# Zeroth-Order Optimization for Composite Problems with Functional Constraints

**Zichong Li[1], Pin-Yu Chen[2], Sijia Liu[3], Songtao Lu[2], Yangyang Xu[1]**

[1]Rensselaer Polytechnic Institute, {liz19, xuy21}@rpi.edu
[2]IBM Research, {Pin-Yu.Chen, songtao}@ibm.com
[3]Michigan State University, liusiji5@msu.edu

## Abstract

In many real-world problems, first-order (FO) derivative evaluations are too expensive or even inaccessible. For solving these problems, zeroth-order (ZO) methods that only need function evaluations are often more efficient than FO methods or sometimes the only options. In this paper, we propose a novel zeroth-order inexact augmented Lagrangian method (ZO-iALM) to solve black-box optimization problems, which involve a composite (i.e., smooth+nonsmooth) objective and functional constraints. This appears to be the first work that develops an iALM-based ZO method for functional constrained optimization and meanwhile achieves query complexity results matching the best-known FO complexity results up to a factor of variable dimension.With an extensive experimental study, we show the effectiveness of our method. The applications of our method span from classical optimization problems to practical machine learning examples such as resource allocation in sensor networks and adversarial example generation.

## Introduction

In many practical optimization problems such as black-box attack (Chen et al. 2017), we only have access to zeroth-order (ZO) function values but no access to first-order (FO) or higher order derivatives (Liu et al. 2020a). In this paper, we consider *nonconvex* problems with *possibly nonconvex* constraints:

$$f_0^* := \min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f_0(\mathbf{x}) := g(\mathbf{x}) + h(\mathbf{x}),\ \text{s.t. } \mathbf{c}(\mathbf{x}) = \mathbf{0} \right\}, \quad (1)$$

where $g$ is smooth but possibly nonconvex, $\mathbf{c} = (c_1, \ldots, c_l) : \mathbb{R}^d \to \mathbb{R}^l$ is a vector function with continuously differentiable components, and $h$ is closed convex but possibly nonsmooth and has a coordinate structure, i.e., $h(\mathbf{x}) = \sum_{i=1}^d h(x_i)$. This formulation follows from (Li et al. 2021), except that in this paper, only the function evaluations of $g$ and $\mathbf{c}$, but not their gradients, are accessible. Such a formulation includes a large class of nonlinear constrained problems. We remark that an inequality constraint $t(\mathbf{x}) \leq 0$ can be equivalently formulated as an equality constraint $t(\mathbf{x})+s = 0$ by enforcing the nonnegativity of $s$, with equivalent stationarity conditions (c.f. (Li et al. 2021)). Note in (1), the inclusion of a coordinate-separable constraint set $\mathcal{X}$ is equivalent to the addition of $I_{\mathcal{X}}$ to the nonsmooth term $h$ in the objective $f_0$, where $I_{\mathcal{X}}(\mathbf{x}) = 0$ if $\mathbf{x} \in \mathcal{X}$ and $+\infty$ otherwise.

## Contributions

Our contributions are three-fold. First, we design a zeroth-order accelerated proximal coordinate update (ZO-APCU) method for solving coordinate-structured strongly-convex composite (i.e., smooth+nonsmooth) problems. ZO-APCU appears to be the first PCU method *with acceleration* by just using function values. It can be viewed as a ZO variant of the APCG in (Lin, Lu, and Xiao 2014). To solve black-box optimization in the form of (1), we propose a novel zeroth-order inexact augmented Lagrangian method (ZO-iALM), by using ZO-APCU to design a zeroth-order inexact proximal point method (ZO-iPPM) to approximately solve each ALM subproblem. Though any ZO method can be used as a subroutine in the iALM, the use of ZO-iPPM with the developed ZO-APCU is crucial to yield best-known query complexity results and also good numerical performance, as we demonstrate in the experiments.

Second, query complexity analysis is conducted on the proposed methods. We show that ZO-APCU needs $O(d\sqrt{\kappa}\log\frac{1}{\varepsilon})$ queries to produce an $\varepsilon$-stationary point of a $d$-dimensional strongly-convex composite problem with a condition number $\kappa$. The ZO-iPPM has an $\tilde{O}(d\varepsilon^{-2})$ complexity to give an $\varepsilon$-stationary point of a nonconvex composite problem. On solving (1) that satisfies a regularity condition, the ZO-iALM has an $\tilde{O}(d\varepsilon^{-3})$ overall complexity to produce an $\varepsilon$-KKT point, and the complexity can be reduced to $\tilde{O}(d\varepsilon^{-\frac{5}{2}})$ if the constraints are affine. All our complexity results are (near) optimal or the best known. To the best of our knowledge, complexity of ZO methods on nonconvex functional constrained problem (1) has not been studied in the literature, thus our $\tilde{O}(d\varepsilon^{-3})$ result is completely new.

Third, we use a coordinate gradient estimator while implementing the core solver ZO-APCU. To be able to yield high-accuracy solutions, we give a multi-point coordinate-wise gradient estimator and analyze its error bound. Under the $j$-th order smoothness assumption for some $j \in \mathbb{Z}^+$, we show that the error of a $\max\{2, 2(j-1)\}$-point coordinate-wise gradient estimator is upper bounded by $O(a^j)$, where $a$ is the sampling radius. This result is meaningful and important to yield high accuracy, because in practice $a$ cannot be too small, otherwise round-off errors will dominate.

Overall, we conduct a comprehensive study on ZO methods on solving nonconvex functional constrained black-box optimization, from multiple perspectives including complexity analysis, gradient estimator, and significantly improved performance on practical machine learning tasks and classical optimization problems.

## Related Works

In this subsection, we review previous works on the inexact augmented Lagrangian methods (iALMs) in the usual FO settings and the zeroth-order methods (ZOMs).

The iALM is one of the most common methods for solving constrained problems. It alternatingly updates the primal variable by inexactly minimizing the augmented Lagrangian (AL) function and the Lagrangian multiplier by dual gradient ascent. If the multiplier is fixed to zero, then iALM reduces to a standard penalty method, which usually has a worse practical performance than iALM. Previous works on iALM assume that FO derivatives of the objective function can be evaluated, therefore use FOMs to inexactly minimize the AL function in each primal update.

For convex nonlinear constrained problems, the iALM in (Li and Xu 2021) and the proximal-iALM in (Li and Qu 2021) can produce an $\varepsilon$-KKT point with $O(\varepsilon^{-1}|\log\varepsilon|)$ gradient evaluations, and the AL-based FOMs in (Xu 2021b,a; Ouyang et al. 2015; Li and Qu 2021; Nedelcu, Necoara, and Tran-Dinh 2014) can produce an $\varepsilon$-optimal solution with $O(\varepsilon^{-1})$ complexity. For strongly-convex problems, the complexity results can be respectively reduced to $O(\varepsilon^{-\frac{1}{2}}|\log\varepsilon|)$ for an $\varepsilon$-KKT point and $O(\varepsilon^{-\frac{1}{2}})$ for an $\varepsilon$-optimal solution, e.g., (Li and Xu 2021; Li and Qu 2021; Xu 2021b; Nedelcu, Necoara, and Tran-Dinh 2014; Necoara and Nedelcu 2014).

For nonconvex problems with nonlinear convex constraints, when Slater's condition holds, $\tilde{O}(\varepsilon^{-\frac{5}{2}})$ complexity results have been obtained by the AL or quadratic-penalty based FOMs in (Li and Xu 2021; Lin, Ma, and Xu 2019) and the proximal ALM in (Melo, Monteiro, and Wang 2020a). When a regularity condition (see Assumption 5 below) holds, the ALM in (Li et al. 2021) achieves $\tilde{O}(\varepsilon^{-\frac{5}{2}})$ complexity for nonconvex problems with nonlinear convex constraints and $\tilde{O}(\varepsilon^{-3})$ complexity for problems with nonconvex constraints.

When gradients of the objective function are unavailable, ZOMs are the only tools available. Previous ZO works mainly focus on problems *without* nonlinear functional constraints. Many existing ZOMs are modified from some gradient descent type FOMs, replacing the exact gradient $\nabla f(\mathbf{x})$ by some gradient estimator $\tilde{\nabla} f(\mathbf{x})$. In the next section, we briefly review some existing gradient estimation frameworks including random search and finite difference. A more detailed overview can be found in (Liu et al. 2020a) for ZOMs.

## Notations, Definitions, and Assumptions

We use $\|\cdot\|$ for the Euclidean norm of a vector and the spectral norm of a matrix. $[n]$ denotes the set $\{1,\ldots,n\}$. We use $\tilde{O}$ to suppress all log terms of $\varepsilon$ from the big-$O$ notation. We denote $J_c(\mathbf{x})$ as the Jacobian of $\mathbf{c}$ at $\mathbf{x}$. The distance between a vector $\mathbf{x}$ and a set $\mathcal{X}$ is denoted as $\operatorname{dist}(\mathbf{x},\mathcal{X}) = \min_{\mathbf{y}\in\mathcal{X}}\|\mathbf{x}-\mathbf{y}\|$. For a function $f$, we use $\partial f$ to denote the subdifferetial of $f$. For a differentiable function $f$, we use $\tilde{\nabla} f$ as an estimator of the gradient $\nabla f$. The AL function of (1) is

$$\mathcal{L}_\beta(\mathbf{x},\mathbf{y}) = f_0(\mathbf{x}) + \mathbf{y}^\top \mathbf{c}(\mathbf{x}) + \tfrac{\beta}{2}\|\mathbf{c}(\mathbf{x})\|^2, \qquad (2)$$

where $\beta > 0$ is the penalty parameter, and $\mathbf{y}\in\mathbb{R}^l$ is the multiplier or the dual variable.

**Definition 1 ($\varepsilon$-KKT point)** *A point $\mathbf{x}\in\mathbb{R}^d$ is an $\varepsilon$-KKT point of* (1) *if there is $\mathbf{y}\in\mathbb{R}^l$ such that*

$$\|\mathbf{c}(\mathbf{x})\| \le \varepsilon, \quad \operatorname{dist}\left(\mathbf{0}, \partial f_0(\mathbf{x}) + J_c^\top(\mathbf{x})\,\mathbf{y}\right) \le \varepsilon. \qquad (3)$$

**Definition 2 ($k$-smoothness)** *For some $k \ge 1$, we say $f$ is $M_k$ $k$-smooth, if the $k$-th derivative of $f$ is $M_k$ Lipschitz continuous.*

**Remark 1** *Letting $k = 1$ above corresponds to the standard smoothness assumption.*

**Definition 3 (coordinate $k$-smooth)** *For some $k \ge 1$, we say $f$ is $M_k$ coordinate $k$-smooth, if the partial function $F_i(\mathbf{x}_i) := f(\mathbf{x}_{<i}, \mathbf{x}_i, \mathbf{x}_{>i})$ is $M_k$ $k$-smooth, $\forall\, i\in[d]$, where $\mathbf{x}_{<i} := (x_1,\ldots,x_{i-1})$ and $\mathbf{x}_{>i} := (x_{i+1},\ldots,x_d)$ are fixed.*

**Remark 2** *If $f$ is $M_k$ $k$-smooth, then it must be $M_k^c$ coordinate $k$-smooth with some $M_k^c \le M_k$.*

**Definition 4 ($\rho$-weakly convex)** *A function $f$ is $\rho$-weakly convex if $f + \frac{\rho}{2}\|\cdot\|^2$ is convex.*

**Remark 3** *A function that is $L$-smooth is also $L$-weakly convex. However, its weak convexity constant can be much smaller than its smoothness constant.*

Throughout this paper, we make the following assumptions.

**Assumption 1 (smoothness and weak convexity)** *In* (1)*, $g$ is $L_0$-smooth and $\rho_0$-weakly convex. For each $j\in[l]$, $c_j$ is $L_j$-smooth and $\rho_j$-weakly convex.*

**Assumption 2 (bounded domain)** *In* (1)*, $h$ is closed convex with a compact domain, i.e.,*

$$D := \max_{\mathbf{x},\mathbf{x}'\in\operatorname{dom}(h)} \|\mathbf{x}-\mathbf{x}'\| < \infty, \qquad (4a)$$

$$D_i := \max_{\substack{\mathbf{x},\mathbf{x}'\in\operatorname{dom}(h)\\ \mathbf{x}_{[d]\setminus i}=\mathbf{x}'_{[d]\setminus i}}} \|\mathbf{x}-\mathbf{x}'\| \le D, \forall i\in[d], \qquad (4b)$$

*where $\mathbf{x}_{[d]\setminus i} = (\mathbf{x}_1,\ldots,\mathbf{x}_{i-1},\mathbf{x}_{i+1},\ldots,\mathbf{x}_d)$.*

Due to the page limit, all proofs in this paper are given in the supplementary material.

# Multi-point Gradient Estimator

In this section, we provide backgrounds on gradient estimators and propose the zeroth-order multi-point coordinate gradient estimator.

## Backgrounds on Gradient Estimators

Let $a$ denote the *sampling radius* (also called the *smoothing parameter*) of a random gradient estimator, and $\mathbf{u} \sim p$ denote some random direction sampled from a distribution $p$. Denote $f_a$ as the *smoothed* version of $f$ defined as $f_a(\mathbf{x}) := \mathbb{E}_{\mathbf{u}\sim p'}[f(\mathbf{x} + a\mathbf{u})]$, where $p'$ is a certain distribution determined by $p$. All random gradients in this subsection are biased with respect to $\nabla f$ but unbiased with respect to $\nabla f_a$, satisfying $\mathbb{E}_{\mathbf{u}\sim p}[\tilde{\nabla} f(\mathbf{x})] = \nabla f_a(\mathbf{x})$,

The 1-*point random gradient estimator* of $f$ has the form

$$\tilde{\nabla} f(\mathbf{x}) := \tfrac{\phi(d)}{a} f(\mathbf{x} + a\mathbf{u})\mathbf{u}, \tag{5}$$

where $\phi(d)$ is a dimension-dependent factor given by the distribution of $\mathbf{u}$. If $p = \mathcal{N}(\mathbf{0}, \mathbf{I})$, then $\phi(d) = 1$; if $p = \mathcal{U}(\mathcal{S}(\mathbf{0}, \mathbf{I}))$ is the uniform distribution on the unit sphere, then $\phi(d) = d$. In practice, the 1-point estimator in (5) is not commonly used due to high variance (Flaxman, Kalai, and McMahan 2004). This motivates the 2-*point random gradient estimator* (Nesterov and Spokoiny 2017; Duchi et al. 2015)

$$\tilde{\nabla} f(\mathbf{x}) := \tfrac{\phi(d)}{a}(f(\mathbf{x} + a\mathbf{u}) - f(\mathbf{x}))\mathbf{u}, \tag{6}$$

where $\mathbb{E}_{\mathbf{u}\sim p}[\mathbf{u}] = \mathbf{0}$ is required for unbiasedness to hold. The 2-point estimator has the following upper bound of expected estimation error (Berahas et al. 2021; Liu et al. 2018)

$$\mathbb{E}[\|\tilde{\nabla} f(\mathbf{x}) - \nabla f(\mathbf{x})\|] = O(\sqrt{d})\|\nabla f(\mathbf{x})\| + O\left(\tfrac{ad^{1.5}}{\phi(d)}\right). \tag{7}$$

Note that the $O(\sqrt{d})\|\nabla f(\mathbf{x})\|$ term in (7) does not vanish even if $a \to 0$. Mini-batch sampling can be used to reduce the estimation error, leading to the *multi-point random gradient estimator* (Duchi et al. 2015; Liu et al. 2018)

$$\tilde{\nabla} f(\mathbf{x}) := \frac{\phi(d)}{a} \sum_{i=1}^{b} (f(\mathbf{x} + a\mathbf{u}_i) - f(\mathbf{x}))\mathbf{u}_i, \tag{8}$$

where $b$ is the mini-batch size, and $\{\mathbf{u}_i\}_{i=1}^{b}$ are random directions drawn from some zero-mean distribution $p$. The multi-point estimator has the improved error bound (Berahas et al. 2021)

$$\begin{aligned} &\mathbb{E}[\|\tilde{\nabla} f(\mathbf{x}) - \nabla f(\mathbf{x})\|] \\ &= O\left(\sqrt{\frac{d}{b}}\right)\|\nabla f(\mathbf{x})\| + O\left(\frac{ad^{1.5}}{\phi(d)b}\right) + O\left(\frac{ad^{0.5}}{\phi(d)}\right). \end{aligned}$$

To further reduce the estimation error, one can use *coordinate-wise gradient* which requires $O(d)$ queries per gradient estimate. Existing works use *forward difference* $\tilde{\nabla} f(\mathbf{x}) := \frac{1}{a}\sum_{i=1}^{d}(f(\mathbf{x} + a\mathbf{e}_i) - f(\mathbf{x}))\mathbf{e}_i$ or *central difference* $\tilde{\nabla} f(\mathbf{x}) := \frac{1}{2a}\sum_{i=1}^{d}(f(\mathbf{x} + a\mathbf{e}_i) - f(\mathbf{x} - a\mathbf{e}_i))\mathbf{e}_i$ as the coordinate gradient, where $\mathbf{e}_i$ is the $i$th basis vector. Under the standard smoothness assumption, both forward difference and central difference have the error bounds (Kiefer, Wolfowitz et al. 1952; Berahas et al. 2021; Lian et al. 2016)

$$\begin{aligned} \mathbb{E}[|\tilde{\nabla}_i f(\mathbf{x}) - \nabla_i f(\mathbf{x})|] &= O(a), \\ \mathbb{E}[\|\tilde{\nabla} f(\mathbf{x}) - \nabla f(\mathbf{x})\|] &= O\left(a\sqrt{d}\right). \end{aligned}$$

## Zeroth-order Multi-point Coordinate Gradient Estimator

In this subsection, assuming $f$ to be coordinate $p$-smooth, we construct the zeroth-order multi-point coordinate gradient estimator (ZO-MCGE) $\tilde{\nabla}_i f(\mathbf{x})$ with $p = \max\{2(j-1), 2\}$ function value queries at $\mathbf{x} + \frac{p}{2}a\mathbf{e}_i, \ldots, \mathbf{x} + a\mathbf{e}_i, \mathbf{x} - a\mathbf{e}_i, \ldots, \mathbf{x} - \frac{p}{2}a\mathbf{e}_i$, and analyze its error bound. The main difference between our proposed ZO-MCGE and the estimators in the previous subsection is that the use of multi-point function evaluation allows for a better control for the gradient estimation error. We observe numerically that using more points in the gradient estimator enables us to reach a higher accuracy; see the logistic regression experiment in the Appendix.

The following lemma directly follows from the coordinate $j$-smoothness of $f$.

**Lemma 1** *Assume $f$ is $M_j$ coordinate $j$-smooth. Let $\nabla_i^l f(\mathbf{x}) := \frac{\partial^l f(\mathbf{x})}{(\partial x_i)^l}$ be the $l$-th order derivative of $f$ at $\mathbf{x}$ with respect to $x_i$. Then*

$$\begin{aligned} &\left| f(\mathbf{x} + b\mathbf{e}_i) - f(x) - b\nabla_i f(\mathbf{x}) - \cdots - \frac{b^j}{j!}\nabla_i^j f(\mathbf{x}) \right| \\ &\le \frac{M_j}{(j+1)!}|b|^{j+1}, \forall\, \mathbf{x} \in \mathbb{R}^d, \text{ and } b \in \mathbb{R}. \end{aligned} \tag{9}$$

Let $a$ be the sampling radius. The following theorem states how to estimate the coordinate gradient $\nabla_i f(\cdot)$ of a $M_j$ coordinate $j$-smooth function $f$ by $p = \max\{2(j-1), 2\}$ queries at $\mathbf{x} + \frac{p}{2}a\mathbf{e}_i, \ldots, \mathbf{x} + a\mathbf{e}_i, \mathbf{x} - a\mathbf{e}_i, \ldots, \mathbf{x} - \frac{p}{2}a\mathbf{e}_i$, and provides the error bound.

**Theorem 1 (multi-point coordinate gradient estimator)** *Assume $f$ is $M_j$ coordinate $j$-smooth for some $j \in \mathbb{Z}^+$. Let $p = \max\{2(j-1), 2\}$ and $m = \frac{p}{2}$. Define the $p$-point coordinate gradient estimator of $f$ with respect to some $i \in [d]$ as*

$$\begin{aligned} \tilde{\nabla}_i f(\mathbf{x}) := {}& C_{\frac{p}{2}} f(\mathbf{x} + \frac{p}{2}ae_i) + \cdots + C_1 f(\mathbf{x} + ae_i) \\ & - C_1 f(\mathbf{x} - ae_i) - \cdots - C_{\frac{p}{2}} f(\mathbf{x} - \frac{p}{2}ae_i), \end{aligned} \tag{10}$$

*where*

$$\begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_{\frac{p}{2}} \end{bmatrix} = \begin{bmatrix} 1 & 2 & \cdots & \frac{p}{2} \\ 1 & 2^3 & \cdots & (\frac{p}{2})^3 \\ & & \vdots & \\ 1 & 2^{p-1} & \cdots & (\frac{p}{2})^{p-1} \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{2a} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

*Then we have the following error bound*

$$|\tilde{\nabla}_i f(\mathbf{x}) - \nabla_i f(\mathbf{x})| \le \sum_{q=1}^{m} |C_q| \frac{M_j q^{j+1}}{(j+1)!} a^{j+1}. \tag{11}$$

**Remark 4** *Theorem 1 implies that if $f$ is $M_1$ coordinate 1-smooth (which holds if $f$ is $M_1$-smooth in the standard notion) or $M_2$ coordinate 2-smooth, then the coordinate gradient estimator given in* (10) *corresponds to the central difference $\tilde{\nabla}_i f(\mathbf{x}) = \frac{1}{2a}(f(\mathbf{x} + ae_i) - f(\mathbf{x} - ae_i))$, with error bounds of $\frac{M_1}{2}a$ and $\frac{M_2}{6}a^2$ respectively, because $C_1 = \frac{1}{2a}$.*

**Algorithm 1:** Zeroth-order inexact augmented Lagrangian method (ZO-iALM) for (1)

```
Initialization: choose $\mathbf{x}^0 \in \mathrm{dom}(f_0), \mathbf{y}^0 = \mathbf{0}$,
    $\beta_0 > 0$ and $\sigma > 1$
for $k = 0, 1, \ldots,$ do
    Let $\beta_k = \beta_0 \sigma^k$, $\phi(\cdot) = \mathcal{L}_{\beta_k}(\cdot, y^k) - h(\cdot)$, and
        $$\hat{\rho}_k = \rho_0 + \bar{L}\|\mathbf{y}^k\| + \beta_k \rho_c, \quad \hat{L}_k = L_0 + \bar{L}\|\mathbf{y}^k\| + \beta_k L_c. \quad (13)$$
    $\mathbf{x}^{k+1} \leftarrow$ ZO-iPPM$(\phi, h, \mathbf{x}^k, \hat{\rho}_k, \hat{L}_k, \varepsilon)$
    Update $\mathbf{y}$ by
        $$\mathbf{y}^{k+1} = \mathbf{y}^k + w_k \mathbf{c}(\mathbf{x}^{k+1}). \quad (14)$$

subroutine ZO-iPPM$(\phi, \psi, \mathbf{x}^0, \rho, L_\phi, \varepsilon)$
    for $t = 0, 1, \ldots,$ do
        Let $G(\cdot) = \phi(\cdot) + \rho\|\cdot - \mathbf{x}^t\|^2$
        Obtain $\mathbf{x}^{t+1}$ by a ZOM such that
            $\mathrm{dist}(\mathbf{0}, \partial(G + \psi)(\mathbf{x}^{t+1})) \le \frac{\varepsilon}{4}$
        if $2\rho\|\mathbf{x}^{t+1} - \mathbf{x}^t\| \le \frac{\varepsilon}{2}$, then return $\mathbf{x}^{t+1}$.
```

*In general, we establish that under the $j$-th order smoothness assumption for some $j \in \mathbb{Z}^+$, the error of the $\max\{2, 2(j-1)\}$-point coordinate gradient estimator is upper bounded by $O(a^j)$, where $a$ is the sampling radius.*

## A Novel AL-based ZOM

In this section, we present a novel ZOM for solving (1) under the ALM framework, with each ALM subproblem approximately solved by an inexact proximal point method (iPPM).

The pseudocode of our AL-based ZOM for (1) is shown in Algorithm 1 that uses the following notations

$$B_0 = \max_{\mathbf{x} \in \mathrm{dom}(h)} \max\left\{|f_0(\mathbf{x})|, \|\nabla g(\mathbf{x})\|\right\},$$

$$B_c = \max_{\mathbf{x} \in \mathrm{dom}(h)} \|J_c(\mathbf{x})\|, \tag{12a}$$

$$B_i = \max_{\mathbf{x} \in \mathrm{dom}(h)} \max\left\{|c_i(\mathbf{x})|, \|\nabla c_i(\mathbf{x})\|\right\}, \forall\, i \in [l], \tag{12b}$$

$$\bar{B}_c = \sqrt{\textstyle\sum_{i=1}^l B_i^2}, \quad \bar{L} = \sqrt{\textstyle\sum_{i=1}^l L_i^2},$$

$$\rho_c = \sum_{i=1}^l B_i \rho_i, \quad L_c = \sum_{i=1}^l B_i L_i + B_i^2, \tag{12c}$$

where $\{\rho_i\}$ and $\{L_i\}$ are given in Assumption 1.

Notice that Algorithm 1 follows the standard framework of the ALM and uses ZO-iPPM to solve each ALM subproblem. In principle, one can use any ZOM as a subroutine to solve ALM subproblems, such as ZO-AdaMM (Chen et al. 2019) and ZO-proxSGD (Ghadimi, Lan, and Zhang 2016). However, the use of ZO-iPPM (together with our developed zeroth-order accelerated proximal coordinate update) not only leads to best known complexity results, but also gives better numerical performance, as we show in Section 11.

The proposed ZO-iALM is triple-looped. An algorithm with fewer loops would be preferable. However, we are not aware of any existing simpler ZOMs with the same theoretical guarantees as our method for solving functional constrained black-box optimization. An important future direction is to reduce the number of loops and achieve the same theoretical guarantees. Nevertheless, as we demonstrate in Section 11, our algorithm can be efficiently implemented without much difficulty. Specifically, to have a good practical performance, all parameters except the smoothness constant do not require much tuning at all, and most can be constant across different problems. Even with triple loops, the proposed ZO-iALM performs well numerically. Furthermore, some existing FOMs are also triple-looped and can perform better than double-looped FOMs; see (Li et al. 2021) for example.

The kernel problems that we solve within the iPPM are strongly-convex composite problems. Below, we design a zeroth-order accelerated proximal coordinate update (ZO-APCU) method.

### Core subsolver: ZO-APCU

In this subsection, we give our core ZO subsolver, called ZO-APCU, to obtain $\mathbf{x}^{t+1}$ in the ZO-iPPM subroutine. Though ZO-APCU will be used for solving subproblems of our proposed ZOM for (1), it has its own merit and appears to be the first proximal coordinate update method *with acceleration* by only using function values of the smooth part. It solves strongly-convex composite problems in the form of

$$\min_{\mathbf{x} \in \mathbb{R}^d} F(\mathbf{x}) := G(\mathbf{x}) + H(\mathbf{x}), \tag{15}$$

where $G$ is a *black-box* $\mu$-strongly convex and $L$-smooth function, and $H$ is a *white-box* closed convex function.

We make the following assumptions on $G$ and $H$.

**Assumption 3 (coordinate smooth)** *$G$ is $M_j$ coordinate $j$-smooth, for some $j \in \mathbb{Z}^+$.*

Note that if $G$ is $L$-smooth, Assumption 3 must hold for $j = 1$ and $M_1 = L$.

**Assumption 4** *The function $H$ is coordinate-separable, i.e., $H(\mathbf{x}) = \sum_{i=1}^d H_i(x_i)$, where each $H_i(\cdot)$ is convex.*

The pseudocode of ZO-APCU is shown in Algorithm 2, with its equivalent and efficient implementation (which avoids full-dimensional vector operations) given in the Appendix. The design is inspired from the APCG method in (Lin, Lu, and Xiao 2014). A zeroth-order accelerated random search (ZO-ARS) method has been designed in (Nesterov and Spokoiny 2017) to solve (15). Although our ZO-APCU has the same-order query complexity as ZO-ARS, it significantly outperforms ZO-ARS in practice, because ZO-APCU exploits the coordinate-structure and uses more accurate coordinate gradient estimator.

In Algorithm 2, to obtain the required (coordinate) gradient estimates, we use the $p$-point coordinate gradient estimator defined in (10), where $p = \max\{2(j-1), 2\}$. Let

$$E_i = \sum_{q=1}^m |C_q| \frac{M_j q^{j+1}}{(j+1)!} a^{j+1}, \forall i \in [d]; \; E = \sqrt{\sum_{i=1}^d E_i^2}, \tag{16}$$

**Algorithm 2:** Zeroth-order accelerated proximal coordinate update for (15): ZO-APCU$(G, H, \mu, L, \varepsilon)$

```
Input: $\mathbf{x}^0 \in \mathrm{dom}(H)$, tolerance $\varepsilon$, smoothness $L$, strong convexity $\mu$, and epoch length $l$.
Initialization: $\mathbf{z}^0 = \mathbf{x}^0, \alpha = \frac{1}{d}\sqrt{\frac{\mu}{L}}$
for $k = 0, 1, \dots$ do
    Let $y^k = \frac{\mathbf{x}^k + \alpha\mathbf{z}^k}{1+\alpha}$
    Sample $i_k \in [d]$ uniformly; compute $\tilde{\nabla}_{i_k} G(\mathbf{y}^k)$ such that $\|\tilde{\nabla}_{i_k} G(\mathbf{y}^k) - \nabla_{i_k} G(\mathbf{y}^k)\| \le E_{i_k}$.
    Compute $\mathbf{z}^{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^d}\{\frac{dL\alpha}{2}\|\mathbf{x} - (1-\alpha)\mathbf{z}^k - \alpha\mathbf{y}^k\|^2 + \langle\tilde{\nabla}_{i_k} G(\mathbf{y}^k), \mathbf{x}_{i_k} - \mathbf{y}^k_{i_k}\rangle + H_{i_k}(\mathbf{x}_{i_k})\}$.
    $\mathbf{x}^{k+1} = \mathbf{y}^k + d\alpha(\mathbf{z}^{k+1} - \mathbf{z}^k) + d\alpha^2(\mathbf{z}^k - \mathbf{y}^k)$.
    if $k + 1 \equiv 0 \pmod{l}$ then
        Compute $\tilde{\nabla} G(\mathbf{x}^{k+1})$ such that $\|\tilde{\nabla} G(\mathbf{x}^{k+1}) - \nabla G(\mathbf{x}^{k+1})\| \le E$
        $\hat{\mathbf{x}}^{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^d}\{\langle\tilde{\nabla} G(\mathbf{x}^{k+1}), \mathbf{x} - \mathbf{x}^{k+1}\rangle + \frac{L}{2}\|\mathbf{x} - \mathbf{x}^{k+1}\|^2 + H(\mathbf{x})\}$
        Return $\hat{\mathbf{x}}^{k+1}$ and stop if $\mathrm{dist}\big(\mathbf{0}, \tilde{\nabla} G(\hat{\mathbf{x}}^{k+1}) + \partial H(\hat{\mathbf{x}}^{k+1})\big) \le \frac{3\varepsilon}{4}$.
```

where $m = \frac{p}{2}$ and $a$ is the sampling radius. By Theorem 1, $E$ and $E_i$ are upper bounds of the gradient estimation errors for $\nabla G(\cdot)$ and $\nabla_i G(\cdot)$. Let $\bar{\varepsilon} = \frac{\mu}{512L}\varepsilon^2$. We choose $a > 0$ and $p$ such that the error bounds $E$ and $\{E_i\}_{i=1}^d$ in (16) satisfy

$$2L\sqrt{\frac{2ED}{\mu}} + E \le \frac{\varepsilon}{4}, \quad ED + \sum_{i=1}^{d} E_i D_i \le \frac{\bar{\varepsilon}}{2}. \qquad (17)$$

## Complexity Results

In this subsection, we establish the total query complexity result of Algorithm 1. We first show that the core subsolver ZO-APCU can produce $\mathbf{x}^{t+1}$ desired in the ZO-iPPM subroutine. The theorem below gives the complexity result of ZO-APCU to produce an expected $\varepsilon$-stationary point of (15). The proof is highly nontrivial and given in the appendix.

**Theorem 2** *Let* $\{\mathbf{x}^k\}, \{\hat{\mathbf{x}}^k\}$ *be generated from Algorithm 2. Suppose the gradient error bounds* $E$ *and* $\{E_i\}_{i=1}^d$ *satisfy* (17). *Let* $\bar{\varepsilon} = \frac{\mu}{512L^2}\varepsilon^2$. *Then* $T = \left\lceil d\sqrt{\frac{L}{\mu}}\log\frac{2(F(\mathbf{x}^0) - F^*) + \mu\|\mathbf{x}^0 - \mathbf{x}^*\|^2}{\bar{\varepsilon}}\right\rceil$ *iterations of ZO-APCU suffice to generate* $\hat{\mathbf{x}}^T$ *satisfying* $\mathbb{E}[\mathrm{dist}(\mathbf{0}, \partial F(\hat{\mathbf{x}}^T))] \le \varepsilon$.

Theorem 2 only guarantees that the output $\hat{\mathbf{x}}^T$ nearly satisfies the stationarity condition *in expectation*. In order to show the complexity results of Algorithm 1, we need, in Line 4 of ZO-iALM, the iterate $\mathbf{x}^{k+1}$ obtained from ZO-iPPM *deterministically* satisfies the near-stationarity condition of $\mathcal{L}_{\beta_k}(\cdot, y^k)$ so that we can uniformly bound the AL function at the generated iterates. For this technical reason, we will require the output from Algorithm 2 to satisfy the near-stationarity condition *deterministically* instead of in an expectation sense. Theorem 3 below serves as a bridge to convert deterministic iteration bound until expected convergence to expected iteration bound until deterministic convergence, by only sacrificing a log factor in the iteration bound. The result is not difficult to prove but is essential in our complexity analysis of ZO-iALM.

**Theorem 3 (expected complexity)** *For a sequence of nonnegative random numbers* $\{q_k\}_{k=1}^\infty$, *suppose* $\mathbb{E}[q_k] \le C\eta^k, \forall k \ge 1$ *for some* $\eta \in (0, 1)$ *and* $C > 0$. *Given* $\varepsilon > 0$, *define* $K(\varepsilon) = \min_{k\in\mathbb{Z}^+}\{k : q_k \le \varepsilon\}$. *Then* $\mathbb{E}[K(\varepsilon)] \le \frac{2-\eta}{1-\eta}\log\frac{C}{\varepsilon(1-\eta)^2} + 3 - \eta$.

By Theorem 2 (and its proof for linear convergence) and Theorem 3, we have the following expected iteration complexity result of Algorithm 2 until deterministic convergence.

**Corollary 1** *Under the same assumptions as Theorem 2, T iterations of ZO-APCU are enough to generate* $\hat{\mathbf{x}}^T$ *satisfying* $\mathrm{dist}(\mathbf{0}, \partial F(\hat{\mathbf{x}}^T)) \le \varepsilon$, *where* $\mathbb{E}[T] = \tilde{O}\left(d\sqrt{\frac{L}{\mu}}\right)$.

Relying on Corollary 1, the next theorem gives the complexity result of the subroutine ZO-iPPM applied on the nonconvex composite problem

$$\Phi^* = \min_{\mathbf{x}\in\mathbb{R}^d} \left\{\Phi(\mathbf{x}) := \phi(\mathbf{x}) + \psi(\mathbf{x})\right\}, \qquad (18)$$

where $\phi$ is a *black-box* $L_\phi$-smooth and $\rho$-weakly convex function, and $\psi$ is a *white-box* closed convex function.

**Theorem 4** *Suppose* $\Phi^*$ *in* (18) *is finite. Then the subroutine ZO-iPPM in Algorithm 1 must stop within T iterations, where* $T = \left\lceil\frac{32\rho}{\varepsilon^2}(\Phi(\mathbf{x}^0) - \Phi^*)\right\rceil$. *The output* $\mathbf{x}^T$ *must satisfy* $\mathrm{dist}(\mathbf{0}, \partial\Phi(\mathbf{x}^T)) \le \varepsilon$. *In addition, if* $\mathrm{dom}(\psi)$ *has diameter* $D_\psi < \infty$ *and ZO-APCU is applied to find each* $\mathbf{x}^{t+1}$ *in ZO-iPPM, then the expected total query complexity is* $\tilde{O}\left(\frac{d\sqrt{\rho L_\phi}}{\varepsilon^2}[\Phi(\mathbf{x}^0) - \Phi^*]\log\frac{D_\psi}{\varepsilon}\right)$.

Now we are ready to establish the query complexity of the proposed ZO-iALM. Due to the difficulty of the possibly nonconvex constraints, a certain regularity condition must be made in order to guarantee (near) feasibility in a polynomial time. Following (Li et al. 2021; Lin, Ma, and Xu 2019; Sahin et al. 2019) that study FOMs, we assume the following regularity condition on (1).

**Assumption 5 (regularity)** *There is some* $v > 0$ *such that for any* $k \ge 1$,

$$v\|\mathbf{c}(\mathbf{x}^k)\| \le \mathrm{dist}\left(-J_c(\mathbf{x}^k)^\top\mathbf{c}(\mathbf{x}^k), \frac{\partial h(\mathbf{x}^k)}{\beta_{k-1}}\right). \qquad (19)$$

**Remark 5** *Notice that we only need the existence of* $v$ *in Assumption 5 but do not need to know its value in our algorithm. The assumption ensures that a near-stationary point of the AL function is near feasible. In (Li et al. 2021), the regularity condition is proven to hold for all affine-equality constrained problems possibly with either an additional polyhedral or ball constraint set. Moreover, several nonconvex examples satisfying Assumption 5 are given in (Lin, Ma, and Xu 2019; Sahin et al. 2019).*

*With Assumption 5, we can simply solve a quadratic-penalty problem of* (1) *with a large enough penalty parameter,* *in order to find a near-KKT point of* (1)*. However, this approach is numerically much slower than the iALM framework in Algorithm 1; see the tests in (Li et al. 2021) for example.*

**Remark 6** *To solve the nonconvex constrained problem* (1)*, a few existing works about FOMs have made key assumptions different from Assumption 5. For example, the uniform Slater's condition was assumed in (Ma, Lin, and Yang 2020), and a strong MFCQ condition was assumed in (Boob, Deng, and Lan 2019). These assumptions are neither strictly stronger nor strictly weaker than Assumption 5.*

The theorem below gives the total query complexity of ZO-iALM with general dual step sizes.

**Theorem 5 (total complexity of ZO-iALM)** *Suppose that Assumptions 1, 2, and 5 hold. In Algorithm 1, for some fixed* $q \in \mathbb{Z}^+ \cup \{0\}$ *and* $M > 0$*, let* $w_k = \frac{M(k+1)^q}{\|\mathbf{c}(\mathbf{x}^{k+1})\|}, \forall k \geq 0$*. Then given* $\varepsilon > 0$*, Algorithm 1 can produce an* $\varepsilon$*-KKT solution of* (1) *with* $\tilde{O}(d\varepsilon^{-3})$ *queries to* $g$ *and* $\mathbf{c}$ *in expectation, by using Algorithm 2 to find each* $\mathbf{x}^{t+1}$ *in ZO-iPPM. In addition, if* $\mathbf{c}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}$*, then* $\tilde{O}(d\varepsilon^{-\frac{5}{2}})$ *queries in expectation are enough for Algorithm 1 to produce an* $\varepsilon$*-KKT solution of* (1)*.*

**Remark 7** *The results in Theorem 5 are novel. To the best of our knowledge, they are the first such results for ZOMs on solving functional constrained black-box optimization. The order-dependence on* $\varepsilon$ *matches with the best-known results for FOMs on solving nonconvex composite optimization with convex or nonconvex constraints, e.g., see (Lin, Ma, and Xu 2019; Melo, Monteiro, and Wang 2020b; Li et al. 2021). For the affine-constrained case, we conjecture that the* $\tilde{O}(d\varepsilon^{-\frac{5}{2}})$ *query complexity may be reduced to* $\tilde{O}(d\varepsilon^{-2})$ *if the nonsmooth term* $h$ *has some special structure.*

## Numerical Results

In this section, we conduct numerical experiments to demonstrate the performance of our proposed ZO-iALM. We consider the problem of resource allocation in sensor networks and the adversarial example generation problem. All the tests were performed in MATLAB 2019b on a Macbook Pro with 4 cores and 16GB memory. Due to the page limitation, we put additional numerical experiments in the Appendix. They are on nonconvex linearly-constrained quadratic programs (LCQP), on the unconstrained strongly-convex quadratic programs (USCQP) to test the core solver ZO-APCU, and on the logistic regression to test different multi-point coordinate gradient estimators. We emphasize here that the proposed ZO-APCU requires significantly fewer queries to reach a near-stationary point to the USCQP problem compared to a few existing methods, and that the use of more points in coordinate gradient estimator can lead to higher accuracy.

### Resource Allocation in Sensor Networks

In this subsection, we test our proposed ZO-iALM on the resource allocation problem in sensor networks (Liu et al. 2016). The problem aims at minimizing the estimation error of a random vector with a Gaussian prior probability density function, subject to a constraint on the total number of sensor activations. It can be formulated as

$$\begin{aligned}&\min_{\mathbf{w}\in\mathbb{R}^d} \ \mathrm{tr}(\Sigma^{-1} + \mathbf{H}^\top(\mathbf{w}\mathbf{w}^\top \circ \mathbf{R}^{-1})\mathbf{H})^{-1},\\ &\ \ \text{s.t. } \ \mathbf{1}^\top\mathbf{w} \leq s, \mathbf{w} \in \{0,1\}^d,\end{aligned} \tag{20}$$

where each $w_i \in \{0,1\}$ denotes whether the $i$th sensor is selected, $\mathbf{H} \in \mathbb{R}^{d\times d}$ is the observation matrix, $\Sigma \in \mathbb{R}^{d\times d}$ is the MSE source statistics, and $\mathbf{R} \in \mathbb{R}^{d\times d}$ is the noise covariance matrix. We assume that $\Sigma$ and $\mathbf{R}$ are symmetric, and $\mathbf{R}$ has small off-diagonal entries. Details of the formulation (20) can be found in (Liu et al. 2016).

ZO optimization methods have been applied in the literature to problem (20), in order to avoid the involved first-order gradient computation (Liu et al. 2018). The use of a ZO solver enables the design of resource management with least prior knowledge, e.g., without having access to the sensing model information encoded in $\mathbf{H}$. The constraint $\mathbf{w} \in \{0,1\}^d$ is combinatorial. Below, we rewrite the 0-1 constraint to $\mathbf{w}^2 - \mathbf{w} = \mathbf{0}$ and also incorporate the constraint $\mathbf{1}^\top\mathbf{w} \leq s$ into the objective by introducing a (fixed) multiplier $\lambda > 0$. More precisely, we apply our ZO-iALM to the problem:

$$\begin{aligned}&\min_{\mathbf{w}\in\mathbb{R}^d} \ \mathrm{tr}(\Sigma^{-1} + \mathbf{H}^\top(\mathbf{w}\mathbf{w}^\top \circ \mathbf{R}^{-1})\mathbf{H})^{-1} + \lambda\mathbf{1}^\top\mathbf{w},\\ &\ \ \text{s.t. } \ \mathbf{w}^2 - \mathbf{w} = \mathbf{0}.\end{aligned} \tag{21}$$

Since no existing ZOMs are able to handle nonconvex constrained problems, we compare the proposed ZO-iALM to two other methods that replace our ZO-iPPM subroutine with ZO-AdaMM (Chen et al. 2019) and ZO-ProxSGD (Ghadimi, Lan, and Zhang 2016) respectively.

We set $d = 80$, $\lambda = 0.5$, and $\varepsilon = 0.5$. Following (Liu et al. 2016), we construct $\mathbf{H} = \frac{1}{2}(\bar{\mathbf{H}} + \bar{\mathbf{H}}^\top)$ with each entry of $\bar{\mathbf{H}} \in \mathbb{R}^{d\times d}$ generated from the uniform distribution $\mathcal{U}(0,1)$, $\mathbf{R} = (\frac{1}{2}(\bar{\mathbf{R}}+\bar{\mathbf{R}}^\top))^{-1}$ with each entry of $\bar{\mathbf{R}} \in \mathbb{R}^{d\times d}$ generated from $\mathcal{U}(0, 10^{-3})$, and $\Sigma = \mathbf{I}$. In each call to the ZO-iPPM subroutine, we set the smoothness parameter to $\hat{L}_k = 50 + 0.3\beta_k$. We tune the parameters of ZO-AdaMM to $\alpha = 1, \beta_1 = 0.75, \beta_2 = 1$, and fix the step size to 0.01 in ZO-ProxSGD. For each method, we choose $a = 10^{-6}$ as the sampling radius and $w_k = \frac{1}{\|\mathbf{c}(\mathbf{x}^k)\|}$ as the dual step size.

In Figure 1, we compare the primal residual trajectories of the proposed ZO-iALM, and the iALM with subroutine ZO-AdaMM in (Chen et al. 2019) and ZO-ProxSGD in (Ghadimi, Lan, and Zhang 2016). The dual residuals by all compared methods are below the error tolerance $\varepsilon$ at the end of each outer loop. In Table 5 in the supplementary material, we also report the primal residual, dual residual, running time (in seconds), and the query count, shortened as `pres`, `dres`, `time`, and `#Obj`, for each method. From the results, we conclude that the proposed ZO-iALM with any of the three subroutines is able to reach an $\varepsilon$-KKT point to the resource allocation problem (21). Moreover, the proposed ZO-iPPM subroutine requires fewer queries than other compared ZOMs to find a specified-accurate stationary point to the nonconvex subproblems.

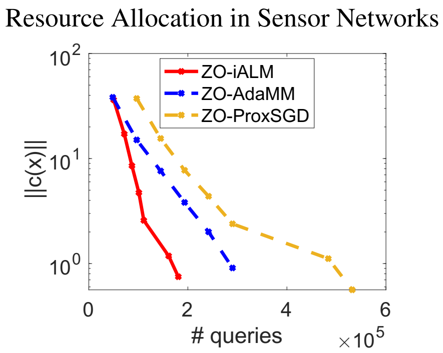

Figure 1: Comparison of iALM on solving (21) with different subroutines: the proposed ZO-iALM, ZO-AdaMM in (Chen et al. 2019), and ZO-ProxSGD in (Ghadimi, Lan, and Zhang 2016). The plots show primal residuals. The markers denote the outer iterations in iALM. Dual residuals for all methods are below the given tolerance $\varepsilon$.

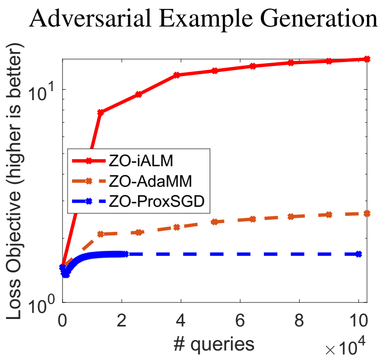

Figure 2: Comparison of iALM on solving (23) with different subroutines: the proposed ZO-iALM, ZO-AdaMM in (Chen et al. 2019), and ZO-ProxSGD in (Ghadimi, Lan, and Zhang 2016). The plots show the loss objective that we attack under the same $L_0$ and $L_\infty$ constraints.

## Adversarial Example Generation

The problem of adversarial example generation for a black-box regression model (Liu et al. 2020b) under both $L_0$ and $L_\infty$-norm constraints can be formulated as

$$\max_{\|\Delta\|_\infty \le \varepsilon_\infty} f_\theta(\mathbf{x}+\Delta), \text{ s.t. } \|\Delta\|_0 \le \varepsilon_0, \tag{22}$$

where $f_\theta(\cdot)$ is a loss function of a black-box regression model parameterized by $\theta$ that is trained over the dataset $\mathbf{x} = [\mathbf{x}_1^\top; \dots; \mathbf{x}_m^\top] \in \mathbb{R}^{m\times d}$, $\Delta \in \mathbb{R}^d$ is the data perturbation, and $\mathbf{x}+\Delta$ denotes adding $\Delta$ to each $\mathbf{x}_i$.

The constraint $\|\Delta\|_0 \le \varepsilon_0$ is combinatorial. To relax it to a continuous one, we introduce a binary vector $\hat{M}$ as a mask and put the constraint onto $\hat{M}$. More precisely, replace $\Delta$ in (22) by $\hat{M} \circ \Delta$, where $\circ$ denotes the Hadamard (component-wise) product. Then the constraint $\|\Delta\|_0 \le \varepsilon_0$ is relaxed to $\hat{M}_i \in \{0,1\}, \forall\, i$ and $\mathbf{1}^\top \hat{M} \le \varepsilon_0$. By further incorporating the constraint $\mathbf{1}^\top \hat{M} \le \varepsilon_0$ into the objective by introducing a (fixed) multiplier $-\lambda < 0$ and rewrite $\hat{M}_i \in \{0,1\}, \forall\, i$ into $\hat{M}^2 - \hat{M} = \mathbf{0}$, where $\hat{M}^2$ denotes the component-wise square of $\hat{M}$, we have the following reformulation:

$$\max_{\substack{\hat{M},\Delta\in\mathbb{R}^d \\ \|\Delta\|_\infty \le \varepsilon_\infty}} f_\theta(\mathbf{x}_0 + \hat{M}\circ\Delta) - \lambda \mathbf{1}^\top \hat{M}, \text{ s.t. } \hat{M}^2 - \hat{M} = \mathbf{0}. \tag{23}$$

We test the proposed ZO-iALM on the adversarial example generation problem (23). In the test, we use the ovarian cancer dataset (Conrads et al. 2004; Petricoin III et al. 2002) that are from $m = 216$ patients. Each data point has $d = 4,000$ features and a label indicating whether the corresponding patient has ovarian cancer. We first use MATLAB's built-in lasso function (with $\lambda = 0.01$) to train a LASSO regression model parameterized by $\theta$. With the trained model, we treat the regression loss $f_\theta(\cdot)$ as a ZO oracle and perform black-box attack on it. Let $\mathbf{x} \in \mathbb{R}^{m\times d}$ denote the data matrix. We then solve the ZO formulation (23) to find an adversarial perturbation $M \circ \Delta$ to each row of $\mathbf{x}$ that near-maximally increases the regression loss $f_\theta(\cdot)$. In (23), we set $\lambda = 0.01$ and $\varepsilon_\infty = 0.1$. Due to the large variable dimension, we set $\varepsilon = 1$ in stopping conditions.

The same as the previous test, we compare the proposed ZO-iALM to two other methods that replace our ZO-iPPM subroutine with ZO-AdaMM (Chen et al. 2019) and ZO-ProxSGD (Ghadimi, Lan, and Zhang 2016) respectively. In each method, we set $a = 10^{-6}$ as the sampling radius and $w_k = \frac{1}{\|\mathbf{c}(\mathbf{x}^k)\|}$ as the dual step size.

Let $(\hat{M}, \Delta)$ be one iterate obtained by one method on solving (23). Then $\tilde{\Delta} \leftarrow \hat{M} \circ \Delta$ is the data perturbation. To recover the solution to (22), we project $\tilde{\Delta}$ to the set $\{\Delta : \|\Delta\|_0 \le 20, \|\Delta\|_\infty \le 0.1\}$. In Figure 2, we plot the trajectory of the loss objective $f_\theta$ by all methods at the processed iterates of perturbed data. From the results, we see that the data perturbation created by the proposed ZO-iALM increases the loss function faster (namely, creates more successful attacks) than other compared methods.

## Conclusion

In this paper, we propose a novel zeroth-order inexact augmented Lagrangian method (ZO-iALM) to solve black-box optimization problems that involve a composite (i.e., smooth+nonsmooth) objective and nonlinear functional constraints. The kernel subproblems that we solve during the ZO-iALM are black-box strongly-convex composite problems with coordinate structure. To most efficiently solve these subproblems, we design a zeroth-order accelerated proximal coordinate update (ZO-APCU) method. In addition, in order to be able to produce high-accurate solutions, we give a new multi-point coordinate gradient estimator and use it in our designed ZO-APCU. All our proposed zeroth-order methods achieve similar-order complexity results as the best-known results obtained by first-order methods, with a difference up to a factor of variable dimension. Besides the novel and best theoretical results, our proposed ZO-iALM can also perform well numerically, which is demonstrated by experiments on practical machine learning tasks and classical optimization problems.

## Acknowledgements

This work was supported by the Rensselaer-IBM AI Research Collaboration (http://airc.rpi.edu), part of the IBM AI Horizons Network (http://ibm.biz/AIHorizons). The work of Y. Xu is partly supported by NSF Award 2053493 and the RPI-IBM AIRC faculty fund.

## References

Berahas, A. S.; Cao, L.; Choromanski, K.; and Scheinberg, K. 2021. A theoretical and empirical comparison of gradient approximations in derivative-free optimization. *Foundations of Computational Mathematics*, 1–54.

Boob, D.; Deng, Q.; and Lan, G. 2019. Stochastic first-order methods for convex and nonconvex functional constrained optimization. *arXiv preprint arXiv:1908.02734*.

Chen, P.-Y.; Zhang, H.; Sharma, Y.; Yi, J.; and Hsieh, C.-J. 2017. ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks Without Training Substitute Models. In *ACM Workshop on Artificial Intelligence and Security*, 15–26.

Chen, X.; Liu, S.; Xu, K.; Li, X.; Lin, X.; Hong, M.; and Cox, D. 2019. Zo-adamm: Zeroth-order adaptive momentum method for black-box optimization. *arXiv preprint arXiv:1910.06513*.

Conrads, T. P.; Fusaro, V. A.; Ross, S.; Johann, D.; Rajapakse, V.; Hitt, B. A.; Steinberg, S. M.; Kohn, E. C.; Fishman, D. A.; Whitely, G.; et al. 2004. High-resolution serum proteomic features for ovarian cancer detection. *Endocrine-related cancer*, 11(2): 163–178.

Duchi, J. C.; Jordan, M. I.; Wainwright, M. J.; and Wibisono, A. 2015. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5): 2788–2806.

Flaxman, A. D.; Kalai, A. T.; and McMahan, H. B. 2004. Online convex optimization in the bandit setting: gradient descent without a gradient. *arXiv preprint cs/0408007*.

Ghadimi, S.; Lan, G.; and Zhang, H. 2016. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1-2): 267–305.

Kiefer, J.; Wolfowitz, J.; et al. 1952. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3): 462–466.

Li, F.; and Qu, Z. 2021. An inexact proximal augmented Lagrangian framework with arbitrary linearly convergent inner solver for composite convex optimization. *Mathematical Programming Computation*, 1–62.

Li, Z.; Chen, P.-Y.; Liu, S.; Lu, S.; and Xu, Y. 2021. Rate-improved inexact augmented Lagrangian method for constrained nonconvex optimization. In *International Conference on Artificial Intelligence and Statistics*, 2170–2178. PMLR.

Li, Z.; and Xu, Y. 2021. Augmented Lagrangian–Based First-Order Methods for Convex-Constrained Programs with Weakly Convex Objective. *INFORMS Journal on Optimization*, 3(4): 373–397.

Lian, X.; Zhang, H.; Hsieh, C.-J.; Huang, Y.; and Liu, J. 2016. A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order. *arXiv preprint arXiv:1606.00498*.

Lin, Q.; Lu, Z.; and Xiao, L. 2014. An accelerated proximal coordinate gradient method. In *Advances in Neural Information Processing Systems*, 3059–3067.

Lin, Q.; Ma, R.; and Xu, Y. 2019. Inexact Proximal-Point Penalty Methods for Non-Convex Optimization with Non-Convex Constraints. *arXiv preprint arXiv:1908.11518*.

Liu, S.; Chen, J.; Chen, P.-Y.; and Hero, A. 2018. Zeroth-order online alternating direction method of multipliers: Convergence analysis and applications. In *International Conference on Artificial Intelligence and Statistics*, 288–297. PMLR.

Liu, S.; Chen, P.-Y.; Kailkhura, B.; Zhang, G.; Hero, A.; and Varshney, P. K. 2020a. A Primer on Zeroth-Order Optimization in Signal Processing and Machine Learning. *IEEE Signal Processing Magazine*.

Liu, S.; Chepuri, S. P.; Fardad, M.; Maşazade, E.; Leus, G.; and Varshney, P. K. 2016. Sensor selection for estimation with correlated measurement noise. *IEEE Transactions on Signal Processing*, 64(13): 3509–3522.

Liu, S.; Lu, S.; Chen, X.; Feng, Y.; Xu, K.; Al-Dujaili, A.; Hong, M.; and O'Reilly, U.-M. 2020b. Min-max optimization without gradients: Convergence and applications to black-box evasion and poisoning attacks. In *International Conference on Machine Learning*, 6282–6293. PMLR.

Ma, R.; Lin, Q.; and Yang, T. 2020. Quadratically Regularized Subgradient Methods for Weakly Convex Optimization with Weakly Convex Constraints. In *International Conference on Machine Learning*, 6554–6564. PMLR.

Melo, J. G.; Monteiro, R. D.; and Wang, H. 2020a. Iteration-complexity of an inexact proximal accelerated augmented Lagrangian method for solving linearly constrained smooth nonconvex composite optimization problems. *Optimization Online*.

Melo, J. G.; Monteiro, R. D.; and Wang, H. 2020b. Iteration-complexity of an inexact proximal accelerated augmented Lagrangian method for solving linearly constrained smooth nonconvex composite optimization problems. *arXiv preprint arXiv:2006.08048*.

Necoara, I.; and Nedelcu, V. 2014. Rate analysis of inexact dual first-order methods application to dual decomposition. *IEEE Transactions on Automatic Control*, 59(5): 1232–1243.

Nedelcu, V.; Necoara, I.; and Tran-Dinh, Q. 2014. Computational complexity of inexact gradient augmented Lagrangian methods: application to constrained MPC. *SIAM Journal on Control and Optimization*, 52(5): 3109–3134.

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

Ouyang, Y.; Chen, Y.; Lan, G.; and Pasiliao Jr, E. 2015. An accelerated linearized alternating direction method of multipliers. *SIAM Journal on Imaging Sciences*, 8(1): 644–681.

Petricoin III, E. F.; Ardekani, A. M.; Hitt, B. A.; Levine, P. J.; Fusaro, V. A.; Steinberg, S. M.; Mills, G. B.; Simone, C.; Fishman, D. A.; Kohn, E. C.; et al. 2002. Use of proteomic patterns in serum to identify ovarian cancer. *The lancet*, 359(9306): 572–577.

Sahin, M. F.; Alacaoglu, A.; Latorre, F.; Cevher, V.; et al. 2019. An inexact augmented lagrangian framework for nonconvex optimization with nonlinear constraints. In *Advances in Neural Information Processing Systems*, 13943–13955.

Xu, Y. 2021a. First-order methods for constrained convex programming based on linearized augmented Lagrangian function. *INFORMS Journal on Optimization*, 3(1): 89–117.

Xu, Y. 2021b. Iteration complexity of inexact augmented lagrangian methods for constrained convex programming. *Mathematical Programming*, 185(1): 199–244.

# A Proofs

In this section, we provide detailed proofs of our theorems.

## A.1 Proof of Theorem 1

Denote $m = \frac{p}{2}$. Note that the constants $C_1, \cdots, C_m$ in Theorem 1 satisfy the following $m$ equalities:

$$\begin{aligned} C_1 + 2C_2 + \cdots + mC_m &= \frac{1}{2a}, \\ C_1 + 2^3C_2 + \cdots + m^3C_m &= 0, \\ &\vdots \\ C_1 + 2^{2m-1}C_2 + \cdots + m^{2m-1}C_m &= 0. \end{aligned} \tag{24}$$

Since $f$ is $M_j$ coordinate $j$-smooth, by plugging $b \in \{ma, \cdots, a, -a, \cdots, -ma\}$ into Lemma 1, we have the following $2m$ inequalities:

$$\begin{aligned} -\tfrac{|C_m|M_j}{(j+1)!}(ma)^{j+1} \le\, & C_m\Big(f(\mathbf{x} + mae_i) - f(\mathbf{x}) \\ & - ma\nabla_i f(\mathbf{x}) - \cdots - \frac{m^j a^j}{j!}\nabla_i^j f(\mathbf{x})\Big) \\ \le\, & \frac{|C_m|M_j}{(j+1)!}(ma)^{j+1}, \\ & \vdots \\ -\tfrac{|C_1|M_j}{(j+1)!}a^{j+1} \le\, & C_1\Big(f(\mathbf{x} + ae_i) - f(\mathbf{x}) - a\nabla_i f(\mathbf{x}) \\ & - \cdots - \frac{a^j}{j!}\nabla_i^j f(\mathbf{x})\Big) \\ \le\, & \frac{|C_1|M_j}{(j+1)!}a^{j+1}, \\ -\tfrac{|C_1|M_j}{(j+1)!}a^{j+1} \le\, & -C_1\Big(f(\mathbf{x} - ae_i) - f(\mathbf{x}) + a\nabla_i f(\mathbf{x}) \\ & - \cdots - \frac{(-1)^j a^j}{j!}\nabla_i^j f(\mathbf{x})\Big) \\ \le\, & \frac{|C_1|M_j}{(j+1)!}a^{j+1}, \\ & \vdots \\ -\tfrac{|C_m|M_j}{(j+1)!}(ma)^{j+1} \le\, & -C_m\Big(f(\mathbf{x} - mae_i) - f(\mathbf{x}) \\ & + ma\nabla_i f(\mathbf{x}) - \cdots - \frac{(-1)^j m^j a^j}{j!}\nabla_i^j f(\mathbf{x})\Big) \\ \le\, & \frac{|C_m|M_j}{(j+1)!}(ma)^{j+1}. \end{aligned}$$

Summing up the above $2m$ inequalities, we have

$$\begin{aligned} & \Big|\tilde{\nabla}_i f(\mathbf{x}) - (C_1 + 2C_2 + \cdots + mC_m)2a\nabla_i f(\mathbf{x}) \\ & - (C_1 + 2^3C_2 + \cdots + m^3C_m)\frac{2a^3}{3!}\nabla_i^3 f(\mathbf{x}) - \\ & \cdots - (C_1 + 2^{2m-1}C_2 + \cdots + m^{2m-1}C_m) \\ & \frac{2a^{2m-1}}{(2m-1)!}\nabla_i^{2m-1} f(\mathbf{x})\Big| \\ \le & \sum_{q=1}^{m} |C_q| \frac{M_j}{(j+1)!}(qa)^{j+1}. \end{aligned}$$

The above equality combined with (24) gives us

$$|\tilde{\nabla}_i f(\mathbf{x}) - \nabla_i f(\mathbf{x})| \le \sum_{q=1}^{m} |C_q| \frac{M_j q^{j+1}}{(j+1)!}a^{j+1},$$

which is exactly (11).

## A.2 Proof of Theorem 2

To prove Theorem 2, first we bound the objective error in Theorem 6 below, next we bound the stationarity gap by the objective error in Theorem 7 below, then we combine these two theorems with the assumed parameter settings and get the desired results. Below, we present the detailed proof of Theorem 2.

Denote

$$\tilde{\partial} F(\mathbf{x}) = \tilde{\nabla} G(\mathbf{x}) + \partial H(\mathbf{x}).$$

By the updates of Algorithm 2, following the proof of Lemmas 2 and 3 in (Lin, Lu, and Xiao 2014), we immediately have the following two lemmas.

**Lemma 2** *Let $\{\mathbf{x}^k\}$ be generated from Algorithm 2. Then we have $\mathbf{x}^k = \sum_{l=0}^{k} \theta_l^k \mathbf{z}^l$, where $\theta_0^0 = 1, \theta_0^1 = 1 - \sqrt{\mu}, \theta_1^1 = \sqrt{\frac{\mu}{L}}$, and $\forall k \ge 1$,*

$$\theta_l^{k+1} = \begin{cases} \sqrt{\mu}, \ \mathit{if}\, l = k+1, \\ (1 - \frac{\mu}{dL})\frac{(d+1)\alpha}{\alpha+1} - \frac{(1-\alpha)\mu}{dL\alpha}, \ \mathit{if}\, l = k, \\ (1 - \frac{\mu}{dL})\frac{1}{\alpha+1}\theta_l^k, \ \mathit{if}\, l = 0, \ldots, k-1. \end{cases}$$

**Lemma 3** *Let $\hat{\psi}_k := \sum_{l=0}^{k} \theta_l^k H(\mathbf{z}^l)$. Then $\forall k \ge 0$, we have $H(\mathbf{x}^k) \le \hat{\psi}_k$ and*

$$\mathbb{E}_{i_k}[\hat{\psi}_{k+1}] \le \alpha H(\tilde{\mathbf{z}}^{k+1}) + (1-\alpha)\hat{\psi}_k,$$

*where*

$$\begin{aligned} \tilde{\mathbf{z}}^{k+1} := \mathop{\arg\min}_{\mathbf{x} \in \mathbb{R}^d} \Big\{ & \frac{d\alpha L}{2}\|\mathbf{x} - (1-\alpha)\mathbf{z}^k - \alpha\mathbf{y}^k\|^2 \\ & + \langle \tilde{\nabla} G(\mathbf{y}^k), \mathbf{x} - \mathbf{y}^k \rangle + H(\mathbf{x}) \Big\}. \end{aligned} \tag{25}$$

Theorem 6 below extends from Theorem 1 in (Lin, Lu, and Xiao 2014), and gives the convergence rate of Algorithm 2.

**Theorem 6 (ZO-APCU convergence rate)** *Let $\{\mathbf{x}^t\}_{t=0}^{K}$ be generated from Algorithm 2. Then*

$$\begin{aligned} \mathbb{E}[F(\mathbf{x}^K)] - F^* \le & \Big(1 - \frac{1}{d}\sqrt{\frac{\mu}{L}}\Big)^K \Big(F(\mathbf{x}^0) - F^* \\ & + \frac{\mu}{2}\|\mathbf{x}^0 - \mathbf{x}^*\|^2\Big) + ED + \sum_{i=1}^{d} E_i D_i. \end{aligned} \tag{26}$$

**Proof:** By updates of $\mathbf{z}^{k+1}$ and $\mathbf{x}^{k+1}$ in Algorithm 2,

$$x_i^{k+1} = \begin{cases} y_i^k + d\alpha(z_i^{k+1} - z_i^k) + \frac{\mu}{dL}(z_i^k - y_i^k), \text{ if } i = i_k \\ y_i^k, \text{ if } i \neq i_k. \end{cases} \tag{27}$$

By the update of $\mathbf{y}^k$,

$$\mathbf{z}^k - \mathbf{y}^k = -\frac{1}{\alpha}(\mathbf{x}^k - \mathbf{y}^k). \tag{28}$$

Also by the update of $\mathbf{x}^{k+1}$ and $\alpha = \frac{1}{d}\sqrt{\frac{\mu}{L}}$, we have

$$\begin{aligned} \mathbf{x}^{k+1} - \mathbf{y}^k &= \sqrt{\frac{\mu}{L}}\mathbf{z}^{k+1} - (1-\alpha)\sqrt{\frac{\mu}{L}}\mathbf{z}^k - \frac{\mu}{dL}\mathbf{y}^k \\ &= \sqrt{\frac{\mu}{L}}\mathbf{z}^{k+1} - (1-\alpha)\sqrt{\frac{\mu}{L}}(\mathbf{z}^k - \mathbf{y}^k) \\ &\quad - \left((1-\alpha)\sqrt{\frac{\mu}{L}} + \frac{\mu}{dL}\right)\mathbf{y}^k, \end{aligned}$$

which combined with (28) gives us

$$\mathbf{x}^{k+1} - \mathbf{y}^k = d\big(\alpha(\mathbf{z}^{k+1} - \mathbf{y}^k) + (1-\alpha)(\mathbf{x}^k - \mathbf{y}^k)\big).$$

Combining the above equation with (27) and $L$ smoothness of $G$, we have

$$\begin{aligned} G(\mathbf{x}^{k+1}) &\leq G(\mathbf{y}^k) + \nabla_{i_k}G(\mathbf{y}^k)(x_{i_k}^{k+1} - y_{i_k}^k) \\ &\quad + \frac{L}{2}\|x_{i_k}^{k+1} - y_{i_k}^k\|^2 \\ &\leq (1-\alpha)(G(\mathbf{y}^k) + d\nabla_{i_k}G(\mathbf{y}^k)(x_{i_k}^k - y_{i_k}^k)) \\ &\quad + \alpha(G(\mathbf{y}^k) + d\tilde{\nabla}_{i_k}G(\mathbf{y}^k)(z_{i_k}^{k+1} - y_{i_k}^k)) \\ &\quad + \frac{d^2L}{2}[\alpha(\mathbf{z}^{k+1} - \mathbf{y}^k) + (1-\alpha)(\mathbf{x}^k - \mathbf{y}^k)]_{i_k}^2 \\ &\quad + \alpha d E_{i_k} D_{i_k}. \end{aligned}$$

Thus by $\mu$-strong convexity of $G$, the choice of $i_k$, and the definition of $\tilde{\mathbf{z}}^{k+1}$, it holds

$$\begin{aligned} \mathbb{E}_{i_k}[G(\mathbf{x}^{k+1})] &\leq (1-\alpha)G(\mathbf{x}^k) + \alpha\Big(G(\mathbf{y}^k) + \langle\tilde{\nabla}G(\mathbf{y}^k), \\ &\quad \tilde{\mathbf{z}}^{k+1} - \mathbf{y}^k\rangle\Big) + \frac{dL}{2}\|\alpha(\tilde{\mathbf{z}}^{k+1} - \mathbf{y}^k) \\ &\quad + (1-\alpha)(\mathbf{x}^k - \mathbf{y}^k)\|^2 + \alpha\sum_{i=1}^n E_i D_i. \end{aligned} \tag{29}$$

In addition, by (28),

$$\begin{aligned} &\frac{dL}{2}\|\alpha(\tilde{\mathbf{z}}^{k+1} - \mathbf{y}^k) + (1-\alpha)(\mathbf{x}^k - \mathbf{y}^k)\|^2 \\ &\quad = \frac{\mu}{2d}\|\tilde{\mathbf{z}}^{k+1} - (1-\alpha)\mathbf{z}^k - \alpha\mathbf{y}^k\|^2. \end{aligned} \tag{30}$$

Combining the above equality with (29) and $\alpha = \frac{1}{d}\sqrt{\frac{\mu}{L}}$, we have

$$\begin{aligned} &\mathbb{E}_{i_k}[G(\mathbf{x}^{k+1})] \\ &\leq (1-\alpha)G(\mathbf{x}^k) + \alpha\Big[G(\mathbf{y}^k) + \langle\tilde{\nabla}G(\mathbf{y}^k), \tilde{\mathbf{z}}^{k+1} - \mathbf{y}^k\rangle \\ &\quad + \frac{\sqrt{\mu L}}{2}\|\tilde{\mathbf{z}}^{k+1} - (1-\alpha)\mathbf{z}^k - \alpha\mathbf{y}^k\|^2\Big] + \alpha\sum_{i=1}^d E_i D_i, \end{aligned}$$

which combined with Lemma 3 gives

$$\begin{aligned} \mathbb{E}_{i_k}[G(\mathbf{x}^{k+1}) + \hat{\psi}_{k+1}] &\leq (1-\alpha)(G(\mathbf{x}^k) + \hat{\psi}_k) \\ &\quad + \alpha V(\tilde{\mathbf{z}}^{k+1}) + \alpha\sum_{i=1}^d E_i D_i. \end{aligned} \tag{31}$$

In the above

$$\begin{aligned} V(\mathbf{x}) &:= G(\mathbf{y}^k) + \langle\tilde{\nabla}G(\mathbf{y}^k), \mathbf{x} - \mathbf{y}^k\rangle \\ &\quad + \frac{\sqrt{\mu L}}{2}\|\mathbf{x} - (1-\alpha)\mathbf{z}^k - \alpha\mathbf{y}^k\|^2 + H(\mathbf{x}). \end{aligned}$$

By (25),

$$\tilde{\mathbf{z}}^{k+1} = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^d} V(\mathbf{x}). \tag{32}$$

Note $V$ is $\sqrt{\mu L}$-strongly convex, so by (32), $V(\mathbf{x}^*) \geq V(\tilde{\mathbf{z}}^{k+1}) + \frac{\sqrt{\mu L}}{2}\|\mathbf{x}^* - \tilde{\mathbf{z}}^{k+1}\|^2$. Thus,

$$\begin{aligned} V(\tilde{\mathbf{z}}^{k+1}) &\leq V(\mathbf{x}^*) - \frac{\sqrt{\mu L}}{2}\|\mathbf{x}^* - \tilde{\mathbf{z}}^{k+1}\|^2 \\ &= G(\mathbf{y}^k) + \langle\tilde{\nabla}G(\mathbf{y}^k), \mathbf{x}^* - \mathbf{y}^k\rangle + \frac{\sqrt{\mu L}}{2}\|\mathbf{x}^* \\ &\quad - (1-\alpha)\mathbf{z}^k - \alpha\mathbf{y}^k\|^2 + H(\mathbf{x}^*) \\ &\quad - \frac{\sqrt{\mu L}}{2}\|\mathbf{x}^* - \tilde{\mathbf{z}}^{k+1}\|^2 \\ &\leq G(\mathbf{y}^k) + \langle\nabla G(\mathbf{y}^k), \mathbf{x}^* - \mathbf{y}^k\rangle + \frac{\sqrt{\mu L}}{2}\|\mathbf{x}^* \\ &\quad - (1-\alpha)\mathbf{z}^k - \alpha\mathbf{y}^k\|^2 + H(\mathbf{x}^*) \\ &\quad - \frac{\sqrt{\mu L}}{2}\|\mathbf{x}^* - \tilde{\mathbf{z}}^{k+1}\|^2 + ED \\ &\leq G(\mathbf{x}^*) - \frac{\mu}{2}\|\mathbf{x}^* - \mathbf{y}^k\|^2 + \frac{\sqrt{\mu L}}{2}\|\mathbf{x}^* \\ &\quad - (1-\alpha)\mathbf{z}^k - \alpha\mathbf{y}^k\|^2 + H(\mathbf{x}^*), \\ &\quad - \frac{\sqrt{\mu L}}{2}\|\mathbf{x}^* - \tilde{\mathbf{z}}^{k+1}\|^2 + ED, \end{aligned}$$

where the last inequality holds by $\mu$-strong convexity of $G$. Combining the last inequality with (31), we have

$$\begin{aligned} &\mathbb{E}_{i_k}[G(\mathbf{x}^{k+1}) + \hat{\psi}_{k+1}] \\ &\leq (1-\alpha)(G(\mathbf{x}^k) + \hat{\psi}_k) + \alpha F^* - \frac{\alpha\mu}{2}\|\mathbf{x}^* - \mathbf{y}^k\|^2 \\ &- \frac{\mu}{2d}\|\mathbf{x}^* - \tilde{\mathbf{z}}^{k+1}\|^2 + \frac{\mu}{2d}\|\mathbf{x}^* - (1-\alpha)\mathbf{z}^k - \alpha\mathbf{y}^k\|^2 \\ &\quad + \alpha ED + \alpha\sum_{i=1}^d E_i D_i. \end{aligned} \tag{33}$$

Now, by the convexity of $\|\cdot\|^2$, it holds

$$\begin{aligned} &\|\mathbf{x}^* - (1-\alpha)\mathbf{z}^k - \alpha\mathbf{y}^k\|^2 \\ &\leq (1-\alpha)\|\mathbf{x}^* - \mathbf{z}^k\|^2 + \alpha\|\mathbf{x}^* - \mathbf{y}^k\|^2. \end{aligned} \tag{34}$$

Note from the updates of Algorithm 2,

$$\mathbf{z}_i^{k+1} = \begin{cases} \tilde{\mathbf{z}}_i^{k+1}, \text{ if } i = i_k, \\ (1-\alpha)\mathbf{z}_i^k + \alpha\mathbf{y}_i^k, \text{ if } i \neq i_k, \end{cases} \tag{35}$$

which implies

$$\begin{aligned}
&\mathbb{E}_{i_k}[\frac{\mu}{2}\|\mathbf{x}^* - \mathbf{z}^{k+1}\|^2]\\
&= \frac{\mu}{2}\left[\frac{d-1}{d}\|\mathbf{x}^* - (1-\alpha)\mathbf{z}^k - \alpha\mathbf{y}^k\|^2 + \frac{1}{d}\|\mathbf{x}^* - \tilde{\mathbf{z}}^{k+1}\|^2\right]\\
&= \frac{\mu(d-1)}{2d}\|\mathbf{x}^* - (1-\alpha)\mathbf{z}^k - \alpha\mathbf{y}^k\|^2 + \frac{\mu}{2d}\|\mathbf{x}^* - \tilde{\mathbf{z}}^{k+1}\|^2\\
&= \frac{\mu}{2}\|\mathbf{x}^* - (1-\alpha)\mathbf{z}^k - \alpha\mathbf{y}^k\|^2 - \frac{\mu}{2d}\|\mathbf{x}^* - (1-\alpha)\mathbf{z}^k\\
&\quad - \alpha\mathbf{y}^k\|^2 + \frac{\mu}{2d}\|\mathbf{x}^* - \tilde{\mathbf{z}}^{k+1}\|^2\\
&\le \frac{(1-\alpha)\mu}{2}\|\mathbf{x}^* - \mathbf{z}^k\|^2 + \frac{\alpha\mu}{2}\|\mathbf{x}^* - \mathbf{y}^k\|^2\\
&\quad - \frac{\mu}{2d}\|\mathbf{x}^* - (1-\alpha)\mathbf{z}^k - \alpha\mathbf{y}^k\|^2 + \frac{\mu}{2d}\|\mathbf{x}^* - \tilde{\mathbf{z}}^{k+1}\|^2,
\end{aligned}$$

where the last inequality follows from (34). Combining the last inequality with (33), we get

$$\begin{aligned}
&\mathbb{E}_{i_k}\left[G(\mathbf{x}^{k+1}) + \hat{\psi}_{k+1} + \frac{\mu}{2}\|\mathbf{x}^* - \mathbf{z}^{k+1}\|^2\right]\\
&\le (1-\alpha)(G(\mathbf{x}^k) + \hat{\psi}_k + \frac{\mu}{2}\|\mathbf{x}^* - \mathbf{z}^k\|^2) + \alpha F^*\\
&\quad + \alpha ED + \alpha\sum_{i=1}^d E_i D_i,
\end{aligned}$$

which implies

$$\begin{aligned}
&\mathbb{E}_{i_k}\left[G(\mathbf{x}^{k+1}) + \hat{\psi}_{k+1} - F^* + \frac{\mu}{2}\|\mathbf{x}^* - \mathbf{z}^{k+1}\|^2\right]\\
&\le (1-\alpha)(G(\mathbf{x}^k) + \hat{\psi}_k - F^* + \frac{\mu}{2}\|\mathbf{x}^* - \mathbf{z}^k\|^2)\\
&\quad + \alpha ED + \alpha\sum_{i=1}^d E_i D_i.
\end{aligned}$$

Thus,

$$\begin{aligned}
&\mathbb{E}\left[G(\mathbf{x}^k) + \hat{\psi}_k - F^* + \frac{\mu}{2}\|\mathbf{x}^* - \mathbf{z}^k\|^2\right]\\
&\le (1-\alpha)^k\left[F(\mathbf{x}^0) - F^* + \frac{\mu}{2}\|\mathbf{x}^* - \mathbf{x}^0\|^2\right]\\
&\quad + \left(\alpha ED + \alpha\sum_{i=1}^d E_i D_i\right)\sum_{t=0}^{k-1}(1-\alpha)^t.
\end{aligned}$$

Hence,

$$\begin{aligned}
&\mathbb{E}\left[F(\mathbf{x}^k) - F^* + \frac{\mu}{2}\|\mathbf{x}^* - \mathbf{z}^k\|^2\right] \le (1-\alpha)^k[F(\mathbf{x}^0) - F^*\\
&+ \frac{\mu}{2}\|\mathbf{x}^* - \mathbf{x}^0\|^2] + ED + \sum_{i=1}^d E_i D_i.
\end{aligned}$$

where the inequality holds by $F(\mathbf{x}^k) \le G(\mathbf{x}^k) + \hat{\psi}_k$, $\mathbf{x}^k = \sum_{l=0}^k \theta_l^k \mathbf{z}^l$, and the definition of $\hat{\psi}_k$. □

Theorem 7 below bounds the subdifferential by the objective error.

**Theorem 7** *Let*

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}'\in\mathbb{R}^d}\langle\tilde{\nabla}G(\mathbf{x}), \mathbf{x}' - \mathbf{x}\rangle + \frac{L}{2}\|\mathbf{x}' - \mathbf{x}\|^2 + H(\mathbf{x}'), \quad (36)$$

*as in the postprocessing step of Algorithm 2, where* $\|\tilde{\nabla}G(\mathbf{x}) - \nabla G(\mathbf{x})\| \le E$. *Then*

$$\operatorname{dist}(0, \partial F(\hat{\mathbf{x}})) \le 4L\sqrt{\frac{2(F(\mathbf{x}) - F^*)}{\mu}} + 2L\sqrt{\frac{2ED}{\mu}} + E. \quad (37)$$

**Proof:** First, observe that

$$\begin{aligned}
F(\hat{\mathbf{x}}) &\le G(\mathbf{x}) + \langle\nabla G(\mathbf{x}), \hat{\mathbf{x}} - \mathbf{x}\rangle + \frac{L}{2}\|\hat{\mathbf{x}} - \mathbf{x}\|^2 + H(\hat{\mathbf{x}})\\
&\le G(\mathbf{x}) + \langle\tilde{\nabla}G(\mathbf{x}), \hat{\mathbf{x}} - \mathbf{x}\rangle + \frac{L}{2}\|\hat{\mathbf{x}} - \mathbf{x}\|^2 + H(\hat{\mathbf{x}})\\
&\quad + ED\\
&\le G(\mathbf{x}) + H(\mathbf{x}) + ED\\
&= F(\mathbf{x}) + ED, \qquad (38)
\end{aligned}$$

where in above, the first inequality follows from $L$ smoothness of $G$, and the third inequality follows from (36).

Then, by the $\mu$-strong convexity of $G$, we have

$$\frac{\mu}{2}\|\mathbf{x}' - \mathbf{x}^*\|^2 \le F(\mathbf{x}') - F^*, \forall\mathbf{x}' \in \mathbb{R}^d. \quad (39)$$

Furthermore, by (36), we have

$$\mathbf{0} \in \tilde{\nabla}G(\mathbf{x}) + L(\hat{\mathbf{x}} - \mathbf{x}) + \partial H(\hat{\mathbf{x}}). \quad (40)$$

Thus,

$$\begin{aligned}
&\operatorname{dist}(0, \partial F(\hat{\mathbf{x}}))\\
&\le \|\nabla G(\hat{\mathbf{x}}) - \nabla G(\mathbf{x}) + \nabla G(\mathbf{x}) - \tilde{\nabla}G(\mathbf{x}) - L(\hat{\mathbf{x}} - \mathbf{x})\|\\
&\le \|\nabla G(\hat{\mathbf{x}}) - \nabla G(\mathbf{x})\| + \|\nabla G(\mathbf{x}) - \tilde{\nabla}G(\mathbf{x})\|\\
&\quad + \|L(\hat{\mathbf{x}} - \mathbf{x})\|\\
&\le 2L\|\hat{\mathbf{x}} - \mathbf{x}\| + E\\
&\le 2L(\|\hat{\mathbf{x}} - \mathbf{x}^*\| + \|\mathbf{x} - \mathbf{x}^*\|) + E\\
&\le 2L\sqrt{\frac{2}{\mu}}(\sqrt{F(\hat{\mathbf{x}}) - F^*} + \sqrt{F(\mathbf{x}) - F^*}) + E\\
&\le 4L\sqrt{\frac{2(F(\mathbf{x}) - F^*)}{\mu}} + 2L\sqrt{\frac{2ED}{\mu}} + E
\end{aligned}$$

where in above, the first inequality follows from (40), the third inequality follows from $L$ smoothness of $G$, the fifth inequality follows from (39), and the last inequality uses (38). □

Based on Theorem 6 and Theorem 7 above, now we are ready to prove Theorem 2.

By Theorem 6, (17), and the definition of $T$, we have

$$\mathbb{E}[F(\mathbf{x}^T)] - F^* \le \frac{\bar{\varepsilon}}{2} + \frac{\bar{\varepsilon}}{2} = \bar{\varepsilon},$$

where $\bar{\varepsilon} = \frac{\mu}{512L^2}\varepsilon^2$. Combining above inequality with Theorem 7 and (17), we have

$$\begin{aligned}
&\mathbb{E}[\mathrm{dist}(\mathbf{0}, \partial F(\hat{\mathbf{x}}^T))] \\
&\leq 4L\sqrt{\frac{2(F(\mathbf{x}^T) - F^*)}{\mu}} + 2L\sqrt{\frac{2ED}{\mu}} + E \\
&\leq \frac{\varepsilon}{4} + \frac{\varepsilon}{4} = \frac{\varepsilon}{2}.
\end{aligned}$$

Thus,

$$\mathbb{E}[\mathrm{dist}(\mathbf{0}, \tilde{\partial} F(\hat{\mathbf{x}}^T))] \leq \mathbb{E}[\mathrm{dist}(\mathbf{0}, \partial F(\hat{\mathbf{x}}^T))] + E \leq \frac{3\varepsilon}{4},$$

and Algorithm 2 must stop within $T$ iterations. This completes the proof of Theorem 2.

## A.3 Proof of Theorem 3

Observe that

$$\begin{aligned}
\mathbb{E}[K(\varepsilon)] &= \sum_{k=1}^{\infty} kP(K(\varepsilon) = k) \\
&\leq t + \sum_{k=t+1}^{\infty} kP(K(\varepsilon) = k), \forall t \in \mathbb{Z}^+.
\end{aligned}$$

Note

$$\begin{aligned}
P(K(\varepsilon) = k) &= P(q_1 > \varepsilon, \ldots, q_{k-1} > \varepsilon, q_k \leq \varepsilon) \\
&\leq P(q_{k-1} > \varepsilon) \leq \frac{\mathbb{E}[q_{k-1}]}{\varepsilon} \leq \frac{C\eta^{k-1}}{\varepsilon}.
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathbb{E}[K(\varepsilon)] &\leq t + \sum_{k=t}^{\infty}(k+1)\frac{C\eta^k}{\varepsilon} \\
&= t + \frac{C}{\varepsilon}\sum_{k=t}^{\infty}(k+1)\eta^k \\
&= t + \frac{C}{\varepsilon}(\frac{\eta^t}{1-\eta} + \sum_{k=t}^{\infty} k\eta^k). \qquad (41)
\end{aligned}$$

Let $S_t = \sum_{k=t}^{\infty} k\eta^k$. So $\eta S_t = \sum_{k=t}^{\infty} k\eta^{k+1}$, and

$$\begin{aligned}
S_t - \eta S_t &= \sum_{k=t}^{\infty} k\eta^k - \sum_{k=t}^{\infty} k\eta^{k+1} = t\eta^t + \sum_{k=t+1}^{\infty} \eta^k \\
&= t\eta^t + \frac{\eta^{t+1}}{1-\eta}.
\end{aligned}$$

Thus $S_t = \frac{1}{1-\eta}(t\eta^t + \frac{\eta^{t+1}}{1-\eta})$. Combining the above equation with (41), we have $\forall t \in \mathbb{Z}^+$,

$$\begin{aligned}
\mathbb{E}[K(\varepsilon)] &\leq t + \frac{C}{\varepsilon}(\frac{\eta^t}{1-\eta} + \frac{1}{1-\eta}(t\eta^t + \frac{\eta^{t+1}}{1-\eta})) \\
&= t + \frac{C}{\varepsilon(1-\eta)}(t + \frac{1}{1-\eta})\eta^t.
\end{aligned}$$

Let $\psi(t) = t + \frac{C}{\varepsilon(1-\eta)}(t + \frac{1}{1-\eta})\eta^t$. Now we want to choose some $t \in \mathbb{Z}^+$ to bound $\psi(t)$ well. Here we choose $t = \lceil s \rceil$, where $s = \log_{\frac{1}{\eta}}[\frac{C}{\varepsilon(1-\eta)^2}]$. So we have $t \in [s, s+1)$ and $\eta^t \in (\eta^{s+1}, \eta^s] = (\frac{\eta\varepsilon(1-\eta)^2}{C}, \frac{\varepsilon(1-\eta)^2}{C}]$. Hence,

$$\begin{aligned}
&\mathbb{E}[K(\varepsilon)] \leq \psi(t) \\
&\quad \leq s + 1 + \frac{C}{\varepsilon(1-\eta)}(s + 1 + \frac{1}{1-\eta})\frac{\varepsilon(1-\eta)^2}{C} \\
&\quad = (2-\eta)(s+1) + 1 \\
&\quad = \frac{2-\eta}{\log\frac{1}{\eta}}\log\frac{C}{\varepsilon(1-\eta)^2} + 3 - \eta \\
&\quad \leq \frac{2-\eta}{1-\eta}\log\frac{C}{\varepsilon(1-\eta)^2} + 3 - \eta.
\end{aligned}$$

## A.4 Proof of Theorem 4

Let $\Phi_t(\mathbf{x}) := \Phi(\mathbf{x}) + \rho\|\mathbf{x} - \mathbf{x}^t\|^2$ and $\Phi_t^* = \min_{\mathbf{x}} \Phi_t(\mathbf{x})$ for each $t \geq 0$. Note we have $\mathrm{dist}(\mathbf{0}, \partial\Phi_t(\mathbf{x}^{t+1})) \leq \delta = \frac{\varepsilon}{4}$, and also $\Phi_t$ is $\rho$-strongly convex. Hence $\Phi_t(\mathbf{x}^{t+1}) - \Phi_t^* \leq \frac{\delta^2}{2\rho}$, and $\Phi(\mathbf{x}^{t+1}) + \rho\|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 - \Phi(\mathbf{x}^t) \leq \frac{\delta^2}{2\rho}$. Thus,

$$\begin{aligned}
&\Phi(\mathbf{x}^T) - \Phi(\mathbf{x}^0) + \rho\sum_{t=0}^{T-1}\|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \leq \frac{T\delta^2}{2\rho} \\
&T\min_{0\leq t\leq T-1}\|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \leq \frac{1}{\rho}\left(\frac{T\delta^2}{2\rho} + [\Phi(\mathbf{x}^0) - \Phi(\mathbf{x}^T)]\right) \\
&2\rho\min_{0\leq t\leq T-1}\|\mathbf{x}^{t+1} - \mathbf{x}^t\| \leq 2\sqrt{\frac{\delta^2}{2} + \frac{\rho[\Phi(\mathbf{x}^0) - \Phi^*]}{T}}.
\end{aligned} \qquad (42)$$

Since $T \geq \frac{32\rho}{\varepsilon^2}[\Phi(\mathbf{x}^0) - \Phi^*]$ and $\delta = \frac{\varepsilon}{4}$, we have

$$\frac{\rho}{T}[\Phi(\mathbf{x}^0) - \Phi^*] \leq \frac{\varepsilon^2}{32}, \qquad (43)$$

and thus (42) implies

$$2\rho\min_{0\leq t\leq T-1}\|\mathbf{x}^{t+1} - \mathbf{x}^t\| \leq \frac{\varepsilon}{2}. \qquad (44)$$

Therefore, the ZO-iPPM subroutine in Algorithm 1 must stop within $T$ iterations, from its stopping condition, and when it stops, the output $\mathbf{x}^S$ satisfies $2\rho\|\mathbf{x}^S - \mathbf{x}^{S-1}\| \leq \frac{\varepsilon}{2}$.

Now recall $\mathrm{dist}(0, \partial\Phi_t(\mathbf{x}^{t+1})) \leq \delta = \frac{\varepsilon}{4}$, i.e.,

$$\mathrm{dist}(0, \partial\Phi(\mathbf{x}^{t+1}) + 2\rho(\mathbf{x}^{t+1} - \mathbf{x}^t)) \leq \frac{\varepsilon}{4}, \forall t \geq 0. \qquad (45)$$

The above inequality together with $2\rho\|\mathbf{x}^S - \mathbf{x}^{S-1}\| \leq \frac{\varepsilon}{2}$ gives

$$\mathrm{dist}(\mathbf{0}, \partial\Phi(\mathbf{x}^S)) \leq \varepsilon,$$

which implies that $\mathbf{x}^S$ is an $\varepsilon$-stationary point to (18).

Finally, we apply Corollary 1 to obtain the expected overall complexity and complete the proof.

## A.5 Proof of Theorem 5

To prove Theorem 5, first we bound the dual variable $\|\mathbf{y}^k\|$, next we establish upper and lower bounds of the AL objective value inside every outer iteration, then we combine above results with Theorem 4 and Corollary 1 to show the total query complexity to reach a near-KKT point, finally we establish the improved query complexity in the special case when the constraints are convex. Below, we present the detailed proof of Theorem 5.

First, by (14), $\mathbf{y}^0 = \mathbf{0}$, and the definition of $w_k$ in Theorem 5, we have

$$\begin{aligned} \|\mathbf{y}^k\| \le \sum_{t=0}^{k-1} w_t \|\mathbf{c}(\mathbf{x}^{t+1})\| &= \sum_{t=0}^{k-1} M(t+1)^q =: y_k \\ &= O(k^{q+1}), \forall k \ge 0. \end{aligned} \tag{46}$$

Following the first part of the proof of Theorem 2 in (Li et al. 2021), we can easily show that at most $K = O(\log \varepsilon^{-1})$ outer iALM iterations are needed to guarantee $\mathbf{x}^K$ to be an $\varepsilon$-KKT point of (1). Hence, $\beta_k = O(\varepsilon^{-1}), \forall\, 0 \le k \le K$.

Combining the above bound on $K$ with (46), we have

$$\begin{aligned} \|\mathbf{y}^k\| \le y_K := \sum_{t=0}^{K-1} M(K+1)^q = O(K^{q+1}) \\ = O\big((\log \varepsilon^{-1})^{q+1}\big),\ \forall 1 \le k \le K. \end{aligned}$$

Hence from (13), we have $\hat{\rho}_k = O(\beta_k) = O(\varepsilon^{-1}), \hat{L}_k = O(\beta_k) = O(\varepsilon^{-1}),\ \forall\, 0 \le k \le K$.

Notice that equations (41) and (42) in (Li et al. 2021) still hold with $y_{\max}$ replaced by $y_k$. Hence, $\forall k \le K, \forall \mathbf{x} \in \operatorname{dom}(h)$,

$$\mathcal{L}_{\beta_k}(\mathbf{x}^k, \mathbf{y}^k) - \mathcal{L}_{\beta_k}(\mathbf{x}, \mathbf{y}^k) = O\left(y_k \left(1 + \frac{y_k}{\beta_k}\right)\right).$$

The above equation together with Theorem 4 gives that for any $k \le K$, at most $T_k^{\mathrm{PPM}}$ iPPM iterations are needed to terminate the ZO-iPPM subroutine in Algorithm 1 at the $k$-th outer iALM iteration, where

$$\begin{aligned} T_k^{\mathrm{PPM}} &= \left\lceil \frac{32\hat{\rho}_k}{\varepsilon^2} \left(\mathcal{L}_{\beta_k}(\mathbf{x}^k, \mathbf{y}^k) - \min_{\mathbf{x}} \mathcal{L}_{\beta_k}(\mathbf{x}, \mathbf{y}^k)\right) \right\rceil \\ &= O\left(\frac{\hat{\rho}_k y_k \left(1 + \frac{y_k}{\beta_k}\right)}{\varepsilon^2}\right). \end{aligned}$$

Also, by Corollary 1, at most $T_k^{\mathrm{APCU}}$ function value queries are needed to terminate Algorithm 2, where

$$\mathbb{E}[T_k^{\mathrm{APCU}}] = \tilde{O}\left(d\sqrt{\frac{\hat{L}_k}{\hat{\rho}_k}}\right), \forall k \ge 0.$$

Therefore, for all $k \le K$,

$$\begin{aligned} \mathbb{E}[T_k^{\mathrm{PPM}} T_k^{\mathrm{APG}}] &= \tilde{O}\left(\frac{d\sqrt{\hat{L}_k \hat{\rho}_k}}{\varepsilon^2} y_k \left(1 + \frac{y_k}{\beta_k}\right)\right) \\ &= \tilde{O}\left(\frac{d y_k}{\varepsilon^2}(\beta_k + y_k)\right) \\ &= \tilde{O}\left(\frac{d k^{q+1}}{\varepsilon^2}(\sigma^k + k^{q+1})\right) \\ &= \tilde{O}\left(\frac{d K^{q+1}}{\varepsilon^2}(\sigma^K + K^{q+1})\right) \\ &= O\left(\frac{d(\log \varepsilon^{-1})^{q+2}}{\varepsilon^3}\right) \\ &= \tilde{O}\left(\frac{d}{\varepsilon^3}\right), \end{aligned}$$

where the second equation is from $\hat{L}_k = O(\beta_k)$ and $\hat{\rho}_k = O(\beta_k)$, and the fifth one is obtained by $K = O(\log \varepsilon^{-1})$.

Consequently, for a general nonlinear $\mathbf{c}(\cdot)$, at most $T$ function value queries in total are needed to find the $\varepsilon$-KKT point $\mathbf{x}^K$, where

$$\begin{aligned} \mathbb{E}[T] &= \sum_{k=0}^{K-1} \mathbb{E}[T_k^{\mathrm{PPM}} T_k^{\mathrm{APG}}] \\ &= \tilde{O}\left(dK\varepsilon^{-3}(\log \varepsilon^{-1})^{q+2}\right) = \tilde{O}\left(d\varepsilon^{-3}\right). \end{aligned}$$

In the special case when $\mathbf{c}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}$, the term $\|\mathbf{c}(\mathbf{x})\|^2 = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ is convex, so we have $\rho_c = 0$. Hence, by (13), $\hat{\rho}_k = \tilde{O}(1), \forall k \ge 0$. Then following the same arguments as above, we obtain that for any $k \le K$,

$$\begin{aligned} \mathbb{E}[T_k^{\mathrm{PPM}} T_k^{\mathrm{APG}}] &= O\left(\frac{\sqrt{\hat{L}_k \hat{\rho}_k}}{\varepsilon^2}(\log \varepsilon^{-1})^{q+2}\right) \\ &= \tilde{O}\left(\varepsilon^{-\frac{5}{2}}\right). \end{aligned}$$

Therefore, at most $T$ total function value queries are needed to find the $\varepsilon$-KKT point $\mathbf{x}^K$, where

$$\mathbb{E}[T] = \sum_{k=0}^{K-1} \mathbb{E}[T_k^{\mathrm{PPM}} T_k^{\mathrm{APG}}] = \tilde{O}\left(\varepsilon^{-\frac{5}{2}}\right),$$

which completes the proof.

## B Efficient Implementation of Algorithm 2

In this section, we provide Algorithm 3, which is a practical efficient implementation of the equivalent Algorithm 2. Algorithm 3 is efficient in the sense that it avoids the full-dimensional vector operations which exist in Algorithm 2. Algorithm 3 is equivalent to Algorithm 2 because their iterates satisfy the relations

$$\begin{aligned} \mathbf{x}^k &= \rho^k \mathbf{u}^k + \mathbf{v}^k, \\ \mathbf{y}^k &= \rho^{k+1} \mathbf{u}^k + \mathbf{v}^k, \\ \mathbf{z}^k &= -\rho^k \mathbf{u}^k + \mathbf{v}^k, \end{aligned}$$

which are proven in Proposition 1 of (Lin, Lu, and Xiao 2014).

**Algorithm 3:** Efficient implementation of ZO-APCG for (15)

```
Input: $\mathbf{x}^{-1} \in \mathrm{dom}(\psi)$, tolerance $\varepsilon$, smoothness $L$, strong convexity $\mu$, and epoch length $l$.
Initialization:
    $\mathbf{u}^0 = \mathbf{0}, \mathbf{v}^0 = \mathbf{x}^0, \alpha = \frac{1}{d}\sqrt{\frac{\mu}{L}}, \rho = \frac{1-\alpha}{1+\alpha}$
for $k = 0, 1, \ldots, K-1$ do
    Sample $i_k \in [d]$ uniformly and compute $\tilde{\nabla}_{i_k} G(\mathbf{y}^k),$ s.t. $\|\tilde{\nabla}_{i_k} G(\mathbf{y}^k) - \nabla_{i_k} G(\mathbf{y}^k)\| \leq E_{i_k}$.
    Compute $\mathbf{h}_{i_k}^k = \arg\min_{\mathbf{h} \in \mathbb{R}^{d_{i_k}}} \{\frac{d\alpha L}{2}\|\mathbf{h}\|^2 + \langle \tilde{\nabla}_{i_k} G(\rho^{k+1}\mathbf{u}^k + \mathbf{v}^k), \mathbf{h}\rangle + H_{i_k}(-\rho^{k+1}\mathbf{u}_{i_k}^k + \mathbf{v}_{i_k}^k + \mathbf{h})\}$.
    $\mathbf{u}^{k+1} = \mathbf{u}^k, \mathbf{v}^{k+1} = \mathbf{v}^k,$
    $\mathbf{u}_{i_k}^{k+1} = \mathbf{u}_{i_k}^k - \frac{1-d\alpha}{2\rho^{k+1}}\mathbf{h}_{i_k}^k, \mathbf{v}_{i_k}^{k+1} = \mathbf{v}_{i_k}^k + \frac{1+d\alpha}{2}\mathbf{h}_{i_k}^k.$
    $\mathbf{x}^{k+1} = \rho^{k+1}\mathbf{u}^{k+1} + \mathbf{v}^{k+1}$.
    if $k + 1 \equiv 0 \pmod{l}$ then
        Compute $\tilde{\nabla} G(\mathbf{x}^{k+1}),$ s.t. $\|\tilde{\nabla} G(\mathbf{x}^{k+1}) - \nabla G(\mathbf{x}^{k+1})\| \leq E$
        $\hat{\mathbf{x}}^{k+1} = \arg\min_{\mathbf{x} \in \mathbb{R}^d}\{\langle \tilde{\nabla} G(\mathbf{x}^{k+1}), \mathbf{x} - \mathbf{x}^{k+1}\rangle + \frac{L}{2}\|\mathbf{x} - \mathbf{x}^{k+1}\|^2 + H(\mathbf{x})\}$.
        Return $\hat{\mathbf{x}}^{k+1}$ and stop if $\mathrm{dist}(\mathbf{0}, \tilde{\partial} F(\hat{\mathbf{x}}^{k+1})) \leq \frac{3\varepsilon}{4}$.
```

# C Additional Numerical Experiments

In this section, we provide additional numerical experiments to demonstrate the empirical performance of the proposed ZO-iALM. All the tests were performed in MATLAB 2019b on a Macbook Pro with 4 cores and 16GB memory.

## C.1 Nonconvex Linearly Constrained Quadratic Programs (LCQP)

In this subsection, we test the proposed method on solving nonconvex LCQP:

$$\begin{aligned} &\min_{\mathbf{x} \in \mathbb{R}^n} \tfrac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{c}^\top \mathbf{x}, \\ &\text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b},\ x_i \in [l_i, u_i],\ \forall\, i \in [n], \end{aligned} \tag{47}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, and $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is symmetric and indefinite (thus the objective is nonconvex). In the test, we generated all data randomly. The smallest eigenvalue of $\mathbf{Q}$ is $-\rho < 0$, and thus the problem is $\rho$-weakly convex. For all tested instances, we set $l_i = -5$ and $u_i = 5$ for each $i \in [n]$.

We generated an LCQP instance with $m = 10$, $n = 100$, and $\rho = 1$. Since no other existing ZOMs are able to handle nonconvex constrained problems, we compare our proposed ZO-iALM to two other methods that replace our ZO-iPPM subroutine with ZO-AdaMM (Liu et al. 2018) and ZO-ProxSGD (Ghadimi, Lan, and Zhang 2016) respectively. We set $\beta_k = \sigma^k \beta_0$ with $\sigma = 3$ and $\beta_0 = 0.01$ for all iALM outer loops. In each method, we set $a = 10^{-4}$ to be the sampling radius. In ZO-AdaMM, we set $\alpha = 1, \beta_1 = 0.75, \beta_2 = 1$. In ZO-ProxSGD, we fix the step size to be $\frac{1}{nL}$, where $L$ is the smoothness constant of each subproblem.The tolerance was set to $\varepsilon = 10^{-3}$ for the proposed ZO-iALM and $\varepsilon = 0.5$ for all compared methods since they could not converge with a tolerance as low as $0.001$. We also conducted experiments replacing our ZO-APCU inner solver with ZO-ARS in (Nesterov and Spokoiny 2017), but ZO-ARS in (Nesterov and Spokoiny 2017) failed to converge.

Table 1: Results by the proposed ZO-iALM with ZO-iPPM, the ZO-AdaMM in (Chen et al. 2019), and the ZO-ProxSGD in (Ghadimi, Lan, and Zhang 2016) on solving a black-box 1-weakly convex LCQP (47) of size $m = 10$ and $n = 100$.

| method | pres | dres | time | #Obj |
|---|---|---|---|---|
| ZO-iALM | 9.61e-4 | 6.83e-4 | 11.42 | 2344400 |
| ZO-AdaMM | 0.42 | 0.47 | 61.86 | 8060000 |
| ZO-ProxSGD | 0.24 | 0.49 | 86.98 | 16520000 |

In Table 1, we report, for each method, the primal residual, dual residual, running time (in seconds), and the number of queries, shortened as `pres`, `dres`, `time`, and `#Obj`.

From the results, we conclude that, to reach an $\varepsilon$-KKT point to the black-box LCQP problem, the proposed ZO-iALM needs significantly fewer queries to reach a significantly higher accuracy than all other compared methods.

## C.2 Unconstrained Strongly-convex Quadratic Programs (USCQP)

In this subsection, we test the proposed core subsolver ZO-APCU on solving USCQP:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \tfrac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{c}^\top \mathbf{x}, \tag{48}$$

where $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is symmetric with $\mu > 0$ as its smallest eigenvalue (thus the objective is $\mu$ strongly-convex). In the test, we generated all data randomly.

We generated an USCQP instance with $n = 100$, and $\mu = 1$. We compare our proposed ZO-APCU to ZO-AdaMM (Chen et al. 2019) and ZO-ARS (Nesterov and Spokoiny 2017) respectively. In each method, we set $\varepsilon = 10^{-3}$ to be the error tolerance and $a = 10^{-5}$ to be the sampling radius. In ZO-AdaMM, we set $\alpha = 1, \beta_1 = 0.75, \beta_2 = 1$. In ZO-ProxSGD, we fix the step size to be $\frac{1}{nL}$, where $L$ is the smoothness constant of each subproblem. In ZO-ARS, we set $\theta = \frac{1}{16L(n+4)^2}, h = \frac{1}{4L(n+4)}, \alpha = \sqrt{L\theta}$, where $L$ is the smoothness constant of the objective.

In Figure 3, we compare the objective error trajectories of our method, ZO-AdaMM in (Chen et al. 2019), and ZO-ARS in (Nesterov and Spokoiny 2017). For each method, we also report the objective error, gradient norm, running time (in seconds), and the query count, shortened as `objErr`, `normGrad`, `time`, and `#Obj` in Table 2. From the results, we conclude that the proposed ZO-APCU reaches an $\varepsilon$-stationary point to the USCQP problem (48) with at least 3 times fewer number of queries compared to all other methods.

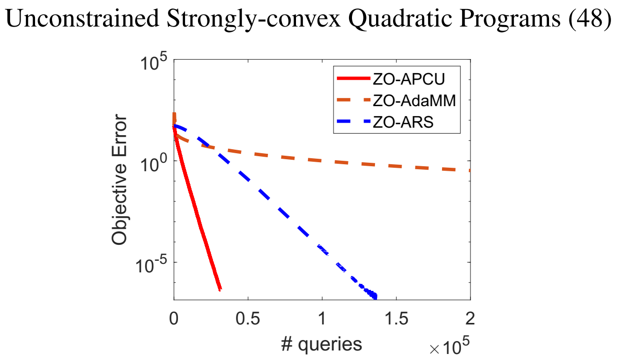

Figure 3: Comparison of our proposed ZO-APCU, ZO-AdaMM in (Chen et al. 2019), and ZO-ARS in (Nesterov and Spokoiny 2017). The plot shows the objective error.

Table 2: Results by the proposed ZO-APCU, the ZO-AdaMM in (Chen et al. 2019), and the ZO-ProxSGD in (Ghadimi, Lan, and Zhang 2016) on solving the unconstrained strongly-convex quadratic programs (48).

| method | objErr | normGrad | time | #Obj |
|---|---|---|---|---|
| ZO-APCU | 4.29e-7 | 1.00e-3 | 0.50 | 31400 |
| ZO-AdaMM | 4.93e-7 | 9.99e-4 | 52.17 | 11576470 |
| ZO-ARS | 1.80e-7 | 4.96e-4 | 3.68 | 136200 |

## C.3 Logistic Regression (LR)

In this subsection, we compare different multi-point coordinate gradient estimators proposed in Section  in the high accuracy setting. We use the proposed subsolver ZO-APCU on solving the logistic regression problem:

$$\min_{\mathbf{w},b} \frac{1}{N}\sum_{i=1}^{N} \log\left(1+\exp[-y_i(\mathbf{w}^\top \mathbf{x}_i+b)]\right)+\frac{\lambda}{2}\|\mathbf{w}\|_2^2+\frac{\lambda}{2}b^2, \tag{49}$$

where we are given the training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ with $y_i \in \{+1, -1\}$ for each $i = 1, \dots, N$. Note that $\lambda$ is the strong convexity constant of the objective function. In the test, we use the *spamdata* data set with $N = 100$ as the number of randomly chosen data points and $n = 57$ as the variable dimension. In this subsection, we run two independent tests.

In the first test, we compare the final accuracy using our proposed ZO-APCU with 2-point and 4-point coordinate gradient estimators respectively. In each method, we set $\lambda = 1$ as the strong convexity constant, $\varepsilon = 10^{-11}$ as the error tolerance, $a = 10^{-5}$ as the sampling radius, and $K = 114000$ to be the maximum number of queries.

In Figures 4 and 5, we compare the gradient norm trajectories of the proposed ZO-APCU under 2-point and 4-point settings. For each setting, we also report the gradient norm, running time (in seconds), and the query count, shortened as `normGrad`, `time`, and `#Obj` in Table 3. From the results, we conclude that using the 4-point gradient estimator enables ZO-APCU to reach a more accurate solution to the LR problem (49) than using the 2-point gradient estimator.

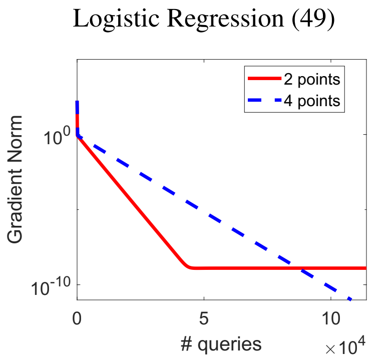

Figure 4: Comparison of ZO-APCU with 2-point and 4-point gradient estimators. The plot shows the gradient norm versus the query count. The sampling radius is $a = 10^{-5}$.

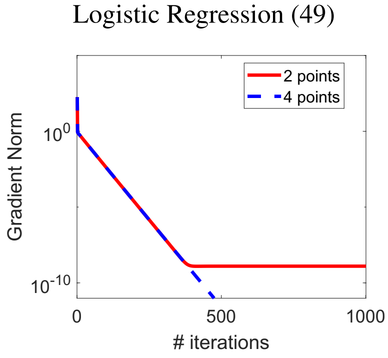

Figure 5: Comparison of ZO-APCU with 2-point and 4-point gradient estimators. The plot shows the gradient norm versus the iteration count. The sampling radius is $a = 10^{-5}$.

In the second test, we compare the final accuracy using our proposed ZO-APCU with 2-point, 4-point, and 6-point gradient estimators respectively, by a larger sampling radius $a = 10^{-2}$. In each method, we set $\lambda = 1$ as the strong convexity constant, $\varepsilon = 10^{-7}$ as the error tolerance, and $K = 114000$ to be the maximum number of queries.

In Figures 6 and 7, we compare the gradient norm trajectories of the proposed ZO-APCU under three settings. For each setting, we also report the gradient norm, running time (in seconds), and the query count, shortened as `normGrad`, `time`, and `#Obj` in Table 4. From the results, we conclude that when the sampling radius is large, to reach a decent accuracy to the LR problem (49), it is beneficial to use more

Table 3: Results by the proposed ZO-APCU with 2-point and 4-point gradient estimators respectively on solving the logistic regression problem (49).

| #Points | normGrad | time | #Obj |
|---|---|---|---|
| 2-pt | 1.26e-9 | 3.12 | 114000 |
| 4-pt | 9.68e-12 | 1.38 | 108072 |

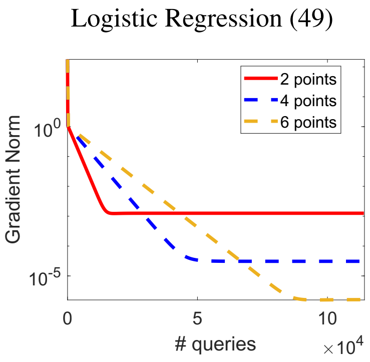

Figure 6: Comparison of ZO-APCU with $(2, 4, 6)$-point gradient estimators. The plot shows the gradient norm versus the query count. The sampling radius is $a = 10^{-2}$.

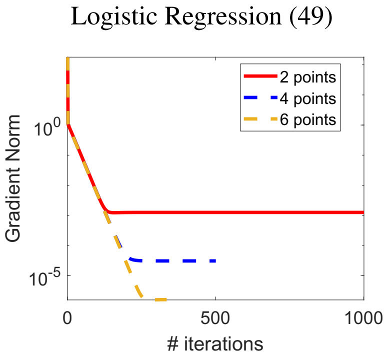

Figure 7: Comparison of ZO-APCU with $(2, 4, 6)$-point gradient estimators. The plot shows the gradient norm versus the iteration count. The sampling radius is $a = 10^{-2}$.

points in the gradient estimators.

## D Additional Table

In Table 5, for the resource allocation problem in sensor networks (21), we report the primal residual, dual residual, running time (in seconds), and the query count, shortened as `pres`, `dres`, `time`, and `#Obj`.

Table 4: Results by the proposed ZO-APCU with $(2, 4, 6)$-point gradient estimators respectively on solving the logistic regression problem (49).

| #Points | normGrad | time | #Obj |
|---|---|---|---|
| 2-pt | 1.3e-3 | 2.71 | 114000 |
| 4-pt | 3.08e-5 | 1.38 | 114000 |
| 6-pt | 1.60e-6 | 1.12 | 114000 |

Table 5: Results by the proposed ZO-iALM with ZO-iPPM, the ZO-AdaMM in (Chen et al. 2019), and the ZO-ProxSGD in (Ghadimi, Lan, and Zhang 2016) on solving the resource allocation problem in sensor networks (21).

| method | pres | dres | time | #Obj |
|---|---|---|---|---|
| ZO-iALM | 4.86e-2 | 7.01e-2 | 53.21 | 303790 |
| ZO-AdaMM | 2.11e-2 | 5.24e-2 | 80.09 | 659340 |
| ZO-ProxSGD | 4.97e-2 | 5.14e-2 | 427.38 | 3590277 |