq f (x^\ast)$ and $\left( \frac{1}{(2 - L \gamma_k)} \right) \geq 0$, we obtain

$$
\sum_{k=1}^N \gamma_k \mathbb{E}_{\xi[k]} \|\nabla f (x_k)\|^2
\leq \frac{2 \left[ f (x_1) - f (x^\ast) \right]}{(2 - L \gamma_1)} + 2 \sum_{k=1}^N \mathcal{E}_k B \left( \gamma_k + L \gamma_k^2 \right)
$$

$$
+ \frac{L \gamma_k^2}{2 - L \gamma_k} \left[ d \mathcal{E}_k^2 + c_2 / \eta_k^2 \right].
$$

(26)

The bound in (20) follows by using the distribution of $R$ [specified in (19)] in the abovementioned equation’s RHS.

We now specialize the result obtained in the abovementioned proposition, to derive the bounds in Theorem 1.

**Proof:** [Theorem 1 i)] Recall the stepsize $\gamma$, perturbation constant $\eta$ from (7), and batch size $m_k \equiv m = m_0N \forall k \geq 1$, for some constant $m_0 > 0$. Let $\mathcal{E} = c_1 \eta^2 + \frac{c_3}{\eta \sqrt{m}}$. Using $\gamma_k \equiv \gamma$ in (20), we obtain

$$
\mathbb{E} \left[ \|\nabla f (x_R)\|^2 \right]
$$

$$
\leq \frac{1}{N \gamma} \left[ 2 D f + 4 N \gamma B \mathbb{E} + L N \gamma^2 \left[ d \mathcal{E}^2 + c_2 / \eta^2 \right] \right]
$$

$$
\leq \frac{2 D f}{N} \max \left\{ L, \frac{N^{2/3}}{\gamma_0} \right\}
$$

$$
+ 4 B \left( \frac{c_1 \eta_0^2}{N^{1/3}} + \frac{c_3}{\eta_0 \sqrt{m_0 N^{1/3}}} \right) + \frac{L \gamma_0}{N^{2/3}} \left[ d \mathcal{E}^2 + c_2 / \eta_0^4 \right]
$$

$$
+ 2 d c_1 \gamma_0 \mathbb{E} \left[ \nabla f \right] + \frac{d c_2}{\eta_0^2 m_0 N^{2/3}} + \frac{c_2}{\eta_0^2 N^{1/3}}.
$$

(27)

In the above, inequality (25) follows by using the fact that $\gamma \leq 1 / L$, while the inequality (26) follows by using the definition
of $E, \gamma, \eta$, and $m$. The bound in (8) follows by rearranging the terms in (26).

**Proof:** (Theorem 1 ii) Recall the stepsize $\gamma$ and perturbation constant $\eta$ from (7). Let $m_k = m_0 k^3 \gamma_k \geq 1$, for some constant $\beta \in (0, 1)$ and $m_0 > 0$. Using $\gamma_k \equiv \gamma$ in (20), we obtain

\[
\mathbb{E} \left[ \| \nabla f(x_R) \|^2 \right] \leq \frac{1}{N \gamma} \left( 2D^2 f + 4N \gamma B c_3 \eta^2 + 4 \frac{\gamma B c_3}{\eta \sqrt{m_0}} \sum_{k=1}^{N} k^{-\beta} \right)
\]

\[
+ L N \gamma^2 \left( \frac{dc_2^2 \eta^4}{\eta^2} + \frac{c_2}{\eta^2} \right) + \frac{2 L d^3 \gamma N c_3 \eta}{\sqrt{m_0}} \left( \frac{N - \frac{2}{\gamma} + 1}{\gamma \sqrt{m_0}} \right)
\]

\[
+ \frac{L d^3 c_3^2}{N \eta^2 m_0} \left( \frac{N - \frac{2}{\gamma} + 1}{\gamma \sqrt{m_0}} \right)
\]

\[
+ \frac{2 D f}{N \gamma} + 4 B c_3 \eta^2 + 4 B c_3 \left( \frac{N - \frac{2}{\gamma} + 1}{\gamma \sqrt{m_0}} \right)
\]

\[
+ L \gamma \left( \frac{dc_2^3 \eta^4}{\eta^2} + \frac{c_2}{\eta^2} \right) + \frac{2 L d^3 \gamma N c_3 \eta}{\sqrt{m_0}} \left( \frac{N - \frac{2}{\gamma} + 1}{\gamma \sqrt{m_0}} \right)
\]

\[
+ \frac{L d^3 c_3^2}{N \eta^2 m_0} \left( \frac{N - \frac{2}{\gamma} + 1}{\gamma \sqrt{m_0}} \right)
\]

In the above, inequality (27) follows by using the fact that $\gamma \leq 1/L$ and $m_k = k^3$. The inequality (28) follows by using the definition of $\gamma$ and $\eta$. The bound in (10) follows by rearranging the terms in (28). \[\Box\]

**2) Proof of Theorem 2:** Proof: By a parallel argument to the proof of Proposition 2, we obtain

\[
\mathbb{E} \left[ \| \nabla f(x_R) \|^2 \right] \leq \frac{1}{N \gamma} \left( \frac{2Df}{(2 - L \gamma_1)} \right)
\]

\[
+ 2 \sum_{k=1}^{N} \mathbb{E}_k \left( \frac{\gamma_k + L \gamma_k^2}{2 - L \gamma_k} \right) \mathbb{E}_{\xi(N)} \| \nabla f(x_k) \|_1
\]

\[
+ L \sum_{k=1}^{N} \left( \frac{\gamma_k}{2 - L \gamma_k} \right) \left[ d \nabla^2 \gamma_k + c_2 \eta_k^2 + \tilde{c}_2 \right]
\]

where $\mathbb{E}_k = c_1 \eta_k + \frac{c_2}{\eta_k \sqrt{m_0}}$. Using arguments similar to those employed in the proof of Theorem 1, we obtain

\[
\mathbb{E} \left[ \| \nabla f(x_R) \|^2 \right] \leq \frac{2Df}{N \gamma} + 4 B \mathbb{E} + L \gamma \left[ d \nabla^2 \gamma + c_2 \eta^2 + \tilde{c}_2 \right].
\]

The main claim follows by plugging values of $\gamma, \eta$, and $m$, as defined in Theorem 2 in the abovementioned equation. \[\Box\]

**B. Proofs for the RSG-BGO Algorithm With Convex Objective**

1) **Proof of Theorem 3 3:** Proof: Letting $w_k = \| x_k - x^* \|$, and using the update iteration (6), we have

\[
\omega_{k+1}^2 = \omega_k^2 - 2 \gamma_k (g_k, x_k - x^*) + \frac{\gamma_k^2}{2} \| g_k \|^2.
\]

Taking expectations with respect to $\xi_{k}$ on the both sides of (29) and using (21) and (22), we obtain

\[
\mathbb{E} [\omega_{k+1}^2] \leq \mathbb{E} [\omega_k^2]
\]

\[
- 2 \gamma_k \langle \nabla f(x_k), x_k - x^* \rangle + 2 \gamma_k \mathbb{E}_k \| x_k - x^* \|_1
\]

\[
+ \gamma_k^2 \left[ \| \nabla f(x_k) \|^2 + 2 \sqrt{d \mathbb{E}} \| \nabla f(x_k) \| + d \mathbb{E}_k^2 + \frac{c_2}{\eta_k^2} \right]
\]

where $\mathbb{E}_k = c_1 \sqrt{\eta_k} + \frac{c_2}{\eta_k \sqrt{m_0}}$ and constants $c_1$, $c_2$, and $c_3$ are as defined in (O1). Since $f$ is convex, we have $\| \nabla f(x_k) \|^2 \leq L \langle \nabla f(x_k), x_k - x^* \rangle \leq L (f(x^*) - f(x_k))$. Using this fact in the abovementioned equation’s RHS, rearranging terms, and then summing over $k = 1, \ldots, N$, we obtain

\[
\mathbb{E} \left[ f(x_R) - f(x^*) \right]
\]

\[
\leq \frac{1}{\sum_{k=1}^{N} \gamma_k} \left( \frac{D^2}{(2 - L \gamma_1)} + 4 \sqrt{d \mathbb{E}} \mathbb{E} + \gamma_k \left[ d \mathbb{E}_k^2 + \frac{c_2}{\eta_k^2} \right] \right)
\]

\[
+ \sum_{k=1}^{N} \left( \frac{\gamma_k^2}{2 - L \gamma_k} \right) \left[ d \mathbb{E}_k^2 + \frac{c_2}{\eta_k^2} \right]
\]

where $D$ is as defined in (12). Now, following the steps used in the proof of Theorem 1, we obtain

\[
\mathbb{E} \left[ f(x_R) - f(x^*) \right]
\]

\[
\leq \frac{L d^2}{N \gamma} + \frac{1}{N^{1/3}} \left[ \frac{D^2}{70} + 4 \sqrt{d \mathbb{E}} (c_1 \eta_k^2 + \frac{c_3}{\eta_k \sqrt{m_0}}) \right]
\]

\[
+ \gamma_0 \left( \frac{d \eta_k^4}{N} + \frac{2 d c_1 \eta_k c_3}{\sqrt{m_0} N} + \frac{d \mathbb{E}_k^2}{\eta_k^2 m_0 N} + \frac{c_2}{\eta_k^2} \right)
\]

where the last inequality follows by plugging values of $\gamma, \eta$, and $m$, which are defined in (7). \[\Box\]

2) **Proof of Theorem 4:** Proof: Following the initial passage in the proof of Theorem 3 with (O1), we obtain the following inequality for the case of (O2) considered here:

\[
\mathbb{E} \left[ f(x_R) - f(x^*) \right]
\]

\[
\leq \frac{1}{\sum_{k=1}^{N} \gamma_k} \left( \frac{D^2}{(2 - L \gamma_1)} + 2 \sqrt{d \mathbb{E}} \mathbb{E} + \gamma_k \left[ d \mathbb{E}_k^2 + \frac{c_2}{\eta_k^2} \right] \right)
\]

\[
+ \sum_{k=1}^{N} \left( \frac{\gamma_k^2}{2 - L \gamma_k} \right) \left[ d \mathbb{E}_k^2 + \frac{c_2}{\eta_k^2} \right]
\]

3Reader is referred to the longer version of this article, available in [34] for a complete proof.
where \( E_k = c_1 \eta_k + \frac{c_2}{\eta_k \sqrt{N}} \). Now, using the choice of parameters in (11), followed by simplifications similar to those in the proof of Theorem 3, we obtain

\[
E [ f (x_R) ] - f (x^*) 
\leq \frac{D^2}{N \gamma} + 4 \sqrt{d} dE + \gamma [ dE^2 + c_2 \eta^2 + c_2 ]
\]

\[
\leq \frac{LD^2}{N} + \frac{1}{\sqrt{N}} \left[ \frac{P^2}{\gamma_0} + 4 \sqrt{d} D \left( c_1 \eta_0 + \frac{c_3}{\eta_0 \sqrt{m_0}} \right) \right]
\]

\[
+ \gamma_0 \left( \frac{d c_1 N_0}{N} + \frac{d c_2}{\eta_0 m_0} + \frac{c_2 N_0 + \eta_2}{N} + \epsilon \right).
\]

C. Proofs for the SGD-BGO Algorithm

1) Proof of Theorem 5: We follow the technique from [10], and our proof involves significant deviations owing to the fact that unbiased gradient information is not available, leading to additional terms involving perturbation constants and batch sizes.

We start with a variant of Lemma 1 from [10].

**Lemma 1:** Assume (A2) and (A3). With the oracle (O1), suppose that the SGD-BGO algorithm is run with stepsize sequence \( \{ \gamma_k \}_{k=1}^K \). Then, given any \( 1 < k_0 < k_1 \leq N \), we have

\[
\sum_{k=k_0}^{k_1} 2 \gamma_k E [ f (x_k) - f (x_{k_0}) ] \leq \sum_{k=k_0}^{k_1} 2 \sqrt{d} \gamma_k dE_k + \gamma_k^2 B_k^2
\]

where \( E_k = c_1 \eta_k^2 + \frac{c_2}{\eta_k \sqrt{m_0}} \), \( B_k^2 = \left[ B^2 + 2 \sqrt{d} \gamma_k dE_k + d \eta^2_k + c_2 \right] \), constants \( c_1 \) and \( c_2 \) are as defined in (O1), and \( D \) is as defined in (A3).

**Proof:** Let \( \Delta_k = g_k - \nabla f (x_k) \) and \( \omega_k = \| x_k - x_{k_0} \| \), \( \forall k \geq 1 \). Then, for \( k = 1, \ldots, N \), we have

\[
\omega_k^2 = \| x_{k+1} - x_k \| \leq \| x_k - x_{k_0} \| \leq \| x_k - g_k \| \\
\leq \| x_k - \gamma_k g_k - x_{k_0} \| \leq \| \Delta_k + \gamma_k g_k - x_{k_0} \| \\
= \omega_k^2 - 2 \gamma_k \| \Delta_k \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
= \omega_k^2 - 2 \gamma_k \| \nabla f (x_k) + \Delta_k, x_k - x_{k_0} \| + \gamma_k^2 \| g_k \| \\
\[-2\gamma_k(x_{k+1} - x^*)^T \Delta_k] + \frac{\gamma_k B^2}{2}.

Taking expectations and using \(\|y_{k+1} - x^*\| \geq \|x_{k+1} - x^*\|\) (see [35, Lem. 3.1]), we obtain
\[
\mathbb{E}[f(x_k) - f(x^*)] \
\leq \frac{1}{2\gamma_k} \left( \mathbb{E}[\|x_k - x^*\|^2] - \mathbb{E}[\|x_{k+1} - x^*\|^2] \right) + 2\gamma_k \mathbb{E}\left[\| (x_{k+1} - x^*)_1 \|_2 \| \Delta_k \|_\infty \right] + \frac{\gamma_k B^2}{2}
\leq \frac{1}{2\gamma_k} \left( \mathbb{E}[\|x_k - x^*\|^2] - \mathbb{E}[\|x_{k+1} - x^*\|^2] \right) + 2\gamma_k \sqrt{d \mathbb{E}[\|x_{k+1} - x^*\|]} + \frac{\gamma_k B^2}{2}
\leq \frac{1}{2\gamma_k} \left( \mathbb{E}[\|x_k - x^*\|^2] - \mathbb{E}[\|x_{k+1} - x^*\|^2] \right) + 2\gamma_k \sqrt{d \mathbb{E}[\|x_{k+1} - x^*\|]} \tag{33}
\]

where the last inequality follows from the fact that \(-\sum_{i=1}^d x_i \leq \|X\| \leq \sqrt{d} \|X\|\) for any vector \(X\). We conclude by summing (33) over \(k\), with \(\gamma_k = \gamma, \eta_k = \eta\), and using (A3), i.e., \(\|x_k - x^*\| \leq D\ \forall k \geq 1\).

\(\Box\)

Proof: (Theorem 5) Recall the definition of \(N_i, l\) from (13) and let \(n_i, 0 \leq i \leq l + 1\), be defined as follows:
\[n_i = \arg\inf_{N_i < k \leq N_{i+1}} \mathbb{E}[f(x_k)], i \in [l + 1]
and \(n_0 = \arg\inf_{\frac{K}{2} \leq k \leq N_1} \mathbb{E}[f(x_k)].\)

We split the horizon \(N\) into \(l\) phases, then show that the function value for the final iterate \(x_{N_i}\) in the last phase \((N_{i+1} = N)\) is close to optima \(x^*\). Using the fact that \(n_{l+1} = N\), we have
\[\mathbb{E}[f(x_N)] = \mathbb{E}[f(x_{n_0})] + \sum_{i=0}^l \mathbb{E}[f(x_{n_i+1}) - f(x_{n_i})]. \tag{34}\]

Now to bound \(\mathbb{E}[f(x_{n_i+1}) - f(x_{n_i})]\), we first consider the case when \(i \geq 1\). Using Lemma 1 with \(k_0 = n_i\) and \(k_1 = N_{i+2}\), we obtain
\[\sum_{k=n_i}^{N_{i+2}} \frac{2\gamma_k \mathbb{E}[f(x_k) - f(x_{n_i})]}{N_{i+2} - n_i + 1} \leq \frac{\sum_{k=n_i}^{N_{i+2}} \left( \frac{\sqrt{d} \mathbb{E}[x_k - x^*] + \gamma_k^2 B^2}{\gamma_k} \right)}{N_{i+2} - n_i + 1} \leq 2\sqrt{\gamma N_{i+1} \mathbb{E}[x_{n_i+1}] + B^2 \gamma_{N_{i+1}}^2} + \frac{2\gamma_{N_{i+1}} B^2}{\gamma_{N_{i+1}}^2} \tag{35}\]
\[= \frac{2\sqrt{\gamma N_{i+1} \mathbb{E}[x_{n_i+1}] + B^2 \gamma_{N_{i+1}}^2}}{N_{i+1}^2} + \frac{2\gamma_{N_{i+1}} B^2}{\gamma_{N_{i+1}}^2} \tag{36}\]
\[\left[ N_{i+1}^2 - 2N_{i+2} + 2\mathbb{E}[x_{n_i+1}] - 2N_{i+2}ight). \tag{36}\]

The inequality in (35) follows from the fact that \(\gamma_k\) and \(\eta_k\) are decaying in a phase-dependent manner [see (14)]. Note that from the definition of \(n_i, \mathbb{E}[f(x_k) - f(x_{n_i})] \geq 0\) whenever \(N_i < k \leq N_{i+1}\). Thus, we have
\[\sum_{k=n_i}^{N_{i+2}} 2\gamma_k \mathbb{E}[f(x_k) - f(x_{n_i})] \leq \frac{\sum_{k=n_i}^{N_{i+2}} 2\gamma_k \mathbb{E}[f(x_k) - f(x_{n_i})]}{N_{i+2} - n_i + 1} \leq \frac{\sum_{k=n_i}^{N_{i+2}} 2\gamma_k \mathbb{E}[f(x_k) - f(x_{n_i})]}{N_{i+2} - n_i + 1} \leq 2\gamma_{N_{i+1}} B^2 + \frac{2\gamma_{N_{i+1}} B^2}{\gamma_{N_{i+1}}^2} + \frac{2\gamma_{N_{i+1}} B^2}{\gamma_{N_{i+1}}^2} \tag{37}\]

where the second inequality follows from the assumption that \(\mathbb{E}[f(x_{n_i+1})] \geq \mathbb{E}[f(x_{n_i})]\) and the fact that \(N_{i+2} - N_i \geq N_{i+2} - n_i + 1\). The last inequality follows from the [10, Lem. 4]. Combining (36) and (37), we obtain
\[\mathbb{E}[f(x_{n_i+1}) - f(x_{n_i})] \leq \frac{10\sqrt{d} \gamma N_{i+1}}{N_1^2/3} + \frac{10\sqrt{d} \gamma \gamma_{N_{i+1}}^2}{N_1^1/3} + \frac{5\gamma_{N_{i+1}}^2}{N_2^2/3} \tag{38}\]

This completes the proof for the case when \(i \geq 1\). The proof for the case, when \(i = 0\), follows in a similar manner. Plugging (38) into (34), we obtain
\[\mathbb{E}[f(x_N)] = \mathbb{E}[f(x_{n_i+1})] \leq \mathbb{E}[f(x_{n_0})] + \sum_{i=0}^l \mathbb{E}[f(x_{n_i+1}) - f(x_{n_i})] \leq \mathbb{E}[f(x_{n_0})] + \sum_{i=0}^l \mathbb{E}[f(x_{n_i+1}) - f(x_{n_i})] \leq \mathbb{E}[f(x_{n_0})] + \sum_{i=1}^l \left( \frac{10\sqrt{d} \gamma N_{i+1} \gamma_{N_{i+1}}^2}{N_1^2/3} \right) \tag{39}\]
\[+ \frac{10\sqrt{d} \gamma \gamma_{N_{i+1}}^2}{N_1^1/3} + \frac{5\gamma_{N_{i+1}}^2}{N_2^2/3} \tag{39}\]
\[+ \frac{2\gamma_{N_{i+1}} B^2}{\gamma_{N_{i+1}}^2} + \frac{2\gamma_{N_{i+1}} B^2}{\gamma_{N_{i+1}}^2} \tag{39}\]
\[+ \frac{2\gamma_{N_{i+1}} B^2}{\gamma_{N_{i+1}}^2} + \frac{2\gamma_{N_{i+1}} B^2}{\gamma_{N_{i+1}}^2} \tag{39}\]
\[+ \frac{2\gamma_{N_{i+1}} B^2}{\gamma_{N_{i+1}}^2} + \frac{2\gamma_{N_{i+1}} B^2}{\gamma_{N_{i+1}}^2} \tag{39}\]
\[\leq \inf_{\frac{K}{2} \leq k \leq N_1} \mathbb{E}[f(x_k)] + \sqrt{d \gamma N_{i+1}} \gamma_{N_{i+1}}^2 + 6\gamma_{N_{i+1}} \tag{39}\]
\[+ \frac{10\gamma_{N_{i+1}} B^2}{N_2^2/3} + \frac{20\gamma_{N_{i+1}} B^2}{N_2^2/3} + \frac{20\gamma_{N_{i+1}} B^2}{N_2^2/3} \tag{39}\]
\[
\frac{10\gamma_0 d (c_1 \eta_0^2 + \frac{c_2}{\eta_0})^2}{N^{4/3}} + \frac{17.5\gamma_0 c_2}{\eta_0^2 N^{1/3}}. \tag{39}
\]

Note that for all \(k \leq N_1\), we have stepsize \(\gamma_k = \frac{\eta_{k-1}}{N_{1-k}}\) and perturbation constant \(\eta_k = \frac{\eta_0}{N_{1-k}}\). Let \(x_k\) be the output of the SGD-BGO algorithm, then using the fact that infimum is smaller than any weighted average, we have
\[
\inf_{\frac{k}{N}} \mathbb{E}[f(x_k) - f(x^*)] \leq \frac{2}{N_1} \sum_{k=1}^{N_1} \mathbb{E}[f(x_k) - f(x^*)] \leq \frac{2}{N_1} \left( \frac{D^2 N^{2/3}}{2\eta_0} + \frac{B^2 N^{2/3}}{2\eta_0^2} + \frac{2N_1 D \sqrt{d} c_1 \eta_0^2}{\eta_0 N} \right) + \frac{2N_1 D \sqrt{d} c_1 \eta_0^2}{\eta_0 N} \tag{40}
\]
\[
\leq \frac{2}{N_1} \left( \frac{D^2 N^{2/3}}{2\eta_0} + \frac{B^2 N^{2/3}}{2\eta_0^2} + \frac{2N_1 D \sqrt{d} c_1 \eta_0^2}{\eta_0 N} \right) \tag{41}
\]
\[
= \frac{1}{N_1^{1/3}} \left( \frac{4D^2}{\gamma_0} + \frac{\gamma_0 B^2}{N_1^{1/3}} + \frac{4D \sqrt{d} c_1 \eta_0^2}{\eta_0 N} \right) \tag{42}
\]
where the inequality in (40) follows from the fact that \(N_1 \leq 2(N_1 - \frac{N}{2}) + 1\), the inequality in (41) follows from the Lemma 2, and the final inequality follows from the fact that \(\frac{N}{2} \leq N_1 \leq \frac{N}{3}\). The main claim follows by plugging (42) into (39).

\section{Proof of Theorem 6:} The proof proceeds through a sequence of lemmas, similar to the proof of Theorem 5 in Section VI-C1 under the oracle (O1).

\textbf{Lemma 3:} Assume (A2). With the oracle (O2), suppose that the SGD-BGO algorithm is run with stepsize sequence \(\{\gamma_k\}_{k=1}^{N}\). Then, given any \(1 < k_0 < k_1 \leq N\), we have
\[
\sum_{k=k_0}^{k_1} 2\gamma_k \mathbb{E}[f(x_k) - f(x_{k_0})] \leq \sum_{k=k_0}^{k_1} 2d^2 \gamma_k \mathcal{E}_k + \gamma_k^2 B_k^2
\]
where \(B_k^2 = \left[ B^2 + 2d \mathbb{E}E_k^2 + dE_k^2 + c_2 \eta_k^2 + c_2 \right] \), \(E_k = c_1 \eta_k + \frac{c_1}{\eta_0 \sqrt{m}}\), constants \(c_1\) and \(c_2\) are as defined in (O2), \(B\) is as defined in (A2), and \(D\) is as defined in (A3).

\textbf{Proof:} Follows by a completely parallel argument to the proof of Lemma 1, after observing that \(\mathbb{E}[\xi_1](g(x_k, \xi_k)] \leq \nabla f(x_k) + c_1 \eta_k 1_{d \times 1} + \frac{c_1}{\eta_0 \sqrt{m}} 1_{d \times 1} \) and \(\mathbb{E}[\xi_1]\|g(x_k, \xi_k)\|^2 \leq \|\mathbb{E}[\xi_1](g(x_k, \xi_k)]\|^2 + c_2 \eta_k^2 + c_2 \). \qed

\textbf{Lemma 4:} Assume (A2). With the oracle (O2), suppose that the SGD-BGO algorithm is run with a constant stepsize and perturbation constant, i.e., \(\gamma_k = \gamma, \eta_k = \eta \forall k \geq 1\). Then, for any \(k \geq 1\), we have
\[
\sum_{k=1}^{N} \mathbb{E}[f(x_k) - f(x^*)] \leq \frac{D^2}{2\gamma} + 2N D \sqrt{d} \mathcal{E} + \frac{N \gamma B^2}{2}
\]
where \(\mathcal{E} = c_1 \eta + \frac{c_1}{\eta_0 \sqrt{m}}\), \(c_1\) is as defined in (O2), \(B\) is as defined in (A2), and \(D\) is as defined in (A3).

\textbf{Proof:} The proof follows in a similar manner as that of the Lemma 2, with the following modification: \(\mathbb{E}[\Delta g_k] = c_1 \eta_k 1_{d \times 1} + \frac{c_1}{\eta_0 \sqrt{m}} 1_{d \times 1}\). \qed

\textbf{Proof: (Theorem 6)} Using a parallel argument to the initial passage in the proof of Theorem 5 leading up to (38), we obtain
\[
\mathbb{E}[f(x_{n+1}) - f(x_n)] \leq \frac{10\sqrt{d} C_1 \eta_0^2}{N} + \frac{10\sqrt{d} C_3 D^{2-i/2}}{\eta_0 \sqrt{N}} + \frac{5\gamma_0^2}{\sqrt{N}} \left[ B^2 + \frac{2\sqrt{d} B C_1 \eta_0}{N^{1/3}} \right] + \frac{2\sqrt{d} B c_1^2 \eta_0^2}{N^{2/3}} \tag{43}
\]
\[
+ \frac{2\sqrt{d} c_2 \eta_0^2}{N^{3/3}} + \frac{2\sqrt{d} c_2^2}{\eta_0^2} \tag{44}
\]
Plugging (43) into (34), we get
\[
\mathbb{E}[f(x_N)] = \mathbb{E}[f(x_{n+1})] \leq \inf_{\frac{k}{N}} \mathbb{E}[f(x_k)] \leq \frac{1}{\sqrt{N}} \left( \frac{4D^2}{\gamma_0} + \frac{\gamma_0 B^2}{N_1^{1/3}} + \frac{4D \sqrt{d} c_1 \eta_0^2}{\eta_0 N} \right) \tag{45}
\]
where we used Lemma 4 and the fact that \(\frac{N}{3} \leq N_1 \leq \frac{N}{2}\). The main claim follows by plugging (45) into (39).

\section{D. Proof for Risk-Sensitive Reinforcement Learning}

\textbf{Proof of Proposition 1:} Let \(F\) denote the distribution of \(K_x(x^0)\). Then, we have
\[
\mathbb{E}[\rho_m - \rho(K_x(x^0))] = \mathbb{E}[\rho(F_m) - \rho(F)] \leq LW_1(F, G) \leq \frac{c_1 L B}{\sqrt{m}} \tag{36, Th. 3.1} \]
\section{VII. CONCLUSION}

Motivated by practical applications involving biased function measurements, we formulated two biased gradient oracles with an additive estimation error component. The first oracle featured a bias-variance tradeoff for the gradient estimates, while the second one did not have such a tradeoff. We studied the nonasymptotic performance of gradient-based algorithms with inputs from a biased gradient oracle in convex as well as nonconvex optimization regimes. Furthermore, we highlighted
the applicability of our proposed biased gradient oracles in a risk-sensitive reinforcement learning setting.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewer for providing a motivating example involving biased function measurements, and for a careful proofreading of earlier versions of this manuscript.

REFERENCES

[1] R. T. Rockafellar and S. Uryasev, “Optimization of conditional value-at-risk,” J. Risk, vol. 2, no. 3, pp. 21–41, 2000.
[2] M. C. Fu, Ed., Handbook of Simulation Optimization. Berlin, Germany: Springer, 2015.
[3] S. Bhatnagar, H. L. Prasad, and L. A. Prashanth, Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods, vol. 434. Berlin, Germany: Springer, 2013.
[4] Y. Nesterov and V. Spokoiny, “Random gradient-free minimization of convex functions,” Found. Comput. Math., vol. 17, no. 2, pp. 527–566, 2017.
[5] J. C. Spall, “Multivariate stochastic approximation using a simultaneous perturbation gradient approximation,” IEEE Trans. Autom. Control, vol. 37, no. 3, pp. 332–341, Mar. 1992.
[6] J. C. Spall, Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control, vol. 65. New York, NY, USA: Wiley, 2005.
[7] X. Hu, L. A. Prashanth, A. György, and C. Szepesvári, “(Bandit) convex optimization with biased noisy gradient oracles,” in Proc. Artif. Intell. Statist., 2016, pp. 819–828.
[8] S. Ghadimi and G. Lan, “Stochastic first-and zeroth-order methods for nonconvex stochastic programming,” SIAM J. Optim., vol. 23, no. 4, pp. 2341–2368, 2013.
[9] K. Balasubramanian and S. Ghadimi, “Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 3459–3468.
[10] P. Jain, D. Nagaraj, and P. Netrapalli, “Making the last iterate of SGD information theoretically optimal,” SIAM J. Optim., vol. 31, no. 2, pp. 1108–1130, 2021.
[11] O. Devolder, F. Glineur, and Y. Nesterov, “First-order methods of smooth convex optimization with inexact oracle,” Math. Program., vol. 146, no. 1, pp. 37–75, 2014.
[12] M. Baes, “Estimate sequence methods: Extensions and approximations,” Inst. Operations Res., ETH Zürich, Switzerland, 2009, p. 5.
[13] A. d’Aspremont, “Smoothing optimization with approximate gradient,” SIAM J. Optim., vol. 19, pp. 1171–1183, 2008.
[14] L. M. Nguyen, K. Scheinberg, and M. Takáč, “Inexact SARAH algorithm for stochastic optimization,” Optim. Methods Softw., vol. 36, no. 1, pp. 237–258, 2021.
[15] R. Bollaapragada, R. H. Byrd, and J. Nocedal, “Exact and inexact subsampled Newton methods for optimization,” IMA J. Numer. Anal., vol. 39, no. 2, pp. 545–578, 2019.
[16] R. Pasupathy, P. Glynn, S. Ghosh, and F. S. Hashemi, “On sampling rates in simulation-based recursions,” SIAM J. Optim., vol. 28, no. 1, pp. 45–73, 2018.
[17] S. Liu, B. Kaiklkhura, P.-Y. Chen, P. Ting, S. Chang, and L. Amini, “Zeroth-order stochastic variance reduction for nonconvex optimization,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 3727–3737.
[18] R. Johnson and T. Zhang, “Accelerating stochastic gradient descent using predictive variance reduction,” Adv. Neural Inf. Process. Syst., vol. 26, pp. 315–323, 2013.
[19] R. J. Serfling, Approximation Theorems of Mathematical Statistics, vol. 162. New York, NY, USA: Wiley, 2009.
[20] S. P. Bhat and L. A. Prashanth, “Concentration of risk measures: A Wasserstein distance approach,” Adv. Neural Inf. Process. Syst., vol. 32, pp. 11762–11771, 2019.
[21] V. V. Vakhover and Y. Kulchitsky, “Convergence of a class of random search algorithms,” Automat. Remote Control, vol. 8, pp. 1321–1326, 1972.
[22] H. J. Kushner and D. S. Clark, Stochastic Approximation Methods for Constrained and Unconstrained Systems. Berlin, Germany: Springer, 1978.
[23] L. A. Prashanth, S. Bhatnagar, M. Fu, and S. Marcus, “Adaptive system optimization using random directions stochastic approximation,” IEEE Trans. Autom. Control, vol. 62, no. 5, pp. 2223–2238, May 2017.
[24] L. A. Prashanth, S. Bhatnagar, N. Bhavsar, M. C. Fu, and S. I. Marcus, “Random directions stochastic approximation with deterministic perturbations,” IEEE Trans. Autom. Control, vol. 65, no. 6, pp. 2450–2465, Jun. 2020.
[25] S. Asmussen and P. W. Glynn, Stochastic Simulation: Algorithms and Analysis, vol. 57. Berlin, Germany: Springer, 2007.
[26] P. Glasserman and Y.-C. Ho, Gradient Estimation Via Perturbation Analysis, vol. 116. Berlin, Germany: Springer, 1991.
[27] L. Bottou, F. E. Curtis, and J. Nocedal. “Optimization methods for large-scale machine learning,” SIAM Rev., vol. 60, no. 2, pp. 223–311, 2018.
[28] Z. Shen, A. Ribeiro, H. Hassani, H. Qian, and C. Mi, “Hessian aided policy gradient,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 5729–5738.
[29] H. Föllmer and A. Schied, “Convex measures of risk and trading constraints,” Finance Stochastics, vol. 6, no. 4, pp. 429–447, 2002.
[30] C. Acerbi, “Spectral measures of risk: A coherent representation of subjective risk aversion,” J. Bank. Finance, vol. 26, no. 7, pp. 1505–1518, 2002.
[31] A. Cassel, S. Mannor, and A. Zeevi, “A general approach to multi-armed bandits under risk criteria,” in Proc. 31st Conf. Learn. Theory, 2018, pp. 1295–1306.
[32] L. A. Prashanth and S. P. Bhat, “Concentration of risk measures: A Wasserstein distance approach,” 2020, arXiv:1902.10709.
[33] A. Tamar, Y. Chow, M. Ghavamzadeh, and S. Mannor, “Policy gradient for coherent risk measures,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 1468–1476.
[34] N. Bhavsar and L. A. Prashanth, “Non-asymptotic bounds for stochastic optimization with biased noisy gradient oracles,” 2021, arXiv:2002.11440.
[35] S. Bubeck, “Convex optimization: Algorithms and complexity,” Found. Trends Mach. Learn., vol. 8, no. 3/4, pp. 231–357, 2015.
[36] J. Lei, “Convergence and concentration of empirical measures under Wasserstein distance in unbounded functional spaces,” Bernoulli, vol. 26, no. 1, pp. 767–798, 2020.

Nirav Bhavsar received the M.S. degree in computer science and engineering from the Indian Institute of Technology-Madras, Chennai, India, in 2020. He is currently a Senior Data Scientist at Unacademy. His research interests are in simulation optimization and reinforcement learning. He received the Biswajit Sain Endowment Prize for the best M.S. Thesis in Computer Science and Engineering at IIT Madras in 2021.

L. A. Prashanth received the master’s and Ph.D. degrees in computer science and automation from the Indian Institute of Science, Bengaluru, India, in 2008 and 2013, respectively. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Indian Institute of Technology-Madras, Chennai, India. He has coauthored a book titled Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods (Springer, 2013). His research interests include reinforcement learning, stochastic optimization, and multiarmed bandits, with applications in transportation, networks, and recommendation systems. Dr. Prashanth was the recipient of the third prize for his Ph.D. dissertation by the IEEE Intelligent Transportation Systems Society (ITSS).