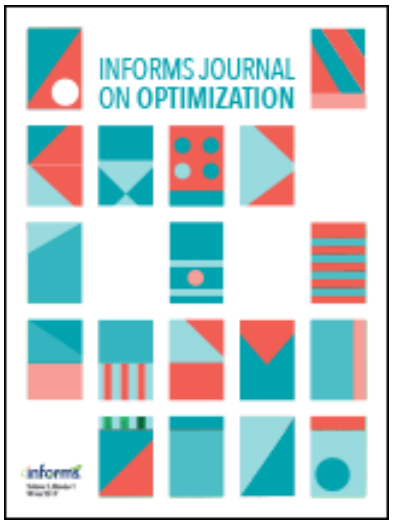


INFORMS Journal on Optimization

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Stochastic Zeroth-Order Functional Constrained Optimization: Oracle Complexity and Applications

Anthony Nguyen, Krishnakumar Balasubramanian

To cite this article:
Anthony Nguyen, Krishnakumar Balasubramanian (2022) Stochastic Zeroth-Order Functional Constrained Optimization: Oracle Complexity and Applications. INFORMS Journal on Optimization

Published online in Articles in Advance 30 Nov 2022

. https://doi.org/10.1287/ijoo.2022.0085

https://pubsonline.informs.org/journal/ijoo

INFORMS JOURNAL ON OPTIMIZATION
Articles in Advance, pp. 1–17
ISSN 2575-1484 (print), ISSN 2575-1492 (online)

Stochastic Zeroth-Order Functional Constrained Optimization: Oracle Complexity and Applications

Anthony Nguyen,[a] **Krishnakumar Balasubramanian**[b,*]

[a] Department of Mathematics, University of California–Davis, Davis, California 95616; [b] Department of Statistics, University of California–Davis, Davis, California 95616

*Corresponding author

Contact: antngu@ucdavis.edu, https://orcid.org/0000-0003-2588-3340 (AN); kbala@ucdavis.edu, https://orcid.org/0000-0003-1472-9559 (KB)

Received: January 14, 2022
Revised: April 23, 2022
Accepted: October 6, 2022
Published Online in Articles in Advance: November 30, 2022

https://doi.org/10.1287/ijoo.2022.0085

Abstract. Functionally constrained stochastic optimization problems, where neither the objective function nor the constraint functions are analytically available, arise frequently in machine learning applications. In this work, assuming we only have access to the noisy evaluations of the objective and constraint functions, we propose and analyze stochastic zeroth-order algorithms for solving this class of stochastic optimization problem. When the domain of the functions is $\mathbb{R}^n$, assuming there are m constraint functions, we establish oracle complexities of order $\mathcal{O}((m+1)n/\epsilon^2)$ and $\mathcal{O}((m+1)n/\epsilon^3)$ in the convex and nonconvex settings, respectively, where ϵ represents the accuracy of the solutions required in appropriately defined metrics. The established oracle complexities are, to our knowledge, the first such results in the literature for functionally constrained stochastic zeroth-order optimization problems. We demonstrate the applicability of our algorithms by illustrating their superior performance on the problem of hyperparameter tuning for sampling algorithms and neural network training.

Funding: K. Balasubramanian was partially supported by a seed grant from the Center for Data Science and Artificial Intelligence Research, University of California–Davis, and the National Science Foundation [Grant DMS-2053918].

Keywords: stochastic optimization • zeroth-order optimization • nonlinear constraints

1. Introduction

We develop and analyze stochastic zeroth-order algorithms for solving the following nonlinear optimization problem with functional constraints:

$$\min_{x\in X} \; f_0(x) \quad \text{such that} \quad f_i(x) \leqslant 0, \quad i \in \{0,1,\ldots,m\}, \tag{1}$$

where for $i \in \{0,1,\ldots,m\}$, $f_i : \mathbb{R}^n \to \mathbb{R}$ are continuous functions that are not necessarily convex defined as $f_i(x) = \mathbb{E}_{\xi_i}[F_i(x,\xi_i)]$, with ξ_i denoting the noise vector associated with function f_i, and $X \subseteq \mathbb{R}^n$ is a convex compact set that represents *known* constraints (i.e., constraints that are analytically available). In the stochastic zeroth-order setting, we *neither observe the objective function f_0 nor the constraint functions f_i analytically*. We only have access to noisy function evaluations of them. The study of stochastic zeroth-order optimization algorithms for unconstrained optimization problems goes back to the early works of Kiefer and Wolfowitz (1952), Blum (1954), Hooke and Jeeves (1961), Spendley et al. (1962), Powell (1964), Nelder and Mead (1965), Nemirovski and Yudin (1983), and Spall (1987). Such zeroth-order algorithms have proved to be extremely useful for hyperparameter tuning (Snoek et al. 2012, Gelbart et al. 2014, Hernández-Lobato et al. 2015, Golovin et al. 2017, Ruan et al. 2020), reinforcement learning (Salimans et al. 2017, Mania et al. 2018, Choromanski et al. 2020, Gao et al. 2020), and robotics (Jaquier and Rozo 2020, Jaquier et al. 2020). However, the study of zeroth-order algorithms and their oracle complexities for the constrained problem as in (1) is limited, despite the fact that several real-world machine learning problems fall under the setting of (1). We now describe two such applications that serve as our main motivation for developing stochastic zeroth-order optimization algorithms for solving (1) and then analyzing their oracle complexity.

1.1. Motivating Application I

The Hamiltonian Monte Carlo (HMC) algorithm, proposed by Duane et al. (1987) and popularized in the statistical machine learning community by Neal (2011), is a gradient-based sampling algorithm that works by discretizing the continuous-time degenerate Langevin diffusion (Leimkuhler and Matthews 2015). It has been used

successfully as a state-of-the-art sampler or a numerical integrator in the Bayesian statistical machine learning community by Hoffman and Gelman (2014), Wang et al. (2013), Girolami and Calderhead (2011), Chen et al. (2014), and Carpenter et al. (2017). However, in order to obtain successful performance in practice using HMC, several hyperparameters need to be tuned optimally. Typically, the functional relationship between the hyperparameters that need to be tuned and the performance measure used is not available in an analytical form. We can only evaluate the performance of the sampler for various settings of the hyperparameter. Furthermore, in practice several constraints, for example, constraints on running times and constraints that enforce the generated samples to pass certain standard diagnostic tests (Geweke 1991, Gelman and Rubin 1992) are enforced in the hyperparameter tuning process. The functional relationship between such constraints and the hyperparameters is also not available analytically. This makes the problem of optimally setting the hyperparameters for HMC a constrained zeroth-order optimization problem. As a preview, in Section 4.1, we show that our approach provides significant improvements over the existing methods of Mahendran et al. (2012), Gelbart et al. (2014), and Hernández-Lobato et al. (2015), which are based on Bayesian optimization techniques for tuning HMC, when we measure the performance adopting the widely used *effective sample size* metric (Kass et al. 1998).

1.2. Motivating Application II

Deep learning has achieved state-of-the-art performance in recent years for various prediction tasks (Goodfellow et al. 2016). Among the various factors involved behind the success of deep learning, hyperparameter tuning is one of primary factors (Bergstra and Bengio 2012, Snoek et al. 2012, Li et al. 2017, Hazan et al. 2018, Elsken et al. 2019). However, most of the existing methods for tuning the hyperparameters do not enforce any constraints on the prediction time required on the validation set or memory constraints on the training algorithm. Such constraints are typically required to make deep learning methods widely applicable to problems arising in several consumer applications based on tiny devices (Yang et al. 2008, Latré et al. 2011, Perera et al. 2015). As in the aforementioned motivating application, the functional relationship between such constraints and the hyperparameters is not available analytically. As a preview, in Section 4.2, we show that our approach provides significant improvements over the existing works of Gelbart et al. (2014), Hernández-Lobato et al. (2015), and Ariafar et al. (2019) that developed hyperparameter tuning techniques that explicitly take into account time/memory constraints.

1.3. Related Works

In the operations research and statistics communities, zeroth-order optimization techniques are well studied under the name of derivative-free optimization. Interested readers are referred to Conn et al. (2009) and Audet and Hare (2017). In the machine learning community, Bayesian optimization techniques have been developed for optimizing functions with only noisy function evaluations. We refer the reader to Mockus (1994), Kolda et al. (2003), Spall (2005), Conn et al. (2009), Mockus (2012), Brent (2013), Shahriari et al. (2015), Audet and Hare (2017), Larson et al. (2019), Frazier (2018), Archetti and Candelieri (2019), and Liu et al. (2020) for more details. In what follows, we focus on relevant literature from zeroth-order optimization and Bayesian optimization literature for *known constrained optimization* problems (i.e., problems with constraints that are analytically available). When the constraint set is analytically available and only the objective function is not, Lewis and Torczon (2002) and Bueno et al. (2013) considered an augmented Lagrangian approach and an inexact restoration method, respectively, and provided convergence analysis. Furthermore, Kolda et al. (2003), Amaioua et al. (2018), and Audet et al. (2015) extended the popular mesh adaptive direct search to this setting. Projection-free methods based on Frank–Wolfe methods have been considered in Balasubramanian and Ghadimi (2018) and Sahu et al. (2019) for the case when the constraint set is a convex subset of $\mathbb{R}^n$. Furthermore, Li et al. (2022) considered the case when the constraint set is a Riemannian submanifold embedded in $\mathbb{R}^n$ (and the function is defined only over the manifold). None of these works is directly applicable to the case of unknown constraints that we consider in this work.

We now discuss some existing methods for solving (variants of) Problem (1) in the zeroth-order setting. For solving (1) in the deterministic setting (i.e., we could obtain exact evaluations of the objective and the constraint functions at a given point), *filter methods* that reduce the objective function while trying to reduce constraint violations were proposed and analyzed in Audet and Dennis (2004), Echebest et al. (2017), and Pourmohamad and Lee (2020). Barrier methods in the zeroth-order setting were considered in Audet and Dennis (2006, 2009), Liuzzi and Lucidi (2009), Gratton and Vicente (2014), Fasano et al. (2014), Liuzzi et al. (2010), and Dzahini et al. (2022), with some works also developing line search approaches for setting the tuning parameters. Model-based approaches were considered in the works of Müller and Woodbury (2017), Tröltzsch (2016), Augustin and Marzouk (2014), Gramacy et al. (2016), and Conn and Le Digabel (2013). Furthermore, Bűrmen et al. (2006) and Audet and Tribes (2018) developed extensions of the Nelder–Mead algorithm to the constrained setting.

Several works in the statistical machine learning community also considered Bayesian optimization methods in the constrained setting, in both the noiseless and noisy settings. We refer the reader, for example, to Gardner et al. (2014), Gelbart et al. (2016), Ariafar et al. (2019), Balandat et al. (2020), Bachoc et al. (2020), Greenhill et al. (2020), Eriksson and Poloczek (2021), Letham et al. (2019), Hernández-Lobato et al. (2015), Lam and Willcox (2017), Picheny et al. (2016), and Acerbi and Ma (2017). On one hand, these works demonstrate the interest in the optimization and machine learning communities for developing algorithms for constrained zeroth-order optimization problems. On the other hand, most of the above-mentioned works are not designed to handle the *stochastic* zeroth-order constrained optimization that we consider. Furthermore, a majority of the aforementioned works are methodological, and the few works that develop convergence analysis do so only in the asymptotic setting. A recent work by Usmanova et al. (2019) considered the case when the constraints are linear functions (but unknown) and provided a Frank–Wolfe-based algorithm with *estimated* constraints. However, the proposed approach is limited to only linear constraints, and the oracle complexities established are highly suboptimal. To the best of our knowledge, there is no rigorous nonasymptotic analysis of the oracle complexity of stochastic zeroth-optimization when the constraints and the objective values are available only via *noisy* function evaluations.

1.4. Methodology and Main Contributions

Our methodology is based on a novel constraint extrapolation technique developed for the zeroth-order setting, extending the work of Boob et al. (2022) in the first-order setting, and the Gaussian smoothing-based zeroth-order stochastic gradient estimators. Specifically, we propose the stochastic zeroth-order constraint extrapolation (`SZO-ConEX`) method in Algorithm 1 for solving problems of the form in (1). We theoretically characterize how to set the *tuning parameters* of the algorithm so as to mitigate the issues caused by the *bias* in the stochastic zeroth-order gradient estimates and obtain the oracle complexity of our algorithm. More specifically, we make the following main contributions:

- When the functions f_i, $i = 0, \ldots, m$ are convex, in Theorem 1, we show that the number of calls to the stochastic zeroth-order oracle to achieve an appropriately defined ϵ-optimal solution of (1) (see Definition 2) is of order $\mathcal{O}((m+1)n/\epsilon^2)$.
- When the functions are nonconvex, in Proposition 1, we show that the number of calls to the stochastic zeroth-order oracle to achieve an appropriately defined ϵ-optimal Karush-Kuhn-Tucker (KKT) solution of (1) (see Definition 3) is of order $\mathcal{O}((m+1)n/\epsilon^3)$.

To our knowledge, these are the first nonasymptotic oracle complexity results for stochastic zeroth-order optimization with stochastic zeroth-order functional constraints. We illustrate the practical applicability of the developed methodology by testing its performance on hyperparameter tuning for HMC sampling algorithm (Section 4.1) and three-layer neural network (Section 4.2).

2. Preliminaries and Methodology

We start by introducing the notation used in this paper. Let $\mathbf{0}$ denote the vector of elements 0 and $[m] := \{1, \ldots, m\}$. Let $f(x) := [f_1(x), \ldots, f_m(x)]^T$; then, the constraints in (1) can be expressed as $f(x) \leqslant \mathbf{0}$. We use $\xi := [\xi_1, \ldots, \xi_m]$ to denote the random vectors in the constraints. Furthermore, $\|\cdot\|$ denotes a general norm, and $\|\cdot\|_*$ denotes its dual norm defined as $\|z\|_* := \sup\{z^T x : \|x\| \leq 1\}$. Furthermore, $[x]_+ := \max\{x, 0\}$ for any $x \in \mathbb{R}$. For any vector $x \in \mathbb{R}^k$, we define $[x]_+$ as the elementwise application of $[\cdot]_+$.

We now describe the precise assumption made on the *stochastic zeroth-order oracle* in this work.

Assumption 1. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. For $i \in \{0, \ldots, m\}$ and for any $x \in \mathbb{R}^n$, the zeroth-order oracle outputs an estimator $F_i(x, \xi_i)$ of $f_i(x)$ such that $\mathbb{E}[F_i(x, \xi_i)] = f_i(x)$, $\mathbb{E}[F_i(x, \xi_i)^2] \leq \sigma_{f_i}^2$, $\mathbb{E}[\nabla F_i(x, \xi_i)] = \nabla f_i(x)$, $\mathbb{E}[\|\nabla F_i(x, \xi_i) - \nabla f_i(x)\|_*^2] \leqslant \sigma_i^2$, where $\|\cdot\|_*$ denotes the dual norm.*

This assumption assumes that we have access to a stochastic zeroth-order oracle that provides unbiased function evaluations with bounded variance. It is worth noting that in the preceding assumption, we do not necessarily assume the noise ξ_i is additive. Furthermore, we allow for different noise models for the objective function and the m constraint functions, which is a significantly general model compared with several existing works such as Le Digabel and Wild (2015). Our gradient estimator is then constructed by leveraging the Gaussian smoothing technique proposed in Nemirovski and Yudin (1983) and Nesterov and Spokoiny (2017). For $\nu_i \in (0, \infty)$ we introduce the smoothed function $f_{i,\nu_i}(x) = \mathbb{E}_{u_i}[f_i(x + \nu_i u_i)]$, where $u_i \sim \mathrm{N}(0, I_n)$ and independent across i. We can estimate the gradient of this smoothed function using function evaluations of f_i. Specifically, we define the stochastic zeroth-order gradient of $f_{i,\nu_i}(x)$ as

$$G_{i,\nu_i}(x, \xi_i, u_i) = \frac{F_i(x + \nu_i u_i, \xi_i) - F_i(x, \xi_i)}{\nu_i} u_i, \tag{2}$$

which is an unbiased estimator of $\nabla f_{i,\nu_i}(x)$; that is, we have $\mathbb{E}_{u,\xi_i}[G_{i,\nu_i}(x,\xi_i,u)] = \nabla f_{i,\nu_i}(x)$. However, it is well known that $G_{i,\nu_i}(x,\xi_i,u_i)$ is a biased estimator of $\nabla f_i(x)$. An interpretation of the gradient estimator in (2) as a consequence of Gaussian Stein's identity, popular in the statistics literature (Stein 1972), was provided in Balasubramanian and Ghadimi (2022).

The gradient estimator in (2) is referred to as the two-point estimator in the literature. The reason is that, for a given random vector ξ_i, it is assumed that the stochastic function in (2) could be evaluated at two points, $F_i(x+\nu_i u_i,\xi_i)$ and $F_i(x,\xi_i)$. Such an assumption is satisfied in several statistics, machine learning, simulation-based optimization, and sampling problems; see, for example, Spall (2005), Mokkadem and Pelletier (2007), Dippon (2003), Agarwal et al. (2010), Duchi et al. (2015), Ghadimi and Lan (2013), and Nesterov and Spokoiny (2017). Yet another estimator in the literature is the one-point estimator, which assumes that for each ξ_i, we observe only one noisy function evaluation, $F_i(x+\nu_i u_i,\xi_i)$. It is well known that the one-point setting is more challenging than the two-point setting (Shamir 2013). From a theoretical point of view, the use of two-point evaluation-based gradient estimator is primarily motivated by the suboptimality (in terms of oracle complexity) of one-point feedback-based stochastic zeroth-order optimization methods in terms of either the approximation accuracy or dimension dependency. For the rest of this work, we focus on the two-point setting and leave the question of obtaining results in the one-point setting as future work. We now describe our assumptions on the objective and constraint functions.

Assumption 2. *Function F_i has a Lipschitz continuous gradient with constant L_i almost surely for any ξ_i; that is, $\|\nabla F_i(y,\xi_i)-\nabla F_i(x,\xi_i)\|_* \le L_i\|y-x\|$, which consequently implies that $|F_i(y,\xi_i)-F_i(x,\xi_i)-\langle\nabla F_i(x,\xi_i),y-x\rangle| \le \frac{L_i}{2}\|y-x\|^2$ for $i\in\{0,1,\dots,m\}$.*

Assumption 3. *Function F_i is Lipschitz continuous with constant M_i almost surely for any ξ_i; that is, $|F_i(y,\xi_i)-F_i(x,\xi_i)| \le M_i\|y-x\|$ for $i\in\{0,1,\dots,m\}$.*

The above-mentioned smoothness assumptions are standard in the literature on stochastic zeroth-order optimization and are made in several works (Ghadimi and Lan 2013, Nesterov and Spokoiny 2017, and Balasubramanian and Ghadimi 2022) for obtaining oracle complexity results. It is easy to see that Assumption 2 implies that for $i\in\{0,\dots,m\}$, f_i has a Lipschitz continuous gradient with constant L_i because $\|\nabla f_i(y)-\nabla f_i(x)\|_* \le \mathbb{E}[\|\nabla F(y,\xi)-\nabla F(x,\xi)\|_*] \le L_i\|y-x\|$ as a result of Jensen's inequality for the dual norm. By similar reasoning and Assumption 3, we also see that f_i is Lipschitz continuous with constant M_i. From Assumptions 2 and 3, we also have $\|f(x_1)-f(x_2)\|_2 \le M_f\|x_1-x_2\|$, $\|\nabla f(x_2)^T(x_1-x_2)\|_2 \le M_f\|x_1-x_2\|$, and $\|f(x_1)-f(x_2)-\nabla f(x_2)^T(x_1-x_2)\|_2 \le \frac{L_f}{2}\|x_1-x_2\|^2$ for all $x_1,x_2\in\mathbb{R}^n$, where $\nabla f(\cdot):=[\nabla f_1(\cdot),\dots,\nabla f_m(\cdot)]\in\mathbb{R}^{n\times m}$, and constants M_f and L_f are defined as

$$M_f := \sqrt{\sum_{i=1}^m M_i^2} \quad \text{and } L_f := \sqrt{\sum_{i=1}^m L_i^2}. \tag{3}$$

We now state the definitions of the **prox**-function and the **prox**-operator. The class of algorithms based on **prox**-operators are called proximal algorithms. Such algorithms have turned out to be particularly useful for efficiently solving various machine learning problems in the recent past. We refer the interested reader to Parikh and Boyd (2014) and Beck (2017) for more details.

Definition 1. Let $\omega: X\to\mathbb{R}$ be continuously differentiable, L_ω-Lipschitz gradient smooth, and 1-strongly convex with respect to $\|\cdot\|$ function. We define the **prox**-function associated with $\omega(\cdot)$, $\forall x,y\in\mathbb{R}^n$ as $W(y,x):=\omega(y)-\omega(x)-\langle\nabla\omega(x),y-x\rangle$. From the smoothness and strong convexity of $\omega(x)$, we have $W(y,x)\le\frac{L_\omega}{2}\|x-y\|^2\le L_\omega W(x,y)$, $\forall x,y\in\mathbb{R}^n$. For any $v\in\mathbb{R}^n$, we define the following **prox**-operator as $\mathbf{prox}(v,\tilde{x},\eta):=\arg\min_{x\in X}\{\langle v,x\rangle+\eta W(x,\tilde{x})\}$.

The function W is also called the Bregman divergence in the literature. A canonical example of W is that of the Euclidean distance function $\|x-y\|^2$, which is useful when $X=\mathbb{R}^n$. We will see in Section 2.1 that our algorithm is based on the above-mentioned **prox**-operator. Finally, we have the following results, which will prove to be useful for subsequent calculations. Let $u:=[u_1,\dots,u_m]$, and let $D_X:=\sup_{x,y\in X}\sqrt{W(x,y)}$ be the diameter of the set X.

Lemma 1. *Let $\nu:=[\nu_1,\dots,\nu_m]$, $F_\nu(x,\xi,u):=[F_1(x+\nu_1u_1,\xi_1),\dots,F_m(x+\nu_mu_m,\xi_m)]^T$, and $f_\nu(x):=[f_{1,\nu_1}(x),\dots,f_{m,\nu_m}(x)]^T$. Under Assumption 3, we have $\mathbb{E}_{u,\xi}[\|F_\nu(x,\xi,u)-f_\nu(x)\|^2]\le\sigma_{f,\nu}^2$, where $\sigma_{f,\nu}^2:=\left(\sum_{i=1}^m 4(n+2)M_i^2\nu_i^2+L_i^2\nu_i^4n^2\right)+2\sigma_f^2$, where $\sigma_f^2=\sum_{i=1}^m\sigma_{f_i}^2$.*

Lemma 2. *Let* $\tilde{B}_i := \frac{\nu_i}{2} L_i (n+3)^{3/2} + L_i D_X + M_i$. *Under Assumptions 1 and 2, we have*

$$\mathsf{E}_{u,\xi}[\|G_{i,\nu_i}(x,\xi,u) - \nabla f_{i,\nu_i}(x)\|^2] \leqslant \sigma_{i,\nu_i}^2, \tag{4}$$

where $\sigma_{i,\nu_i}^2 := \nu_i^2 L_i^2 (n+6)^3 + 10(n+4)[\sigma_i^2 + \tilde{B}_i^2]$.

2.1. Algorithmic Methodology

We now present the SZO-ConEX algorithm for solving the stochastic zeroth-order functional constrained optimization problem (1). The constraint extrapolation framework is a novel primal-dual method that proceeds by (i) considering the Lagrangian formulation of (1), (ii) constructing linear approximations for the constraint functions, and (iii) constructing an *extrapolation operation* that enables acceleration. Such an approach has the advantages that (i) it does not require the projection of Lagrangian multipliers onto a possibly unknown bounded set (which is required by several other primal-dual methods), and (ii) it is a single-loop algorithm with a built-in acceleration step. It is worth remarking that Boob et al. (2022) and Hamedani and Aybat (2021) showed that such an approach helps achieve a better rate of convergence than existing methods for solving Lagrangian problems (of the form in (5)) in the stochastic first-order setting. However, their approach is not directly applicable to the zeroth-order setting where the estimated stochastic gradients are biased and have variances that are not uniformly bounded.

Recall the problem in (1) and notice that there are two types of constraints. The set X represents *known* constraints (i.e., constraints that are analytically available), and the inequality constraints defined by the functions f_i, $i \in [m]$ are the *unknown* or *zeroth-order constraints*. The Lagrangian of (1) is given by

$$\min_{x \in X} \max_{y \geqslant \mathbf{0}} \left\{ \mathcal{L}(x,y) := f_0(x) + \sum_{i=1}^{m} y_i f_i(x) \right\}. \tag{5}$$

In other words, (x^*, y^*) is a *saddle point* of the Lagrange function $\mathcal{L}(x,y)$ such that

$$\mathcal{L}(x^*, y) \leqslant \mathcal{L}(x^*, y^*) \leqslant \mathcal{L}(x, y^*) \tag{6}$$

for all $x \in X, y \geqslant \mathbf{0}$ whenever the optimal dual, y^*, exists. Throughout this work, we assume the existence of y^* satisfying (6). In order to handle the zeroth-order setting, we also define the Lagrangian with the smoothed functions as

$$\mathcal{L}_\nu(x,y) := f_{0,\nu_0}(x) + \sum_{i=1}^{m} y_i f_{i,\nu_i}(x). \tag{7}$$

Now, we describe the linearization in the context of the iterates directly as it will be easier to understand in the stochastic setting that we are in. Let $x^{(t)}$ be the sequence produced by the algorithm (to be discussed later). The linearization of $f(\cdot)$ at the point $x^{(t)}$, with respect to the point $x^{(t-1)}$, is given by

$$\ell_f(x^{(t)}) := f_\nu(x^{(t-1)}) + \nabla f_\nu(x^{(t-1)})^T (x^{(t)} - x^{(t-1)}),$$

where similar to ∇f, we define $\nabla f_\nu(x^{(t-1)}) := [\nabla f_{1,\nu_1}(x^{(t-1)}), \ldots, \nabla f_{m,\nu_m}(x^{(t-1)})]$. For the implementation, we use the version of linearization with the Gaussian smoothing-based stochastic zeroth-order gradients. In particular, we define $\ell_F(x^{(t)}) := F_\nu(x^{(t-1)}, \overline{\xi}^{(t-1)}, \overline{u}^{(t-1)}) + G_\nu(x^{(t-1)}, \overline{\xi}^{(t-1)}, \overline{u}^{(t-1)})^T (x^{(t)} - x^{(t-1)})$, where $G_\nu(x^{(t-1)}, \overline{\xi}^{(t-1)}, \overline{u}^{(t-1)}) \in \mathsf{R}^{n \times m}$ is given by

$$[G_{1,\nu_1}(x^{(t-1)}, \overline{\xi}_1^{(t-1)}, \overline{u}_1^{(t-1)}), \ldots, G_{m,\nu_m}(x^{(t-1)}, \overline{\xi}_m^{(t-1)}, \overline{u}_m^{(t-1)})].$$

Here, by $\overline{\xi}^{(t-1)}, \overline{u}^{(t-1)}$, we mean an independent (of $\xi^{(t-1)}, u^{(t-1)}$, respectively) realization of random objects ξ, u, respectively.

Based on this, the overall procedure, termed SZO-ConEx, is provided in Algorithm 1.

Algorithm 1 (SZO-ConEx (Stochastic Zeroth-Order Constraint Extrapolation) Method)

Input: $\nu_0 > 0$, $\nu > \mathbf{0}$, $(x^{(0)}, y^{(0)})$, $\{\gamma_t, \tau_t, \eta_t, \theta_t\}_{t \geqslant 0}$, T.

1: Set $(x^{(-1)}, y^{(-1)}) \leftarrow (x^{(0)}, y^{(0)})$, $F_\nu(x^{(-1)}, \overline{\xi}^{(-1)}, \overline{u}^{(-1)}) \leftarrow F_\nu(x^{(0)}, \overline{\xi}^{(0)}, \overline{u}^{(0)})$, $\ell_F(x^{(-1)}) \leftarrow \ell_F(x^{(0)})$.

2: **for** $t = 0, \ldots, T-1$ **do**

3: $\quad s^{(t)} \leftarrow (1+\theta_t)\ell_F(x^{(t)}) - \theta_t \ell_F(x^{(t-1)})$.

4: $\quad y^{(t+1)} \leftarrow [y^{(t)} + \frac{1}{\tau_t} s^{(t)}]_+$.

5: $\quad x^{(t+1)} \leftarrow \mathbf{prox}\Big(G_{0,\nu_0}(x^{(t)}, \xi_0^{(t)}, u_0^{(t)}) + \sum_{i=1}^m G_{i,\nu_i}(x^{(t)}, \xi_i^{(t)}, u_i^{(t)}) y_i^{(t+1)}, x^{(t)}, \eta_t\Big)$.

6: **return** $\overline{x}_T = \left(\sum_{t=0}^{T-1} \gamma_t\right)^{-1} \sum_{t=0}^{T-1} \gamma_t x^{(t+1)}$.

We now explain the individual steps in more detail.

• *Step* 3: This extrapolation step, considered by Boob et al. (2022) (see also Hamedani and Aybat (2021)) for the stochastic first-order setting, forms the main methodological innovation over the existing primal-dual method. First, note that instead of working with constraint functions, we work with a stochastic linearization of them. The extrapolation or moving average is essentially a way to incorporate momentum in the $s^{(t)}$ sequence. From the analysis, it turns out that the choice of constant θ_t (which we set as $\theta_t = 1$, without any loss of generality) gives the best possible oracle complexity in our analysis.

It is also worth remarking that the extrapolation/moving-average approach has been also used recently in the stochastic optimization of the composition of two functions in Ghadimi et al. (2020). Furthermore, the linearization technique is also used in the stochastic optimization of the composition of T functions for any $T \geq 1$ in Ruszczynski (2021) and Balasubramanian et al. (2022).

• *Step* 4: This step corresponds to the gradient ascent step to address the maximization problem in the Lagrangian formulation. We let parameter τ_t depend on t in the algorithm. However, the analysis in Section 3 reveals that a constant step size of $\tau_t = \tau$ suffices to obtain the derived oracle complexity.

• *Step* 5: This step corresponds to the descent step-or, more precisely, the proximal gradient descent step-to solve the minimization part of the saddle point problem in the Lagrangian formulation. We remark that one could potentially replace the proximal gradient step with a conditional gradient step when performing linear minimization over the set X is computationally efficient. We leave a rigorous oracle complexity analysis of this modification to future work.

• *Step* 6: This step corresponds to the averaging of the iterates. As we demonstrate later in the analysis in Section 3, in the convex and nonconvex settings that we consider, the best oracle complexities obtained correspond to the case of constant choice (i.e., $\gamma_t = 1$, without loss of generality). However, we suspect that there might be advantages of considering time-varying γ_t for the challenging case of adaptive algorithms that do not necessarily know the structure of the optimization problem at hand. We leave a detailed analysis of such adaptive algorithms as future work.

Finally, it is worth noting that Gramacy et al. (2016) proposed an augmented Lagrangian approach for solving the problem in (1) in the nonnoisy setting. However, they did not propose the above-mentioned constraint extrapolation technique. In our experiments in Section 4, we show that our constraint extrapolation approach significantly outperforms the approach in Gramacy et al. (2016) in simulations and real-world problems.

3. Main Results

We now present our main results on the oracle complexity of `SZO-ConEX` algorithm. Recall the definition of the stochastic zeroth-order gradient estimators from (2). At a high level, the algorithm could be interpreted as using the constraint extrapolation method of Boob et al. (2022) for solving (5) with $\mathcal{L}(x,y)$ replaced by $\mathcal{L}_\nu(x,y)$ as defined in (7), as the stochastic zeroth-order gradients used in Algorithm 1 are essentially unbiased estimators of the smoothed functions $f_{\nu,i}$ (for $i \in [m]$). However, they have unbounded variance. Hence, the analysis of Boob et al. (2022), which is for the stochastic first-order setting under the assumption of unbiased stochastic gradient and uniformly bounded variance, is not directly applicable. Furthermore, on the one hand, as the smoothing parameters ν_i (for $i \in [m]$) tend to 0, $\mathcal{L}_\nu(x,y)$ converges to $\mathcal{L}(x,y)$ defined in (5). However, on the other hand, the parameters ν_i are in the denominator of the stochastic zeroth-order gradient estimators (see (2)). Hence, we cannot let them tend to 0 at any arbitrary rate. Picking the tuning parameters ν_i carefully to balance this tension and get the best possible oracle complexity forms the crux of our analysis. Finally, we also point out that general strategies for picking the smoothing parameters (as proposed in Beck and Teboulle (2012) for dealing with nonsmooth stochastic first-order optimization problems) are also not directly applicable for analyzing stochastic zeroth-order algorithms, and specialized approaches are often required-we refer the reader to Duchi et al. (2015), Nesterov and Spokoiny (2017), Ghadimi and Lan (2013), and Balasubramanian and Ghadimi (2022) for several related techniques for analyzing unconstrained stochastic zeroth-order optimization algorithms.

3.1. Convex Setting

We first provide our theoretical results for the case when the functions f_i for $i \in [m]$ are convex. We start by describing the measure of optimality we consider for solving (1).

Definition 2. A point $\overline{x}$ is an ϵ-approximately optimal solution in expectation, for (1), if it satisfies $\mathbb{E}[f_0(\overline{x}) - f_0^*] \leq \epsilon$ and $\mathbb{E}[\|[f(\overline{x})]_+\|_2] \leq \epsilon$, where f_0^* is the optimal value of (1) and the expectation is with respect to the randomness arising due to ξ_i and u_i across all iterations.

The first part of Definition 2 corresponds to the standard optimality condition for the convex problem. The next part corresponds to the constraint violation. Our main result is described next. We define $M_X := \sup_{x\in X}\|x\|$. Furthermore, we define $\sigma_v := [\sigma_{1,v_1}, \ldots, \sigma_{m,v_m}]$, where σ_{i,v_i} for $i \in [m]$ are as defined in Lemma 2, and $\sigma_{X,f} := \left(\sigma_{f,v}^2 + D_X^2\|\sigma_v\|_2^2\right)^{1/2}$ (where $\sigma_{f,v}^2$ is as defined in Lemma 1).

Theorem 1. *Suppose the functions f_i, for $i \in [m]$, are convex and satisfy Assumptions 1–3. Define $\mathcal{H}_* := (L_f D_X\|y^*\|_2)/2$. Set $y_0 = \mathbf{0}$ and $\{\gamma_t, \theta_t, \eta_t, \tau_t\}$ in Algorithm 1 according to the following: $\gamma_t = 1$, $\eta_t = L_0 + L_f + \eta$, and $\theta_t = 1$, $\tau_t = \tau$, where*

$$\eta := \max\left\{\frac{\sqrt{2T[\mathcal{H}_*^2 + \sigma_{0,v_0}^2 + 48\|\sigma_v\|_2^2]}}{D_X}, \frac{6\max\{2M_f, 4\|\sigma_v\|_2\}}{D_X}\right\},$$
$$\tau := \max\left\{\sqrt{96T}\sigma_{X,f}, 2D_X\max\{M_f, 4\|\sigma_v\|_2\}\right\}.$$

Then, we have

$$\mathbb{E}[f_0(\bar{x}_T) - f_0(x^*)] \leq \frac{(L_0 + L_f)D_X^2 + \max\{12M_f, 24\|\sigma_v\|_2\}D_X}{T} + \frac{1}{\sqrt{T}}\sqrt{2(\mathcal{H}_*^2 + \sigma_{0,v_0}^2 + 48\|\sigma_v\|_2^2)}D_X + \frac{1}{\sqrt{T}}\left\{\frac{\sqrt{2}\zeta^2 D_X}{\sqrt{\mathcal{H}_*^2 + \sigma_{0,v_0}^2 + 48\|\sigma_v\|_2^2}} + \frac{\sqrt{3}\sigma_{X,f}}{\sqrt{2}}\right\} + \left[v_0^2 L_0 n + M_X n\left(\sum_{i=1}^m v_i^4 L_i^2\right)^{1/2}\right], \tag{8}$$

and

$$\mathbb{E}[\|[f(\bar{x}_T)]_+\|_2] \leq \frac{1}{\sqrt{T}}\left\{\left[12\sqrt{6}(\|y^*\|_2 + 1)^2 + \frac{13}{4\sqrt{6}}\right]\sigma_{X,f} + \left[v_0^2 L_0 n + M_X n\left(\sum_{i=1}^m v_i^4 L_i^2\right)^{1/2}\right] + \sqrt{2}D_X\left[\sqrt{\mathcal{H}_*^2 + \sigma_{0,v_0}^2 + 48\|\sigma_v\|_2^2} + \frac{\zeta^2 + \mathcal{H}_*^2}{\sqrt{\mathcal{H}_*^2 + \sigma_{0,v_0}^2 + 48\|\sigma_v\|_2^2}}\right]\right\} + \frac{(L_0 + L_f)D_X^2 + \max\{12M, 24\|\sigma_v\|_2\}D_X(1 + (\|y^*\|_2 + 1)^2)}{T}, \tag{9}$$

where $\zeta := 2e\{\sigma_{0,v_0}^2 + \|\sigma_v\|_2^2(14\|y^*\|_2^2 + 75) + 2\sqrt{3}\|\sigma_v\|_2(2\mathcal{H}_* + \sigma_{0,v_0} + \sqrt{48}\|\sigma_v\|_2) + \sqrt{6}D_X^{-1}\|\sigma_v\|_2[v_0^2 L_0 n + M_X n\left(\sum_{i=1}^m v_i^4 L_i^2\right)^{1/2}]\sqrt{T}\}^{1/2}$. *Hence, by choosing*

$$v_0 \leq \min\left\{\frac{1}{\sqrt{2L_0 n\sqrt{T}}}, \frac{2}{(n+3)^{3/2}}, \frac{1}{L_i(n+6)^{3/2}}\right\}, \tag{10}$$

$$v_i \leq \min\left\{\frac{2}{(n+3)^{3/2}}, \frac{1}{2M_i\sqrt{(n+2)m}}, \frac{1}{\sqrt{L_i n\sqrt{m}}}, \frac{1}{\sqrt{2L_i n M_X\sqrt{Tm}}}, \frac{1}{L_i(n+6)^{3/2}\sqrt{m}}\right\}, \tag{11}$$

for $i \in [m]$, the number of calls to the stochastic zeroth-order oracle required by Algorithm 1 to find an ε-approximately optimal solution of (1) is of the order $\mathcal{O}(((m+1)n)/\epsilon^2)$.

Remark 1. Although the parameter settings of Theorem 1 and the right-hand side of (8) and (9) appear complicated to parse, the important takeaway message is that the right-hand side of (8) and (9) are of the order $\mathcal{O}(1/\sqrt{T})$, which leads to the oracle complexity described in the preceding. Furthermore, the order of ϵ in the oracle complexity is of the same order as that in Boob et al. (2022) for the stochastic first-order setting. The $(m+1)n$ factor in the oracle complexity appears because we are required to estimate $m+1$ gradient vectors, each of

dimension n. The dimension dependency is unavoidable even in the unconstrained setting, as shown via lower bounds in Jamieson et al. (2012) and Duchi et al. (2015). For a fixed dimensionality n, the oracle complexity in the zeroth-order setting is linear in the number of constraints m.

Remark 2. A word is in order regarding the choice of the tuning parameters ν_i, $i \in [m]$ in (11). If one follows the standard analysis for selecting the tuning parameters for stochastic zeroth-order algorithms, which are predominantly developed for unconstrained problems, the m related factors appearing in the choice of ν_i would be missed. This subsequently would lead to an increased dependency of the oracle complexity on m, instead of the linear dependency that we obtain now. A main part of our proof involves obtaining the choice of the smoothing parameters ν_i as in (11), which helps us to obtain oracle complexity as stated in Theorem 1.

3.2. Proximal-Point Based Meta-Algorithm for the Nonconvex Setting

We now consider the case when objective function f_0 and the constraint functions $f_1, \ldots, f_m$ are nonconvex. In this case, Boob et al. (2022) analyzed a two-step meta-algorithm, which is based on the standard proximal method; see, for example, Drusvyatskiy (2017) for a survey.

The basic idea behind the method (as stated in Algorithm 2) consists of the following two steps: (i) we construct a sequence of convex relaxations for the nonconvex problem, and (ii) we leverage the algorithm developed for the convex setting. Given Algorithm 1, we leverage this framework to solve (1) in the nonconvex setting.

Algorithm 2 (Meta-Algorithm for Nonconvex Setting)

Input: Input x_0, parameters μ_o, μ_i, $i \in [m]$.

1: **for** $k = 1, \ldots, K$ **do**

2: For $i \in [m]$, set

$$f_0(x; x_{k-1}) := f_0(x) + 2\mu_0 W(x, x_{k-1}),$$
$$f_i(x; x_{k-1}) := f_i(x) + 2\mu_i W(x, x_{k-1}).$$

3: Obtain an ϵ-approximately optimal solution to the problem:

$$\arg\min_{x \in X} f_0(x; x_{k-1}) \quad \text{s.t.} \quad f_i(x; x_{k-1}) \leqslant 0, \quad i \in [m] \tag{12}$$

by using `SZO-ConEx` in Algorithm 1. Denote it by x_k for $k = 1, \ldots, K$.

4: Randomly choose $\hat{k} \in \{1, \ldots, K\}$

5: **return** $x_{\hat{k}}$.

We first define the exact KKT condition for (1) as follows. For a convex set X, we denote its interior as int$X$, the normal cone at $x \in X$ as $N_X(x)$, and its dual cone as $N_X^*(x)$. For convenience, we recall the definition of a normal cone: for a convex set X, we have $N_X^*(x) := \{v \in \mathbb{R}^n : \langle y, z - x \rangle \leq 0 \text{ for all } z \in X\}$; see parts I and II of Rockafellar (2015) for additional properties and examples. Let $\oplus$ denote the Minkowski sum of two sets $A, B \subset \mathbb{R}^n$, defined as $A \oplus B = \{a + b : a \in A \text{ and } b \in B\}$. We refer to the distance between two sets $A, B \subset \mathbb{R}^n$ as $d(A, B) := \inf_{a \in A, b \in B} \|a - b\|$.

Definition 3. We say that $x^* \in X$ is a critical KKT point of (1) if $f_i(x^*) \leqslant 0$ and $\exists y^* := [y_1^*, \ldots, y_m^*]^T \geqslant \mathbf{0}$ such that

$$y_i^* f_i(x^*) = 0, \quad i \in [m],$$
$$d(\nabla f_0(x^*) + \sum_{i=1}^m y_i^* \nabla f_i(x^*) \oplus N_X(x^*), \mathbf{0}) = 0.$$

The parameters $\{y_i^*\}_{i \in [m]}$ are called *Lagrange multipliers*. For brevity, we use the notation y^* and $[y_1^*, \ldots, y_m^*]^T$ interchangeably. With this definition, we also have the following approximate KKT condition, which is the standard approximate optimality condition for solving (1) in the nonconvex setting.

Definition 4. We say that a point $\hat{x} \in X$ is an (ε, δ)-KKT point in expectation for (1) if there exists $(\overline{x}, \overline{y})$ such that $f(\overline{x}) \leqslant \mathbf{0}, \overline{y} \geqslant \mathbf{0}$ and

$$\mathbb{E}\left[\sum_{i=1}^m |\overline{y}_i f_i(\overline{x})|\right] \leqslant \varepsilon, \ \mathbb{E}\|\overline{x} - \hat{x}\|^2 \Bigg] \leqslant \delta$$
$$\mathbb{E}\left[\left(d(\nabla f_0(\overline{x}) + \sum_{i=1}^m \overline{y}_i \nabla f_i(\overline{x}) \oplus N_X(\overline{x}), \mathbf{0})\right)^2\right] \leqslant \varepsilon.$$

Proposition 1. *Consider solving* (1) *with both the objective and the constraint function being nonconvex and satisfying Assumptions* 1–3. *Then, by running Algorithm* 2 *with* $K = \mathcal{O}(1/\epsilon)$, *we obtain an* $(\epsilon, 2\epsilon/2\mu_0\mu_{\max})$*-KKT point, where* $\mu_{\max} := \max\{\mu_1, \ldots, \mu_m\}$. *Hence, the total number of calls to the stochastic zeroth-order oracle is given by* $\mathcal{O}\big(((m+1)n)/\epsilon^3\big)$.

The proof of this proposition follows immediately by Theorem 1 and corollary 3.19 from Boob et al. (2022) and is hence omitted. The parameters μ_0 and μ_i, $i \in [m]$, in Algorithm 2 are set according to the desired level of accuracy based on Proposition 1. To the best of our knowledge, we are not aware of a nonasymptotic result on the oracle complexity of stochastic zeroth-order optimization with stochastic zeroth-order functional constraints in both the convex and nonconvex settings.

3.3. Detailed Comparison with Boob et al. (2022)

In this subsection, we highlight the main differences between our work and Boob et al. (2022). As mentioned previously, our methodological and theoretical results build on the work of Boob et al. (2022).

- *Methodology:* At a methodological level, our work focuses on the case when we only have noisy function evaluations, whereas Boob et al. (2022) focused on the case when we have access to noisy gradients. To deal with this, we use the Gaussian smoothing-based zeroth-order gradient estimator in combination with the constraint extrapolation technique from Boob et al. (2022).
- *Biased gradients:* The use of the Gaussian smoothing-based zeroth-order gradient estimator leads to stochastic gradients that are biased. Although Boob et al. (2022) considered noisy gradients, they assumed that their stochastic gradients were unbiased. This complicates the analysis of the zeroth-order setting we work with.
- *Nonuniform variance:* Apart from the unbiased stochastic gradient assumption, Boob et al. (2022) required the variance of their stochastic gradient to be *uniformly bounded* over the entire parameter space. However, the Gaussian smoothing-based gradient estimator does not satisfy this assumption. A major technical part of our analysis involves dealing with stochastic gradients that are not uniformly bounded.
- *Smoothing parameters:* Our method requires dealing with the additional tuning parameters (ν_i's) that determine the level of smoothing in the zeroth-order gradient estimator. Dealing with this requires a careful analysis, as otherwise, one would end up with a worse oracle complexity than what we have established in this work; see Remark 2 for details. By contrast, Boob et al. (2022) did not require dealing with any tuning parameters for their stochastic gradient because of their generic set of assumptions.
- *Experiments:* Boob et al. (2022) did not provide any experimental verification of their algorithm. By contrast, in Section 4, we provide a detailed experimental evaluation, comparing it with the existing state-of-the-art methods for constrained zeroth-order optimization, and demonstrate the advantages of the proposed approach.

4. Experimental Results

We compare the performance of our algorithm (Algorithm 1) with the following widely used algorithms for constrained zeroth-order optimization.

- The `ALBO` method by Gramacy et al. (2016): This method takes a hybrid approach for constrained zeroth-order optimization, based on combining Bayesian optimization (i.e., Gaussian process-based approaches) with augmented Lagrangian methods. Specifically, the objective function of augmented Lagrangian (which is similar in spirit to (5)) is estimated using Gaussian process priors. This method has various tuning parameters, which makes the implementation a bit difficult. In fact, Gramacy et al. (2016) did not provide the full implementation details and mention that "many specifics have been omitted for space considerations." We use the implementation provided in Gramacy (2016, p. 7) as recommended by Gramacy et al. (2016).
- The `Slack-AL` method by Picheny et al. (2016): This method builds on the `ALBO` method and is also a hybrid method. Specifically, a particular step in estimating the objective function using Gaussian process technique (referred to as the Expected-Improvement step) is avoided by using slack variables. Similar to the previous method, we use the implementation provided in Gramacy (2016).
- The `ADMMBO` method by Ariafar et al. (2019): This method is also a hybrid method that uses Bayesian optimization methods. However, the authors used an approach based on an alternating direction method of multipliers (ADMM) to solve the augmented Lagrangian problem. We follow the recommendation in section 5.1 of Ariafar et al. (2019) for the implementation.
- The `PESC` method by Hernández-Lobato et al. (2015): This is a purely Bayesian optimization method that uses predictive entropy search for solving constrained zeroth-order optimization methods. As mentioned in Hernández-Lobato et al. (2015, p. 5), "One disadvantage of `PESC` is that it is relatively difficult to implement." Furthermore, all the implementation details are not provided in detail in Ariafar et al. (2019). Hence, we follow the implementation provided in Ariafar et al. (2019) for our experiments.

Compared with the above-mentioned methods, our algorithm comes with a theoretical guarantee for setting the various tuning parameters of the proposed algorithm.

We first report simulation experiments on (i) the oracle complexity of `SZO-ConEX` on two different test case objective and constraint functions and (ii) the effect of the smoothing parameters (corresponding to the zeroth-order gradient estimation process) on the oracle complexity. For our experiments, we consider the following optimization problem (termed quadratically constrained quadratic programming (QCQP) in the literature) where the objective function and the constraint function are quadratic functions:

$$\min_{x\in\mathbb{R}^n} f_0(x) := x^\top A_0 x + b_0^\top x + c_0$$
$$\text{such that } f_1(x) := x^\top A_1 x + b_1^\top x + c_1 \leqslant 1.$$

Here, $A_0, A_1 \in \mathbb{R}^{n\times n}$, $b_0, b_1 \in \mathbb{R}^n$, and $c_0, c_1 \in \mathbb{R}$. When the matrices $A_0, A_1 \in \mathbb{R}^{n\times n}$ are further assumed to be symmetric and positive semidefinite, the preceding problem is a convex optimization problem with convex constraints. In the general case, nonconvex quadratically constrained quadratic programs form a rich class of optimization problems. For example, every polynomial optimization problem with polynomial constraint could be turned into a nonconvex QCQP at the expense of increasing the number of the optimization variables (d'Aspremont and Boyd 2003). Furthermore, it is also known that it is NP-hard to find global minimizers of the nonconvex QCQP problem in the worst case.

We now elaborate on the two settings we consider.

1. We first consider the *convex setting*. Here, we set A_0 and A_1 to be random but fixed symmetric positive semidefinite matrices. Similarly, b_0, b_1, c_0 and c_1 were generated randomly but fixed. Hence, the problem instance is fixed. In our experiments, we only use (noisy) function evaluations of both the objective and constraint functions. We used the standard normal distribution and Student's t-distribution with five degrees of freedom for the noise in the function evaluations. For Algorithm 1, θ_t was set to 1 based on the theoretical result. Furthermore, τ and η, the parameters corresponding to the ascent step and the descent step, respectively, were set based on trial and error to achieve the best performance. We remark that one could potentially use principled approaches such as line search for setting the step-size parameters (Berahas et al. 2021). As we are working in the zeroth-order setting, in our experimentation we provide additional attention to the smoothing parameters (ν_0 and ν_1) corresponding to the zeroth-order gradient estimators. We set them both to 0.05, 0.1, and 2 and report our performance.

In Figure 1, we report the function value difference (corresponding to Theorem 1) versus the number of calls to the (noisy) zeroth-order oracle for various algorithms and our algorithm with the three choices of smoothing parameters. We work with dimensions $n = 200$ and $n = 500$ for our problem. Note here that it is easy to obtain the function value at the optimal solution for the convex QCQP by using standard solvers (we use `cvxpy` to calculate it). The curves in Figure 1 correspond to an average of over 100 trials. We notice that the performance of our algorithm is uniformly better than the compared algorithms in terms of the number of function calls required to obtain a prescribe accuracy. Furthermore, we notice that our algorithm is robust to the choice of smoothing parameters: as long as it is small enough, we have fast convergence, but the iterates diverge when the smoothing parameter value is large.

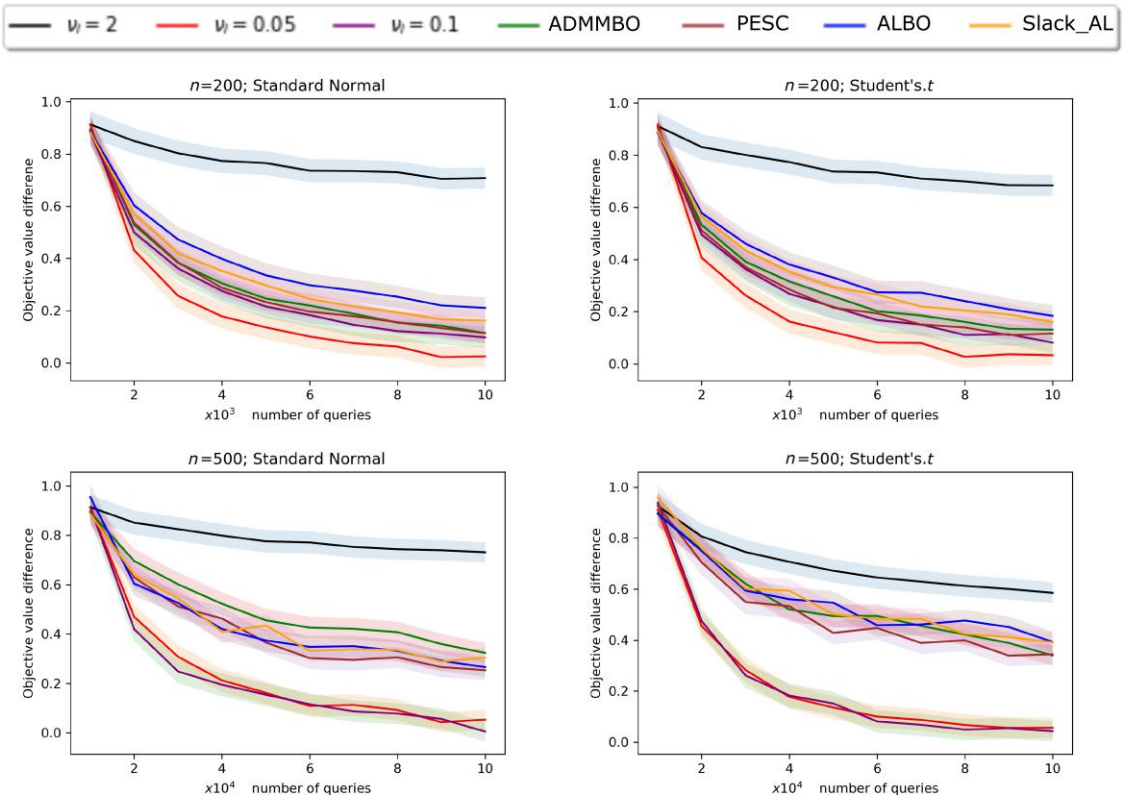
2. We now list the changes we make for the *nonconvex setting*. First, although the matrices are still random but fixed, we make them non-positive-definite. Furthermore, for Algorithm 2, we set $K = 50$. In the bottom two rows of Figure 2, we report the norm of the gradient of the objective function (corresponding to Theorem 1) versus the number of calls to the (noisy) zeroth-order oracle for various algorithms and our algorithm with the three choices of smoothing parameters. The curves in Figure 2 correspond to an average of over 100 trials. We notice that, similar to the convex case, the performance of our algorithm is uniformly better than the compared algorithms in terms of the number of function calls required to obtain a prescribed accuracy.

A brief summary of the observations follows: (i) The oracle complexity of `SZO-ConEX` method is consistently lower than other existing techniques including `ALBO` (Gramacy et al. 2016), `Slack-AL` (Picheny et al. 2016), `ADMMBO` (Ariafar et al. 2019), and `PESC` (Hernández-Lobato et al. 2015), highlighting the benefit of the *constraint extrapolation* step. (ii) The `SZO-ConEX` method is robust to the smoothing parameters as long as it is less than a particular threshold. Next, we report the performance of our algorithm on the two motivating examples from Section 1.

4.1. Application I: Tuning the HMC Algorithm

We now consider the problem of optimizing the hyperparameters of the HMC algorithm. A brief description of the HMC algorithm is provided in the supplementary material for completeness. We follow Gelbart et al. (2014) and Hernández-Lobato et al. (2015) closely for the experimental setup. The specific hyperparameters that we

Figure 1. (Color online) Performance Comparison on Simulation Experiment: Plot of Number of Queries vs. Objective Value Difference



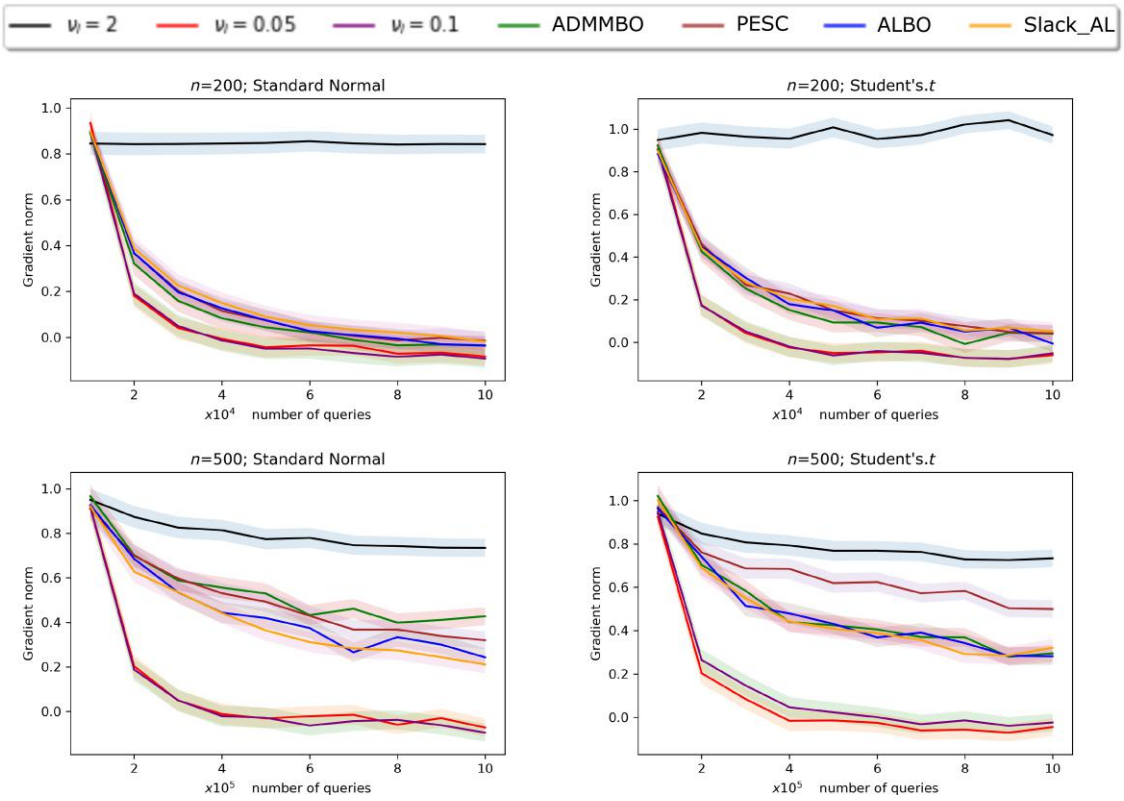
Notes. The plots represent average curves over 100 trials, and the shaded region corresponds to the standard errors. In the legend, the curves corresponding to ν_i correspond to the SZO-ConEX algorithm.

consider for this experiment are (i) the number of leapfrog steps, denoted by τ, (ii) the step-size parameter, denoted by η, (iii) the scalar coefficient of the mass matrix, denoted by κ (here, following Neal (2011), we parametrize the mass matrix as κ times an identity matrix), and (iv) the fraction of the allotted time the algorithm spends in the burning phase. Hence, the optimization variables are given by $x \in \mathbb{R}^4$. We remark that although the number of leap-frog steps is an integer, for our experiments, we consider it to be real-valued number. In practice, we round it off to the closest integer, with ties broken randomly.

The objective function we maximize is the number of effective samples in a fixed computation time. This is a widely used diagnostic metric for measuring the performance of sampling algorithms in Bayesian statistical machine learning (Kass et al. 1998, Lenth 2001). For sampling problems, an effective sample size is defined as follows. First note that the samples outputted by a sampling algorithm are typically correlated. The effective sample size is defined as the number of *independent* samples from the target density that achieves the same performance as the correlated samples outputted by the sampling algorithm. However, there is no closed-form analytical relationship between this performance measure and the optimization variable x. For our experiments, we use the CODA package (Plummer et al. 2006) for calculating the effective sample size. The constraint functions that we use are as follows: (i) the generated samples must pass the Geweke diagnostics (Geweke (1991)), where the worst Geweke test score across all variables and chains could be at most 2.0, and (ii) the generated samples must pass the Gelman–Rubin convergence diagnostics (Gelman and Rubin 1992), where the worst Gelman–Rubin score between variables and chains could be at most be 1.2. The analytical form of the above-mentioned convergence diagnostics and the optimization variable x is also not available in closed form. We use the PyMC package (Patil et al. 2010) for evaluating the above-mentioned diagnostic metrics.

We tune the HMC sampling algorithm with the aforementioned setup for the problem of sampling from the posterior distribution of a logistic regression binary classification problem on the German credit data set from

Figure 2. (Color online) Performance Comparison on Simulation Experiment: Plot of Number of Queries vs. Norm of the Gradient



Notes. The plots represent average curves over 100 trials, and the shaded region corresponds to the standard error. In the legend, the curves corresponding to ν_i correspond to the SZO-ConEX algorithm.

the UCI Machine Learning Repository (Dua and Graff 2017). The data set contains 1,000 observations that are normalized to have unit variance. We initialize each chain randomly with independent draws from a Gaussian distribution with mean 0 and standard deviation 10^{-3}. For each set of inputs, we compute two chains, each with 5 minutes of computation time. As mentioned previously, all our simulation settings following those of Gelbart et al. (2014) and Hernández-Lobato et al. (2015). We conduct our experiments by subsampling data sets of size 800 from the original data set and repeating the procedure for 100 trials. We compare the performance of our algorithm (with $K = 50$) with that of the ALBO method by Gramacy et al. (2016), the Slack-AL method by Picheny et al. (2016), the ADMMBO method by Ariafar et al. (2019), and the PESC method by Hernández-Lobato et al. (2015). The tuning parameters of the respective methods were set according to the guidelines provided in the papers. For our algorithm, we found the performance was robust to the choice of the smoothing parameters, as long as it was sufficiently small. For the performance reported in Table 1, we set it to $\nu_i = 0.05$. In Table 1, we report the average effective sample size (ESS) for the various methods, along with the standard deviation. We notice that the performance of SZO-ConEx is significantly better than that of the other methods, thereby demonstrating the effectiveness of our method for the problem of hyperparameter tuning for the HMC sampling algorithm.

Table 1. ESS of Hamiltonian Monte Carlo Sampling Algorithm Tuned by Various Methods, Along with Their Standard Errors.

Algorithm	ALBO	Slack-AL	ADMMBO	PESC	SZO-ConEx
ESS	$9.4 \times 10^4 \pm 924$	$9.3 \times 10^4 \pm 982$	$9.4 \times 10^4 \pm 884$	$9.9 \times 10^4 \pm 998$	$10.8 \times 10^4 \pm 992$

Table 2. VE Along the Standard Error of Three-Layer Neural Network Trained Using SGD with Momentum for 5,000 Iterations on MNIST and CIFAR-10 Data Sets by Picking Hyperparameters Tuned by Various Methods

Algorithm	ALBO	Slack-AL	ADMMBO	PESC	SZO-ConEx
VE on MNIST	3.4 ± 0.05	3.1 ± 0.08	3.0 ± 0.05	2.9 ± 0.03	1.9 ± 0.04
VE on CIFAR-10	4.7 ± 0.02	4.0 ± 0.03	3.9 ± 0.05	3.4 ± 0.03	2.2 ± 0.02

Note. The numbers reported are related to the constraint that the prediction time is not greater than 0.050 seconds on a Nvidia Tesla K20 GPU.

4.2. Application II: Tuning a Three-Layer Neural Network

Next, we turn to the problem of tuning the hyperparameters of a three-layer neural network with the rectified linear unit activation function trained by the stochastic gradient descent algorithm with momentum (Sutskever et al. 2013) for 5,000 iterations. We follow Hernández-Lobato et al. (2015) and Ariafar et al. (2019) closely for the experimental setup. The specific hyperparameters that we consider for this experiment are as follows: (i) two learning rate parameters (initial and decay rate), (ii) momentum parameters (initial and final), (iii) dropout parameters (input layer and hidden layers), (iv) regularization parameters corresponding to the weight decay and max weight norm, and (v) the number of hidden units in each of the three hidden layers. Hence, the optimization variables are given by $x \in \mathsf{R}^{11}$. Similar to the previous experiment, we treat the number of hidden layers as a real-valued variable and use the same rounding technique in practice.

The objective function we minimize is the classification error on the validation set (which we call the validation error (VE)). Indeed, there is no good closed-form expression connecting the above-mentioned hyperparameters and the VE. The constraint function that we use is that the prediction time must not exceed 0.050 seconds. Here, we compute the prediction time as the average time of 1,000 predictions, over a batch of size 128 (Hernández-Lobato et al. 2015, Ariafar et al. 2019). The number 0.050 seconds is set based on the computing resource we use (Nvidia Tesla K20 GPU) so that we can see an active trade-off between the objective function (the VE) and the constraint function (prediction time). As highlighted by Hernández-Lobato et al. (2015) and Ariafar et al. (2019), this specific choice is highly dependent on the computing resource used. Clearly, there is no analytical form for the function describing the relationship between the hyperparameters and the constraint function. All our implementations for this experiment were based on the PyTorch open source machine learning library (Paszke et al. 2019).

We tune the SGD algorithm with momentum with the above-mentioned setup for the problem of classification on MNIST (LeCun and Cortes 2010) and CIFAR-10 data sets (Krizhevsky 2009). For both data sets, we conduct our experiments by subsampling 90% of the training data and report our error over 100 trials. Similar to the previous case, we compare the performance of our algorithm (with $K = 50$) with that of the ALBO method by Gramacy et al. (2016), the Slack-AL method by Picheny et al. (2016), the ADMMBO method by Ariafar et al. (2019), and the PESC method by Hernández-Lobato et al. (2015). The tuning parameters of the respective methods were set as suggested in the respective papers. The smoothing parameter for our algorithm was set as $\nu_i = 0.03$. In Table 2, we report the validation error achieved such that the constraint on the prediction time is respected for the various algorithms. From the results, we notice that the SZO-ConEX method outperforms the other methods on both the MNIST and CIFAR-10 data sets.

5. Conclusion

In this paper, we proposed and analyzed stochastic zeroth-order optimization algorithms for nonlinear optimization problems with functional constraints. We consider the case when both the objective function and the constraint functions are observed only via noisy function queries. Our algorithm is based on leveraging the constraint extrapolation technique proposed by Boob et al. (2022) and the Gaussian smoothing technique. We characterize the oracle complexity of the proposed algorithm in both the convex and nonconvex settings. We also apply our methodology to the problem of hyperparameter tuning for the HMC algorithm and three-layer neural networks trained using SGD with momentum, and we demonstrate its superior performance.

For future work, we plan to develop parallel versions of our algorithm for the case when the objective functions and the constraint functions are available only locally in different machines. We also plan to develop lower bounds on the oracle complexity of stochastic zeroth-order optimization algorithms in the constrained setting. It is of great interest to find other applications of the proposed methodology in statistical machine learning, reinforcement learning, and other scientific and engineering fields. Finally, it is also interesting to extend our methodology

to the case of mixed constraints (i.e., equality and inequality constraints) and to develop novel methodology and analysis for constrained zeroth-order optimization with both binary and real-valued decision variables.

Acknowledgments

The authors are grateful to the anonymous reviewers for their constructive comments that greatly helped improve the presentation of this paper.

References

Acerbi L, Ma WJ (2017) Practical Bayesian optimization for model fitting with Bayesian adaptive direct search. Guyon I, Von Luxburg U, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, eds. *Advances in Neural Information Processing Systems*, vol. 30 (Curran Associates, Red Hook, NY), 1836–1846.

Agarwal A, Dekel O, Xiao L (2010) Optimal algorithms for online convex optimization with multi-point bandit feedback. Kalai AT, Mohri M, eds. *23rd Conf. Learn. Theory* (Omnipress, Madison, WI), 28–40.

Amaioua N, Audet C, Conn AR, Le Digabel S (2018) Efficient solution of quadratically constrained quadratic subproblems within the mesh adaptive direct search algorithm. *Eur. J. Oper. Res.* 268(1):13–24.

Archetti F, Candelieri A (2019) *Bayesian Optimization and Data Science* (Springer, Cham, Switzerland).

Ariafar S, Coll-Font J, Brooks DH, Dy JG (2019) ADMMBO: Bayesian optimization with unknown constraints using ADMM. *J. Machine Learn. Res.* 20(123):1–26.

Audet C, Tribes C (2018) Mesh-based Nelder–Mead algorithm for inequality constrained optimization. *Comput. Optim. Appl.* 71(2):331–352.

Audet C, Dennis JE Jr (2004) A pattern search filter method for nonlinear programming without derivatives. *SIAM J. Optim.* 14(4):980–1010.

Audet C, Dennis JE Jr (2006) Mesh adaptive direct search algorithms for constrained optimization. *SIAM J. Optim.* 17(1):188–217.

Audet C, Dennis JE Jr (2009) A progressive barrier for derivative-free nonlinear programming. *SIAM J. Optim.* 20(1):445–472.

Audet C, Hare W (2017) *Derivative-Free and Blackbox Optimization* (Springer, Cham, Switzerland).

Audet C, Le Digabel S, Peyrega M (2015) Linear equalities in blackbox optimization. *Comput. Optim. Appl.* 61(1):1–23.

Augustin F, Marzouk YM (2014) NOWPAC: A provably convergent derivative-free nonlinear optimizer with path-augmented constraints. Preprint, submitted March 8, https://doi.org/10.48550/arXiv.1403.1931.

Bachoc F, Helbert C, Picheny V (2020) Gaussian process optimization with failures: Classification and convergence proof. *J. Global Optim.* 78(3):483–506.

Balandat M, Karrer B, Jiang D, Daulton S, Letham B, Wilson AG, Bakshy E (2020) BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, eds. *Advances in Neural Information Processing Systems*, vol. 33 (Curran Associates, Red Hook, NY), 21524–21538.

Balasubramanian K, Ghadimi S (2018) Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates. Bengio S, Wallach HM, Larochelle H, Grauman K, Cesa-Bianchi N, eds. *Proc. 32nd Internat. Conf. Neural Inform. Processing Systems* (Curran Associates, Red Hook, NY), 3459–3468.

Balasubramanian K, Ghadimi S (2022) Zeroth-order nonconvex stochastic optimization: Handling constraints, high-dimensionality and saddle-points. *Foundations Comput. Math.* 22(1):35–76.

Balasubramanian K, Ghadimi S, Nguyen A (2022) Stochastic multilevel composition optimization algorithms with level-independent convergence rates. *SIAM J. Optim.* 32(2):519–544.

Beck A (2017) *First-Order Methods in Optimization* (SIAM, Philadelphia).

Beck A, Teboulle M (2012) Smoothing and first order methods: A unified framework. *SIAM J. Optim.* 22(2):557–580.

Berahas AS, Cao L, Scheinberg K (2021) Global convergence rate analysis of a generic line search algorithm with noise. *SIAM J. Optim.* 31(2):1489–1518.

Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. *J. Machine Learn. Res.* 13(2):281–305.

Blum JR (1954) Multidimensional stochastic approximation methods. *Ann. Math. Statist.* 25(4):737–744.

Boob D, Deng Q, Lan G (2022) Stochastic first-order methods for convex and nonconvex functional constrained optimization. *Math. Programming*, ePub ahead of print January 21, https://doi.org/10.1007/s10107-021-01742-y.

Brent RP (2013) *Algorithms for Minimization Without Derivatives* (Dover Publications, Mineola, NY).

Bueno LF, Friedlander A, Martinez JM, Sobral FNC (2013) Inexact restoration method for derivative-free optimization with smooth constraints. *SIAM J. Optim.* 23(2):1189–1213.

Bűrmen Á, Puhan J, Tuma T (2006) Grid restrained Nelder-Mead algorithm. *Comput. Optim. Appl.* 34(3):359–375.

Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A (2017) Stan: A probabilistic programming language. *J. Statist. Software* 76(1):1–32.

Chen T, Fox E, Guestrin C (2014) Stochastic gradient Hamiltonian Monte Carlo. Xing EP, Jebara T, eds. *Proc. 31st Internat. Conf. Machine Learn.* (PMLR), 1683–1691.

Choromanski K, Pacchiano A, Parker-Holder J, Tang Y, Jain D, Yang Y, Iscen A, Hsu J, Sindhwani V (2020) Provably robust blackbox optimization for reinforcement learning. Kaelbling LP, Kragic D, Sugiura K, eds. *Proc. Conf. Robot Learn.*, vol. 100 (PMLR), 683–696.

Conn AR, Le Digabel S (2013) Use of quadratic models with mesh-adaptive direct search for constrained black box optimization. *Optim. Methods Software* 28(1):139–158.

Conn A, Scheinberg K, Vicente L (2009) *Introduction to Derivative-Free Optimization*, vol. 8 (SIAM, Philadelphia).

d'Aspremont A, Boyd S (2003) Relaxations and randomized methods for nonconvex QCQPs. EE392o Class Notes, Stanford University, Stanford, CA. https://web.stanford.edu/class/ee392o/relaxations.pdf.

Dippon J (2003) Accelerated randomized stochastic optimization. *Ann. Statist.* 31(4):1260–1281.

Drusvyatskiy D (2017) The proximal point method revisited. Preprint, submitted December 17, https://doi.org/10.48550/arXiv.1712.06038.

Dua D, Graff C (2017) UCI Machine Learning Repository. Accessed August 1, 2022, http://archive.ics.uci.edu/ml.

Duane S, Kennedy AD, Pendleton BJ, Roweth D (1987) Hybrid Monte Carlo. *Phys. Lett. B* 195(2):216–222.

Duchi JC, Jordan MI, Wainwright MJ, Wibisono A (2015) Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Trans. Inform. Theory* 61(5):2788–2806.

Dzahini KJ, Kokkolaras M, Le Digabel S (2022) Constrained stochastic blackbox optimization using a progressive barrier and probabilistic estimates. *Math. Programming*, ePub ahead of print March 30, https://doi.org/10.1007/s10107-022-01787-7.

Echebest N, Schuverdt ML, Vignau RP (2017) An inexact restoration derivative-free filter method for nonlinear programming. *Comput. Appl. Math.* 36(1):693–718.

Elsken T, Metzen JH, Hutter F (2019) Neural architecture search: A survey. *J. Machine Learn. Res.* 20(1):1997–2017.

Eriksson D, Poloczek M (2021) Scalable constrained Bayesian optimization. *Internat. Conf. Artificial Intelligence Statist*. (PMLR), 730–738.

Fasano G, Liuzzi G, Lucidi S, Rinaldi F (2014) A line-search based derivative-free approach for nonsmooth constrained optimization. *SIAM J. Optim.* 24(3):959–992.

Frazier PI (2018) A tutorial on Bayesian optimization. Preprint, submitted July 8, https://doi.org/10.48550/arXiv.1807.02811.

Gao W, Graesser L, Choromanski K, Song X, Lazic N, Sanketi P, Sindhwani V, Jaitly N (2020) Robotic table tennis with model-free reinforcement learning. Preprint, submitted March 31, https://arxiv.org/abs/2003.14398.

Gardner J, Kusner M, Weinberger K, Cunningham J (2014) Bayesian optimization with inequality constraints. Xing EP, Jebara T, eds. *Proc. 31st Internat. Conf. Machine Learn.* (PMLR), 937–945.

Gelbart MA, Snoek J, Adams RP (2014) Bayesian optimization with unknown constraints. Zhang N, Tian J, eds. *Proc. 30th Conf. Uncertainty Artificial Intelligence* (AUAI Press, Arlington, VA), 250–259.

Gelbart MA, Adams RP, Hoffman MW, Ghahramani Z (2016) A general framework for constrained Bayesian optimization using information-based search. *J. Machine Learn. Res.* 17(160):1–53.

Gelman A, Rubin DB (1992) A single series from the Gibbs sampler provides a false sense of security. Bernardo JM, Berger JO, Dawid AP, Smith AFM, eds. *Bayesian Statistics 4* (Oxford University Press, Oxford, UK), 625–632.

Geweke J (1991) Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. Research Development Staff Report 148, Federal Reserve Bank of Minneapolis, Minneapolis.

Ghadimi S, Lan G (2013) Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM J. Optim.* 23(4):2341–2368.

Ghadimi S, Ruszczynski A, Wang M (2020) A single timescale stochastic approximation method for nested stochastic optimization. *SIAM J. Optim.* 30(1):960–979.

Girolami M, Calderhead B (2011) Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J. Roy. Statist. Soc. Ser. B* 73(2):123–214.

Golovin D, Solnik B, Moitra S, Kochanski G, Karro J, Sculley D (2017) Google Vizier: A service for black-box optimization. *Proc. 23rd ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (ACM, New York), 1487–1495.

Goodfellow I, Bengio Y, Courville A, Bengio Y (2016) *Deep Learning*, vol. 1 (MIT Press, Cambridge, MA).

Gramacy RB (2016) laGP: Large-scale spatial modeling via local approximate Gaussian processes in R. *J. Statist. Software* 72(1):1–46.

Gramacy RB, Gray GA, Le Digabel S, Lee H, Ranjan P, Wells G, Wild SM (2016) Modeling an augmented Lagrangian for blackbox constrained optimization. *Technometrics* 58(1):1–11.

Gratton S, Vicente LN (2014) A merit function approach for direct search. *SIAM J. Optim.* 24(4):1980–1998.

Greenhill S, Rana S, Gupta S, Vellanki P, Venkatesh S (2020) Bayesian optimization for adaptive experimental design: A review. *IEEE Access* 8:13937–13948.

Hamedani EY, Aybat NS (2021) A primal-dual algorithm with line search for general convex-concave saddle point problems. *SIAM J. Optim.* 31(2):1299–1329.

Hazan E, Klivans A, Yuan Y (2018) Hyperparameter optimization: A spectral approach. *6th Internat. Conf. Learn. Representations* (Vancouver, Canada), https://openreview.net/forum?id=H1zriGeCZ.

Hernández-Lobato JM, Gelbart M, Hoffman M, Adams R, Ghahramani Z (2015) Predictive entropy search for Bayesian optimization with unknown constraints. Bach F, Blei D, eds. *Proc. 32nd Internat. Conf. Machine Learn.* (PMLR), 1699–1707.

Hoffman MD, Gelman A (2014) The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Machine Learn. Res.* 15(1):1593–1623.

Hooke R, Jeeves TA (1961) Direct search solution of numerical and statistical problems. *J. ACM* 8(2):212–229.

Jamieson KG, Nowak RD, Recht B (2012) Query complexity of derivative-free optimization. Pereira F, Burges CJC, Bottou L, Weinberger KQ, eds. *Proc. 25th Internat. Conf. Neural Inform. Processing Systems* (Curran Associates, Red Hook, NY), 2672–2680.

Jaquier N, Rozo L (2020) High-dimensional Bayesian optimization via nested Riemannian manifolds. Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, eds. *Advances in Neural Information Processing Systems*, vol. 33 (Curran Associates, Red Hook, NY), 20939–20951.

Jaquier N, Rozo L, Calinon S, Bürger M (2020) Bayesian optimization meets Riemannian manifolds in robot learning. Kaelbling LP, Kragic D, Sugiura K, eds. *Proc. Conf. Robot Learn.*, vol. 100 (PMLR), 233–246.

Kass RE, Carlin BP, Gelman A, Neal RM (1998) Markov chain Monte Carlo in practice: A roundtable discussion. *Amer. Statist.* 52(2):93–100.

Kiefer J, Wolfowitz J (1952) Stochastic estimation of the maximum of a regression function. *Ann. Math. Statist.* 23(3):462–466.

Kolda TG, Lewis RM, Torczon V (2003) Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Rev.* 45(3):385–482.

Krizhevsky A (2009) Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, Toronto.

Lam R, Willcox K (2017) Lookahead Bayesian optimization with inequality constraints. Von Luxburg U, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, eds. *Advances in Neural Information Processing Systems*, vol. 30 (Curran Associates, Red Hook, NY), 1890–1900.

Larson J, Menickelly M, Wild SM (2019) Derivative-free optimization methods. *Acta Numer.* 28(May):287–404.

Latré B, Braem B, Moerman I, Blondia C, Demeester P (2011) A survey on wireless body area networks. *Wireless Networks* 17(1):1–18.

LeCun Y, Cortes C (2010) MNIST handwritten digit database. Accessed August 1, 2022, http://yann.lecun.com/exdb/mnist/.

Le Digabel S, Wild SM (2015) A taxonomy of constraints in simulation-based optimization. Preprint, submitted May 28, https://doi.org/10.48550/arXiv.1505.07881.

Leimkuhler B, Matthews C (2015) *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods* (Springer, Cham, Switzerland).

Lenth RV (2001) Some practical guidelines for effective sample size determination. *Amer. Statist.* 55(3):187–193.

Letham B, Karrer B, Ottoni G, Bakshy E (2019) Constrained Bayesian optimization with noisy experiments. *Bayesian Anal.* 14(2):495–519.

Lewis RM, Torczon V (2002) A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM J. Optim.* 12(4):1075–1089.

Li J, Balasubramanian K, Ma S (2022) Stochastic zeroth-order Riemannian derivative estimation and optimization. *Math. Oper. Res.*, epub ahead of print September 8, https://doi.org/10.1287/moor.2022.1302.

Li L, Jamieson K, DeSalvo G, Rostamizadeh A, Talwalkar A (2017) Hyperband: A novel bandit-based approach to hyperparameter optimization. *J. Machine Learn. Res.* 18(1):6765–6816.

Liu S, Chen P-Y, Kailkhura B, Zhang G, Hero AO III, Varshney PK (2020) A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine* 37(5):43–54.

Liuzzi G, Lucidi S (2009) A derivative-free algorithm for inequality constrained nonlinear programming via smoothing of an ℓ_∞ penalty function. *SIAM J. Optim.* 20(1):1–29.

Liuzzi G, Lucidi S, Sciandrone M (2010) Sequential penalty derivative-free methods for nonlinear constrained optimization. *SIAM J. Optim.* 20(5):2614–2635.

Mahendran N, Wang Z, Hamze F, De Freitas N (2012) Adaptive MCMC with Bayesian optimization. Teh YW, Titterington M, eds. *Proc. 13th Conf. Artificial Intelligence Statist*. (PMLR), 751–760.

Mania H, Guy A, Recht B (2018) Simple random search provides a competitive approach to reinforcement learning. Preprint, submitted March 19, https://doi.org/10.48550/arXiv.1803.07055.

Mockus J (1994) Application of Bayesian approach to numerical methods of global and stochastic optimization. *J. Global Optim.* 4(4):347–365.

Mockus J (2012) *Bayesian Approach to Global Optimization: Theory and Applications* (Springer Science & Business Media, New York).

Mokkadem A, Pelletier M (2007) A companion for the Kiefer–Wolfowitz–Blum stochastic approximation algorithm. *Ann. Statist.* 35(4):1749–1772.

Müller J, Woodbury JD (2017) GOSAC: Global optimization with surrogate approximation of constraints. *J. Global Optim.* 69(1):117–136.

Neal RM (2011) MCMC using Hamiltonian dynamics. Brooks S, Gelman A, Jones G, Meng X-L, eds. *Handbook of Markov Chain Monte Carlo* (CRC Press, Boca Raton, FL), 113–162.

Nelder JA, Mead R (1965) A simplex method for function minimization. *Comput. J.* 7(4):308–313.

Nemirovski AS, Yudin DB (1983) *Problem Complexity and Method Efficiency in Optimization* (John Wiley & Sons, Chichester, UK).

Nesterov Y, Spokoiny V (2017) Random gradient-free minimization of convex functions. *Foundations Comput. Math.* 17(2):527–566.

Parikh N, Boyd S (2014) Proximal algorithms. *Foundations Trends Optim.* 1(3):127–239.

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, et al. (2019) Pytorch: An imperative style, high-performance deep learning library. Wallach HM, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox EA, Garnett R, eds. *Advances in Neural Information Processing Systems*, vol. 32 (Curran Associates, Red Hook, NY), 8024–8035.

Patil A, Huard D, Fonnesbeck CJ (2010) PyMC: Bayesian stochastic modelling in Python. *J. Statist. Software* 35(4):1–81.

Perera C, Liu CH, Jayawardena S (2015) The emerging internet of things marketplace from an industrial perspective: A survey. *IEEE Trans. Emerging Topics Comput.* 3(4):585–598.

Picheny V, Gramacy RB, Wild S, Le Digabel S (2016) Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian. Lee DD, Sugiyama M, von Luxburg U, Guyon I, Garnett R, eds. *Advances in Neural Information Processing Systems*, vol. 29 (Curran Associates, Red Hook, NY), 1435–1443.

Plummer M, Best N, Cowles K, Vines K (2006) CODA: Convergence diagnosis and output analysis for MCMC. *R News* 6(1):7–11.

Pourmohamad T, Lee H (2020) The statistical filter approach to constrained optimization. *Technometrics* 62(3):303–312.

Powell MJD (1964) An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J.* 7(2):155–162.

Rockafellar RT (2015) *Convex Analysis* (Princeton University Press, Princeton, NJ).

Ruan Y, Xiong Y, Reddi S, Kumar S, Hsieh C-J (2020) Learning to learn by zeroth-order oracle. *Internat. Conf. Learn. Representations*.

Ruszczynski A (2021) A stochastic subgradient method for nonsmooth nonconvex multilevel composition optimization. *SIAM J. Control Optim.* 59(3):2301–2320.

Sahu AK, Zaheer M, Kar S (2019) Toward gradient free and projection free stochastic optimization. Chaudhuri K, Sugiyama M, eds. *22nd Internat. Conf. Artificial Intelligence Statist.* (PMLR), 3468–3477.

Salimans T, Ho J, Chen X, Sidor S, Sutskever I (2017) Evolution strategies as a scalable alternative to reinforcement learning. Preprint, submitted March 10, https://doi.org/10.48550/arXiv.1703.03864.

Shahriari B, Swersky K, Wang Z, Adams RP, De Freitas N (2015) Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* 104(1):148–175.

Shamir O (2013) On the complexity of bandit and derivative-free stochastic convex optimization. Shalev-Shwartz S, Steinwart I, eds. *26th Annual Conf. Learn. Theory Proc.* (PMLR), 3–24.

Snoek J, Larochelle H, Adams RP (2012) Practical Bayesian optimization of machine learning algorithms. Pereira F, Burges CJ, Bottou L, Weinberger KQ, eds. *Advances in Neural Information Processing Systems*, vol. 25 (Curran Associates, Red Hook, NY), 2951–2959.

Spall JC (1987) A stochastic approximation technique for generating maximum likelihood parameter estimates. *Proc.* 1987 *Amer. Control Conf.* (IEEE, Piscataway, NJ), 1161–1167.

Spall JC (2005) *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control* (John Wiley & Sons, Hoboken, NJ).

Spendley W, Hext GR, Himsworth FR (1962) Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics* 4(4):441–461.

Stein C (1972) A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. Le Cam LM, Neyman J, Scott EL, eds. *Proc. Sixth Berkeley Sympos. Math. Statist. Probab.*, vol. 2: Probab. Theory (Regents of the University of California–Berkeley, Berkeley, CA), 583–602.

Sutskever I, Martens J, Dahl G, Hinton G (2013) On the importance of initialization and momentum in deep learning. Dasgupta S, McAllester D, eds. *Proc. 30th Internat. Conf. Machine Learning* (PMLR), 1139–1147.

Tröltzsch A (2016) A sequential quadratic programming algorithm for equality-constrained optimization without derivatives. *Optim. Lett.* 10(2):383–399.

Usmanova I, Krause A, Kamgarpour M (2019) Safe convex learning under uncertain constraints. *Proc. 22nd Internat. Conf. Artificial Intelligence Statist.* (PMLR), 2106–2114.

Wang Z, Mohamed S, Freitas N (2013) Adaptive Hamiltonian and Riemann manifold Monte Carlo. Dasgupta S, McAllester D, eds. *Proc. 30th Internat. Conf. Machine Learn.*, vol. 89 (PMLR), 1462–1470.

Yang AY, Iyengar S, Sastry S, Bajcsy R, Kuryloski P, Jafari R (2008) Distributed segmentation and classification of human actions using a wearable motion sensor network. *2008 IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognition Workshops* (IEEE, Piscataway, NJ), 1–8.