ZO-BCD: A Zeroth-Order Block Coordinate Descent Algorithm for Huge-Scale Black-Box Optimization

HanQin Cai (UCLA) Yuchen Lou (U. Hong Kong) Daniel McKenzie (UCLA) Wotao Yin (Alibaba US)

Highlights

- Novel Zeroth-Order (ZO) Optimization algorithm using compressed sensing, randomized finite differencing and block coordinate descent.
- Query *and* computational complexity grow sub-linearly in d.
- Convergence is theoretically guaranteed.
- Benchmark on adversarial attacks on image/audio ($d > 10^6$).

Huge-Scale Zeroth-Order Optimization

Zeroth-Order Optimization. Use only noisy function queries (no gradients) to find $x_\star \approx \mathrm{argmin}_{x\in\mathbb{R}^d} f(x)$. This problem arises in many domains, *e.g.* simulation-based optimization, adversarial attacks, hyperparameter tuning and reinforcement learning. Our focus is *adversarial attacks*: perturbing a signal to fool a (neural-network) classifier (see Fig. 1).

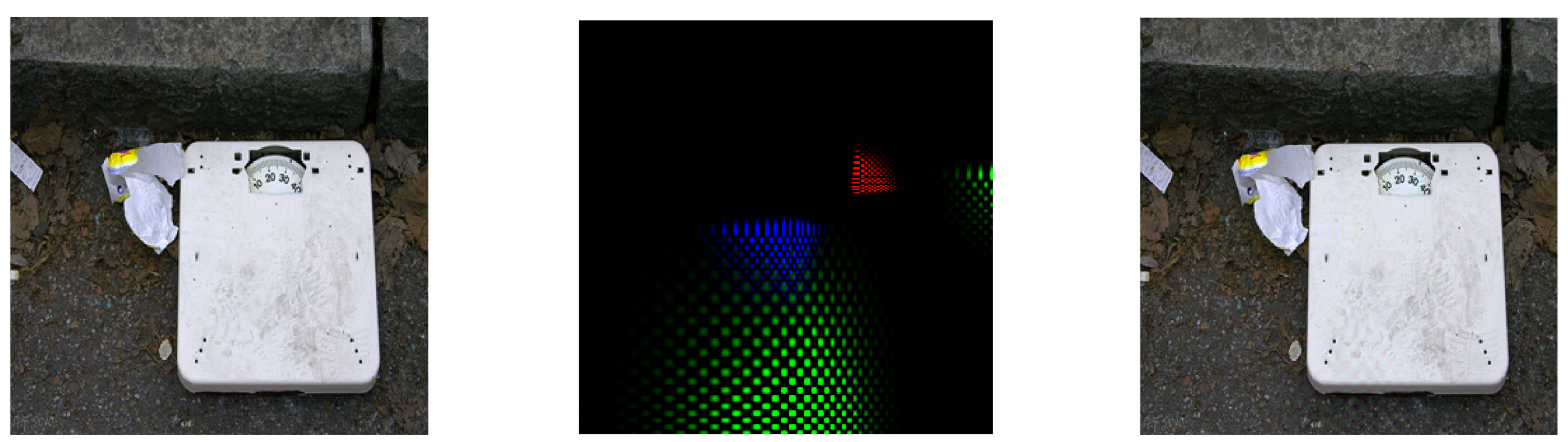

Figure 1: **Left:** Clean signal correctly labelled 'scale'. **Middle:** Attacking perturbation found using ZO-BCD (scaled up 50×). **Right:** Attacked signal incorrectly labelled 'switch'.

Function queries usually *expensive* so algorithms need to be *query-efficient*.

Standard Gradient Estimators. Many ZO algorithms use finite differences to approximate $\nabla f(x)$:

$$\nabla_i f(x) \approx \frac{f(x+\delta e^i)-f(x)}{\delta} \quad \text{or} \quad \nabla f(x) \approx \sum_{i=1}^m \frac{f(x+\delta z^i)-f(x)}{\delta} z_i \qquad (1)$$

where $\delta > 0$, e^i is a coordinate vector and z^i is a random vector. This requires $\mathcal{O}(d)$ function queries to accurately approximate $\nabla f(x)$ [1]. For large problems, *e.g.* $d > 10^7$, this is undesirable. Recent work [2], [3] exploits *compressed sensing* to reduce query complexity to $\mathcal{O}(s \log d)$ assuming $\nabla f(x)$ is $s-$sparse, *i.e.*, $\|\nabla f(x)\|_0 := |\{i : \nabla_i f(x) \neq 0\}| \leq s$.

The ZORO Gradient Estimator. [3] proposes the following scheme for approximating $\nabla f(x)$ using Rademacher random vectors z^i:

$$y_i = \frac{f(x+\delta z^i)-f(x)}{\delta} \approx (z^i)^\top \nabla f(x) \text{ for } i = 1, \cdots, m \approx s \log d$$

$$y = [y_1 \quad \ldots \quad y_m] \text{ and } Z = \frac{1}{\sqrt{m}} [z^1 \quad \ldots \quad z^m]^\top$$

$$\nabla f(x) \approx \hat{g} \triangleq \mathrm{argmin}_{v\in\mathbb{R}^d} \|Zv - y\|_2 \text{ s.t. } \|v\|_0 \leq s \text{ using CoSaMP [4]}$$

although query efficient, this is *computationally intractible* for large d.

ZO-BCD: Algorithm Description

ZO-BCD. We propose a novel algorithm, coined ZO-BCD, exhibiting favorable query *and* computational complexity. Key features:

- Randomized blocks ensure approximately equi-sparse block gradients.
- ZORO gradient estimator tractable when applied to block gradients.
- Recent work [5] guarantees inexact BCD converges.

A sketch version of the algorithm is as follows:

Algorithm 1 Zeroth-Order Block Coordinate Descent (ZO-BCD)

for $j = 1, \cdots, m$ ◁ Create sample directions
  $z^j \leftarrow \texttt{randvec}(d)$ ◁ Random vector
$\pi \leftarrow \texttt{randperm}(d)$ ◁ Random permutation
for $j = 1, \cdots, J$ ◁ Create J blocks
  $x^{(j)} \leftarrow [x_{\pi((j-1)\frac{d}{J}+1)}, \cdots, x_{\pi(j\frac{d}{J}+1)}]$ ◁ Assign variables to blocks
for $k = 1, \cdots, K$ ◁ Do K iterations
  $j \leftarrow \texttt{randint}(\{1, \cdots, J\})$ ◁ Randomly select a block
  for $i = 1, \cdots, m$ ◁ Query objective function
    $y_i = \frac{f(x+\delta z^i)-f(x)}{\delta}$ ◁ Approximate $z_i^\top \nabla f(x)$
  $\hat{g}^{(j)} \leftarrow \arg\min_{v:\|v\|_0\leq s} \|Zv - y\|_2$ ◁ Estimate block gradient
  $x_{k+1} \leftarrow x_k - \alpha \hat{g}^{(j)}$ ◁ Step of BCD
return x_K ◁ Approximated minimizer

Decreasing the Memory Footprint. ZO-BCD stores m d-dim vectors in memory. For large d, this may be infeasible. Thus, we also propose a memory-efficient variant, ZO-BRD-RC. Here $z^1, \ldots, z^m$ are randomly selected rows from the **R**ademacher **C**irculant matrix:

$$\mathcal{C}(z) = \begin{pmatrix} z_1 & z_2 & \cdots & z_{d/J} \\ z_{d/J} & z_1 & \cdots & z_{d/J-1} \\ \vdots & \ddots & \ddots & \vdots \\ z_2 & \cdots & z_{d/J} & z_1 \end{pmatrix}. \qquad (2)$$

and only $z \in \mathbb{R}^d$ needs to be stored. Circulant matrices also afford a *fast multiplication*: $\mathcal{C}(z) \cdot x = \mathcal{F}\left(\mathcal{F}(z) \cdot \mathcal{F}^{-1}(x)\right)$ where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the (Fast) Fourier transform and its inverse.

Convergence. With the analysis of inexact block coordinate descent in [5], CoSaMP & compressed sensing in [4], error bounds in [3], and further analysis of randomization, we have the following theorem. Note $f_\star = \mathrm{argmin}\, f(x)$ and $\tilde{\mathcal{O}}(\cdot)$ hides logarithmic factors.

Theorem 1 *Assume f is convex. Assume $\|\nabla f(x)\|_0 \leq s$ for all $x \in \mathbb{R}^d$. Choose $J \ll d$ random blocks. ZO-BCD returns x_K satisfying $f(x_K) - f_\star \leq \varepsilon$ using $\tilde{\mathcal{O}}(s/\varepsilon)$ total queries and $\tilde{\mathcal{O}}(sd/J^2)$ FLOPS per iteration with high probability. And ZO-BCD-RC returns x_K satisfying $f(x_K) - f_\star \leq \varepsilon$ using $\tilde{\mathcal{O}}(s/\varepsilon)$ total queries and $\tilde{\mathcal{O}}(d/J)$ FLOPS per iteration with high probability.*

See our paper for the proof.

Numerical Experiments

Synthetic Experiments. We consider two test functions exhibiting gradient sparsity: 1. Sparse quadric: $f(x) = \sum_{i=1}^s x_i^2$; 2. Max-$s$-squared-sum: $f(x) = \sum_{i=1}^s x_{\sigma(i)}^2$ where $|x_{\sigma(1)}| \geq |x_{\sigma(2)}| \geq \cdots$.
ZO-BCD matches/exceeds state-of-the-art query complexity with superior computational complexity (Fig. 2).

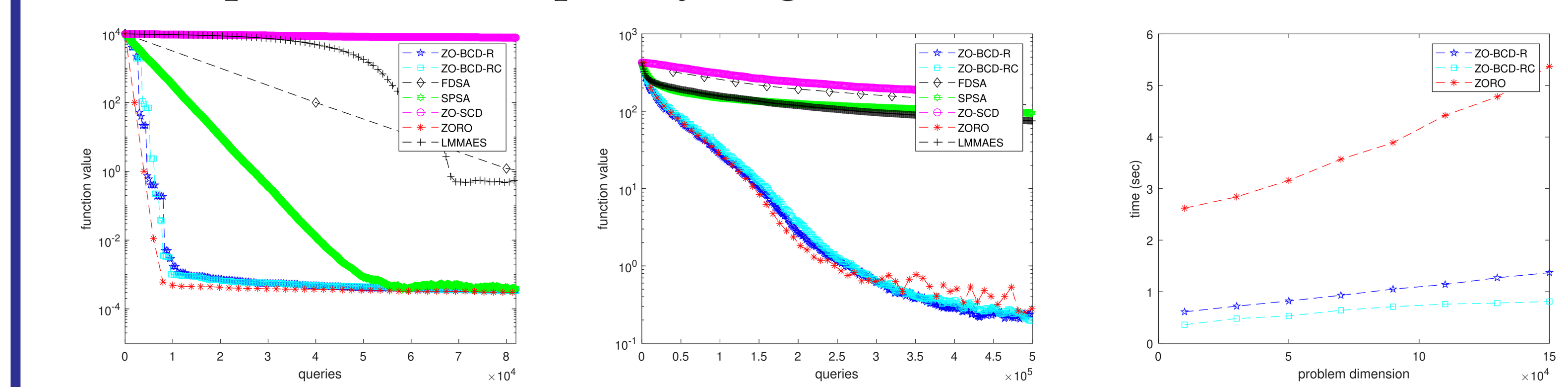

Figure 2: **Left:** Sparse quadric. **Center:** Max-s-squared-sum. **Right:** runtime for sparse quadric.

Adversarial Attack. We use ZO-BCD for *sparse wavelet transform attack*:

$$x_\star = \arg\min_x f(\mathrm{IWT}(\mathrm{WT}(\tilde{x}) + x)) \qquad (3)$$

where $\tilde{x}$ = clean image/audio signal, f = C-W loss function [6], WT & IWT are fixed wavelet transform and inverse. Switching to wavelet domain increases problem dimension *and* solution quality. We consider: **1. Image Attack.** Model: `Inception-v3`, trained on `ImageNet`. Wavelet: 'db45'. $d \approx 675,000$. **2. Audio Attack.** Model: `commandNet`, trained on `SpeechCommand` dataset. Wavelet: Morse. $d \approx 1,700,000$.

Image Attack			
Method	ASR	ℓ_2 dist	Queries
ZO-SCD	78%	57.5	2400
ZO-SGD	78%	37.9	**1590**
ZO-AdaMM	81%	28.2	1720
ZORO	90%	21.1	2950
ZO-BCD	**96%**	**13.7**	1662

Audio Attack	
Method	ASR
Alzantot *et al*, 2018	89.0%
Vadillo *et al*, 2019	70.4%
Li *et al*, 2020	96.8%
Xie *et al*, 2020	97.8%
ZO-BCD	**97.9%**

Attack **S**uccess **R**ate = fraction of signals successfully perturbed within 10,000 queries.

References

[1] A. S. Berahas, L. Cao, K. Choromanski, and K. Scheinberg, "A theoretical and empirical comparison of gradient approximations in derivative-free optimization," *arXiv preprint arXiv:1905.01332*, 2019.

[2] Y. Wang, S. Du, S. Balakrishnan, and A. Singh, "Stochastic zeroth-order optimization in high dimensions," in *International Conference on Artificial Intelligence and Statistics*, 2018.

[3] H. Cai, D. Mckenzie, W. Yin, and Z. Zhang, "Zeroth-order regularized optimization (ZORO): Approximately sparse gradients and adaptive sampling," *arXiv preprint arXiv:2003.13001*, 2020.

[4] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and computational harmonic analysis*, 2009.

[5] R. Tappenden, P. Richtárik, and J. Gondzio, "Inexact coordinate descent: Complexity and preconditioning," *Journal of Optimization Theory and Applications*, 2016.

[6] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE symposium on security and privacy*, IEEE, 2017.