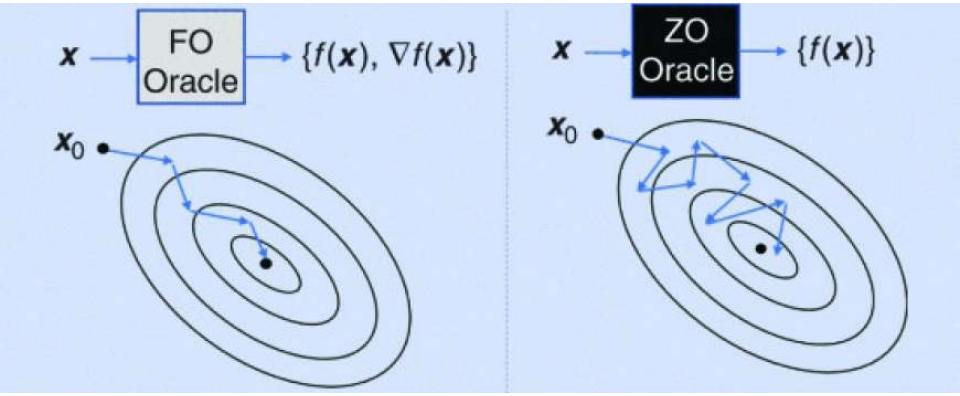
Zeroth-Order Optimization

Presented by Ruizhong Qiu

ILLINOIS

Zeroth-Order Optimization (ZOO)

- Multivariate function $f: \mathbb{R}^d \to \mathbb{R}$.
- Query: given a point $x \in \mathbb{R}^d$, outputs $f(x)$.
- Task: to minimize $f(x)$ using only queries.
- Goal: using as few queries as possible (*query complexity*).
- Basic idea: estimating $\nabla f(x)$ via queries and mimicking gradient descent.



Taxonomy of ZOO Methods

- Full-gradient methods:
  - Isotropic search: RS, RSPG , TPGE.
  - Step size searching: GLD.
  - Gradient postprocessing: ZO-signSGD, ZO-AdaMM.
- Sparse-gradient methods:
  - Masking: TruncZSGD, SZOHT, SparseSZO.
  - Sparse regression: LASSO, ZORO.
  - Block-wise estimation: ZO-BCD, GraCe.

Isotropic Search: RS & RSPG

- Basic idea (directional derivative): given $v \in \mathbb{R}^d$,

$$\partial_v f(x) := \lim_{\epsilon \to 0} \frac{f(x+\epsilon v) - f(x)}{\epsilon} = v^{\mathrm{T}} \nabla f(x).$$

  - We can gain information about $\nabla f(x)$ via queries $f(x+\epsilon v)$.

- How to design a Monte Carlo estimator of $\nabla f(x)$?
  - Consider a random exploration $v \sim \mathcal{D}$ and a small ϵ:

$$\mathbb{E}_{v \sim \mathcal{D}} \left[v \frac{f(x+\epsilon v) - f(x)}{\epsilon} \right] = \mathbb{E}_{v \sim \mathcal{D}}[v v^{\mathrm{T}}] \nabla f(x) + O(\epsilon).$$

  - We can use *isotropic* random vectors: $\mathbb{E}_{v \sim \mathcal{D}}[v v^{\mathrm{T}}] = I_d$.
  - Common choices: $\mathcal{D} = \mathcal{N}(0, I_d)$; $\mathcal{D} = \mathrm{Unif}\left(\sqrt{d}\mathbb{S}^{d-1}\right)$.
  - RS [1]: using a single $v \sim \mathcal{N}(0, I_d)$.
  - RSPG [2]: reducing variance via multiple $v_1, \ldots, v_K \sim \mathcal{N}(0, I_d)$.

[1] Nesterov et al. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 2015.

[2] Ghadimi et al. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 2016. 5.

Aside: Hessian–Vector Product

- Problem definition:
  - Given $L: \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ and $x \in \mathbb{R}^m$, $y, v \in \mathbb{R}^n$, compute
$$\nabla_x \nabla_y L(x, y) \cdot v.$$
- Hessian–vector products arise in bi-level optimization:
  - For example, [1] needs to estimate
$$\nabla_\alpha \nabla_w L_{\text{train}}(w, \alpha) \cdot \nabla_{w'} L_{\text{val}}(w', \alpha).$$
- Approximation via the directional derivative [1]:
  - Using a small ϵ,
$$\nabla_x \nabla_y L(x, y) \cdot v = \partial_v \Big|_y \nabla_x L(x, y)$$
$$= \frac{\nabla_x L(x, y + \epsilon v) - \nabla_x L(x, y)}{\epsilon} + O(\epsilon).$$

[1] Liu et al. DARTS: Differentiable architecture search. *ICLR*, 2019.

Isotropic Search: TPGE

- Two-point gradient estimator (TPGE) [1]: with two $u, v \sim \mathcal{D}$ and two small δ, ϵ with $\frac{\delta}{\epsilon} \leq \frac{1}{2}$, the gradient estimator is

$$v \frac{f(x + \delta u + \epsilon v) - f(x + \delta u)}{\epsilon}.$$

- Similarly with RS & RSPG,

$$\mathbb{E}_{u,v\sim\mathcal{D}}\left[v \frac{f(x + \delta u + \epsilon v) - f(x + \delta u)}{\epsilon}\right]$$

$$= \mathbb{E}_{u,v\sim\mathcal{D}}\left[v\left(\left(\frac{\delta}{\epsilon}u + v\right)^{\mathrm{T}} \nabla f(x) - \frac{\delta}{\epsilon}u^{\mathrm{T}}\nabla f(x)\right)\right] + O(\epsilon)$$

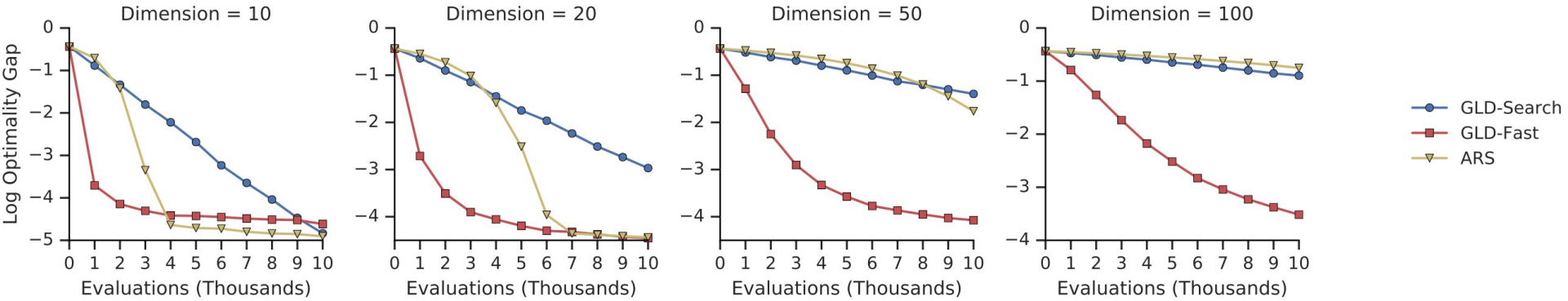
$$= \mathbb{E}_{v\sim\mathcal{D}}[vv^{\mathrm{T}}]\nabla f(x) + O(\epsilon) = \nabla f(x) + O(\epsilon).$$

[1] Duchi et al. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 2015.

Step Size Searching: GLD

- Gradientless descent (GLD) [1]:
  - Let $R > 0$ denote the search radius.
  1. Use K explorations $v_0, \ldots, v_{K-1}$, where $v_k \sim 2^{-k} R \cdot \mathcal{D}$.
  2. Let $x_{t+1} \leftarrow \underset{x \in \{x_t, x_t + v_0, \ldots, x_t + v_{K-1}\}}{\operatorname{argmin}} f(x)$.
  - For strongly convex functions, with constant probability,

$$f(x_T) - f_* \leq (f(x_{T-1}) - f_*)\left(1 - \Omega\left(\frac{1}{d}\right)\right).$$



[1] Golovin et al. Gradientless descent: High-dimensional zeroth-order optimization. *ICLR*, 2020.

Gradient Postprocessing: ZO-signSGD

- signSGD [1] is a method for distributed training of neural networks with compressed gradients:
  - Use $-\text{sign}\big(\nabla f(x)\big)$ as the descent direction.
- ZO-signSGD [2] = signSGD + RSPG.

Table 1: Iteration comparison of attacking black-box DNN on MNIST (image ID 2).

Iteration	0	40	80	120	160	200	240	280	312	356
ZO-SGD										
Classified as	1	1	1	1	1	1	1	1	4	4
Iteration	0	40	80	120	145	202	240	280	320	359
ZO-signSGD										
Classified as	1	1	1	1	4	4	4	4	4	4
Iteration	0	40	80	120	142	200	240	279	320	360
ZO-M-signSGD										
Classified as	1	1	1	1	4	4	4	4	4	4
Iteration	0	40	80	120	160	200	258	287	321	359
ZO-NES										
Classified as	1	1	1	1	1	1	4	4	4	4

[1] Bernstein et al. signSGD: Compressed optimisation for non-convex problems. *ICML*, 2018.

[2] Liu et al. signSGD via zeroth-order oracle. *ICLR*, 2019.

Gradient Postprocessing: ZO-AdaMM

- AMSGrad [1] is an adaptive momentum method aiming to address the divergence issue of Adam.
- ZO-AdaMM [2] = AMSGrad + RS.

Algorithm 1 ZO-AdaMM

Input: $\mathbf{x}_1 \in \mathcal{X}$, step sizes $\{\alpha_t\}_{t=1}^T$, $\beta_{1,t}, \beta_2 \in (0,1]$, and set $\mathbf{m}_0$, $\mathbf{v}_0$ and $\hat{\mathbf{v}}_0$

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

$\quad$ let $\hat{\mathbf{g}}_t = \hat{\nabla} f_t(\mathbf{x}_t)$ by (1), $f_t(\mathbf{x}_t) := f(\mathbf{x}_t; \boldsymbol{\xi}_t)$

$\quad \mathbf{m}_t = \beta_{1,t}\mathbf{m}_{t-1} + (1-\beta_{1,t})\hat{\mathbf{g}}_t$

$\quad \mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1-\beta_2)\hat{\mathbf{g}}_t^2$

$\quad \hat{\mathbf{v}}_t = \max(\hat{\mathbf{v}}_{t-1}, \mathbf{v}_t)$, and $\hat{\mathbf{V}}_t = \mathrm{diag}(\hat{\mathbf{v}}_t)$

$\quad \mathbf{x}_{t+1} = \Pi_{\mathcal{X}, \sqrt{\hat{\mathbf{V}}_t}}(\mathbf{x}_t - \alpha_t \hat{\mathbf{V}}_t^{-1/2}\mathbf{m}_t)$

end for

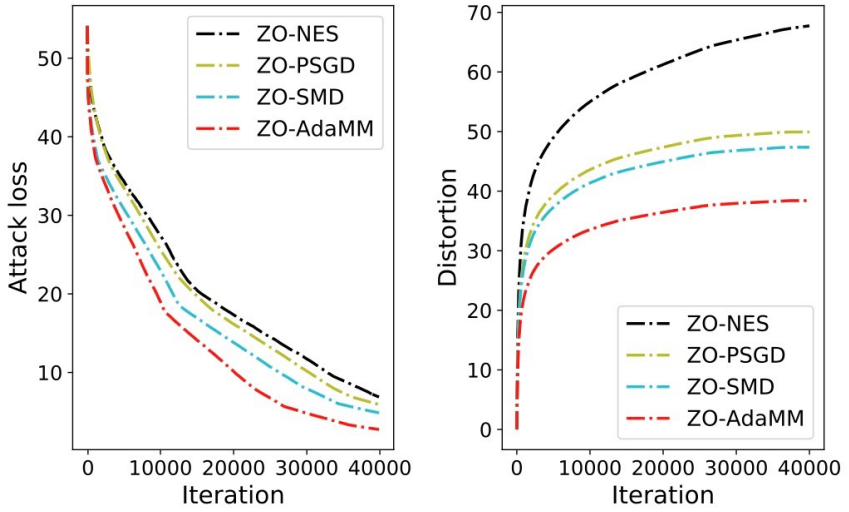


Figure 2: Attack loss and distortion of universal attack.

[1] Reddi et al. On the convergence of Adam and beyond. *ICLR*, 2018.

[2] Chen et al. ZO-AdaMM: Zeroth-order adaptive momentum method for black-box optimization. *NeurIPS*, 2019.

Sparse Gradient Estimation

- For general functions, ZOO suffers from a polynomial dependence on d in the overall query complexity [1].
  - This is undesirable for high-dimensional ZOO.
- For functions with s-sparse gradients, we can achieve a logarithmic or double logarithmic dependence on d.

Table 1. Comparison in nonconvex ZOO. (q, b: hyperparameters; $\tau := \arg\min_{t=1,\dots,T} \|\nabla f(\boldsymbol{x}_t)\|_2$; w.h.p.: with high probability.) To our best knowledge, we are the first to achieve a *double-logarithmic* dependence on d in the query complexity under weaker assumptions.

Type	Method	Queries per step	Rate of convergence
Full Gradient	RS (Ghadimi & Lan, 2012)	$O(1)$	$\mathbb{E}[\|\nabla f(\boldsymbol{x}_\tau)\|_2^2] \le O\big(\frac{\sqrt{d}}{\sqrt{T}} + \frac{d}{T}\big)$
	TPGE (Duchi et al., 2015)	$O(1)$	$\mathbb{E}[\|\nabla f(\boldsymbol{x}_\tau)\|_2^2] \le O\big(\frac{\sqrt{d}}{\sqrt{T}}\big)$
	RSPG (Ghadimi et al., 2016)	$O(q)$	$\mathbb{E}[\|\nabla f(\boldsymbol{x}_\tau)\|_2^2] \le O\big(\frac{d}{q} + \frac{d^2}{qT}\big)$
	ZO-signSGD (Liu et al., 2019)	$O(bq)$	$\mathbb{E}[\|\nabla f(\boldsymbol{x}_\tau)\|_2] \le O\big(\frac{\sqrt{d}\sqrt{q+d}}{\sqrt{bq}} + \frac{\sqrt{d}}{\sqrt{T}}\big)$
	ZO-AdaMM (Chen et al., 2019)	$O(1)$	$\mathbb{E}[\|\nabla f(\boldsymbol{x}_\tau)\|_2^2] \le O\big(\frac{d}{\sqrt{T}} + \frac{d^2}{T}\big)$
Sparse Gradient	ZORO (Cai et al., 2022)	$O\big(s \log \frac{d}{s}\big)$	$\|\nabla f(\boldsymbol{x}_\tau)\|_2^2 \le O\big(\frac{1}{T}\big)$ w.h.p.
	GraCe (ours)	$O\big(s \log\log \frac{d}{s}\big)$	$\mathbb{E}[\|\nabla f(\boldsymbol{x}_\tau)\|_2^2] \le O\big(\frac{1}{T}\big)$

[1] Alabdulkareem et al. Information-theoretic lower bounds for zero-order stochastic gradient estimation. *IEEE International Symposium on Information Theory*, 2021

Masking: TruncZSGD & SZOHT

- SZOHT [1]:
  - Designed for problems with sparse solutions.
  - Keeping only the s largest-magnitude values in the solution.

- TruncZSGD [2]:
  - Designed for functions with sparse gradients.
  - Keeping only the s largest-magnitude values in the gradient estimate.

[1] De Vazelhes et al. Zeroth-order hard-thresholding: Gradient error vs expansivity. *NeurIPS*, 2022.

[2] Balasubramanian et al. Zeroth-order nonconvex stochastic optimization: Handling constraints, high dimensionality, and saddle points. *Foundations of Computational Mathematics*, 2022.

Masking: SparseSZO

- SparseSZO [1] periodically updates a gradient mask.

Algorithm 1 Sparse SZO Optimization

INPUT: dataset $\mathcal{X} = \{\mathbf{x}_0, \cdots, \mathbf{x}_T\}$, sequence of learning rates h, sparsification interval s, number of samples k, smoothing parameter $\mu > 0$
Initialize mask: $\mathbf{m}^{(0)} = \mathbf{1}^n$ $(\bar{n}^{(0)} = n)$
Initialize weights: $\mathbf{w}^{(0)}$
for $t = 0, \cdots, T$ **do**
  if $\mathrm{mod}(t, s) = 0$ **then**
    Update mask $\mathbf{m}^{(t')} = \mathrm{get\_mask}(\mathbf{w}^{(t)})$
    if pruning **then**
      $\mathbf{w}^{(t)} = \mathbf{m}^{(t')} \odot \mathbf{w}^{(t)}$
    end if
  end if
  Observe $\mathbf{x}^{(t)} \in \mathcal{X}$
  for $j = 1, \cdots, k$ **do**
    Sample unit vector $\mathbf{u}^{(t)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
    Apply mask $\bar{\mathbf{u}}^{(t)} = \mathbf{m}^{(t')} \odot \mathbf{u}^{(t)}$ $(\bar{n}^{(t)} \ll n)$
    Compute $g_\mu^{(j)}(\mathbf{w}^{(t)})$
  end for
  Average $g_\mu(\mathbf{w}^{(t)}) = \underset{j}{\mathrm{avg}}(g_\mu^{(j)}(\mathbf{w}^{(t)}))$
  Update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - h^{(t)} g_\mu(\mathbf{w}^{(t)})$
end for
OUTPUT: sequence of $\{\mathbf{w}^{(t)}\}_{t \geq 0}$

[1] Ohta et al. Sparse perturbations for improved convergence in stochastic zeroth-order optimization. *International Conference on Machine Learning, Optimization, and Data Science (LOD)*, 2020.

[2] Frankle et al. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *ICLR*, 2019.

Sparse Regression: LASSO

- LASSO [1] is a classic method for sparse regression by relaxing the ℓ_0 constraint to an ℓ_1 constraint.
- The paper [2] proposes to use LASSO to find a sparse gradient estimate g:
  - Given $v_1, \ldots, v_K \sim \mathrm{Unif}(\{\pm 1\})$ and $\theta > 0$, solve:

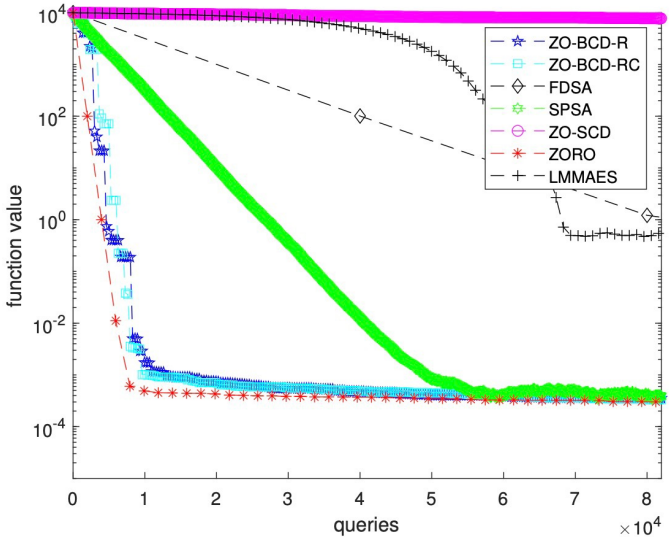
$$\min_{g \in \mathbb{R}^d} \quad \sum_{k=1}^{K} \left(v_k^{\mathrm{T}} g - \frac{f(x + \epsilon v_k) - f(x)}{\epsilon} \right)^2,$$
$$\text{s.t.} \quad \|g\|_1 \leq \theta.$$

[1] Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 1996.

[2] Wang et al. Stochastic zeroth-order optimization in high dimensions. *AISTATS*, 2018.

Sparse Regression: ZORO & ZO-BCD

- ZORO [1] proposes to use CoSaMP [2] for sparse regression instead of LASSO.
  - ZORO needs $O\left(s \log \frac{d}{s}\right)$ queries.
  - ZORO works for noisy queries as well.



(a) Sparse Quadric.

Algorithm 1: CoSaMP Recovery Algorithm

Input: Sampling matrix $\mathbf{\Phi}$, noisy sample vector $\boldsymbol{u}$, sparsity level s
Output: An s-sparse approximation $\boldsymbol{a}$ of the target signal

$\boldsymbol{a}^0 \leftarrow \mathbf{0}, \boldsymbol{v} \leftarrow \boldsymbol{u}, k \leftarrow 0$ { Initialization }

repeat
$k \leftarrow k + 1$
$\boldsymbol{y} \leftarrow \mathbf{\Phi}^* \boldsymbol{v}$ { Form signal proxy }
$\mathrm{W} \leftarrow \mathrm{supp}(\boldsymbol{y}_{2s})$ { Identify large components }
$T \leftarrow \mathrm{W} \cup \mathrm{supp}(\boldsymbol{a}^{k-1})$ { Merge supports }
$\boldsymbol{b}|_T \leftarrow \mathbf{\Phi}_T^{\dagger} u$ { Signal estimation }
$\boldsymbol{b}|_{T^c} \leftarrow \mathbf{0}$
$\boldsymbol{a}^k \leftarrow \boldsymbol{b}_s$ { Prune approximation }
$\boldsymbol{v} \leftarrow \boldsymbol{u} - \mathbf{\Phi} \boldsymbol{a}^k$ { Update current samples }

until halting criterion *true*

- ZO-BCD [3] proposes to divide d dimensions into blocks and apply ZORO to each block.

[1] Cai et al. Zeroth-order regularized optimization (ZORO): Approximately sparse gradients and adaptive sampling. *SIAM Journal on Optimization*, 2022.

[2] Needell et al. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Communications of the ACM*, 2010.

[3] Cai et al. A zeroth-order block coordinate descent algorithm for huge-scale black-box optimization. *ICML*, 2021.

Block-Wise Estimation: GraCe

- Previous methods aims to find a gradient estimate g such that $\|g - \nabla f(x)\|$ is small.
  - This is too strong and may waste queries.
  - To achieve $O\left(\frac{1}{T}\right)$ rate of convergence in nonconvex ZOO, it suffices to ensure:
    - (i) $g^{\mathrm{T}}\nabla f(x) \geq \Omega(\|\nabla f(x)\|^2) - O(\epsilon)$;
    - (ii) $\|g\| \leq \|\nabla f(x)\| + O(\epsilon)$.
- Gradient compressed sensing (GraCe):
  - Employs the IPW algorithm [1] from compressed sensing;
  - Improves the query complexity to $O\left(s \log\log \frac{d}{s}\right)$.

[1] Indyk et al. On the power of adaptivity in sparse recovery. *FOCS*, 2011.

Motivating Case: 1-Sparse & High-SNR

- Suppose a candidate set $S \subseteq [d]$ in which only one dimension $j \in S$ has a large gradient.
- If the signal-to-noise (SNR) ratio $\frac{|\nabla_j f(x)|}{\|\nabla_{S\setminus\{j\}} f(x)\|}$ is sufficiently large,
- then with perturbations $u_i' := \epsilon \cdot 1_{[i\in S]}, \quad v_i' := \epsilon \cdot i \cdot 1_{[i\in S]}, \quad i \in [d]$
- we have

$$\frac{f(\boldsymbol{x}+\boldsymbol{v}')-f(\boldsymbol{x})}{f(\boldsymbol{x}+\boldsymbol{u}')-f(\boldsymbol{x})} \approx \frac{\sum_{i\in[d]} v_i' \cdot \nabla_i f(\boldsymbol{x})}{\sum_{i\in[d]} u_i' \cdot \nabla_i f(\boldsymbol{x})}$$

$$= \frac{\sum_{i\in S} \epsilon \cdot i \cdot \nabla_i f(\boldsymbol{x})}{\sum_{i\in S} \epsilon \cdot \nabla_i f(\boldsymbol{x})} \approx \frac{\epsilon \cdot j \cdot \nabla_j f(\boldsymbol{x})}{\epsilon \cdot \nabla_j f(\boldsymbol{x})} = j.$$

- Thus, we can use $O(1)$ queries to find j by rounding $\frac{f(x+v')-f(x)}{f(x+u')-f(x)}$ to the nearest integer.
- Example: If $f(x) = 0.1x_1 + x_2$, then $\frac{f(0+v')-f(0)}{f(0+u')-f(0)} = \frac{2.1\epsilon}{1.1\epsilon} \approx 2$.

Base Case: 1-Sparse Gradient

- In general, the SNR $\frac{|\nabla_j f(x)|}{\|\nabla_{S\setminus\{j\}} f(x)\|}$ might not be large enough.
  - To increase the SNR, instead of finding j directly, we aim to find a small subgroup $S' \subseteq S$ that contains j.
- Given a division parameter D, we randomly divide S into $B := \left\lceil \frac{|S|}{D} \right\rceil$ blocks and label the blocks $1, \ldots, B$.
  - For each dimension $i \in S$, let h_i denote the label of the block that i belongs to, and let $\sigma_i \sim \mathrm{Unif}(\{\pm 1\})$.
  - Use perturbations: $u_i := \epsilon \cdot \sigma_i \cdot 1_{[i \in S]},\ v_i := \epsilon \cdot \sigma_i \cdot h_i \cdot 1_{[i \in S]},\ i \in [d]$.
- We can show that the SNR increases rapidly.
  - For the r-th iteration, we are allowed to use $D = \exp\left(\Omega(1.5^r)\right)$.
  - This implies that $O(\log\log|S|)$ iterations suffice to find j.

General Case: s-Sparse Gradient

- Dividing the d dimensions into $\Theta(s)$ groups so that every group has at most one large-gradient dimension w.h.p.

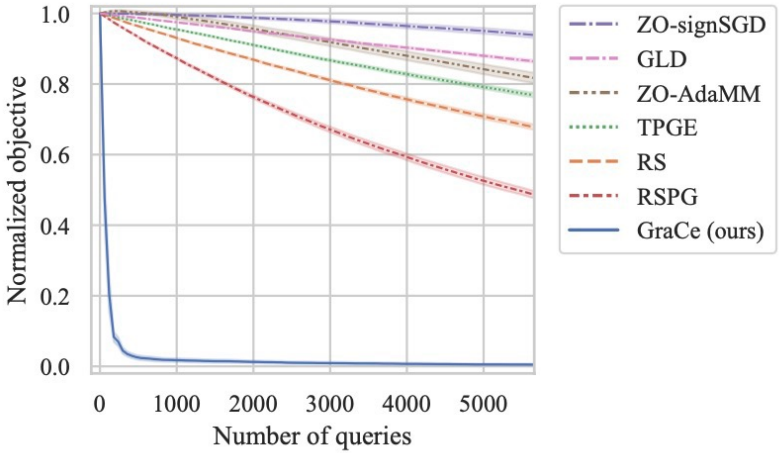
Algorithm 1 Gradient Compressed Sensing (GraCe)

Input: point $\boldsymbol{x}$; sparsity s; finite difference ϵ; number m of repeats; group size n; division schedule $\{D_r\}_{r\geq 1}$

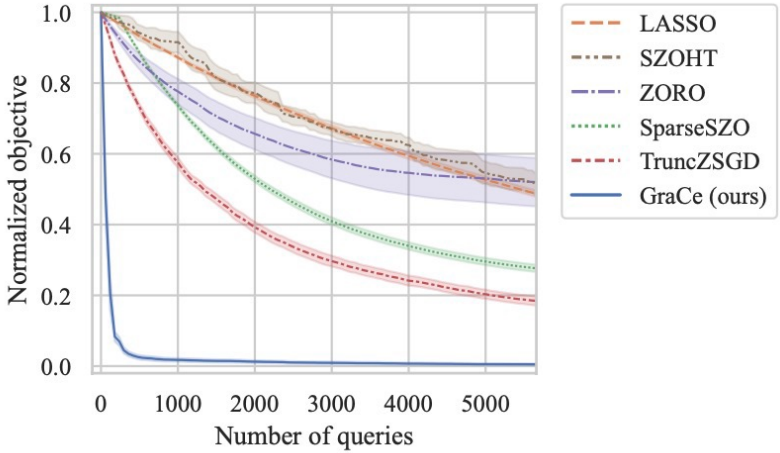
Output: the gradient estimate $\boldsymbol{g} \in \mathbb{R}^d$

candidate set $J \leftarrow \varnothing$
for $l = 1$ **to** m **do**
  random permutation $\omega \sim \mathsf{Unif}(\mathcal{P}_{[d]\to[d]})$
  for $k = 1$ **to** $\lceil d/n \rceil$ **do**
    candidate group $S \leftarrow \left\{i \in [d] : \left\lceil \frac{\omega(i)}{n} \right\rceil = k\right\}$
    iteration number $r \leftarrow 0$
    repeat
      iteration number $r \leftarrow r + 1$
      random permutation $\varpi \sim \mathsf{Unif}(\mathcal{P}_{S\to[|S|]})$
      block size $B \leftarrow \left\lceil \frac{|S|}{D_r} \right\rceil$
      perturbations $\boldsymbol{u} \leftarrow \mathbf{0}_d$, $\boldsymbol{v} \leftarrow \mathbf{0}_d$
      for $i \in S$ **do**
        random sign $\sigma_i \sim \mathsf{Unif}(\{\pm 1\})$
        block label $h_i \leftarrow \left\lceil \frac{\varpi(i)}{B} \right\rceil$
        perturbations $u_i \leftarrow \epsilon \cdot \sigma_i$, $v_i \leftarrow \epsilon \cdot \sigma_i \cdot h_i$
      end for
      target $q \leftarrow \mathrm{round}\left(\frac{f(\boldsymbol{x}+\boldsymbol{v})-f(\boldsymbol{x})}{f(\boldsymbol{x}+\boldsymbol{u})-f(\boldsymbol{x})}\right)$ via 2 queries
      candidate group $S \leftarrow \{i \in S : h_i = q\}$
    until $|S| \leq 2$
    candidate set $J \leftarrow J \cup S$
  end for
end for
gradient estimate $\boldsymbol{g} \leftarrow \mathbf{0}_d$
for $j \in J$ **do**
  finite difference $g_j \leftarrow \frac{f(\boldsymbol{x}+\epsilon \boldsymbol{e}_j)-f(\boldsymbol{x})}{\epsilon}$ via 1 query
end for
return gradient estimate $\boldsymbol{g}$

Comparison of ZOO Methods

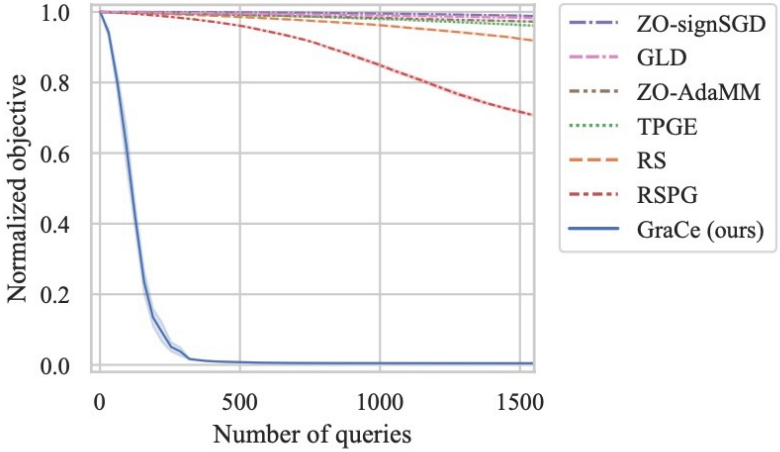


(a) Comparison with full-gradient methods.

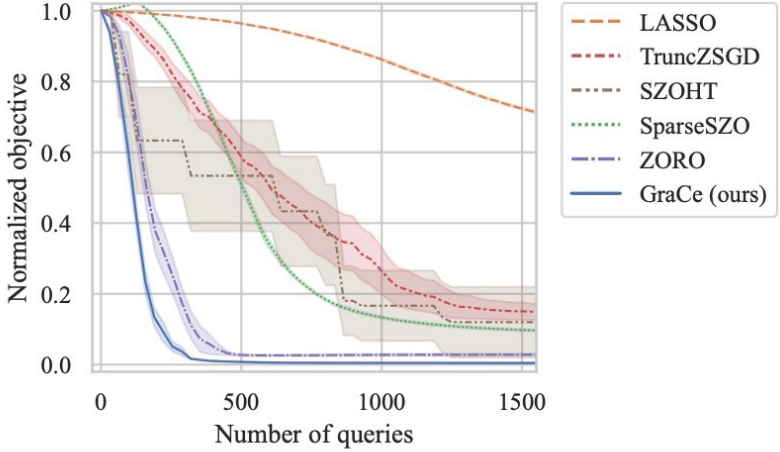


(b) Comparison with sparse-gradient methods.

Figure 1. Convergence plots for DISTANCE (mean ± s.e.).



(a) Comparison with full-gradient methods.



(b) Comparison with sparse-gradient methods.

Figure 2. Convergence plots for MAGNITUDE (mean ± s.e.).

Thanks!

ILLINOIS