On Iterative Hard-Thresholding: Gradient Estimation and Non-Convex Projections
Thesis Plan

by

William de Vazelhes

Candidacy thesis submitted to the
Deanship of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the Ph.D. degree in
Machine Learning

Department of Machine Learning
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

Supervisor: Dr. Bin Gu
Professor, Dept. of ML,
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)

External Examiner: Dr. Xiaotong Yuan
Professor,
Nanjing University of Information Science and Technology (NUIST)

Internal Member: Dr. Chih-Jen Lin
Professor, Dept. of ML,
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)

Internal Member: Dr. Karthik Nandakumar
Professor, Dept. of ML,
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)

Internal Member: Dr. Zhiqiang Xu
Professor, Dept. of ML,
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Learning sparse models is an important topic in machine learning, in particular with the ever increasing dimensionality of data. In this candidacy thesis, we extend the existing work on sparse optimization from the existing literature to various settings.

First, calculating first-order gradients of the objective function in real-world scenarios can be costly or even impossible. In such cases, zeroth-order (ZO) gradients can be a useful alternative. However, the compatibility of ZO gradients with the hard-thresholding operator - a dominant technique for solving ℓ_0 constrained optimization - is still unresolved. To address this challenge, we propose a new algorithm called stochastic zeroth-order gradient hard-thresholding (SZOHT). The algorithm uses a novel random support sampling method to estimate ZO gradients and has a general ZO gradient estimator. We provide convergence analysis for SZOHT under standard assumptions and identify a conflict between the expansivity of the hard-thresholding operator and the deviation of ZO estimators. We also establish a theoretical minimum value for the number of random directions in ZO gradients. Our experiments show that SZOHT's query complexity is independent or weakly dependent on the dimensionality under different settings. Furthermore, we demonstrate the effectiveness of SZOHT in solving portfolio optimization problems and black-box adversarial attacks.

Second, in the special case of sparse linear regression, recently, iterative regularization methods have emerged as a promising fast approach for sparse recovery. Indeed, they can achieve recovery in one pass through early stopping, rather than the tedious grid-search used in the traditional methods. Since directly using the ℓ_0 norm is NP-hard, most of those iterative methods are based on the ℓ_1 norm which requires restrictive applicability conditions and could fail in many cases. Therefore, achieving sparse recovery with iterative regularization methods under a wider range of conditions has yet to be further explored. To address this issue, we propose a novel iterative regularization algorithm, IRKSN, based on the k-support norm regularizer rather than the ℓ_1 norm. We provide conditions for sparse recovery with IRKSN, and compare them with traditional conditions for recovery with ℓ_1 norm regularizers. Additionally, we give an early stopping bound on the model error of IRKSN with explicit constants, achieving the standard linear rate for sparse recovery. Finally, we illustrate the usefulness of our algorithm on several applications, including sparse prediction on gene microarray data, and we show that IRKSN achieves comparable prediction to state of the art methods, with an efficient early stopping rule.

Finally, we present the future directions that we will explore, to provide an even fuller and unified picture of our results.

Acknowledgements

I would like to thank my advisor (Dr. Bin Gu) for his patience and guidance.

Table of Contents

Chapter 1

Preliminaries

Throughout this paper, we denote by $\|\boldsymbol{x}\|$ the Euclidean norm for a vector $\boldsymbol{x} \in \mathbb{R}^d$, by $\|\boldsymbol{x}\|_\infty$ the maximum absolute component of that vector, and by $\|\boldsymbol{x}\|_0$ the ℓ_0 norm (which is not a proper norm). For simplicity, we denote $f_{\boldsymbol{\xi}}(\cdot) := f(\cdot, \boldsymbol{\xi})$. We call $\boldsymbol{u}_F$ (resp. $\nabla_F f(\boldsymbol{x})$) the vector which sets all coordinates $i \notin F$ of $\boldsymbol{u}$ (resp. $\nabla f(\boldsymbol{x})$) to 0. We also denote by $\boldsymbol{x}^*$ the solution of problem (2.1) defined in the introduction, for some target sparsity k^* which could be smaller than k. Additionally, f denotes any convex function. To derive our result, we will need the following assumptions on f.

Assumption 1 ((ν_s, s)-RSC, [43, 50, 55, 63, 67, 80, 96])**.** *f is said to be ν_s restricted strongly convex with sparsity parameter s if it is differentiable, and there exist a generic constant ν_s such that for all $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^d$ with $\|\boldsymbol{x} - \boldsymbol{y}\|_0 \leq s$:*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\nu_s}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2$$

Assumption 2 ((L_s, s)-RSS, [67, 80])**.** *For almost any $\boldsymbol{\xi}$, $f_{\boldsymbol{\xi}}$ is said to be L_s restricted smooth with sparsity level s, if it is differentiable, and there exist a generic constant L_s such that for all $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^d$ with $\|\boldsymbol{x} - \boldsymbol{y}\|_0 \leq s$:*

$$\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}) - \nabla f_{\boldsymbol{\xi}}(\boldsymbol{y})\| \leq L_s \|\boldsymbol{x} - \boldsymbol{y}\|$$

Assumption 3 (σ^2-FGN [39], Assumption 2.3 (Finite Gradient Noise))**.** *f is said to have σ-finite gradient noise if for almost any $\boldsymbol{\xi}$, $f_{\boldsymbol{\xi}}$ is differentiable and the gradient noise $\sigma = \sigma(f, \boldsymbol{\xi})$ defined below is finite:*

$$\sigma^2 = \mathbb{E}_{\boldsymbol{\xi}}[\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|_\infty^2]$$

Remark 1. *Even though the original version of [39] uses the ℓ_2 norm, we use the ℓ_∞ norm here, in order to give more insightful results in terms of k and d, as is done classically in ℓ_0 optimization, similarly to [101]. We also note that in [39], $\boldsymbol{x}^*$ denotes an unconstrained minimum when in our case it denotes the constrained minimum for some sparsity k^*.*

We will also need the more usual smoothness assumption:

Assumption 4 (L-smooth)**.** *For almost any $\boldsymbol{\xi}$, $f_{\boldsymbol{\xi}}$ is said to be L smooth, if it is differentiable, and for all $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^d$:*

$$\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}) - \nabla f_{\boldsymbol{\xi}}(\boldsymbol{y})\| \leq L \|\boldsymbol{x} - \boldsymbol{y}\|$$

Chapter 2

Literature Review

2.1 Sparse Zeroth-Order Optimization

ℓ_0 constrained optimization is prevalent in machine learning, particularly for high-dimensional problems, because it is a fundamental approach to achieve sparse learning. In addition to improving the memory, computational and environmental footprint of the models, these sparse constraints help reduce overfitting and obtain consistent statistical estimation [18, 64, 74, 97]. We formulate the problem as follows:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \{f(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{\xi}} f(\boldsymbol{x}, \boldsymbol{\xi})\}, \quad \text{s.t.} \quad \|\boldsymbol{x}\|_0 \leq k \tag{2.1}$$

where $f(\cdot, \boldsymbol{\xi}) : \mathbb{R}^d \to \mathbb{R}$ is a differentiable function and $\boldsymbol{\xi}$ is a noise term, for instance related to an underlying finite sum structure in f, of the form: $\mathbb{E}_{\boldsymbol{\xi}} f(\boldsymbol{x}, \boldsymbol{\xi}) = \frac{1}{n}\sum_{i=1}^n f_i(\boldsymbol{x})$. Hard-thresholding gradient algorithm [43, 67, 96] is a dominant technique to solve this problem. It generally consists in alternating between a gradient step, and a hard-thresholding operation which only keeps the k-largest components (in absolute value) of the current iterate. The advantage of hard-thresholding over its convex relaxations ([83, 87]) is that it can often attain similar precision, but is more computationally efficient, since it can directly ensure a desired sparsity level instead of tuning an ℓ_1 penalty or constraint. The only expensive computation in hard-thresholding is the hard-thresholding step itself, which requires finding the top k elements of the current iterate. Hard-thresholding was originally developed in its full gradient form [43], but has been later on extended to the stochastic setting by [67], which developed a stochastic gradient descent (SGD) version of hard thresholding (StoIHT), and further more with [101], [80] and [50], which used variance reduction technique to improve upon StoIHT.

However, the first-order gradients used in the above methods may be either unavailable or expensive to calculate in a lot of real-world problems. For example, in certain graphical modeling tasks [90], obtaining the gradient of the objective function is computationally hard. Even worse, in some settings, the gradient is inaccessible by nature, for instance in bandit problems [79], black-box adversarial attacks [25, 26, 86], or reinforcement learning [27, 56, 77]. To tackle those problems, ZO optimization methods have been developed [66]. Those

Table 2.1: Complexity of sparsity-enforcing algorithms. We give the query complexity for a precision ε, up to the system error (see section 3.1.2). For first-order algorithms (FO), we give it in terms of number of first order oracle calls (#IFO), that is, calls to $\nabla f(x, \boldsymbol{\xi})$, and for ZO algorithms, in terms of calls of $f(\boldsymbol{\xi}, \cdot)$. Here κ denotes the condition number $\frac{L}{\nu}$, with L is the smoothness (or RSS) constant and ν is the strong-convexity (or RSC) constant.

Type	Name	Assumptions	#IZO/#IFO	#HT ops.
FO/ℓ_0	StoIHT [67]	RSS, RSC	$\mathcal{O}(\kappa \log(\frac{1}{\varepsilon}))$	$\mathcal{O}(\kappa \log(\frac{1}{\varepsilon}))$
ZO/ℓ_1	RSPGF [34]	smooth[3]	$\mathcal{O}(\frac{d}{\varepsilon^2})$	—
ZO/ℓ_1	ZSCG[2][5]	convex, smooth	$\mathcal{O}(\frac{d}{\varepsilon^2})$	—
ZO/ℓ_1	ZORO [20]	s-sparse gradient, weakly sparse hessian, smooth[3], RSC_{bis}[1]	$\mathcal{O}(s \log(d) \log(\frac{1}{\varepsilon}))$	—
ZO/ℓ_0	**SZOHT**	RSS, RSC	$\mathcal{O}((k + \frac{d}{s_2})\kappa^2 \log(\frac{1}{\varepsilon}))$	$\mathcal{O}(\kappa^2 \log(\frac{1}{\varepsilon}))$
ZO/ℓ_0	**SZOHT**	smooth, RSC	$\mathcal{O}(k\kappa^2 \log(\frac{1}{\varepsilon}))$	$\mathcal{O}(\kappa^2 \log(\frac{1}{\varepsilon}))$

[1] The definition of Restricted Strong Convexity from [20] is different from ours and that of [67], hence the bis subscript.
[2] We refer to the modified version of ZSCG (Algorithm 3 in [5]).
[3] RSPGF and ZORO minimize $f(x) + \lambda\|x\|_1$: only f needs to be smooth.

methods usually replace the inaccessible gradient by its finite difference approximation which can be computed only from function evaluations, following the idea that for a differentiable function $f : \mathbb{R} \to \mathbb{R}$, we have: $f'(x) = \lim_{h\to 0} \frac{f(x+h)-f(x)}{h}$. Later on, ZO methods have been adapted to solve problems constrained to a convex set, such as the ℓ_1 convex relaxation of problem (2.1). To that end, [34], and [20] introduce proximal ZO algorithms, [54] introduce a ZO projected gradient algorithm and [5] introduce a ZO conditional gradient [49] algorithm. We provide a review of those results in Table 2.1. As can be seen from the table, their query complexity is high (linear in d), except [20] that has a complexity of $\mathcal{O}(s \log(d) \log(\frac{1}{\varepsilon}))$, but assumes that gradients are sparse. In addition, those methods must introduce a hyperparameter λ (the strength of the ℓ_1 penalty) or R (the radius of the ℓ_1 ball), which need to be tuned to find which value ensures the right sparsity level. Therefore, it would be interesting to use the hard-thresholding techniques described in the previous paragraph, instead of those convex relaxations.

Unfortunately, ZO hard-thresholding gradient algorithms have not been exploited formally. Even more, whether ZO gradients can work with the hard-thresholding operator is still an unknown problem. Although there was one related algorithm proposed recently by [5], they did not target ℓ_0 constrained optimization problem and importantly have strong assumptions in their convergence analysis. Indeed, they assume that the gradients, as well as the solution of the unconstrained problem, are s-sparse: $\|\nabla f(\boldsymbol{x})\|_0 \leq s$ and $\|\boldsymbol{x}^*\|_0 \leq s^* \approx s$, where $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$. In addition, it was recently shown by [20] that they must in fact assume that the support of the gradient is fixed for all $\boldsymbol{x} \in \mathcal{X}$, for their convergence result to hold, which is a hard limitation, since that amounts to say that the function f depends on s fixed variables.

2.2 Linear Sparse Recovery

In many cases, it is interesting to study a special form of the problem described above, which is when the function f is the mean squared error in linear regression. This is called sparse recovery, and is ubiquitous in machine learning and signal processing, with applications ranging from single pixel camera, to MRI, or radar[1]. In particular, with the ever-increasing amount of information, real-life datasets often contain much more features than samples: this is for instance the case in DNA microarray datasets [38], text data [48], or image data such as fMRI [11], where the number of features is generally much larger than the number of samples. In these high-dimensional settings, finding a linear model is under-specified, and therefore, one often needs to leverage additional assumptions about the true model, such as sparsity, to recover it. Usually, the problem is formulated as follows: we seek to recover a sparse vector $\boldsymbol{w}^* \in \mathbb{R}$ from its noisy linear measurements

$$\boldsymbol{y}^\delta = \boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\epsilon}$$

Here, $\boldsymbol{y}^\delta$ is a noisy measurement vector, i.e. a noisy version of the true target vector $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^*$, $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_d] \in \mathbb{R}^{n \times d}$ is a measurement matrix, also called design matrix, $\boldsymbol{\epsilon} \in \mathbb{R}^n$ is some bounded noise ($\|\boldsymbol{\epsilon}\|_2 \leq \delta$, with $\delta \in \mathbb{R}_+$), and $\boldsymbol{w}^*$ is the unknown k-sparse vector, i.e. containing only k non-zero components, that we wish to estimate with a vector $\hat{\boldsymbol{w}}$ obtained by running some sparse recovery algorithm on observations $\boldsymbol{y}^\delta$ and $\boldsymbol{X}$. Unfortunately, this problem is NP-hard in general, even in the noiseless setting [62].

Due to the NP-hard nature of sparse recovery, existing methods are known to suffer either from restrictive (or even unknown) applicability conditions, or high computational cost. Amongst those methods, a first group of methods can achieve an exact sparsity k of the estimate $\hat{\boldsymbol{w}}$: Iterative Hard Thresholding [14] returns an estimate $\hat{\boldsymbol{w}}$ which recovers $\boldsymbol{w}^*$ up to an error $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\| \leq O(\delta)$, if the design matrix $\boldsymbol{X}$ satisfies some Restricted Isometry Property (RIP) [14]. However, as mentioned in [43], this condition is very restrictive, and does not hold in most high-dimensional problems. Greedy methods, such as Orthogonal Matching Pursuit (OMP) [85], also can return an exactly k-sparse vector, and bounds on the recovery of a (generalized version of) OMP, of the type $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\| \leq O(\delta)$, can be found for instance in [92], under some RIP condition.

A second set of methods for sparse recovery solve the following penalized problem:

$$(P) : \min_{\boldsymbol{w}} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}^\delta\|^2 + \lambda R(\boldsymbol{w})$$

Where R is a regularizer, such as the ℓ_1 norm as is done in the Lasso method [84], and λ is a penalty parameter that needs to be tuned. For a given λ, (P) is usually solved through a convex optimization algorithm, and returns a solution $\hat{\boldsymbol{w}}$ of (P), as an estimate of $\boldsymbol{w}^*$. Amongst those, one of the most important algorithms for sparse recovery, the Lasso [83], has been proven in [40] to give a bound $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\| \leq O(\delta)$ under the so-called *source conditions* (described in Condition 4.3 from [40]) which can be equivalently rewritten as

[1] An introduction to this topic, as well as an extensive review of its applications can be found in [31] and [94].

the fact that: $\boldsymbol{X}_S$ is injective, and $\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^{\dagger} \boldsymbol{x}_\ell, \text{sgn}(\boldsymbol{w}_S^*) \rangle| < 1$, with $S \subset [d]$ the support of $\boldsymbol{w}^*$, i.e. indices of all the nonzero elements of $\boldsymbol{w}^*$, and $\bar{S}$ its complement in $[d]$, $\boldsymbol{X}_S^{\dagger}$ the Moore-Penrose pseudo-inverse of $\boldsymbol{X}_S$, and $\boldsymbol{x}_\ell$ is the ℓ-th column of $\boldsymbol{X}$ (we detail this point further in our paper). Importantly, [40] show that such conditions are even necessary to achieve a linear rate for the Lasso with a priori choice of the penalty. Following the Lasso, the ElasticNet was later developed to solve the problem of a design matrix with possibly high correlations. However, although some conditions for *statistical consistency* exist for the ElasticNet [45], to the best of our knowledge, there is no model error bound (and conditions thereof) for recovery with ElasticNet. Finally, the k-support norm regularization has also been used successfully as a regularizer [3], with even better empirical results than the ElasticNet, but no explicit error bounds on model error (and the conditions thereof) currently exists: indeed, their work was mostly focused on *sparse prediction* and not *sparse recovery*. Efficient solvers have later been derived for the Lasso using for instance coordinate descent and its variants [29] [13]. However, even with efficient solvers, these penalized methods need to tune the parameter λ, which is very costly.

Recently, iterative regularization methods have emerged as a promising fast approach because they can achieve sparse recovery in one pass through early stopping, rather than the tedious grid-search used in traditional methods. They solve the following problem

$$\begin{aligned} (I): \quad & \min_{\boldsymbol{w}} R(\boldsymbol{w}) \\ \text{s.t.} \quad & \boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}^{\delta} \end{aligned}$$

An iterative algorithm is used to solve it, and returns some $\hat{\boldsymbol{w}}$ to estimate $\boldsymbol{w}^*$. Importantly, $\hat{\boldsymbol{w}}$ is obtained by stopping the algorithm before convergence, also called *early stopping*. One of the first amongst these methods, SRDI [68], achieves a rate of $\|\hat{\boldsymbol{w}} - \boldsymbol{w}\| \leq O(\sigma \sqrt{\frac{k \log d}{n}})$ with high probability, assuming $\epsilon \underset{i.i.d.}{\sim} \mathcal{N}(0, \sigma)$, and two conditions: (1) $\exists \gamma \in (0, 1] : \ \boldsymbol{X}_S^{\top} \boldsymbol{X}_S \geq n\gamma I_{d,d}$ (Restricted Strong Convexity) and (2) $\exists \eta \in (0, 1) : \ \|\boldsymbol{X}_{\bar{S}} \boldsymbol{X}_S^{\dagger}\|_{\infty} \leq 1 - \eta$. IROSR [89] uses an iterative regularization scheme that is based on a reparameterization of the problem (I). They prove a high probability model consistency bound of $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\| \leq O(\sigma \sqrt{\frac{k \log d}{n}})$, assuming the $((k+1, c)$-RIP for some constant $c(k, \boldsymbol{w}^*, \boldsymbol{X}, \epsilon)$. Similar to their work is [99]: under similar conditions, they also obtain a similar rate. Finally, [60] provide bounds of the form $\|\hat{\boldsymbol{w}} - \boldsymbol{w}\| \leq O(\delta)$, under the same *source conditions* as in [40].

However, most of those iterative methods are based on the ℓ_1 norm which requires restrictive applicability conditions and could fail in many cases. Indeed, in those cases, the conditions for recovery with the methods described above (e.g. RIP, or the *source condition* we discussed) do not hold anymore. For instance, in gene array data [102], it is known that many columns of the design matrix are correlated, and that RIP does not hold. It is therefore crucial to come up with algorithms for which recovery is provably possible under different conditions.

Table 2.2: Comparison of the existing algorithms for sparse recovery in the literature, including their conditions on $\boldsymbol{X}$ and $\boldsymbol{w}^*$ required for recovery. T is the number of iterations each algorithm is ran for, and Λ is the number of values of λ that need to be tried out (for penalized methods). [(1)] assuming $\epsilon \underset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$. [(2)]: Additionally, $\boldsymbol{X}_S$ should be injective.

Method	Condition on $\boldsymbol{X}$	Bound on $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|$	Complexity
IHT [14]	RIP	$O(\delta)$	$O(T)$
Lasso [83]	$\max_{\ell \in \bar{S}} \lvert\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \operatorname{sgn}(\boldsymbol{w}_S^*)\rangle\rvert < 1$ [(2)]	$O(\delta)$	$O(\Lambda T)$
ElasticNet [102]	-	-	$O(\Lambda T)$
KSN pen. [3]	-	-	$O(\Lambda T)$
OMP [85]	RIP	$O(\delta)$	$O(k)$
SRDI [68]	$\begin{cases} \exists \gamma \in (0,1]: \ \boldsymbol{X}_S^\top \boldsymbol{X}_S \geq n\gamma I_{d,d} \\ \exists \eta \in (0,1): \ \|\boldsymbol{X}_{\bar{S}} \boldsymbol{X}_S^\dagger\|_\infty \leq 1 - \eta \end{cases}$	$O(\sigma\sqrt{\frac{k \log d}{n}})$ [(1)]	$O(T)$
IROSR [89]	RIP	$O(\sigma\sqrt{\frac{k \log d}{n}})$ [(1)]	$O(T)$
IRCR [60]	$\max_{\ell \in \bar{S}} \lvert\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \operatorname{sgn}(\boldsymbol{w}_S^*)\rangle\rvert < 1$ [(2)]	$O(\delta)$	$O(T)$
IRKSN (ours)	$\max_{\ell \in \bar{S}} \lvert\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^*\rangle\rvert < \min_{j \in S} \lvert\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_j, \boldsymbol{w}_S^*\rangle\rvert$	$O(\delta)$	$O(T)$

Chapter 3

Research progress

In this chapter, we provide our research progress: in the first section, we present our result regarding zeroth-order hard thresholding. Then, in the second section, we present our result regarding our k-support norm iterative regularization algorithm.

3.1 Zeroth-Order Hard-Thresholding

As described in the Literature Review, there are currently no algorithm that study global convergence of Iterative Hard Thresholding when the gradients are approximated by zeroth-order.

To fill this gap, we propose a novel stochastic zeroth-order gradient hard-thresholding (SZOHT) algorithm. Specifically, we propose a dimension friendly ZO gradient estimator powered by a novel random support sampling technique, and then embed it into the standard hard-thresholding operator.

We then provide the convergence and complexity analysis of SZOHT under the standard assumptions of sparse learning, which are restricted strong smoothness (RSS), and restricted strong convexity (RSC) [67, 80], to retain generality, therefore providing a positive answer to the question of whether ZO gradients can work with the hard-thresholding operator. Crucial to our analysis is to provide carefully tuned requirements on the parameters q (the number of random directions used to estimate the gradient, further defined in Section 3.1.1) and k. Finally, we illustrate the utility of our method on a portfolio optimization problem as well as black-box adversarial attacks, by showing that our method can achieve competitive performance in comparison to state of the art methods for sparsity-enforcing zeroth-order algorithm described in Table 2.1, such as [5, 20, 34].

Importantly, we also show that in the smooth case, the query complexity of SZOHT is independent of the dimensionality, which is significantly different to the dimensionality dependent results for most existing ZO algorithms. Indeed, it is known from [44] that the worst case query complexity of ZO optimization over the class $\mathcal{F}_{\nu,L}$ of ν-strongly convex and L-smooth functions defined over a convex set $\mathcal{X}$ is linear in d. Our work is thus in line with

other works achieving dimension-insensitive query complexity in zeroth-order optimization such as [5, 19, 20, 20, 36, 44, 52, 81, 93], but contrary to those, instead of making further assumptions (i.e. restricting the class $\mathcal{F}_{\nu,L}$ to a smaller class), we bypass the impossibility result by replacing the convex feasible set $\mathcal{X}$ by a *non-convex* set (the ℓ_0 ball), which is how we can avoid making stringent assumptions on the class of functions f.

Contributions. We summarize our main contributions as follows:

1. We propose a new algorithm SZOHT that is, up to our knowledge, the first zeroth-order sparsity constrained algorithm that is dimension independent under the smoothness assumption, without assuming any gradient sparsity.
2. We reveal an interesting conflict between the error from zeroth-order estimates and the hard-thresholding operator, which results in a minimal value for the number of random directions q that is necessary to ensure at each iteration.
3. We also provide the convergence analysis of our algorithm in the more general RSS setting, providing, up to our knowledge, the first zeroth-order algorithm that can work with the usual assumptions of RSS/RSC from the hard-thresholding literature.

3.1.1 Algorithm

Random support Zeroth-Order estimate

In this section, we describe our zeroth-order gradient estimator. It is basically composed of a random support sampling step, followed by a random direction with uniform smoothing on the sphere supported by this support. We also use the technique of averaging our estimator over q dimensions, as described in [53]. More formally, our gradient estimator is described below:

$$\hat{\nabla} f_{\boldsymbol{\xi}}(\boldsymbol{x}) = \frac{d}{q\mu} \sum_{i=1}^{q} \left(f_{\boldsymbol{\xi}}(\boldsymbol{x} + \mu \boldsymbol{u}_i) - f_{\boldsymbol{\xi}}(\boldsymbol{x}) \right) \boldsymbol{u}_i \tag{3.1}$$

where each random direction $\boldsymbol{u}_i$ is a unit vector sampled uniformly from the set $\mathcal{S}_{s_2}^d := \{\boldsymbol{u} \in \mathbb{R}^d : \|\boldsymbol{u}\|_0 \leq s_2, \|\boldsymbol{u}\| = 1\}$. We can obtain such vectors $\boldsymbol{u}$ by sampling first a random support S (i.e. a set of coordinates) of size s_2 from $[d]$, (denoted as $S \sim \mathcal{U}(\binom{[d]}{s_2})$ in Algorithm 1) and then by sampling a random unit vector $\boldsymbol{u}$ supported on that support S, that is, uniformly sampled from the set $\mathcal{S}_S^d := \{\boldsymbol{u} \in \mathbb{R}^d : \boldsymbol{u}_{[d]-S} = \boldsymbol{0}, \|\boldsymbol{u}\| = 1\}$, (denoted as $\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}_S^d)$ in algorithm 1). The original uniform smoothing technique on the sphere is described in more detail in [32]. However, in our case, the sphere along which we sample is restricted to a random support of size s_2. Our general estimator, through the setting of the variable s_2, can take several forms, which are similar to pre-existing gradient estimators from the literature described below:

- If $s_2 = d$, $\hat{\nabla} f_{\boldsymbol{\xi}}(\boldsymbol{x})$ is the *usual vanilla estimator with uniform smoothing on the sphere* [32].

- If $1 \leq s_2 \leq d$, our estimator is similar to the Random Block-Coordinate gradient estimator from [51], except that the blocks are not fixed at initialization but chosen randomly, and that we use a uniform smoothing with forward difference on the given block instead of a coordinate-wise estimation with central difference. This random support technique allows us to give a convergence analysis under the classical assumptions of the hard-thresholding literature (see Remark 3), and to deal with huge scale optimization, when sampling uniformly from a unit d-sphere is costly [19, 20]: in the distributed setting for instance, each worker would just need to sample an s_2-sparse random vector, and only the centralized server would materialize the full gradient approximation containing up to qs_2 non-zero entries.

Error Bounds of the Zeroth-Order Estimator. We now derive error bounds on the gradient estimator, that will be useful in the convergence rate proof, except that we consider *only the restriction to some support* F (that is, we consider a subset of components of the gradient/estimator). Indeed, proofs in the hard-thresholding literature (see for instance [96]), are usually written only on that support. That is the key idea which explains how the dimensionality dependence is reduced when doing SZOHT compared to vanilla ZO optimization. We give more insight on the shape of the original distribution of gradient estimators, and the distribution of their projection onto a hyperplane F in Figure A.1 in Appendix A.5. We can observe that even if the original gradient estimator is poor, in the projected space, the estimation error is reduced, which we quantify in the proposition below.

Proposition 1. *(Proof in Appendix A.3.3) Let us consider any support $F \subset [d]$ of size s ($|F| = s$). For the Z0 gradient estimator in* (3.1)*, with q random directions, and random supports of size s_2, and assuming that each $f_{\boldsymbol{\xi}}$ is (L_{s_2}, s_2)-RSS, we have, with $\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x})$ denoting the hard thresholding of the gradient $\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x})$ on F (that is, we set all coordinates not in F to 0):*

- $\|\mathbb{E}\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x})\|^2 \leq \varepsilon_\mu \mu^2$
- $\mathbb{E}\|\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x})\|^2 \leq \varepsilon_F \|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x})\|^2 + \varepsilon_{F^c} \|\nabla_{F^c} f_{\boldsymbol{\xi}}(\boldsymbol{x})\|^2 + \varepsilon_{abs}\mu^2$
- $\mathbb{E}\|\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x})\|^2 \leq 2(\varepsilon_F + 1)\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x})\|^2 + 2\varepsilon_{F^c} \|\nabla_{F^c} f_{\boldsymbol{\xi}}(\boldsymbol{x})\|^2 + 2\varepsilon_{abs}\mu^2$

$$\text{with} \quad \varepsilon_\mu = L_{s_2}^2 sd, \quad \varepsilon_F = \frac{2d}{q(s_2+2)}\left(\frac{(s-1)(s_2-1)}{d-1} + 3\right) + 2,$$
$$\varepsilon_{F^c} = \frac{2d}{q(s_2+2)}\left(\frac{s(s_2-1)}{d-1}\right) \quad \text{and} \quad \varepsilon_{abs} = \frac{2dL_{s_2}^2 s s_2}{q}\left(\frac{(s-1)(s_2-1)}{d-1} + 1\right) + L_{s_2}^2 sd \tag{3.2}$$

SZOHT Algorithm

We now present our full algorithm to optimize problem 2.1, which we name SZOHT (Stochastic Zeroth-Order Hard Thresholding). Each iteration of our algorithm is composed

of two steps: (i) the gradient estimation step, and (ii) the hard thresholding step, where the gradient estimation step is the one described in the section above, and the hard-thresholding is described in more detail in the following paragraph. We give the full formal description of our algorithm in Algorithm 1.

In the hard thresholding step, we only keep the k largest (in magnitude) components of the current iterate x^t. This ensures that all our iterates (including the last one) are k-sparse. This hard-thresholding operator has been studied for instance in [80], and possesses several interesting properties. Firstly, it can be seen as a projection on the ℓ_0 ball. Second, importantly, it is not non-expansive, contrary to other operators like the soft-thresholding operator [80]. That expansivity plays an important role in the analysis of our algorithm, as we will see later.

Compared to previous works, our algorithm can be seen as a variant of Stochastic Hard Thresholding (StoIHT from [67]) , where we replaced the true gradient of $f_{\boldsymbol{\xi}}$ by the estimator $\hat{\nabla} f_{\boldsymbol{\xi}}(\boldsymbol{x})$. It is also very close to Algorithm 5 from [5] (Truncated-ZSGD), with just a different zeroth-order gradient estimator: we use a uniform smoothing, random-block estimator, instead of their gaussian smoothing, full support vanilla estimator. This allows us to deal with very large dimensionalities, in the order of millions, similarly to [19]. Furthermore, as described in the Introduction, contrary to [5], we provide the analysis of our algorithm without any gradient sparsity assumption.

The key challenge arising in our analysis is described in Figure 3.1: the hard-thresholding operator being expansive [80], each approximate gradient step must approach the solution enough to stay close to it even after hard-thresholding. Therefore, it is *a priori* unclear whether the zeroth-order estimate can be accurate enough to guarantee the convergence of SZOHT. Hopefully, as we will see in the next section, we can indeed ensure convergence, as long as we carefully choose the value of q.

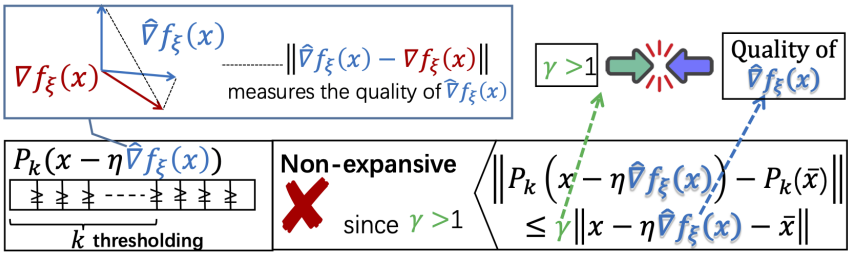


Figure 3.1: Conflict between the hard-thresholding operator and the zeroth-order estimate.

3.1.2 Convergence analysis

In this section, we provide the convergence analysis of SZOHT, using the assumptions from the Preliminaries chapter, and discuss an interesting property of the combination of the zeroth-order gradient estimate and the hard-thresholding operator, providing a positive answer to the question from the previous section.

Theorem 1. *(Proof in Appendix A.4.1) Assume that that each $f_{\boldsymbol{\xi}}$ is $(L_{s'}, s' := \max(s_2, s))$-RSS, and that f is (ν_s, s)-RSC and σ-FGN, with $s = 2k + k^* \leq d$, with $\frac{d-k^*}{2} \geq k \geq$*

Algorithm 1: Stochastic Zeroth-Order Hard-Thresholding (SZOHT)

Initialization: *Learning rate η, maximum number of iterations T, size of the random directions support s_2, number of random directions q, number of coordinates to keep at each iteration $k = \mathcal{O}(\kappa^4 k^*)$, initial point $\boldsymbol{x}^{(0)}$ with $\|\boldsymbol{x}^{(0)}\|_0 \leq k^*$ (typically $\boldsymbol{x}^{(0)} = 0$), .*

Output : $\boldsymbol{x}^T$.

for $t = 1, ..., T$ **do**

  Sample $\boldsymbol{\xi}$ (for instance sample a train sample i)

  for $i = 1, ..., q$ **do**

    Sample a random support $S \sim \mathcal{U}(\binom{[d]}{s_2})$

    Sample a random direction $\boldsymbol{u}_i$ from the unit sphere supported on S: $\boldsymbol{u}_i \sim \mathcal{U}\left(\mathcal{S}_S^d\right)$

    Compute $\hat{\nabla} f_{\boldsymbol{\xi}}(\boldsymbol{x}^{t-1}; \boldsymbol{u}_i) = \frac{d}{\mu}\left(f_{\boldsymbol{\xi}}(\boldsymbol{x} + \mu \boldsymbol{u}_i) - f_{\boldsymbol{\xi}}(\boldsymbol{x})\right) \boldsymbol{u}_i$;

  end

  Compute $\hat{\nabla} f_{\boldsymbol{\xi}}(\boldsymbol{x}^{t-1}) = \frac{1}{q}\sum_{i=1}^{q} \hat{\nabla} f_{\boldsymbol{\xi}}(\boldsymbol{x}^{t-1}; \boldsymbol{u}_j)$

  Compute $\tilde{\boldsymbol{x}}^t = \boldsymbol{x}^{t-1} - \eta \hat{\nabla} f_{\boldsymbol{\xi}}(\boldsymbol{x}^{t-1})$;

  Compute $\boldsymbol{x}^t = \tilde{\boldsymbol{x}}_k^t$ as the truncation of $\tilde{\boldsymbol{x}}^t$ with top k entries preserved;

end

$\rho^2 k^*/(1-\rho^2)^2$, with ρ defined as below. Suppose that we run SZOHT with random supports of size s_2, q random directions, a learning rate of $\eta = \frac{\nu_s}{(4\varepsilon_F+1)L_{s'}^2}$, and k coordinates kept at each iterations. Then, we have a geometric convergence rate, of the following form, with $\boldsymbol{x}^{(t)}$ denoting the t-iterate of SZOHT:

$$\mathbb{E}\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\| \leq (\gamma\rho)^t \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\| + \left(\frac{\gamma a}{1-\gamma\rho}\right)\sigma + \left(\frac{\gamma b}{1-\gamma\rho}\right)\mu$$

$$\text{with} \quad a = \eta\left(\sqrt{(4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k)} + \sqrt{s}\right), \; b = \left(\frac{\sqrt{\varepsilon_\mu}}{L_{s'}} + \eta\sqrt{2\varepsilon_{abs}}\right), \tag{3.3}$$

$$\rho^2 = 1 - \frac{\nu_s^2}{(4\varepsilon_F+1)L_{s'}^2}, \; \text{and } \gamma = \sqrt{1 + \left(k^*/k + \sqrt{(4 + k^*/k)\, k^*/k}\right)/2}$$

and $\varepsilon_F, \varepsilon_{abs}$, *and* ε_μ *are defined in* (3.2).

Remark 2 (System error)**.** *The format of our result is similar to the ones in [96] and [67], in that it contains a linear convergence term, and a system error which depends on the expected norm of the gradient at $\boldsymbol{x}^*$ (through the variable σ). We note that if f has a k^*-sparse unconstrained minimizer, which could happen in sparse reconstruction, or with overparameterized deep networks (see for instance [73, Assumption (2)]), then we would have $\|\nabla f(\boldsymbol{x}^*)\| = 0$, and hence that part of the system error would vanish. In addition to that usual system error, we also have here another system error, which depends on the smoothing radius μ, due to the error from the ZO estimate.*

Remark 3 (Generality)**.** *If we take $s_2 \leq s$, the first assumption of Theorem 1 becomes the requirement that $f_{\boldsymbol{\xi}}$ is (L_s, s)-RSS. Therefore, SZOHT as well as the theorem above*

provides, up to our knowledge, the first algorithm that can work in the usual setting of hard-thresholding algorithms (that is, (L_s, s)-RSS and (ν_s, s)-RSC [67, 80]), as well as its convergence rate.

Interplay between hard-thresholding and zeroth-order error Importantly, contrary to previous works in ZO optimization, q must be chosen carefully here, due to our specific setting combining ZO and hard-thresholding. Indeed, as described in [80], the hard-thresholding operator is not non-expansive (contrary to projection onto the ℓ_1 ball) so it can drive the iterates away from the solution. Therefore, enough descent must be made by the (approximate) gradient step to get close enough to the solution, and it is therefore crucial to limit errors in gradient estimation. This problem arises with any kind of gradient errors: for instance with SGD errors [67, 101], it is generally dealt with either by ensuring some conditions on the function f [67], forming bigger batches of samples [101], and/or considering a larger number of components k kept in hard-thresholding (to make the hard-thresholding less expansive). In our work, similarly to [101], we deal with this problem by relaxing k and sampling more directions $\boldsymbol{u}_i$ (which is the ZO equivalent to taking bigger batch-size in SGD). However, there is an additional effect that happens in our case, specific to ZO estimation: as described in Proposition 1, the quality of our estimator *also depends on* k. Therefore, it may be hard to make the algorithm converge only by considering larger k: *higher k means less expansivity (which helps convergence), but worse gradient estimate (which harms convergence).* We further illustrate this conflict between the non-expansiveness of hard-thresholding (quantified by the parameter γ [80]), and the error from the zeroth-order estimate, in Figure 3.1.
Therefore, it is even more crucial to tune precisely our remaining degree of freedom at hand which is q. More precisely, we provide below the minimal value q, needed to ensure the descent of our algorithm, that is, to ensure $\rho\gamma < 1$ for some $k^* \geq 1$:

Lemma 1 (First condition on q, Proof in Appendix A.4.2)**.** *Let $k^* \in \{1, ..., d\}$. In the case $s_2 > 1$, if $q \geq q_{min}$, with q_{min} defined below, then there exist $k \in \mathbb{N}$ such that $k \geq k^* \frac{\rho^2}{(1-\rho^2)^2}$, which in turn ensures $\rho\gamma \leq 1$. If $s_2 = 1$, no condition on q is needed to ensure such an existence. However, in the case $s_2 = 1$, $q = 1$ does not ensure the second necessary condition in the following Remark 4, as detailed in the proof of the present Lemma. Therefore, a minimal value of q is always necessary, to ensure existence of k such that Theorem 1 applies.*

$$q_{min} = \frac{16d(s_2-1)k^*\kappa^2}{(s_2+2)(d-1)}\left[18\kappa - 1 + 2\sqrt{9\kappa(9\kappa-1) + \frac{1}{2} - \frac{1}{2k^*} + \frac{3}{2}\frac{d-1}{s_2-1}}\right]$$

Remark 4 (Second condition on q)**.** *As mentioned in the Lemma 1 above, there is also a second condition on q necessary to ensure for Theorem 1 to be valid, which is simply to ensure that the smallest valid k above (i.e. from the first condition), is smaller or equal to d (since we cannot keep more components than we have).*

Weak/non dependence on dimensionality of the query complexity.

In this section, we provide Corollaries 1 and 2, following from Theorem 1, which give an example of q that is valid for both these conditions above, allowing to converge (that is, to

obtain $\gamma\rho < 1$ in Theorem 1), and that achieves weak dimensionality dependence in the case of RSS, and complete dimension independence in the case of smoothness.

Corollary 1 (RSS $f_{\boldsymbol{\xi}}$, proof in Appendix A.4.3)**.** *Assume that that almost all $f_{\boldsymbol{\xi}}$ are $(L_{s'}, s' := \max(s_2, s))$-RSS, and that f is (ν_s, s)-RSC and σ-FGN, with $s = 2k + k^* \leq d$, with $\frac{d-k^*}{2} \geq k \geq (86\kappa^4 - 12\kappa^2)k^*$ (with $\kappa := \frac{L_{s'}}{\nu_s}$) . Suppose that we run SZOHT with random support of size s_2, a learning rate of $\eta = \frac{\nu_s}{13L_{s'}^2}$, with k coordinates kept at each iterations by the hard-thresholding, and with $q \geq 2s + 6\frac{d}{s_2}$. Then, we have a geometric convergence rate, of the following form, with $\boldsymbol{x}^{(t)}$ denoting the t-iterate of SZOHT:*

$$\mathbb{E}\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\| \leq (\gamma\rho)^t \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\| + \left(\frac{\gamma a}{1-\gamma\rho}\right)\sigma + \left(\frac{\gamma b}{1-\gamma\rho}\right)\mu$$

with a, b and γ are defined in (3.3)*, and $\rho = \sqrt{1 - \frac{2}{13\kappa^2}}$. Therefore, the query complexity (QC) to ensure that $\mathbb{E}\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\| \leq \varepsilon + \left(\frac{\gamma a}{1-\gamma\rho}\right)\sigma + \left(\frac{\gamma b}{1-\gamma\rho}\right)\mu$ is $\mathcal{O}(\kappa^2(k + \frac{d}{s_2})\log(\frac{1}{\varepsilon}))$.*

We now turn to the case where the functions $f_{\boldsymbol{\xi}}$ are smooth. The key result in that case is that we can have a query complexity independent of the dimension d, which is, up to our knowledge, the first result of such kind for sparse zeroth-order optimization without assuming any gradient sparsity.

Corollary 2 (Smooth $f_{\boldsymbol{\xi}}$, proof in Appendix A.4.4))**.** *Assume that, in addition to the conditions from Corollary 1 above, almost all $f_{\boldsymbol{\xi}}$ are L-smooth, with $\frac{d-k^*}{2} \geq k \geq (86\kappa^4 - 12\kappa^2)k^*$ (with $\kappa := \frac{L}{\nu_s}$), and take $q \geq 2(s+2)$, and $s_2 = d$ (that is, no random support sampling). Then, we have a geometric convergence rate, of the following form, with $\boldsymbol{x}^{(t)}$ denoting the t-iterate of SZOHT:*

$$\mathbb{E}\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\| \leq (\gamma\rho)^t \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\| + \left(\frac{\gamma a}{1-\gamma\rho}\right)\sigma + \left(\frac{\gamma b}{1-\gamma\rho}\right)\mu$$

Therefore, the QC to ensure that $\mathbb{E}\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\| \leq \varepsilon + \left(\frac{\gamma a}{1-\gamma\rho}\right)\sigma + \left(\frac{\gamma b}{1-\gamma\rho}\right)\mu$ is $\mathcal{O}(\kappa^2 k \log(\frac{1}{\varepsilon}))$.

Additionally, our convergence rate highlights an interesting connection between the geometry of f (defined by the condition number $\kappa = L_{s'}/\nu_s$), and the number of random directions that we need to take at each iteration: if the problem is ill-conditioned, that is κ is high, then we need a bigger k. This result is standard in the ℓ_0 litterature (see for instance [96]). But specifically, in our ZO case, it also impacts the query complexity: since the projected gradient is harder to approximate when the dimension k of the projection is larger, q needs to grow too, resulting in higher query complexity. We believe this is an interesting result for the sparse zeroth-order optimization community: it reveals that the query complexity may in fact depend on some notion of intrinsic dimension to the problem, related to both the sparsity of the iterates k, and the geometry of the function f for a given s_2 (through the restricted condition number κ), rather than the dimension of the original space d as in previous works like [34].

3.1.3 Experiments

Sensitivity analysis

We first conduct a sensitivity parameter analysis on a toy example, to highlight the importance of the choice of q, as discussed in Section 3.1.2. We fix a target sparsity $k^* = 5$, choose $k = 74k^*$, and consider a sparse quadric function $f : \mathbb{R}^{5000} \to \mathbb{R}$, with: $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{a} \odot (\boldsymbol{x} - \boldsymbol{b})\|^2$ ($\odot$ denotes the elementwise product), with $\boldsymbol{a}_i = 1$ if $i \geq d - s$ and 0 otherwise (to ensure f is s-RSC and smooth, with $\nu_s = L = 1$), and $\boldsymbol{b}_i = \frac{i}{100d}$ for all $i \in [d - 70k^*]$ and 0 for all $d - 70k^* \leq i \leq d$ (we make such a choice in order to have $\|\nabla f(\boldsymbol{x}^*)\|$ small enough). We choose η as in Theorem 1: $\eta = \frac{1}{(4\varepsilon_F + 1)}$ with ε_F defined in Proposition 1 in terms of s and d (we take $s_2 = d$), $\mu = 1e - 4$, and present our results in Figure 3.2, for six values of q. We can observe on Figure 3.2(b) that the smaller the q, the less $f(\boldsymbol{x})$ can descend. Interestingly, we can also see on Figure 3.2(a) that for $q = 1$ and 20, $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\|$ diverges: we can indeed compute that $\rho\gamma > 1$ for those q, which explains the divergence, from Theorem 1.

Baselines

We compare our SZOHT algorithms with state of the art zeroth-order algorithms that can deal with sparsity constraints, that appear in Table 2.1:

- **ZSCG** [5] is a Frank-Wolfe ZO algorithm, for which we consider an ℓ_1 ball constraint.
- **RSPGF** [34] is a proximal ZO algorithm, for which we consider an ℓ_1 penalty.
- **ZORO** [20] is a proximal ZO algorithm, that makes use of sparsity of gradients assumptions, using a sparse reconstruction algorithm at each iteration to reconstruct the gradient from a few measurements. Similarly, as for ZSCG, we consider an ℓ_1 penalty.

In all the applications below, we will tune the sparsity k of SZOHT, the penalty of RSPGF and ZORO, and the radius of the constraint of ZSCG, such that all algorithms attain a similar converged objective value, for fair comparison.

Applications

We compare the algorithms above on two tasks: a sparse asset risk management task from [23], and an adversarial attack task [25] with a sparsity constraint.

Sparse asset risk management We consider the portfolio management task and dataset from [23], similarly to [20]. We have a given portfolio of d assets, with each asset i giving an expected return $\boldsymbol{m}_i$, and with a global covariance matrix of the return of assets denoted as $\boldsymbol{C}$. The cost function we minimize is the portfolio risk: $\frac{\boldsymbol{x}^T \boldsymbol{C} \boldsymbol{x}}{2(\sum_{i=1}^d \boldsymbol{x}_i)^2}$, where $\boldsymbol{x}$ is a vector

where each component $\boldsymbol{x}_i$ denotes how much is invested in each asset, and we require to minimize it under a constraint of minimal return r: $\frac{\sum_{i=1}^d \boldsymbol{m}_i \boldsymbol{x}_i}{\sum_{i=1}^d \boldsymbol{x}_i}$. We enforce that constraint using the Lagrangian form below. Finally, we add a sparsity constraint, to restrict the investments to only k assets. Therefore, we obtain the cost function below:

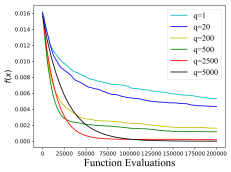
$$\min_{x \in \mathbb{R}^d} \frac{\boldsymbol{x}^\top \boldsymbol{C} \boldsymbol{x}}{2\left(\sum_{i=1}^d \boldsymbol{x}_i\right)^2} + \lambda \left(\min\left\{ \frac{\sum_{i=1}^d \boldsymbol{m}_i \boldsymbol{x}_i}{\sum_{i=1}^d \boldsymbol{x}_i} - r, 0 \right\} \right)^2 \quad \text{s.t.} \quad \|\boldsymbol{x}\|_0 \leq k$$

We use three datasets: port3, port4 and port5 from the OR-library [9], of respective dimensions $d = 89; 98; 225$. We keep r and λ the same for the 4 algorithms: $r = 0.1$, $\lambda = 10$ (for port3 and port4); and $r = 1e-3$, $\lambda = 1e-3$ for port5. For SZOHT, we set $k = 10$, $s_2 = 10$, $q = 10$, and $(\mu, \eta) = (0.015, 0.015)$ for port4, and $(\mu, \eta) = (0.1, 1)$ for port5 (μ and η are both obtained by grid search over the interval $[10^{-3}, 10^3]$). For all other algorithms, we got the optimal hyper-parameters through grid search. We present our results in Figure 3.3.

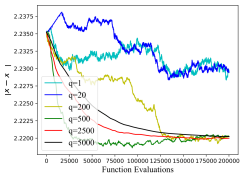
Few pixels adversarial attacks We consider the problem of adversarial attacks with a sparse constraint. Our goal is to minimize $\min_\delta f(\boldsymbol{x} + \boldsymbol{\delta})$ such that $\|\boldsymbol{\delta}\|_0 \leq k$, where f is the Carlini-Wagner cost function [25], that is computed from the outputs of a pre-trained model on the corresponding dataset. We consider three different datasets for the attacks: MNIST, CIFAR, and Imagenet, of dimension respectively $d = 784; 3072; 268203$. All algorithms are initialized with $\boldsymbol{\delta} = \boldsymbol{0}$. We set the hyperparameters of SZOHT as follows: MNIST: $k = 20$, $s_2 = 100$, $q = 100$, $\mu = 0.3$, $\eta = 1$; CIFAR: $k = 60$, $s_2 = 100$, $q = 1000$, $\mu = 1e-3$, $\eta = 0.01$; ImageNet: $k = 100000$, $s_2 = 1000$, $q = 100$, $\mu = 0.01$, $\eta = 0.015$. We present our results in Figure 3.4. All experiments are conducted in the workstation with four NVIDIA RTX A6000 GPUs, and take about one day to run.

Results and Discussion

We can observe from Figures 3.3 and 3.4 that the performance of SZOHT is comparable or better than the other algorithms. This can be explained by the fact that SZOHT has a linear convergence, but the query complexity of ZSCG and RSPGF is in $\mathcal{O}(1/T)$. We can also notice that RSPGF is faster than ZSCG, which is natural since proximal algorithms are faster than Frank-Wolfe algorithms (indeed, in case of possible strong-convexity, vanilla Frank-Wolfe algorithms maintain a $\mathcal{O}(1/T)$ rate [33], when proximal algorithms get a linear rate [10, Theorem 10.29]). Finally, it appears that the convergence of ZORO is sometimes slower, particularly at the early stage of training, which may come from the fact that ZORO assumes sparse gradients, which is not necessarily verified in real-world use cases like the ones we consider; in those cases where the gradient is not sparse, it is possible that the sparse gradient reconstruction step of ZORO does not work well. This motivates even further the need to consider algorithms able to work without those assumptions, such as SZOHT.

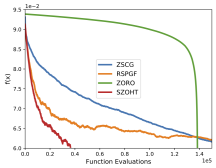


(a) $f(\boldsymbol{x})$

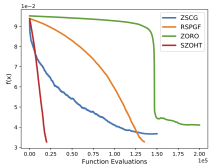


(b) $\|\boldsymbol{x} - \boldsymbol{x}^*\|$

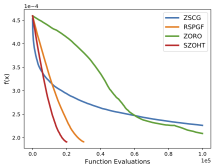
Figure 3.2: Sensitivity analysis



(a) port3

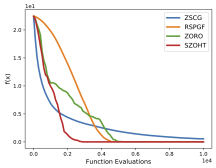


(b) port4

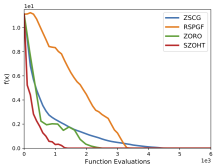


(c) port5

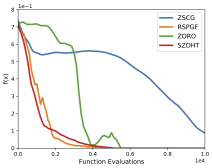
Figure 3.3: $f(\boldsymbol{x})$ vs. # queries (asset management)



(a) MNIST



(b) CIFAR



(c) Imagenet

Figure 3.4: $f(\boldsymbol{x})$ vs. # queries (adversarial attack)

3.2 Sparse recovery: iterative regularization with k-support norm

As described in the Literature Review above, current iterative regularization methods for sparse recovery are based on the ℓ_1 norm regularization, and only under very specific conditions is one guaranteed to retreive the same solution as when using an ℓ_0 regularization. To address this issue, we propose a novel iterative regularization algorithm, IRKSN, based on the k-support norm regularizer rather than the ℓ_1 norm. That norm was first introduced in [3], as a way to improve upon the ElasticNet for sparse prediction. More precisely, we plug the k-support norm regularizer, for which there exist efficient proximal computations [3, 59], into the primal-dual framework for iterative regularization described in [57].

We then provide the conditions for sparse recovery with IRKSN, and discuss on a simple example how they compare with traditional conditions for recovery with ℓ_1 norm regularizers. More precisely, we give the following conditions for recovery with IRKSN: $\max_{i\in\bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_i, \boldsymbol{w}_S^* \rangle| < \min_{j\in S} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_j, \boldsymbol{w}_S^* \rangle|$, and we discuss why this specific condition include cases that are not included in usual conditions for recovery with traditional methods based on the ℓ_1 norm (see Figure **??**). Since those types of conditions are usually slightly opaque to interpret, we do as is common in the literature (such as in [45, 102]), namely, we discuss and compare those solutions with the help of an illustrative example. We also give an early stopping bound on the model error of IRKSN with explicit constants, achieving the standard linear rate for sparse recovery.

Finally, we illustrate the usefulness of our algorithm on several applications, including sparse prediction on gene microarray data, and show that IRKSN achieves comparable prediction to state-of-the-art methods, with an efficient early stopping rule.

Contributions. We summarize the main contributions of our paper as follows:

1. We introduce a new algorithm, IRKSN, which allows recovery of the true sparse vector under conditions for which the *source conditions* for recovery with ℓ_1 norm do not hold. We discuss the difference between those conditions on a detailed example.
2. We give an early stopping bound on the model error of IRKSN with explicit constants, achieving the standard linear rate for sparse recovery.
3. We compare IRKSN with other algorithms on various datasets, including a gene array dataset, and show that it achieves comparable performance with state-of-the-art algorithms, with a fast selection of the regularization strength done by early stopping.

Preliminaries

In this section, we recall a few definitions and notations used in the rest of the paper. We denote all vectors and matrices variables in bold font. For any matrix $\boldsymbol{M} \in \mathbb{R}^{n\times d}$, $\boldsymbol{m}_i$ denotes its i-th column for $i \in \mathbb{N}$, $\boldsymbol{M}^\top$ its transpose, $\boldsymbol{M}^\dagger$ its Moore-Penrose pseudo-inverse [37], and $\|\boldsymbol{M}\|$ its nuclear norm. For a vector $\boldsymbol{w} \in \mathbb{R}^d$, $\mathrm{supp}(\boldsymbol{w})$ denotes its support $\boldsymbol{w}$, that is, the coordinates of the non-zero components of $\boldsymbol{w}$, w_i denotes its i-th component, $|w|_i^\downarrow$ denotes its i-th top absolute value, and $\|\boldsymbol{w}\|$ denotes its ℓ_2 norm. More generally $\|\boldsymbol{w}\|_p$ denotes its ℓ_p norm for $p \in [1, +\infty)$, and $\|\boldsymbol{w}\|_0$ denotes its number of non-zero components. $\boldsymbol{w}_S \in \mathbb{R}^k$ denotes its restriction to a support S of size k, that is, the sub-vector of size k formed by extracting only the components w_i with $i \in S$. $\mathrm{sgn}(\boldsymbol{w})$ denotes the vector of its signs (with the additional convention that if $\boldsymbol{w}_i = 0$, $\mathrm{sgn}(\boldsymbol{w})_i = 0$).

We start by introducing the k-support norm, which is the main component of our algorithm. The k-support norm was first introduced in [3], as the tightest convex relaxation of the intersection of the ℓ_2 ball and the ℓ_0 ball. It was later generalized to the matrix case [58, 59], as well as successfully applied to several problems, including for instance fMRI [11, 35]. We give below its formal definition, with the following variational formula from [3]:

Definition 1 ([3], Definition 2.1)**.** *Let $k \in \{1, ..., d\}$. The k-support norm $\|\cdot\|_k^{sp}$ is defined, for every $\boldsymbol{w} \in \mathbb{R}^d$, as:*

$$\|\boldsymbol{w}\|_k^{sp} = \min\{\sum_{I\in\mathcal{G}_k} \|\boldsymbol{y}_I\| : supp(\boldsymbol{y}_I) \subseteq I, \sum_{I\in\mathcal{G}_k} \boldsymbol{y}_I = \boldsymbol{w}\} \tag{3.4}$$

where $\mathcal{G}_k$ denotes the set of all subsets of $\{1, ..., d\}$ of cardinality at most k.

In other words, the k-support norm is equal to the smallest sum of the norms of some k-sparse *atoms* (the $\boldsymbol{y}_I$ above) that constitute $\boldsymbol{w}$: as studied in [24], the k-support norm is indeed a so-called *atomic norm*. One can also see from this definition that the k-support norm interpolates between the ℓ_1 norm (which it is equal to if $k = 1$) and the ℓ_2 norm (which it is equal to if $k = d$). As discussed in [3], another interpretation of the k-support norm is that it is equivalent to the Group-Lasso penalty with overlaps [42], when the set of overlapping groups is all possible subsets of $\{1, ..., d\}$ of cardinality at most k. An alternative definition that allows an explicit computation of the norm, can also be given as:

Definition 2 ([3], Proposition 2.1)**.**

$$\|\boldsymbol{w}\|_k^{sp} = \left(\sum_{i=1}^{k-r-1} (|w|_i^{\downarrow})^2 + \frac{1}{r+1} \left(\sum_{i=k-r}^{d} |w|_i^{\downarrow} \right)^2 \right)^{\frac{1}{2}} ,$$

where, letting $|x|_0^{\downarrow}$ *denote* $+\infty$*,* r *is the unique integer in* $\{0, \ldots, k-1\}$ *satisfying*

$$|w|_{k-r-1}^{\downarrow} \; > \; \frac{1}{r+1} \sum_{i=k-r}^{d} |w|_i^{\downarrow} \; \geq \; |w|_{k-r}^{\downarrow} \, .$$

Finally, we introduce the proximal operator [70] below, that will be used in our algorithm:

Definition 3 (Proximal operator, [70])**.** *The proximal operator for a function* $h : \mathbb{R}^d \to \mathbb{R}$ *is defined as:*

$$prox_h(\boldsymbol{z}) = \arg\min_{\boldsymbol{w}} h(\boldsymbol{w}) + \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{z}\|_2^2$$

A closed form for the proximal operator of the squared k-support norm was first given in [3], and more efficient computations have been found e.g. in [59], which we will use in IRKSN, as described in the Appendix.

3.2.1 Algorithm

In this section, we describe the IRKSN (Iterative Regularization with k-Support Norm) algorithm. It is based on the general accelerated algorithm from [57], in which we plug a regularization function based on the k-support norm. More precisely, [57] describe a general regularization algorithm for model recovery based on a primal-dual method, and an early stopping rule. As they do, we will solve the following problem approximately (i.e. with early stopping):

$$\begin{aligned} (I_{ks}): \qquad & \min_{\boldsymbol{w}} R(\boldsymbol{w}) \\ \text{s.t.} \quad & \boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}^{\delta} \end{aligned}$$

with a specific regularizer that we introduce: $R(\boldsymbol{w}) = F(\boldsymbol{w}) + \frac{\alpha}{2}\|\boldsymbol{w}\|_2{}^2$ with $F(\boldsymbol{w}) = \frac{1-\alpha}{2}(\|\boldsymbol{w}\|_k^{sp})^2$, for some constant $1 > \alpha > 0$ which will be described later. The algorithm that we will use to solve approximately (I_{ks}) is the Accelerated Dual Gradient Descent (ADGD) described in [57], which is an accelerated version of a primal-dual method that is known in the literature under many names, and that comprises the following steps, with γ being some learning rate, and $\hat{\boldsymbol{v}}_t$ being a dual variable:

`# primal projection step`
$\hat{\boldsymbol{w}}_t \leftarrow \text{prox}_{\alpha^{-1}F}(-\alpha^{-1}\boldsymbol{X}^{\top}\hat{\boldsymbol{v}}_t)$
`# dual update step`
$\hat{\boldsymbol{v}}_{t+1} \leftarrow \hat{\boldsymbol{v}}_t + \gamma(\boldsymbol{X}\hat{\boldsymbol{w}}_t - \boldsymbol{y}^{\delta})$

The method above is most commonly known in the signal processing and image denoising literature as Linearized Bregman Iterations, or Inverse Scale Space Methods [21, 68]. In the optimization literature, it is mostly known as (Lazy) Mirror Descent [17], also called Dual Averaging [65, 95]. The main idea in [57] is to early stop the algorithm at some iteration T, before convergence. We present the full accelerated version, IRKSN, in Algorithm 2.

Algorithm 2: IRKSN

Initialization: $\hat{\boldsymbol{v}}_0 = \hat{\boldsymbol{z}}_{-1} = \hat{\boldsymbol{z}}_0 \in \mathbb{R}^d, \gamma = \alpha \|\boldsymbol{X}\|^{-2}, \theta_0 = 1$ **for** $t = 0$ ***to*** T **do**

$\hat{\boldsymbol{w}}_t \leftarrow \text{prox}_{\alpha^{-1}F}\left(-\alpha^{-1}\boldsymbol{X}^T\hat{\boldsymbol{z}}_{\boldsymbol{t}}\right)$

$\hat{\boldsymbol{r}}_t \leftarrow \text{prox}_{\alpha^{-1}F}\left(-\alpha^{-1}\boldsymbol{X}^T\hat{\boldsymbol{v}}_t\right)$

$\hat{\boldsymbol{z}}_t \leftarrow \hat{\boldsymbol{v}}_t + \gamma\left(\boldsymbol{X}\hat{\boldsymbol{r}}_{\boldsymbol{t}} - \boldsymbol{y}^\delta\right)$

$\theta_{t+1} \leftarrow \left(1 + \sqrt{1 + 4\theta_t^2}\right)/2$

$\hat{\boldsymbol{v}}_{t+1} = \hat{\boldsymbol{z}}_t + \frac{\theta_t - 1}{\theta_{t+1}}\left(\hat{\boldsymbol{z}}_t - \hat{\boldsymbol{z}}_{t-1}\right)$

end

Main results

In this section, we introduce the main result of our paper, which gives specific conditions for robust recovery of $\boldsymbol{w}^*$, and early stopping bounds on $\|\hat{\boldsymbol{w}}_t - \boldsymbol{w}^*\|$ for IRKSN.

Assumptions

We will require several assumptions, which are equivalent to the source conditions needed for ℓ_1-based recovery, but instead here must hold for recovery with the k-support norm. We compare those assumptions to usual source conditions for recovery with ℓ_1 norm in Section 3.2.1. The first assumption below is a more general version of the usual feasibility assumption of the noiseless problem [31]: it simply states that $\boldsymbol{w}^*$ is the true model, that we wish to recover, and that it is k-sparse. Recall from the introduction that $\boldsymbol{y}$ is the true target vector, i.e. uncorrupted by noise.

Assumption 5. *$\boldsymbol{w}^*$ is k-sparse of support $S \subset [d]$, and is a solution of the system $(L) : \boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}$. In addition, $\boldsymbol{w}^*$ is the smallest ℓ_2 norm solution of (L) on its support, that is, $\boldsymbol{w}^*$ is such that:*

$$\boldsymbol{w}_S^* = \arg \min_{\boldsymbol{z} \in \mathbb{R}^k : \boldsymbol{X}_S \boldsymbol{z} = \boldsymbol{y}} \|\boldsymbol{z}\|_2$$

We now provide our main assumption, which is intrinsically linked to the structure of the k-support norm, and which is, up to our knowledge, the first condition of such kind in the sparse recovery literature.

Assumption 6. *$\boldsymbol{w}^*$ verifies:*

$$\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^* \rangle| < \min_{j \in S} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_j, \boldsymbol{w}_S^* \rangle|$$

Up to our knowledge, we are the first to provide such assumptions for recovery with a k-support norm based algorithm: although [24] proposed a k-support norm based algorithm and corresponding conditions for recovery, those conditions only apply in the case of a design matrix $\boldsymbol{X}$ with Gaussian i.i.d. entries.

Discussion on the assumptions

In this paragraph, we attempt to interpret the assumptions above in simple terms, and to compare them to the usual assumptions for recovery with ℓ_1 norm. More precisely, the condition below is equivalent to Condition 4.3 from [40], which is shown in [40] to be a necessary and sufficient condition for achieving a linear rate of recovery with ℓ_1 norm Tikhonov regularization. We prove such equivalence in Appendix B.2.

Assumption 7 (Recovery with ℓ_1 norm.)**.** *Let $\boldsymbol{w}^*$ be supported on a support $S \subset [d]$. $\boldsymbol{w}^*$ is such that:*

(i) $\boldsymbol{X}\boldsymbol{w}^* = \boldsymbol{y}$

(ii) $\boldsymbol{X}_S$ *is injective*

(iii) $\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \operatorname{sgn}(\boldsymbol{w}_S^*)\rangle| < 1$

Below, we now compare this assumption to ours.

The min ℓ_2 norm solution: In our assumption 5, the minimum ℓ_2 norm condition is actually not restrictive, compared to 7: indeed, in 7 $\boldsymbol{X}_S$ needs to be injective, which implies that there needs to be only one solution $\boldsymbol{w}_S^*$ on S such that $\boldsymbol{X}_S \boldsymbol{w}_s^* = \boldsymbol{y}$: we can also work in such situations, but we also include the additional cases where there are several solutions on S (we just require that $\boldsymbol{w}^*$ is the minimum norm one) : $\boldsymbol{X}_S$ does not need to be injective in our case. Importantly we can deal with cases with $n < k$, when Lasso (and ℓ_1 iterative regularization methods) cannot (that we can obtain recovery in a regime where the number of samples n is even lower than the sparsity of the signal k).

Dependence on the sign: As we can observe, 7 is verified or not based on $\operatorname{sgn}(\boldsymbol{w}_S^*)$. This implies that irrespective of the actual values of $\boldsymbol{w}^*$, recovery will be possible or not only based on $\operatorname{sgn}(\boldsymbol{w}_S^*)$. On the contrary, our assumption 6 depends on $\boldsymbol{w}^*$ itself.

Interpretation as regression of columns: In [68], section 3, a related condition to 7 called the Irrepresentability Condition [100] is analyzed in terms of *regressing the columns of* $\boldsymbol{X}$. Here we attempt such a similar explanation for our Assumption 6. It is well known that the minimum ℓ_2 norm solution $\boldsymbol{z}^*$ to a general least square problem $\min_{\boldsymbol{z}} \|\boldsymbol{M}\boldsymbol{z} - \boldsymbol{b}\|^2$ for some matrix $\boldsymbol{M}$ and vector $\boldsymbol{b}$ can be expressed with the pseudo-inverse of $\boldsymbol{M}$: $\boldsymbol{z}^* = \boldsymbol{M}^\dagger \boldsymbol{b}$ [12, 72]. Let us call such a $\boldsymbol{z}^*$ the *best fit for predicting* $\boldsymbol{b}$ *using* $\boldsymbol{M}$. We can then interpret our Assumption 6 in the following way: *"$\boldsymbol{w}_S^*$ must be close (in absolute inner product) to all the best fits for predicting any column of $\boldsymbol{X}_S$ using $\boldsymbol{X}_S$, and far from the best fits for predicting any column of $\boldsymbol{X}_{\bar{S}}$ using $\boldsymbol{X}_S$."*

Case where $\boldsymbol{X}_S$ is injective: In the case where $\boldsymbol{X}_S$ is injective (as happens in most cases in practice when $n > k$), it is even easier to compare Assumptions 6 and 7. Indeed,

since in that case we have that $\boldsymbol{X}_S$ is full column rank, we then have : $\boldsymbol{X}_S^\dagger \boldsymbol{X}_S = \boldsymbol{I}_{k\times k}$. Therefore, Assumption 6 can be rewritten into: $\max_{\ell\in\bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^*\rangle| < \min_{j\in S} |\boldsymbol{w}_S^*|$, which is equivalent to:

$$\max_{\ell\in\bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \frac{\boldsymbol{w}_S^*}{\min_{j\in S} |\boldsymbol{w}_S^*|}\rangle| < 1$$

Therefore, we can notice that if $\boldsymbol{w}_S^* = \gamma \operatorname{sgn}(\boldsymbol{w}_S^*)$ for some $\gamma > 0$ (that is, each component of $\boldsymbol{w}_S^*$ have the same absolute value), both Assumptions 6 and 7 become equivalent (because then: $\frac{\boldsymbol{w}_S^*}{\min_{j\in S} |\boldsymbol{w}_S^*|} = \operatorname{sgn}(\boldsymbol{w}_S^*)$). However, the two conditions 6 and 7 may differ depending on the *relative magnitudes* of the entries in $\boldsymbol{w}_S^*$. In particular, it may happen that our condition 6 is verified even if the condition for recovery with ℓ_1 norm (7) is not verified. We analyze such an example in more details in Section 3.2.2.

3.2.2 Early Stopping Bound

We are now ready to state our main result:

Theorem 2 (Early Stopping Bound)**.** *Let $\delta \in]0, 1]$ and let $(\hat{\boldsymbol{w}}_t)_{t\in\mathbb{N}}$ be the sequence generated by IRKSN. Assuming the design matrix $\boldsymbol{X}$ and the true sparse vector $\boldsymbol{w}^*$ satisfy Assumptions 6 and 5, and with $\alpha < \frac{\eta}{\|\boldsymbol{w}\|_\infty}$ with $\eta := \min_{j\in S} |\langle (\boldsymbol{X}_S \boldsymbol{X}_S^\top)^\dagger \boldsymbol{y}, \boldsymbol{x}_j\rangle| - \max_{\ell\in\bar{S}} |\langle (\boldsymbol{X}_S \boldsymbol{X}_S^\top)^\dagger \boldsymbol{y}, \boldsymbol{x}_\ell\rangle|$, we have for $t \geq 2$:*

$$\|\hat{\boldsymbol{w}}_t - \boldsymbol{w}^*\|_2 \leq at\delta + bt^{-1}$$

$$\textit{with} \quad a = 4\|\boldsymbol{X}\|^{-1} \quad \textit{and} \quad b = \frac{2\|\boldsymbol{X}\|\|(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*\|}{\alpha}$$

In particular (if $\delta > 0$), choosing $t_\delta = \lceil c\delta^{-1/2} \rceil$ for some $c > 0$:

$$\|\hat{\boldsymbol{w}}_t - \boldsymbol{w}^*\|_2 \leq (a(c+1) + bc^{-1})\delta^{1/2}$$

Proof. Proof in Appendix B.3. □

Discussion We can notice in Theorem 2 above that b is large when α is small: therefore, if the inequality in 6 is very tight, as a consequence, α will need to be taken small, and b will become large. Therefore, we can say that the larger the margin by which Assumption 6 is fulfilled is, the better the retrieval of the true vector $\boldsymbol{w}^*$ is (because the larger we can choose α).

Illustrating Example

In this section, we describe a simple example that illustrates the cases where ℓ_1 norm-based regularization will fail, and IRKSN will successfully recover the true vector.

Example 1: We consider a model that consists of three "generating" variables $X^{(0)}, X^{(1)}$ and $X^{(2)}$, that are random i.i.d. variables from standard Gaussian (we denote

$X^{(0)} \sim \mathcal{N}(0,1)$ and $X^{(1)} \sim \mathcal{N}(0,1)$ and $X^{(2)} \sim \mathcal{N}(0,1)$). Two other variables $X^{(3)}$ and $X^{(4)}$, are actually correlated with the previous random variables: they are obtained noiselessly, and linearly from those, with some vectors $\boldsymbol{w}^{(3)}$ and $\boldsymbol{w}^{(4)}$ that will be defined below:

$$X^{(3)} = w_0^{(3)} X^{(0)} + w_1^{(3)} X^{(1)} + w_2^{(3)} X^{(2)}$$

and

$$X^{(4)} = w_0^{(4)} X^{(0)} + w_1^{(4)} X^{(1)} + w_2^{(4)} X^{(2)}$$

In addition, similarly, the actual observations Y are formed noiselessly and linearly from $(X^{(0)}, X^{(1)}, X^{(2)})$, for some vector $\boldsymbol{w}^{(y)}$:

$$Y = w_0^{(y)} X^{(0)} + w_1^{(y)} X^{(1)} + w_2^{(y)} X^{(2)}$$

A graphical visualization of this construction can be seen on Figure 3.5. More precisely, we define the vectors $\boldsymbol{w}^{(3)}, \boldsymbol{w}^{(4)}$ and $\boldsymbol{w}^{(y)}$ are defined as follows:

$$\boldsymbol{w}^{(3)} = \begin{bmatrix} 9/11 \\ 6/11 \\ 2/11 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{w}^{(4)} = \begin{bmatrix} 1/3 \\ 14/15 \\ 2/15 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{w}^{(y)} = \begin{bmatrix} 1 \\ 1 \\ -4 \\ 0 \\ 0 \end{bmatrix}.$$

We will generate such a dataset with $n = 4$: so the dataset will be composed of 4 samples of $X^{(0)}, X^{(1)}, X^{(2)}, X^{(3)}, X^{(4)}$, which form the matrix $\boldsymbol{X} \in \mathbb{R}^{4,5}$, with $\boldsymbol{X} = [\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4]$ and 4 samples of Y, which form the vector $\boldsymbol{y} \in \mathbb{R}^4$. In our case, we have $S = \text{supp}(\boldsymbol{w}^{(y)}) = \{0, 1, 2\}$, and therefore we just ensure that $\boldsymbol{X}_S = [\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2]$ is full column rank (which should be the case with overwhelming probability since those three first vectors are sampled from a Gaussian, and since we have $n = 4 > k = 3$). Our goal is to reconstruct the true linear model of Y, which is $\boldsymbol{w}^{(y)}$ from the observation of $\boldsymbol{X}$ and $\boldsymbol{y}$. We can easily check

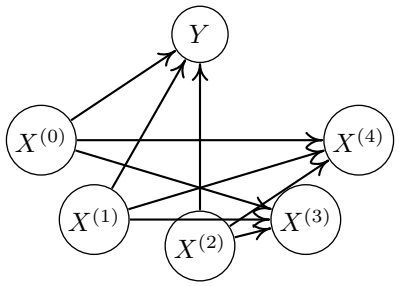


Figure 3.5: $X^{(3)}$ and $X^{(4)}$ are correlated with $X^{(0)}, X^{(1)}$ and $X^{(2)}$

mathematically (using the closed form from the first column of Table 2.2), that this example only verifies our condition (Assumption 6), but that it does not verify Assumption 7 (i.e. it is in the blue area from Figure **??**). Indeed, in that case, $\boldsymbol{X}_S$ is full column rank, which implies $(\boldsymbol{X}_S)^\dagger \boldsymbol{x}_3 = \boldsymbol{w}^{(3)}$ and $(\boldsymbol{X}_S)^\dagger \boldsymbol{x}_4 = \boldsymbol{w}^{(4)}$ [37].

We then have:

$$|\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_3, \text{sgn}(\boldsymbol{w}^{(y)})\rangle| = |\langle \boldsymbol{w}^{(3)}, \text{sgn}(\boldsymbol{w}^{(y)})\rangle| = \frac{13}{11} > 1$$

and

$$|\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_4, \text{sgn}(\boldsymbol{w}^{(y)})\rangle| = |\langle \boldsymbol{w}^{(4)}, \text{sgn}(\boldsymbol{w}^{(y)})\rangle| = \frac{17}{15} > 1$$

Therefore: $\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \text{sgn}(\boldsymbol{w}_S^*)\rangle| = \frac{13}{11} > 1$ Which means that Assumption 7 is not verified. However, on the other hand, we have:

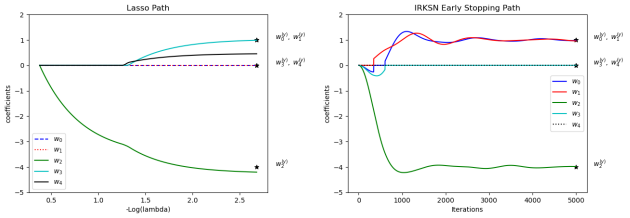
$$|\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_3, \frac{\boldsymbol{w}^{(y)}}{\min_{j\in S} |\boldsymbol{w}^{(y)}|}\rangle| = |\langle \boldsymbol{w}^{(3)}, \frac{\boldsymbol{w}^{(y)}}{\min_{j\in S} |\boldsymbol{w}^{(y)}|}\rangle| = \frac{7}{11}$$

and

$$|\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_4, \frac{\boldsymbol{w}^{(y)}}{\min_{j\in S} |\boldsymbol{w}^{(y)}|}\rangle| = |\langle \boldsymbol{w}^{(4)}, \frac{\boldsymbol{w}^{(y)}}{\min_{j\in S} |\boldsymbol{w}^{(y)}|}\rangle| = \frac{11}{15}$$

Therefore: $\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \frac{\boldsymbol{w}^{(y)}}{\min_{j\in S} |\boldsymbol{w}^{(y)}|}\rangle| = \frac{11}{15} < 1$. Therefore, from Section 3.2.1, paragraph *Case where $\boldsymbol{X}_S$ is injective*, we see that our Assumption 6 is verified here.

Comparison of the IRKSN path with Lasso In Figure 3.6 below, we compare the Lasso path (that is, the solutions found by Lasso for all values of the penalization λ), with the IRKSN path (that is, the solutions found by IRKSN at every timestep). For indicative purposes, we also provide the path of the ElasticNet on the same problem in Appendix B.7.



(a) Lasso path (b) IRKSN path

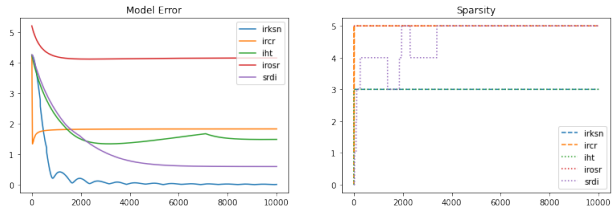
Figure 3.6: Comparison of the path of IRKSN with Lasso. $w_i^{(y)}$ with $i \in [0, ..., 4]$ denotes the i-th component of $\boldsymbol{w}^{(y)}$, and `lambda` is the penalty of the Lasso. We recall $w_0^{(y)} = w_1^{(y)} = 1, w_2^{(y)} = -4, w_3^{(y)} = w_4^{(y)} = 0$: only IRKSN recovers the true $\boldsymbol{w}^{(y)}$.

As we can see, the Lasso is unable to retrieve the true sparse vector, for any λ. However IRKSN can successfully retrieve it, which confirms the theory above.

In addition, this path from Figure 3.6 above illustrates well the optimization dynamics of IRKSN: first, the true support of $\boldsymbol{w}^{(y)}$ is not identified in the first iterations. But after a few iterations, we observe what we could call a phenomenon of *exchange of variable*: $w_0^{(y)}$

is exchanged with $w_1^{(y)}$, and later, $w_3^{(y)}$ is exchanged with $w_0^{(y)}$ (by *exchange*, we mean that at a timestep t, $w_0^{(y)}(t) \neq 0$ but $w_1^{(y)}(t) = 0$, but at timestep $t+1$: $w_0^{(y)}(t+1) = 0$ and $w_1^{(y)}(t+1) \approx w_0^{(y)}(t)$). This can be explained by the fact that when α is small, the proximal operator of the k-support norm approaches the hard-thresholding operator from [14]: hence at a particular timestep the ordering (in absolute magnitude) of the components of $\boldsymbol{X}^\top \hat{\boldsymbol{z}}_t$ suddenly changes (with the components where the change occurs having about the same magnitude at the time of change, if the learning rate is small), which results into such an observed change in primal space.

Additionally, in Figure 3.7 below, we run the iterative methods from Table 2.2 (IRKSN, IRCR, IROSR and SRDI) (as well as IHT for comparison) on Example 1, and measure the recovery error $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^{(y)}\|$ as well as the sparsity $\|\hat{\boldsymbol{w}}\|_0$ of the iterates. As we can see, only IRKSN can achieve 0 error, that is, full recovery in the noiseless setting. In addition, except IHT (which however fails to approach the true solution), no method is able to converge to a 3-sparse solution, which is the true degree of sparsity of the solution.



(a) Model error $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^{(y)}\|$ (b) Model sparsity $\|\hat{\boldsymbol{w}}\|_0$

Figure 3.7: Only IRKSN can recover the true $\boldsymbol{w}^{(y)}$ in this example.

3.2.3 Experiments

In this section, we run some experiments on real-life datasets to illustrate the applicability of IRKSN.

Setting We consider the problem of sparse linear regression, where our goal is to minimize the expected mean squared error (MSE) loss of prediction $\mathbb{E}_{X,Y}(Y - \hat{Y})^2$, where Y is the true regressed target, and $\hat{Y}$ is the predicted target, predicted linearly from the regressors X:

$$\hat{Y} = \langle \hat{\boldsymbol{w}}, X \rangle + b = \sum_{i=1}^{d} \hat{w}_i X_i + b$$

(where b is the intercept, fitted separately (see Appendix B.8 for more details)), and where $\hat{\boldsymbol{w}}$ is a sparse model that we seek to estimate from a training set of n observations of X and Y. For each run, we first randomly split the data into a training set, and a test set which contains 25% of the data. Then, we split the training set into an actual training set and a validation set, with the same proportion (75%/25%). Hyperparameters, including learning

rate parameters and early stopping time are fitted to minimize the MSE on the validation set. Then the empirical MSE on the test set is reported. This procedure is repeated 10 times, and we report in Tables 3.2 and 3.3 the mean and standard deviation of that test set MSE.

Additional details including details on the intercept and a preprocessing step, as well as the values for the grid-search of each algorithm are described in Appendix B.8. Our results are produced on a server of CPUs with 32 cores and 126G RAM, and take 5 hours to run.

Datasets We evaluate the algorithms on the following open source datasets (obtained from the sources LibSVM [22] and OpenML [88]), of which a brief summary is presented in Table 3.1.

Table 3.1: Datasets used in the comparison. *References:* [(1)]: [38], [(2)] [69], [(3)]: [78], [(4)]: [75]. *Sources:* [(1)]: [22], [(2)] [47] downloaded with `scikit-learn` [71], [(3,4)]: [16] .

Dataset	d	n
leukemia[(1)]	7129	38
housing [(2)]	8	20640
scheetz2006[(3)]	18975	120
rhee2006[(4)]	361	842

Results We present our results in Tables 3.2 and 3.3. Generally, we observe that for datasets with a large d (such as `leukemia` and `scheetz2006`), ℓ_1 based methods such as Lasso, IRCR, or SRDI achieve poorer performance: indeed, the Lasso is known to saturate when $d > n$ [102], i.e. its predicted $\boldsymbol{w}^*$ cannot contain more than n nonzero variables. This is not the case for the ElasticNet and k-support norm based algorithms like IRKSN, which is why those latter algorithms achieve a good score in this $d > n$ setting.

Perhaps surprisingly, IROSR also achieves a good score on `scheetz2006` ($d >> n$), even if its reparameterization is supposed to enforce some ℓ_1 regularization [89]. However, the theory in [89] holds for small initializations, so we hypothesize that due to our grid search, IRCR might be able to explore regimes beyond the ℓ_1 norm, with large initialization.

Also, although IRKSN and KSN penalty achieve similar errors on such high dimensional datasets, we recall that for KSN penalized, the strenght of the penalty λ needs to be tuned by cross-validation which is not the case for IRKSN, which just takes a single run to identify the best early stopping iteration. Our results also confirm the findings from [3], namely that the k-support norm regularization often outperforms the ElasticNet: this is also true for iterative regularizations using the k-support norm (namely, IRKSN).

Finally, in the light of section 3.2.1, we hypothesize that this success of the k-support norm regularization might be related to the fact that IRKSN converges to the *minimum* ℓ_2 *norm feasible solution* that is k-sparse (assuming Assumption 6 is verified for this solution)

(see Section 3.2.1). In other words, the k-support norm regularization allows one to benefit *at the same time* from the regularization property of a small support for the solution (i.e. of size k), *and* of a small ℓ_2 norm (which is known to prevent overfitting in many cases [7]).

Table 3.2: Test MSE of the methods of Table 2.2 on the `leukemia` and `housing` datasets. Bold figures have a mean within the standard deviation of the best score from each column.

Method	leukemia	housing
IHT	$\mathbf{0.322 \pm 0.137}$	$\mathbf{0.535 \pm 0.011}$
Lasso	0.450 ± 0.204	$\mathbf{0.535 \pm 0.016}$
ElasticNet	$\mathbf{0.307 \pm 0.154}$	$\mathbf{0.540 \pm 0.031}$
KSN pen.	$\mathbf{0.251 \pm 0.090}$	$\mathbf{0.533 \pm 0.009}$
OMP	0.730 ± 0.376	$\mathbf{0.533 \pm 0.009}$
SRDI	0.396 ± 0.220	$\mathbf{0.533 \pm 0.009}$
IROSR	0.352 ± 0.121	0.655 ± 0.013
IRCR	$\mathbf{0.326 \pm 0.102}$	$\mathbf{0.534 \pm 0.010}$
IRKSN (ours)	$\mathbf{0.264 \pm 0.091}$	$\mathbf{0.538 \pm 0.012}$

Table 3.3: Test MSE of the methods of Table 2.2 on gene array data: the `scheetz2006` dataset and the `rhee2006` dataset.

Method	scheetz2006	rhee2006
IHT	$\mathbf{0.008 \pm 0.003}$	$\mathbf{0.576 \pm 0.053}$
Lasso	0.012 ± 0.008	$\mathbf{0.557 \pm 0.049}$
ElasticNet	$\mathbf{0.009 \pm 0.004}$	$\mathbf{0.541 \pm 0.042}$
KSN pen.	$\mathbf{0.008 \pm 0.003}$	$\mathbf{0.556 \pm 0.035}$
OMP	0.016 ± 0.06	0.684 ± 0.057
SRDI	0.018 ± 0.013	$\mathbf{0.567 \pm 0.043}$
IROSR	$\mathbf{0.007 \pm 0.003}$	$\mathbf{0.583 \pm 0.044}$
IRCR	0.018 ± 0.013	1.389 ± 0.105
IRKSN (ours)	$\mathbf{0.008 \pm 0.003}$	$\mathbf{0.578 \pm 0.038}$

Chapter 4

Ongoing and Future Directions

In this chapter, we present some further directions that we, as well as our collaborators, are currently exploring, and that we will continue to explore further for the rest of the thesis.

4.1 Variance Reduction

As discussed in Section 3 above, the variance from the stochastic gradients estimates (where the stochasticity comes from the zeroth-order or the usual stochastic sampling), is in conflict with the hard-threholding operator. Therefore, we will explore variance reductions techniques, in order to counter such variance. Below we describe the Stochastic Variance Reduced Zeroth-Order Hard-Thresholding (VR-SZHT), which is a variant of SZOHT introduced by Xinzhe Yuan (who we are collaborating with), where the gradient is estimated using a variance reduction technique.

Algorithm 3: Stochastic variance reduced zeroth-order Hard-Thresholding (VR-SZHT)

Initialization: η, T, $\boldsymbol{x}^0$, *SVRG update frequency* m, q, k. **Output :** $\boldsymbol{x}^T$.
for $r = 1, \ldots, T$ **do**
  $\boldsymbol{x}^{(0)} = \boldsymbol{x}^{r-1}$; $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \hat{\nabla} f_i(\boldsymbol{x}^{(0)})$; **for** $t = 0, 1, \ldots, m-1$ **do**
    Randomly sample $i_t \in \{1, 2, \ldots, n\}$;
    Compute ZO estimate $\hat{\nabla} f_{i_t}(\boldsymbol{x}^{(t)})$, $\hat{\nabla} f_{i_t}(\boldsymbol{x}^{(0)})$;
    $\bar{\boldsymbol{x}}^{(t+1)} = \boldsymbol{x}^{(t)} - \eta(\hat{\nabla} f_{i_t}(\boldsymbol{x}^{(t)}) - \hat{\nabla} f_{i_t}(\boldsymbol{x}^{(0)}) + \hat{\mu})$;
    $\boldsymbol{x}^{(t+1)} = \phi_k(\bar{\boldsymbol{x}}^{(t+1)})$;
  end
  $\boldsymbol{x}^r = \boldsymbol{x}^{(m)}$;
end

In a recent submission to ICCV 2023 to which we collaborated, Xinzhe Yuan provided the convergence analysis of VR-SZHT, and found that VR-SZHT removes the restriction on the number of random directions q, which largely improves the applicability of the algorithm. Our plan is to build on such a work, and to integrate it into some unified analysis

of Iterative Hard Thresholding, that incorporates general stochasticity (from zeroth-order and sampling from a finite sum structure), as well as variance reduction, and additional constraints as described below.

4.2 Additional Constraints

Additionally, we plan to incorporate additional constraints in the original Zeroth-Order Hard-Thresholding. Indeed, practitioners may often want not only to control the sparsity of the model, but also, say, its maximum and minimum value, for instance in few pixels adversarial attacks, where if one only constraints the sparsity, one can obtain large pixels perturbation, which is undesirable (such an extra constraints is studied for instance in a submitted paper by one of our collaborators). Therefore, we started studying, first in the first order case (but this might be later extended to the zeroth-order case) a generalized version of hard-thresholding operator where an additional convex set $\mathcal{S}$ is added to the constraints. The general problem can be formalized as follows:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \mathcal{F}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}), \quad s.t.\ \|\boldsymbol{x}\|_0 \leq k \text{ and } \boldsymbol{x} \in \mathcal{S}. \tag{4.1}$$

4.3 Structural sparsity

So far, we only discussed ℓ_0 ball constraints. However, one may want to enforce instead some structured constraints, for instance, constraints of the form: $\{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}_{\mathcal{G}_1}\|_0 < D_1 \wedge \|\boldsymbol{x}_{\mathcal{G}_2}\|_0 < D_2\}$, where $\boldsymbol{x}_{\mathcal{G}_1}$ and $\boldsymbol{x}_{\mathcal{G}_1}$ denote a partition of $\boldsymbol{x}$ into coordinates from a group $\mathcal{G}_1$ and a group $\mathcal{G}_2$.

4.4 Reinforcement learning

One important application of zeroth-order optimization is reinforcement learning. Works such as [77] have demonstrated the high efficiency of zeroth-order like methods (more precisely, evolution strategies, which are similar to zeroth-order) for reinforcement learning, in particular in the distributed setting. However, similarly to usual zeroth-order, such optimization is highly dependent on the dimensionality of the input d. Therefore, it would be interesting to study whether the dimension independence (under some conditions) of SZOHT could be leveraged in order to improve the efficiency of reinforcement learning methods.

4.5 Others

Below we also present other various ideas that we may also explore further in the future:

- **Low-rank Matrices:** it would be interesting to extend our work to the space of matrices, not only the space of vectors.

- **Sparse graphs:** similarly to the above, we could extend our work to the space of (sparse) graphs.

- **Acceleration of ZOHT:** recent works such as [4] allow to obtain algorithms with $k = \mathcal{O}(\kappa)$ instead of $k = \mathcal{O}(\kappa^2)$, which is significant, in particular for large condition number κ. Being able to use such algorithms in our framework would allow to obtain sparser solutions with better objective function values.

- **Relaxed Assumptions:** recent work extend IHT beyond the RSC assumption, including some non-restricted convex settings [73]. This could potentially represent more problems encountered in practice, such as optimization with neural networks, or potentially non-convex problems such as reinforcement learning.

- **Lower bound:** There are already some existing lower bounds on the oracle complexity of zeroth-order optimization (i.e. number of function calls), such as [44]. Similarly, there are some lower bounds on the number of first order oracle calls necessary for convex optimization when it is known that a minimum is sparse [1]. However, up to our knowledge, there are no lower bound on the query complexity of **sparse zeroth-order optimization** (i.e. for instance, ZO optimization when a solution is known to be sparse.). Finding such a bound would answer the open question on whether SZOHT (and its variant) are actually optimal (i.e. ideally it may show that there does not exist any algorithm with a better query complexity than SZOHT for a certain class of problems).

- **Better non-convex regularizations:** From the properties of the k-support norm, we can easily show that the following penalty h is a relaxation of the ℓ_0 ball indicator function, that is, when $\lambda \to \infty$, it equals the ℓ_0 ball indicator function (see figure below). This is because the k-support norm of any vector of sparsity at most k is equal to its ℓ_2 norm.

$$h(\boldsymbol{x}) = \lambda(\frac{1}{2}(\|\boldsymbol{x}\|_k^{sp})^2 - \frac{1}{2}\|\boldsymbol{x}\|_2^2)$$

  Therefore, it would be interesting to compare such a non-convex penalty with existing ones, such as SCAD [28] or MCP [98]. A first interesting property is that such a penalty h would be 0 for any vector of sparsity at most k, and nonzero otherwise.

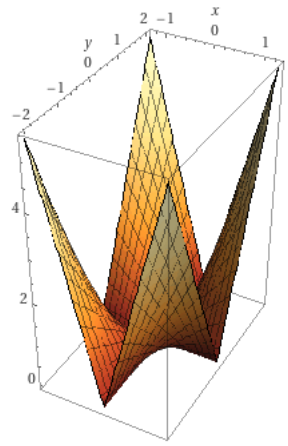


Figure 4.1: Nonconvex penalty based on the k-support norm.

References

[1] Alekh Agarwal, Peter L Bartlett, Pradeep Ravikumar, and Martin J Wainwright. Information-theoretic lower bounds on the oracle complexity of sparse convex optimization.

[2] George B Arfken and Hans J Weber. *Mathematical methods for physicists*. American Association of Physics Teachers, 1999.

[3] Andreas Argyriou, Rina Foygel, and Nathan Srebro. Sparse prediction with the k-support norm. *Advances in Neural Information Processing Systems*, 25, 2012.

[4] Kyriakos Axiotis and Maxim Sviridenko. Iterative hard thresholding with adaptive regularization: Sparser solutions without sacrificing runtime. In *International Conference on Machine Learning*, pages 1175–1197. PMLR, 2022.

[5] Krishnakumar Balasubramanian and Saeed Ghadimi. Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[6] J Baptiste, H Urruty, and C Lemarechal. Fundamentals of convex analysis, 2001.

[7] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

[8] Heinz H Bauschke, Patrick L Combettes, et al. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011.

[9] John E Beasley. Or-library: distributing test problems by electronic mail. *Journal of the operational research society*, 41(11):1069–1072, 1990.

[10] Amir Beck. *First-order methods in optimization*. SIAM, 2017.

[11] Eugene Belilovsky, Katerina Gkirtzou, Michail Misyrlis, Anna B Konova, Jean Honorio, Nelly Alia-Klein, Rita Z Goldstein, Dimitris Samaras, and Matthew B Blaschko. Predictive sparse modeling of fmri data for improved classification, regression, and visualization using the k-support norm. *Computerized Medical Imaging and Graphics*, 46:40–46, 2015.

[12] Adi Ben-Israel and Thomas NE Greville. *Generalized inverses: theory and applications*, volume 15. Springer Science & Business Media, 2003.

[13] Quentin Bertrand and Mathurin Massias. Anderson acceleration of coordinate descent. In *International Conference on Artificial Intelligence and Statistics*, pages 1288–1296. PMLR, 2021.

[14] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

[15] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[16] Patrick Breheny, 2022.

[17] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[18] Peter Bühlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

[19] HanQin Cai, Yuchen Lou, Daniel McKenzie, and Wotao Yin. A zeroth-order block coordinate descent algorithm for huge-scale black-box optimization. In *International Conference on Machine Learning*, pages 1193–1203. PMLR, 2021.

[20] HanQin Cai, Daniel McKenzie, Wotao Yin, and Zhenliang Zhang. Zeroth-order regularized optimization (zoro): Approximately sparse gradients and adaptive sampling. *SIAM Journal on Optimization*, 32(2):687–714, 2022.

[21] Jian-Feng Cai, Stanley Osher, and Zuowei Shen. Linearized bregman iterations for compressed sensing. *Mathematics of computation*, 78(267):1515–1536, 2009.

[22] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[23] T-J Chang, Nigel Meade, John E Beasley, and Yazid M Sharaiha. Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, 27(13):1271–1302, 2000.

[24] Soumyadeep Chatterjee, Sheng Chen, and Arindam Banerjee. Generalized dantzig selector: Application to the k-support norm. *Advances in Neural Information Processing Systems*, 27, 2014.

[25] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.

[26] Xiangyi Chen, Sijia Liu, Kaidi Xu, Xingguo Li, Xue Lin, Mingyi Hong, and David Cox. Zo-adamm: Zeroth-order adaptive momentum method for black-box optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[27] Krzysztof Choromanski, Aldo Pacchiano, Jack Parker-Holder, Yunhao Tang, Deepali Jain, Yuxiang Yang, Atil Iscen, Jasmine Hsu, and Vikas Sindhwani. Provably robust blackbox optimization for reinforcement learning. In *Conference on Robot Learning*, pages 683–696. PMLR, 2020.

[28] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

[29] Huang Fang, Zhenan Fan, Yifan Sun, and Michael Friedlander. Greed meets sparsity: Understanding and improving greedy coordinate descent for sparse optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 434–444. PMLR, 2020.

[30] Samuel Farrens, Antoine Grigis, Loubna El Gueddari, Zaccharie Ramzi, GR Chaithya, S Starck, B Sarthou, Hamza Cherkaoui, Philippe Ciuciu, and J-L Starck. Pysap: Python sparse data analysis package for multidisciplinary image processing. *Astronomy and Computing*, 32:100402, 2020.

[31] Simon Foucart and Holger Rauhut. An invitation to compressive sensing. In *A mathematical introduction to compressive sensing*, pages 1–39. Springer, 2013.

[32] Xiang Gao, Bo Jiang, and Shuzhong Zhang. On the information-adaptive variants of the admm: an iteration complexity perspective. *Journal of Scientific Computing*, 76(1):327–363, 2018.

[33] Dan Garber and Elad Hazan. Faster rates for the frank-wolfe method over strongly-convex sets. In *International Conference on Machine Learning*, pages 541–549. PMLR, 2015.

[34] Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1):267–305, 2016.

[35] Katerina Gkirtzou, Jean Honorio, Dimitris Samaras, Rita Goldstein, and Matthew B Blaschko. fmri analysis of cocaine addiction using k-support sparsity. In *2013 IEEE 10th International Symposium on Biomedical Imaging*, pages 1078–1081. IEEE, 2013.

[36] Daniel Golovin, John Karro, Greg Kochanski, Chansoo Lee, Xingyou Song, and Qiuyi Zhang. Gradientless descent: High-dimensional zeroth-order optimization. In *International Conference on Learning Representations*, 2019.

[37] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

[38] Todd R Golub, Donna K Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P Mesirov, Hilary Coller, Mignon L Loh, James R Downing, Mark A Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531–537, 1999.

[39] Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. SGD: General analysis and improved rates. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 5200–5209. PMLR, 2019.

[40] Markus Grasmair, Otmar Scherzer, and Markus Haltmeier. Necessary and sufficient conditions for linear convergence of ℓ_1-regularization. *Communications on Pure and Applied Mathematics*, 64(2):161–182, 2011.

[41] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

[42] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning*, pages 433–440, 2009.

[43] Prateek Jain, Ambuj Tewari, and Purushottam Kar. On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

[44] Kevin G Jamieson, Robert Nowak, and Ben Recht. Query complexity of derivative-free optimization. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

[45] Jinzhu Jia and Bin Yu. On model selection consistency of the elastic net when p » n. *Statistica Sinica*, pages 595–611, 2010.

[46] Sham Kakade, Shai Shalev-Shwartz, Ambuj Tewari, et al. On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization.

[47] Charles Kooperberg. Statlib: an archive for statistical software, datasets, and information. *The American Statistician*, 51(1):98, 1997.

[48] Ken Lang. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier, 1995.

[49] Evgeny S Levitin and Boris T Polyak. Constrained minimization methods. *USSR Computational mathematics and mathematical physics*, 6(5):1–50, 1966.

[50] Xingguo Li, Raman Arora, Han Liu, Jarvis Haupt, and Tuo Zhao. Nonconvex sparse learning via stochastic optimization with progressive variance reduction. *arXiv preprint arXiv:1605.02711*, 2016.

[51] Xiangru Lian, Huan Zhang, Cho-Jui Hsieh, Yijun Huang, and Ji Liu. A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order. *Advances in Neural Information Processing Systems*, 29, 2016.

[52] Hongcheng Liu and Yu Yang. A dimension-insensitive algorithm for stochastic zeroth-order optimization. *arXiv preprint arXiv:2104.11283*, 2021.

[53] Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O Hero III, and Pramod K Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020.

[54] Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[55] Po-Ling Loh and Martin J Wainwright. Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *Advances in Neural Information Processing Systems*, 26, 2013.

[56] Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search of static linear policies is competitive for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[57] Simon Matet, Lorenzo Rosasco, Silvia Villa, and Bang Long Vu. Don't relax: early stopping for convex regularization. *arXiv preprint arXiv:1707.05422*, 2017.

[58] Andrew McDonald, Massimiliano Pontil, and Dimitris Stamos. Fitting spectral decay with the k-support norm. In *Artificial Intelligence and Statistics*, pages 1061–1069. PMLR, 2016.

[59] Andrew M McDonald, Massimiliano Pontil, and Dimitris Stamos. New perspectives on k-support and cluster norms. *The Journal of Machine Learning Research*, 17(1):5376–5413, 2016.

[60] Cesare Molinari, Mathurin Massias, Lorenzo Rosasco, and Silvia Villa. Iterative regularization for convex regularizers. In *International conference on artificial intelligence and statistics*, pages 1684–1692. PMLR, 2021.

[61] Thomas Moreau, Mathurin Massias, Alexandre Gramfort, Pierre Ablin, Pierre-Antoine Bannier, Benjamin Charlier, Mathieu Dagréou, Tom Dupré La Tour, Ghislain Durif, Cassio F Dantas, et al. Benchopt: Reproducible, efficient and collaborative optimization benchmarks. In *NeurIPS-36th Conference on Neural Information Processing Systems*, 2022.

[62] Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[63] Sahand Negahban, Bin Yu, Martin J Wainwright, and Pradeep Ravikumar. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Advances in neural information processing systems*, 22, 2009.

[64] Sahand N Negahban, Pradeep Ravikumar, Martin J Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical science*, 27(4):538–557, 2012.

[65] Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009.

[66] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

[67] Nam Nguyen, Deanna Needell, and Tina Woolf. Linear convergence of stochastic iterative greedy algorithms with sparse constraints. *IEEE Transactions on Information Theory*, 63(11):6869–6895, 2017.

[68] Stanley Osher, Feng Ruan, Jiechao Xiong, Yuan Yao, and Wotao Yin. Sparse recovery via differential inclusions. *Applied and Computational Harmonic Analysis*, 41(2):436–469, 2016.

[69] R Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

[70] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239, 2014.

[71] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[72] Roger Penrose. On best approximate solutions of linear matrix equations. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 52, pages 17–19. Cambridge University Press, 1956.

[73] Alexandra Peste, Eugenia Iofinova, Adrian Vladu, and Dan Alistarh. Ac/dc: Alternating compressed/decompressed training of deep neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[74] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE transactions on information theory*, 57(10):6976–6994, 2011.

[75] Soo-Yon Rhee, Jonathan Taylor, Gauhar Wadhera, Asa Ben-Hur, Douglas L Brutlag, and Robert W Shafer. Genotypic predictors of human immunodeficiency virus type 1 drug resistance. *Proceedings of the National Academy of Sciences*, 103(46):17355–17360, 2006.

[76] R. Tyrrell Rockafellar. Convex analysis, 1970.

[77] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[78] Todd E Scheetz, Kwang-Youn A Kim, Ruth E Swiderski, Alisdair R Philp, Terry A Braun, Kevin L Knudtson, Anne M Dorrance, Gerald F DiBona, Jian Huang, Thomas L Casavant, et al. Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, 103(39):14429–14434, 2006.

[79] Ohad Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *The Journal of Machine Learning Research*, 18(1):1703–1713, 2017.

[80] Jie Shen and Ping Li. A tight bound of hard thresholding. *The Journal of Machine Learning Research*, 18(1):7650–7691, 2017.

[81] Artem Sokolov, Julian Hitschler, Mayumi Ohta, and Stefan Riezler. Sparse stochastic zeroth-order optimization with an application to bandit structured prediction. *arXiv preprint arXiv:1806.04458*, 2018.

[82] Stanislav Sykora. Surface integrals over n-dimensional spheres. *Stan's Library*, (Volume I), May 2005.

[83] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[84] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[85] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007.

[86] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 742–749, 2019.

[87] Sara A Van de Geer. High-dimensional generalized linear models and the lasso. *The Annals of Statistics*, 36(2):614–645, 2008.

[88] Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.

[89] Tomas Vaskevicius, Varun Kanade, and Patrick Rebeschini. Implicit regularization for optimal sparse recovery. *Advances in Neural Information Processing Systems*, 32, 2019.

[90] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

[91] Christian Walck et al. Hand-book on statistical distributions for experimentalists. *University of Stockholm*, 10:96–01, 2007.

[92] Jian Wang, Suhyuk Kwon, Ping Li, and Byonghyo Shim. Recovery of sparse signals via generalized orthogonal matching pursuit: A new analysis. *IEEE Transactions on Signal Processing*, 64(4):1076–1089, 2015.

[93] Yining Wang, Simon Du, Sivaraman Balakrishnan, and Aarti Singh. Stochastic zeroth-order optimization in high dimensions. In *International Conference on Artificial Intelligence and Statistics*, pages 1356–1365. PMLR, 2018.

[94] John Wright and Yi Ma. *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications*. Cambridge University Press, 2022.

[95] Lin Xiao. Dual averaging method for regularized stochastic learning and online optimization. In *Advances in Neural Information Processing Systems*, volume 22, 2009.

[96] Xiao-Tong Yuan, Ping Li, and Tong Zhang. Gradient hard thresholding pursuit. *Journal of Machine Learning Research*, 18(1):6027–6069, 2017.

[97] Xiaotong Yuan and Ping Li. Stability and risk bounds of iterative hard thresholding. In *International Conference on Artificial Intelligence and Statistics*, pages 1702–1710. PMLR, 2021.

[98] Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942, 2010.

[99] Peng Zhao, Yun Yang, and Qiao-Chu He. High-dimensional linear regression via implicit regularization. *Biometrika*, 2022.

[100] Peng Zhao and Bin Yu. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006.

[101] Pan Zhou, Xiaotong Yuan, and Jiashi Feng. Efficient stochastic gradient hard thresholding. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[102] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

Appendix A

Appendix: Zeroth-Order Hard-Thresholding

A.1 Notations and Definitions

Throughout this appendix, we will use the following notations:

- we denote the vectors in bold letters.
- $\nabla f(\boldsymbol{x})$ denotes the gradient of f at $\boldsymbol{x}$.
- $[d]$ denotes the set of all integers between 1 and d: $\{1, .., d\}$.
- $\boldsymbol{u}_i$ denotes the i-th coordinate of vector $\boldsymbol{u}$, and $\nabla_i f(\boldsymbol{x})$ the i-th coordinate of $\nabla f(\boldsymbol{x})$.
- $\|\cdot\|_0$ denotes the ℓ_0 norm (which is not a proper norm).
- $\|\cdot\|$ denotes the ℓ_2 norm.
- $\|\cdot\|_\infty$ denotes the maximum absolute component of a vector.
- $\boldsymbol{x} \sim \mathcal{P}$ denotes that the random variable $\boldsymbol{X}$ (denoted as $\boldsymbol{x}$), of realization $\boldsymbol{x}$, follows a probability distribution $\mathcal{P}$ (we abuse notation by denoting similarly a random variable and its realization).
- $\boldsymbol{x}_1, .., \boldsymbol{x}_n \overset{i.i.d}{\sim} \mathcal{P}$ denotes that we draw n i.i.d. samples of a random variable $\boldsymbol{x}$, each from the distribution $\mathcal{P}$.
- $P(\boldsymbol{x})$ denotes the value of the probability of $\boldsymbol{x}$ according to its probability distribution.
- $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}}$ (or simply $\mathbb{E}_{\boldsymbol{x}}$ if there is no possible confusion) to denote the expectation of $\boldsymbol{x}$ which follows the distribution $\mathcal{P}$.
- We denote by $\text{supp}(\boldsymbol{x})$ the support of a vector $\boldsymbol{x}$, that is the set of its non-zero coordinates.

- $|F|$ the cardinality (number of elements) of a set F.
- All the sets we consider are subsets of $[d]$. So for a given set F, F^c denotes the complement of F in $[d]$
- $\mathcal{S}^d(R)$ (or $\mathcal{S}^d(R)$ for simplicity if $R = 1$) denotes the d-sphere of radius R, that is $\mathcal{S}^d(R) = \{\boldsymbol{u} \in \mathbb{R}^d / \|\boldsymbol{u}\| = R\}$.
- $\mathcal{U}(\mathcal{S}^d)$ the uniform distribution on that unit sphere.
- $\beta(d)$ is the surface area of the unit d-sphere defined above.
- $\mathcal{S}_S^d$ denotes a set that we call the restricted d-sphere on S, described as: $\{\boldsymbol{u}_S / \boldsymbol{u} \in \{\boldsymbol{v} \in \mathbb{R}^d / \|\boldsymbol{v}_S\| = 1\}\}$, that is the set of unit vectors supported by S.
- $\mathcal{U}(\mathcal{S}_S^d)$ denotes the uniform distribution on that restricted sphere above.
- We denote by $\boldsymbol{u}_F$ (resp. $\nabla_F f(\boldsymbol{x})$) the hard-thresholding of $\boldsymbol{u}$ (resp. $\nabla f(\boldsymbol{x})$) over the support F, that is, a vector which keeps $\boldsymbol{u}$ (resp. $\nabla f(\boldsymbol{x})$) untouched for the set of coordinates in F, but sets all other coordinates to 0.
- $\binom{[d]}{s}$ denotes the set of all subsets of $[d]$ that contain s elements: $\binom{[d]}{s} = \{S : |S| = s, S \subseteq [d]\}$.
- $\mathcal{U}(\binom{[d]}{s})$ denotes the uniform distribution on the set above.
- $\boldsymbol{I}$ denotes the identity matrix $\boldsymbol{I}_{d \times d}$.
- $\boldsymbol{I}_S$ denotes the identity matrix with 1 on the diagonal only at indices belonging to the support S: $\boldsymbol{I}_{i,i} = 1$ if $i \in S$, and 0 elsewhere.
- $S \ni e$ denotes that set S contains the element e.
- $(\boldsymbol{u}_i)_{i=1}^n$ denotes the n-uple of elements $\boldsymbol{u}_1, .., \boldsymbol{u}_n$.
- Γ denotes the Gamma function [2].
- $\int_A f(\boldsymbol{u}) d\boldsymbol{u}$ denotes the integral of f over the set A.
- log denotes the natural logarithm (in base e).

A.2 Auxilliary Lemmas

Lemma A.2.1 ([82] (10)). *Let $\boldsymbol{p} \in \mathbb{N}^d$, and denote $p := \sum_{i=1}^d \boldsymbol{p}_i$, we have:*

$$\int_{\mathcal{S}^d} \prod_{i=1}^d \boldsymbol{u}_i^{\boldsymbol{p}_i} d\boldsymbol{u} = 2 \frac{\prod_{i=1}^n \Gamma(\boldsymbol{p}_i + 1/2)}{\Gamma(p + d/2)}$$

Proof. The proof is given in [82]. □

Lemma A.2.2. *Let F be a subset of $[d]$, of size s, with $(s,d) \in \mathbb{N}_*^2$. We have the following:*

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\|\boldsymbol{u}_F\| \leq \sqrt{\frac{s}{d}} \tag{A.1}$$

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\|\boldsymbol{u}_F\|^2 = \frac{s}{d} \tag{A.2}$$

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\|\boldsymbol{u}_F\|^4 = \frac{(s+2)s}{(d+2)d} \tag{A.3}$$

Proof. We start by proving (A.2). Decomposing the norm onto every component, we get:

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\|\boldsymbol{u}_F\|^2 = \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\sum_{i\in F}\boldsymbol{u}_i^2 = \sum_{i\in F}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\boldsymbol{u}_i^2 \tag{A.4}$$

By symmetry, each $\boldsymbol{u}_i$ has the same marginal probability distribution, so:

$$\forall i \in [d]: \qquad \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\boldsymbol{u}_i^2 = \frac{1}{d}\sum_{i=1}^{d}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\boldsymbol{u}_i^2 \tag{A.5}$$

We also know, from the definition of the ℓ_2 norm, and the fact that $\boldsymbol{u}$ is a unit vector, that:

$$\sum_{i=1}^{d}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\boldsymbol{u}_i^2 = \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\sum_{i=1}^{d}\boldsymbol{u}_i^2 = \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\|\boldsymbol{u}\|^2 = \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}1 = 1 \tag{A.6}$$

Therefore, combining (A.5) and (A.6):

$$\forall i \in [d]: \qquad \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\boldsymbol{u}_i^2 = \frac{1}{d}$$

Plugging this into (A.4), we get (A.2):

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\|\boldsymbol{u}_F\|^2 = \frac{s}{d}$$

Using Jensen's inequality, (A.1) follows from (A.2). Let us now prove (A.3). By definition of the expectation for a uniform distribution on the unit sphere:

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\|\boldsymbol{u}_F\|^4 = \frac{1}{\beta(d)}\int_{\mathcal{S}^d}\|\boldsymbol{u}_F\|^4 d\boldsymbol{u}$$

We further develop the integral as follows:

$$\begin{aligned}\int_{\mathcal{S}^d}\|\boldsymbol{u}_F\|^4 d\boldsymbol{u} &= \int_{\mathcal{S}^d}(\|\boldsymbol{u}_F\|^2)^2 d\boldsymbol{u} = \int_{\mathcal{S}^d}(\sum_{i\in F}\boldsymbol{u}_i^4 + \sum_{(i,j)\in F, j\neq i}\boldsymbol{u}_i^2\boldsymbol{u}_j^2)d\boldsymbol{u} \\ &= s\int_{\mathcal{S}^d}\boldsymbol{u}_1^4 d\boldsymbol{u} + 2\binom{s}{2}\int_{\mathcal{S}^d}\boldsymbol{u}_1^2\boldsymbol{u}_2^2 d\boldsymbol{u} \quad \text{(by symmetry)}\end{aligned}$$

Using Lemma A.2.1 in the expression above, with $\boldsymbol{p}^{(a)} := (2, 0, ..., 0)$, and $\boldsymbol{p}^{(b)} := (1, 1, 0, ..., 0)$, we obtain:

$$\int_{\mathcal{S}^d} \|\boldsymbol{u}_F\|^4 d\boldsymbol{u} = s\frac{\prod_{i=1}^d \Gamma(\boldsymbol{p}_k^{(a)} + \frac{1}{2})}{\Gamma(2 + d/2)} + 2\frac{s(s-1)}{2}2\frac{\prod_{i=1}^d \Gamma(\boldsymbol{p}_k^{(b)} + 1/2)}{\Gamma(2 + d/2)}$$
$$\overset{(a)}{=} \frac{6s\sqrt{\pi}^d}{(d+2)d\Gamma(d/2)} + \frac{2s(s-1)\sqrt{\pi}^d}{(d+2)d\Gamma(d/2)} = \frac{2(s+2)s\sqrt{\pi}^d}{(d+2)d\Gamma(d/2)}$$

Where in (a) we used the fact that $\Gamma(\frac{1}{2}) = \sqrt{\pi}$ and $\Gamma(\frac{3}{2}) = \frac{\sqrt{\pi}}{2}$. So:

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\|\boldsymbol{u}_F\|^4 = \frac{1}{\beta(d)}\int_{S^d} \|\boldsymbol{u}_F\|^4 d\boldsymbol{u} \overset{(b)}{=} \frac{s+2}{d+2}\frac{s}{d}$$

Where (b) comes from the closed form for the area of a d unit sphere: $\beta(d) = \frac{2\sqrt{\pi}^d}{\Gamma(\frac{d}{2})}$ □

Lemma A.2.3 ([32], Lemma 7.3.b)**.**

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\boldsymbol{u}\boldsymbol{u}^T = \frac{1}{d}\boldsymbol{I}$$

Proof. The proof is given in [32]. □

Lemma A.2.4 ([80], Theorem 1; [96], Lemma 17)**.** *Let $\boldsymbol{b} \in \mathbb{R}^d$ be an arbitrary d-dimensional vector and $\boldsymbol{a} \in \mathbb{R}^d$ be any k-sparse vector. Denote $\bar{k} = \|\boldsymbol{a}\|_0 \leq k$, and $\boldsymbol{b}_k$ the vector $\boldsymbol{b}$ with all the $d-k$ smallest components set to 0 (that is, $\boldsymbol{b}_k$ is the best k-sparse approximation of $\boldsymbol{b}$). Then, we have the following bound:*

$$\|\boldsymbol{b}_k - \boldsymbol{a}\|^2 \leq \delta\|\boldsymbol{b} - \boldsymbol{a}\|^2, \quad \delta = 1 + \frac{\beta + \sqrt{(4+\beta)\beta}}{2}, \quad \beta = \frac{\min\{\bar{k}, d-k\}}{k - \bar{k} + \min\{\bar{k}, d-k\}}$$

Proof. The proof is given in [80]. □

Corollary A.2.1. *With the notations and variables above in Lemma A.2.4, we also have the following, simpler bound, from [96]:*

$$\|\boldsymbol{b}_k - \boldsymbol{a}\| \leq \gamma\|\boldsymbol{b} - \boldsymbol{a}\|$$

with

$$\gamma = \sqrt{1 + \left(\bar{k}/k + \sqrt{\left(4 + \bar{k}/k\right)\bar{k}/k}\right)/2}$$

Proof. There are two possibilities for β in Lemma A.2.4: either $\beta = \frac{\bar{k}}{k}$ (if $d - k > \bar{k}$) or $\beta = \frac{d-k}{d-\bar{k}}$ (if $d - k \leq \bar{k}$). In the latter case:

$$d-k \leq \bar{k} \implies d-\bar{k} \leq k \implies \frac{k-\bar{k}}{d-\bar{k}} \geq \frac{k-\bar{k}}{k} \implies 1-\frac{k-\bar{k}}{d-\bar{k}} \leq 1-\frac{k-\bar{k}}{k} \implies \frac{d-k}{d-\bar{k}} \leq \frac{\bar{k}}{k}$$

Therefore, in both cases, $\beta \leq \frac{\bar{k}}{k}$, which, plugging into Lemma A.2.4, gives Corollary A.2.1. □

A.3 Proof of Proposition 1

With an abuse of notation, let us denote by f any function $f_{\boldsymbol{\xi}}$ for some given value of the noise $\boldsymbol{\xi}$. First, we derive in section A.3.1 the error of the gradient estimate if we sample only one direction ($q = 1$). Then, in section A.3.2, we show how sampling q directions reduces the error of the gradient estimator, producing the results of Proposition 1.

A.3.1 One direction estimator

Throughout all this section, we assume that $q = 1$ for the gradient estimator $\hat{\nabla} f(x)$ defined in (3.1).

Expected deviation from the mean

Lemma A.3.1. *For any (L_{s_2}, s_2)-RSS function f, using the gradient estimator $\hat{\nabla} f(x)$ defined in (3.1) with $q = 1$, we have, for any support $F \in [d]$, with $|F| = s$:*

$$\left\| \mathbb{E}\left[\hat{\nabla}_F f(\boldsymbol{x})\right] - \nabla_F f(\boldsymbol{x}) \right\|^2 \leq \varepsilon_\mu \mu^2$$

with $\varepsilon_\mu = L_{s_2}^2 sd$

Proof. From the definition of the gradient estimator in (3.1):

$$\|\mathbb{E}[\hat{\nabla}_F f(\boldsymbol{x})] - \nabla_F f(\boldsymbol{x})\| = \left\| \mathbb{E} d \frac{f(\boldsymbol{x} + \mu \boldsymbol{u}) - f(\boldsymbol{x})}{\mu} \boldsymbol{u}_F - \nabla_F f(\boldsymbol{x}) \right\|$$

Now, (L_{s_2}, s_2)-RSS implies continuous differentiability over an s_2-sparse direction (since (L_{s_2}, s_2)-RSS actually equals Lipschitz continuity of the gradient over any s_2-sparse set, which implies continuity of the gradient over those sets). Therefore, from the mean value theorem, , we have, for some $c \in [0, \mu]$: $\frac{f(\boldsymbol{x}+\mu\boldsymbol{u})-f(\boldsymbol{x})}{\mu} = \langle \nabla f(\boldsymbol{x} + c\boldsymbol{u}), \boldsymbol{u} \rangle$. We now use the following result:

$$\mathbb{E}\boldsymbol{u}\boldsymbol{u}^T = \mathbb{E}_{S \sim \binom{[d]}{s_2}} \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}_S^d)} \boldsymbol{u}\boldsymbol{u}^T \overset{(a)}{=} \mathbb{E}_{S \sim \binom{[d]}{s_2}} \frac{1}{s_2} \boldsymbol{I}_S = \frac{1}{s_2} \mathbb{E}_{S \sim \binom{[d]}{s_2}} \boldsymbol{I}_S \overset{(b)}{=} \frac{1}{s_2} \frac{s_2}{d} \boldsymbol{I} = \frac{1}{d} \boldsymbol{I}$$

Where for (a) comes from applying Lemma A.2.3 to the unit sub-sphere on the support S, and (b) follows by observing that each diagonal element of index i actually follows a Bernoulli distribution of parameter $\frac{s_2}{d}$, since there are $\binom{d-1}{s_2-1}$ arrangements of the support which contain i, over $\binom{d}{s_2}$ total arrangements, which gives a probability $p = \frac{\binom{d-1}{s_2-1}}{\binom{d}{s_2}} = \frac{(d-1)!s_2!(d-s_2)!}{(s_2-1)!(d-1-(s_2-1))!d!} = \frac{s_2}{d}$ to get the value 1 at i.

This allows to factor the true gradient into the scalar product:

$$\begin{aligned}\|\mathbb{E}[\hat{\nabla}_F f(\boldsymbol{x})] - \nabla_F f(\boldsymbol{x})\| &= d\|\mathbb{E}\langle \nabla f(\boldsymbol{x}+c\boldsymbol{u}) - \nabla f(\boldsymbol{x}), \boldsymbol{u}\rangle \boldsymbol{u}_F\| \\ &\leq d\mathbb{E}\|\boldsymbol{u}_F \boldsymbol{u}^T[\nabla f(\boldsymbol{x}+c\boldsymbol{u}) - \nabla f(\boldsymbol{x})]\|\end{aligned}$$

where the last inequality follows from the property $\mathbb{E}\|\boldsymbol{X} - \mathbb{E}\boldsymbol{X}\|^2 = \mathbb{E}\|\boldsymbol{X}\|^2 - \|\mathbb{E}\boldsymbol{X}\|^2$, which implies $\|\mathbb{E}\boldsymbol{X}\| = \sqrt{\mathbb{E}\|\boldsymbol{X}\|^2 - \mathbb{E}\|(\boldsymbol{X} - \mathbb{E}\boldsymbol{X})\|^2} \leq \mathbb{E}\|\boldsymbol{X}\|$, for any multidimensional random variable $\boldsymbol{X}$. Using the Cauchy-Schwarz inequality, we obtain:

$$\|\mathbb{E}[\hat{\nabla}_F f(\boldsymbol{x})] - \nabla_F f(\boldsymbol{x})\| \leq \mathbb{E}_{S\sim\binom{[d]}{s_2}} \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)} \|\boldsymbol{u}_F\|\|\boldsymbol{u}\|\|\nabla_S f(\boldsymbol{x}+c\boldsymbol{u}) - \nabla_S f(\boldsymbol{x})\|$$

Since $f \in (L_{s_2}, s_2)$-RSS and $\|\boldsymbol{u}_s\|_0 \leq s_2$, we have: $\|\nabla_S f(\boldsymbol{x}+c\boldsymbol{u}) - \nabla_S f(\boldsymbol{x})\| \leq L_{s_2}\|c\boldsymbol{u}\|$. We also have $c \in [0, \mu]$, which implies $\|c\boldsymbol{u}\| \leq \mu\|\boldsymbol{u}\|$. Therefore:

$$\begin{aligned}\|\mathbb{E}[\hat{\nabla}_F f(\boldsymbol{x})] - \nabla_F f(\boldsymbol{x})\| &\leq \mathbb{E}_S \mathbb{E}_{\boldsymbol{u}} d L_{s_2}\mu \|\boldsymbol{u}_F\|\|\boldsymbol{u}\|\|\boldsymbol{u}\| = \mathbb{E}_S \mathbb{E}_{\boldsymbol{u}} d L_{s_2}\mu \|\boldsymbol{u}_F\|\|\boldsymbol{u}\|^2 = \mathbb{E}_S \mathbb{E}_{\boldsymbol{u}} d L_{s_2}\mu \|\boldsymbol{u}_F\| \\ &\overset{(a)}{\leq} d L_{s_2}\mu \mathbb{E}_S \mathbb{E}_{\boldsymbol{u}} \sqrt{\frac{|S\cap F|}{s_2}} \\ &\overset{(b)}{\leq} d L_{s_2}\mu \sqrt{\mathbb{E}_S \frac{|S\cap F|}{s_2}} = d L_{s_2}\mu \sqrt{\mathbb{E}_k \mathbb{E}_{S||S\cap F|=k} \frac{k}{s_2}} \\ &= d L_{s_2}\mu \sqrt{\frac{s s_2}{d s_2}} = L_{s_2}\mu\sqrt{sd}\end{aligned}$$

Where (a) follows from Lemma A.2.2, restricted to the support S, and (b) follows from Jensen's inequality. □

Expected norm

Lemma A.3.2. *For any (L_{s_2}, s_2)-RSS function f, using the gradient estimator $\hat{\nabla} f(x)$ defined in (3.1) with $q = 1$, we have, for any support $F \in [d]$, with $|F| = s$:*

$$\mathbb{E}\|\hat{\nabla}_F f(\boldsymbol{x})\|^2 = \varepsilon_F \|\nabla_F f(\boldsymbol{x})\|^2 + \varepsilon_{F^c}\|\nabla_{F^c} f(\boldsymbol{x})\|^2 + \varepsilon_{abs}\mu^2$$

with:
(i) $\varepsilon_F = \frac{2d}{(s_2+2)}\left(\frac{(s-1)(s_2-1)}{d-1} + 3\right)$
(ii) $\varepsilon_{F^c} = \frac{2d}{(s_2+2)}\left(\frac{s(s_2-1)}{d-1}\right)$
(iii) $\varepsilon_{abs} = 2dL_s^2 s s_2\left(\frac{(s-1)(s_2-1)}{d-1} + 1\right)$

Proof.

$$\begin{aligned}\mathbb{E}\|\hat{\nabla}_F f(\boldsymbol{x})\|^2 &= \mathbb{E}\left\|d\frac{f(\boldsymbol{x}+\mu\boldsymbol{u}) - f(\boldsymbol{x})}{\mu}\boldsymbol{u}_F\right\|^2 \\ &= \mathbb{E}\frac{d^2}{\mu^2}|f(\boldsymbol{x}+\mu\boldsymbol{u}) - f(\boldsymbol{x})|^2\|\boldsymbol{u}_F\|^2 \\ &= \frac{d^2}{\mu^2}\mathbb{E}[f(\boldsymbol{x}+\mu\boldsymbol{u}) - f(\boldsymbol{x}) - \langle\nabla f(\boldsymbol{x}), \mu\boldsymbol{u}\rangle + \langle\nabla f(\boldsymbol{x}), \mu\boldsymbol{u}\rangle)]^2\|\boldsymbol{u}_F\|^2\end{aligned}$$

Using the mean value theorem, we obtain that for a certain $c \in (0, \mu)$, we have:

$$f(\boldsymbol{x} + \mu\boldsymbol{u}) - f(\boldsymbol{x}) = \langle \nabla f(\boldsymbol{x} + c), \mu\boldsymbol{u} \rangle$$

Therefore, plugging this in the above:

$$\begin{aligned}
\mathbb{E}\|\hat{\nabla}_F f(\boldsymbol{x})\|^2 &\leq d^2 \mathbb{E}[\langle \nabla f(\boldsymbol{x} + c\boldsymbol{u}) - \nabla f(\boldsymbol{x}), \boldsymbol{u} \rangle + \langle \nabla f(\boldsymbol{x}), \boldsymbol{u} \rangle]^2 \|\boldsymbol{u}_F\|^2 \\
&\overset{(a)}{\leq} d^2 \mathbb{E}\left[2\langle \nabla f(\boldsymbol{x} + c\boldsymbol{u}) - \nabla f(\boldsymbol{x}), \boldsymbol{u} \rangle^2 \|\boldsymbol{u}_F\|^2 + \langle \nabla f(\boldsymbol{x}), \boldsymbol{u} \rangle^2 \|\boldsymbol{u}_F\|^2\right] \\
&\leq 2d^2 \mathbb{E}[\|\nabla f(\boldsymbol{x} + c\boldsymbol{u}) - \nabla f(\boldsymbol{x})\|^2 \|\boldsymbol{u}\|^2 \|\boldsymbol{u}_F\|^2 + \langle \nabla f(\boldsymbol{x}), \boldsymbol{u} \rangle^2 \|\boldsymbol{u}_F\|^2] \\
&\overset{(b)}{\leq} 2d^2 \mathbb{E}[L_s^2 \mu^2 \|\boldsymbol{u}\|^2 \|\boldsymbol{u}\|^2 \|\boldsymbol{u}_F\|^2 + \langle \nabla f(\boldsymbol{x}), \boldsymbol{u} \rangle^2 \|\boldsymbol{u}_F\|^2] \\
&\overset{(c)}{=} 2d^2 \mathbb{E}[L_s^2 \mu^2 \|\boldsymbol{u}_F\|^2 + \langle \nabla f(\boldsymbol{x}), \boldsymbol{u} \rangle^2 \|\boldsymbol{u}_F\|^2] \\
&= 2d^2 [L_s^2 \mu^2 \mathbb{E}\|\boldsymbol{u}_F\|^2 + \nabla f(\boldsymbol{x})^T \left(\mathbb{E}\boldsymbol{u}\boldsymbol{u}^T \|\boldsymbol{u}_F\|^2\right) \nabla f(\boldsymbol{x})] \\
&= 2d^2 [L_{s_2}^2 \mu^2 \mathbb{E}\|\boldsymbol{u}_F\|^2 + \nabla f(\boldsymbol{x})^T (\mathbb{E}_{S \sim \binom{[d]}{s_2}} \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}_S^d)} \boldsymbol{u}\boldsymbol{u}^T \|\boldsymbol{u}_F\|^2) \nabla f(\boldsymbol{x})] \\
&\overset{(d)}{=} 2d^2 [L_{s_2}^2 \mu^2 \mathbb{E}\|\boldsymbol{u}_F\|^2 + \mathbb{E}_{S \sim \binom{[d]}{s_2}} [\nabla f(\boldsymbol{x})^T (\mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}_S^d)} \boldsymbol{u}\boldsymbol{u}^T \|\boldsymbol{u}_F\|^2) \nabla f(\boldsymbol{x})]] \quad \text{(A.7)}
\end{aligned}$$

Where (a) follows from the fact that for any $(a, b) \in \mathbb{R}^2 : (a+b)^2 \leq 2a^2 + 2b^2$, (b) follows from the Cauchy-Schwarz inequality, (c) follows from the fact that $\|\boldsymbol{u}\| = 1$ since $\boldsymbol{u} \in \mathcal{S}_S^d$, and (d) follows by linearity of expectation. Let us turn to computing the following expression above: $\mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}_S^d)} \boldsymbol{u}\boldsymbol{u}^T \|\boldsymbol{u}_F\|^2$. We start by distinguishing the indices that belong to F and those that do not. By symmetry, denoting $i_1, ..., i_s$ the elements of F:

$$\mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}_S^d)} \boldsymbol{u}_{i_1}^2 \|\boldsymbol{u}_F\|^2 = ... = \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}_S^d)} \boldsymbol{u}_{i_s}^2 \|\boldsymbol{u}_F\|^2$$

Therefore, for all $i \in F$:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}_S^d)} \boldsymbol{u}_i^2 \|\boldsymbol{u}_F\|^2 &= \frac{1}{|S \cap F|} \sum_{j=1}^{s} \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}_S^d)} \boldsymbol{u}_{i_j}^2 \|\boldsymbol{u}_F\|^2 \\
&= \frac{1}{|S \cap F|} \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}_S^d)} \sum_{j=1}^{s} \boldsymbol{u}_{i_j}^2 \|\boldsymbol{u}_F\|^2 = \frac{1}{|S \cap F|} \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}_S^d)} \|\boldsymbol{u}_F\|^4 \quad \text{(A.8)}
\end{aligned}$$

By definition of the restricted d-sphere on F (see section A.1), for all $\boldsymbol{u} \in \mathcal{S}_S^d$, if $i \notin S$: $\boldsymbol{u}_i = 0$. Therefore, since the exact indices of the elements of F do not matter in the expected value (A.8), but only their cardinality, (A.8) can be rewritten using a simpler expectation over a unit $|S|$-sphere as follows :

$$\mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}_S^d)} \|\boldsymbol{u}_F\|^4 = \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^{|S|})} \|\boldsymbol{u}_{[|S \cap F|]}\|^4$$

Using Lemma A.2.2 to get a closed form expression of the expected value above, we further obtain:

$$\forall i \in F : \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d)} \boldsymbol{u}_i^2 \|\boldsymbol{u}_F\|^2 = \frac{1}{|S \cap F|} \frac{|S \cap F|(|S \cap F| + 2)}{d(d+2)} = \frac{|S \cap F| + 2}{d(d+2)} \quad \text{(A.9)}$$

Similarly, by symmetry, denoting $i_1, ..., i_{d-s}$ the elements of F^c:

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\boldsymbol{u}_{i_j}^2\|\boldsymbol{u}_F\|^2 = ... = \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\boldsymbol{u}_{i_j}^2\|\boldsymbol{u}_F\|^2$$

Therefore, for all $i \notin F$:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\boldsymbol{u}_i^2\|\boldsymbol{u}_F\|^2 &= \frac{1}{d-s}\sum_{j=1}^{d-s}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\boldsymbol{u}_{i_j}^2\|\boldsymbol{u}_F\|^2 = \frac{1}{d-s}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\sum_{j=1}^{d-s}\boldsymbol{u}_{i_j}^2\|\boldsymbol{u}_F\|^2 \\ &\overset{(a)}{=} \frac{1}{d-s}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}(\|\boldsymbol{u}\|^2 - \|\boldsymbol{u}_F\|^2)\|\boldsymbol{u}_F\|^2 \\ &\overset{(b)}{=} \frac{1}{d-s}(\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\|\boldsymbol{u}_F\|^2 - \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\|\boldsymbol{u}\|^4)\end{aligned}$$

Where (a) follows from the Pythagorean theorem and (b) follows from $\|\boldsymbol{u}\| = 1$. Similarly as before, rewriting those expected values and using Lemma A.2.2, we obtain:

$$\forall i \notin F : \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\boldsymbol{u}_i^2\|\boldsymbol{u}_F\|^2 = \frac{1}{d-|S\cap F|}\frac{|S\cap F|(d+2-(|S\cap F|+2))}{d(d+2)} = \frac{|S\cap F|}{d(d+2)} \tag{A.10}$$

Finally, by symmetry of the distribution $\mathcal{U}(\mathcal{S}_S^d)$, we have, for all $(i,j) \in [d]^2$ with $i \neq j$:

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\boldsymbol{u}_i\boldsymbol{u}_j\|\boldsymbol{u}_F\|^2 = \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}(-\boldsymbol{u}_i)\boldsymbol{u}_j\|\boldsymbol{u}_F\|^2 = -\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\boldsymbol{u}_i\boldsymbol{u}_j\|\boldsymbol{u}_F\|^2$$

Therefore, for all $(i,j) \in [d]^2, i \neq j$:

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\boldsymbol{u}_i\boldsymbol{u}_j\|\boldsymbol{u}_F\|^2 = 0 \tag{A.11}$$

Therefore, combining (A.9), (A.10) and (A.11), we obtain:

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\boldsymbol{u}\boldsymbol{u}^T\|\boldsymbol{u}_F\|^2 = \begin{bmatrix} a_1 & & & \\ & a_2 & & \\ & & \ddots & \\ & & & a_d \end{bmatrix}$$

With, for all $i \in [d] : a_i = \begin{cases} \frac{|S\cap F|+2}{d(d+2)} \text{ if } i \in F \\ \frac{|S\cap F|}{d(d+2)} \text{ if } i \notin F \end{cases}$. Plugging this back into (A.7), we obtain:

$$\begin{aligned}A &:= \mathbb{E}_{S\sim\binom{[d]}{s_2}}[\nabla f(\boldsymbol{x})^T\left(\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\boldsymbol{u}\boldsymbol{u}^T\|\boldsymbol{u}_F\|^2\right)\nabla f(\boldsymbol{x})] \\ &= \mathbb{E}_{S\sim\binom{[d]}{s_2}}\left[\frac{|S\cap F|+2}{s_2(s_2+2)}\|\nabla_{S\cap F}f(\boldsymbol{x})\|^2 + \frac{|S\cap F|}{s_2(s_2+2)}\|\nabla_{S\setminus(S\cap F)}f(\boldsymbol{x})\|^2\right] \\ &= \frac{1}{s_2(s_2+2)}\Big[\mathbb{E}_{S\sim\binom{[d]}{s_2}}\left[|S\cap F|\ \|\nabla_{F\cap S}f(\boldsymbol{x})\|^2\right] \\ &\quad +2\mathbb{E}_{S\sim\binom{[d]}{s_2}}\left[\|\nabla_{F\cap S}f(\boldsymbol{x})\|^2 + |S\cap F|\ \|\nabla_{S\setminus(S\cap F)}f(\boldsymbol{x})\|^2\right]\Big]\end{aligned} \tag{A.12}$$

We will now develop the expected values above using the law of total expectation, to exhibit the role of the random variable k denoting the size of $S \cap F$. Given that we sample s_2 indices from $[d]$ without replacement, k follows a hypergeometric distribution with, as parameters, population size d, number of success states s and number of draws s_2, which we denote $\mathcal{H}(d, s, s_2)$. For simplicity, we will use the following notations for the expected values: $\mathbb{E}_k[\cdot] := \mathbb{E}_{k\sim\mathcal{H}(d,s,s_2)}[\cdot]$, and $\mathbb{E}_{S||S\cap F|=k}[\cdot] = \mathbb{E}_{S\sim\binom{[d]}{s_2}||S\cap F|=k}[\cdot]$. Therefore, rewriting (A.12) using the law of total expectation, we obtain:

$$\begin{aligned}
A &= \frac{1}{s_2(s_2+2)}\Big[\mathbb{E}_k\mathbb{E}_{S||S\cap F|=k}k\|\nabla_{S\cap F}f(\boldsymbol{x})\|^2 + 2\mathbb{E}_k\mathbb{E}_{S||S\cap F|=k}\|\nabla_{S\cap F}f(\boldsymbol{x})\|^2 \\
&\quad +\mathbb{E}_k\mathbb{E}_{S||S\cap F|=k}k\|\nabla_{S\setminus(S\cap F)}f(\boldsymbol{x})\|^2\Big] \\
&= \frac{1}{s_2(s_2+2)}\Big[\mathbb{E}_k k\mathbb{E}_{S||S\cap F|=k}\|\nabla_{S\cap F}f(\boldsymbol{x})\|^2 + 2\mathbb{E}_S\mathbb{E}_{S||S\cap F|=k}\|\nabla_{S\cap F}f(\boldsymbol{x})\|^2 \\
&\quad +\mathbb{E}_k k\mathbb{E}_{S||S\cap F|=k}\|\nabla_{S\setminus(S\cap F)}f(\boldsymbol{x})\|^2\Big] \qquad \text{(A.13)}
\end{aligned}$$

To compute the conditional expectations above, let us consider the first of them (the other ones will follow similarly) : $\mathbb{E}_{S||S\cap F|=k}\|\nabla_{S\cap F}f(\boldsymbol{x})\|^2$. Given some k, from the multiplication principle in combinatorics, we can have $\binom{d}{k}\binom{d-s}{s_2-k}$ arrangements of supports such that k elements of that support are in F (because it means there are k elements in F and $s_2 - k$ elements outside of F). So the conditional probability of each of those supports S, assuming they indeed have at least one element in common with F, is $\left(\binom{d}{k}\binom{d-s}{s_2-k}\right)^{-1}$. Otherwise it is 0. To rewrite it:

$$P(S||S\cap F| = k) = \begin{cases} \left(\binom{d}{k}\binom{d-s}{s_2-k}\right)^{-1} \text{ if } S\cap F \neq \varnothing \\ 0 \text{ if } S\cap F \neq \varnothing \end{cases}$$

So, developing $\mathbb{E}_{S||S\cap F|=k}\|\nabla_{S\cap F}f(\boldsymbol{x})\|^2$ using the definition of conditional probability, we

have:

$$\begin{aligned}
\mathbb{E}_{S||S\cap F|=k}\|\nabla_{S\cap F}f(\boldsymbol{x})\|^2 &= \sum_S P(S|\ |S\cap F|=k)\sum_{i\in S\cap F}\nabla_i f(\boldsymbol{x})^2 \\
&= \sum_{S/|S\cap F|=k}\left(\binom{d}{k}\binom{d-s}{s_2-k}\right)^{-1}\sum_{i\in S\cap F}\nabla_i f(\boldsymbol{x})^2 \\
&= \left(\binom{d}{k}\binom{d-s}{s_2-k}\right)^{-1}\sum_{S/|S\cap F|=k}\sum_{i\in S\cap F}\nabla_i f(\boldsymbol{x})^2 \\
&\overset{(a)}{=} \left(\binom{d}{k}\binom{d-s}{s_2-k}\right)^{-1}\sum_{i\in F}\sum_{S/((|S\cap F|=k),(S\ni i))}\nabla_i f(\boldsymbol{x})^2 \\
&\overset{(b)}{=} \left(\binom{d}{k}\binom{d-s}{s_2-k}\right)^{-1}\sum_{i\in F}\binom{s-1}{k-1}\binom{d-s}{s_2-k}\nabla_i f(\boldsymbol{x})^2 \\
&= \frac{s}{k}\sum_{i\in F}\nabla_i f(\boldsymbol{x})^2 \\
&= \frac{s}{k}\|\nabla_F f(\boldsymbol{x})\|^2 \qquad \text{(A.14)}
\end{aligned}$$

Where (a) follows by re-arranging the sum, and (b) follows by observing that by the multiplication principle, there are $\binom{s-1}{k-1}\binom{d-s}{s_2-k}$ possible arrangements of support such that: $(|S\cap F|=k),(S\ni i)$, since one element of S is already fixed to be i, so there remains $k-1$ indices to arrange over $s-1$ possibilities, and still s_2-k indices to arrange over $d-s$ possibilities. Similarly, to (A.14) we have, for the second expectation:

$$\mathbb{E}_{S||S\cap F|=k}\|\nabla_{S\setminus(S\cap F)}f(\boldsymbol{x})\|^2 = \frac{s_2-k}{d-s}\|\nabla_{F^c}f(\boldsymbol{x})\|^2 \qquad \text{(A.15)}$$

Therefore, plugging (A.14) and (A.15) into (A.13)

$$\begin{aligned}
A &= \frac{1}{s_2(s_2+2)}\left[\mathbb{E}_k k\frac{k}{s}\|\nabla_F f(\boldsymbol{x})\|^2 + 2\mathbb{E}_k\frac{k}{s}\|\nabla_F f(\boldsymbol{x})\|^2 + \mathbb{E}_k k\frac{s_2-k}{d-s}\|\nabla_{F^c}f(\boldsymbol{x})\|^2\right] \\
&= \frac{1}{s_2(s_2+2)}\left[\frac{1}{s}\|\nabla_F f(\boldsymbol{x})\|^2\left[\mathbb{E}_k k^2 + 2\mathbb{E}_k k\right] + \|\nabla_{F^c}f(\boldsymbol{x})\|^2\left[\frac{s_2}{d-s}\left(\mathbb{E}_k k\right) - \frac{1}{d-s}\mathbb{E}_k k^2\right]\right] \qquad \text{(A.16)}
\end{aligned}$$

Since k follows a hypergeometric distribution $\mathcal{H}(d,s,s_2)$, its expected value is given in closed form by: $\mathbb{E}_k k = \frac{ss_2}{d}$ (see [91], section 2.1.3). We can also express the non-centered moment of order 2, using the formula for $\mathrm{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$, which holds for a random variable X, where $Var(X)$ denotes the variance of X:

$$\begin{aligned}
\mathbb{E}_k k^2 = Var(k) + (\mathbb{E}_k[k])^2 &\overset{(a)}{=} \frac{ss_2}{d}\frac{d-s}{d}\frac{d-s_2}{d-1} + \left(\frac{ss_2}{d}\right)^2 = \frac{ss_2}{d}\left(\frac{d-s}{d}\frac{d-s_2}{d-1} + \frac{ss_2}{d}\right) \\
&= \frac{ss_2}{d}\left(\frac{d^2 - sd - s_2 d + ss_2 + ss_2 d - ss_2}{d(d-1)}\right) = \frac{ss_2}{d}\left(\frac{d-s-s_2+ss_2}{d-1}\right) \\
&= \frac{ss_2}{d}\left(\frac{(s-1)(s_2-1)}{d-1} + 1\right)
\end{aligned}$$

Where (a) follows by the closed form for the variance of a hypergeometric variable given in [91]. Therefore, plugging in into (A.16):

$$\begin{aligned}
&\mathbb{E}_S \nabla f(\boldsymbol{x})^T \left(\mathbb{E}_{\mathcal{U}_S|S} \boldsymbol{u}\boldsymbol{u}^T \|\boldsymbol{u}_F\|^2\right) \nabla f(\boldsymbol{x}) \\
&= \frac{1}{s_2(s_2+2)} \left[\frac{1}{s}\|\nabla_F f(\boldsymbol{x})\|^2 \left[\frac{ss_2}{d}\left(\frac{(s-1)(s_2-1)}{d-1}+1\right)+2\frac{ss_2}{d}\right]\right] \\
&\quad + \frac{1}{s_2(s_2+2)}\|\nabla_{F^c} f(\boldsymbol{x})\|^2 \left[\frac{s_2}{d-s}\frac{ss_2}{d} - \frac{1}{d-s}\frac{ss_2}{d}\left(\frac{(s-1)(s_2-1)}{d-1}+1\right)\right] \\
&= \frac{1}{s_2+2}\left[\|\nabla_F f(\boldsymbol{x})\|^2 \left[\frac{1}{d}\left(\frac{(s-1)(s_2-1)}{d-1}+3\right)\right]\right. \\
&\quad \left. + \|\nabla_{F^c} f(\boldsymbol{x})\|^2 \left[\frac{s}{(d-s)d}\left(s_2 - \left(\frac{(s-1)(s_2-1)}{d-1}+1\right)\right)\right]\right] \\
&= \frac{1}{d(s_2+2)}\left[\|\nabla_F f(\boldsymbol{x})\|^2 \left[\left(\frac{(s-1)(s_2-1)}{d-1}+3\right)\right]\right. \\
&\quad \left. + \|\nabla_{F^c} f(\boldsymbol{x})\|^2 \left[\frac{s}{(d-s)}\left(s_2 - \left(\frac{(s-1)(s_2-1)}{d-1}+1\right)\right)\right]\right] \qquad \text{(A.17)}
\end{aligned}$$

Let us simplify the rightmost term:

$$\begin{aligned}
\frac{s}{(d-s)}\left(s_2 - \left(\frac{(s-1)(s_2-1)}{d-1}+1\right)\right) &= \frac{s(s_2-1)}{d-s}\left[1-\frac{s-1}{d-1}\right] \\
&= \frac{s(s_2-1)}{(d-s)}\left[\frac{d-s}{d-1}\right] = \frac{s(s_2-1)}{d-1}
\end{aligned}$$

Plugging it back into (A.17):

$$\begin{aligned}
&\mathbb{E}_S \nabla f(\boldsymbol{x})^T \left(\mathbb{E}_{\mathcal{U}_S|S} \boldsymbol{u}\boldsymbol{u}^T \|\boldsymbol{u}_F\|^2\right) \nabla f(\boldsymbol{x}) \\
&= \frac{1}{d(s_2+2)}\left[\|\nabla_F f(\boldsymbol{x})\|^2 \left(\frac{(s-1)(s_2-1)}{d-1}+3\right) + \|\nabla_{F^c} f(\boldsymbol{x})\|^2 \left(\frac{s(s_2-1)}{d-1}\right)\right]
\end{aligned}$$

Finally, plugging this back into (A.7):

$$\begin{aligned}
\mathbb{E}\|\hat{\nabla}_F f(\boldsymbol{x})\|^2 &= 2d^2\left[L_{s_2}^2\mu^2\mathbb{E}\|\boldsymbol{u}_F\|^2 + \nabla f(\boldsymbol{x})^T\left(\mathbb{E}\boldsymbol{u}\boldsymbol{u}^T\|\boldsymbol{u}_F\|^2\right)\nabla f(\boldsymbol{x})\right] \\
&= 2d^2\left[L_{s_2}^2\mu^2\mathbb{E}_k\mathbb{E}_{\boldsymbol{u}||S\cap F|=k}\|\boldsymbol{u}_F\|^2 + \nabla f(\boldsymbol{x})^T\left(\mathbb{E}\boldsymbol{u}\boldsymbol{u}^T\|\boldsymbol{u}_F\|^2\right)\nabla f(\boldsymbol{x})\right] \\
&= 2d^2\left[L_{s_2}^2\mu^2\mathbb{E}_k k^2 + \nabla f(\boldsymbol{x})^T\left(\mathbb{E}\boldsymbol{u}\boldsymbol{u}^T\|\boldsymbol{u}_F\|^2\right)\nabla f(\boldsymbol{x})\right] \\
&= d2L_{s_2}^2\mu^2 ss_2\left(\frac{(s-1)(s_2-1)}{d-1}+1\right) \\
&\quad + \frac{2d}{(s_2+2)}\left[\|\nabla_F f(\boldsymbol{x})\|^2\left(\frac{(s-1)(s_2-1)}{d-1}+3\right) + \|\nabla_{F^c} f(\boldsymbol{x})\|^2\left(\frac{s(s_2-1)}{d-1}\right)\right]
\end{aligned}$$

□

A.3.2 Batched-version of the one-direction estimator

We now describe how sampling $q \geq 1$ random directions improves the gradient estimate. Our proof is similar to the proof of Lemma 2 in [54], however we make sure that it works for our random support gradient estimator, and with our new expression in A.3.2, which depends on the two terms $\|\nabla_F f(\boldsymbol{x})\|^2$ and $\|\nabla_{F^c} f(\boldsymbol{x})\|^2$. We express our results here in the form of a general lemma, depending only on the general bounding factors ε_F, ε_{F^c}, ε_{abs} and ε_μ defined below, in such a way that the proof of Proposition 1 follows immediately from plugging the results of Lemma A.3.1 and A.3.2 into Lemma A.3.3 below.

Lemma A.3.3. *For any (L_{s_2}, s_2)-RSS function f, we use the gradient estimator $\hat{\nabla} f(x)$ defined in (3.1) with $q \geq 1$. Let us suppose that the estimator $\hat{\nabla} f(x)$ is such that for $q = 1$, it verifies the following bounds for some ε_F, ε_{F^c}, ε_{abs} and ε_μ in $\mathbb{R}_+^*$, for any support $F \in [d]$, with $|F| = s$:*
(i) $\|\mathbb{E}\hat{\nabla}_F f(\boldsymbol{x}) - \nabla_F f(\boldsymbol{x})\|^2 \leq \varepsilon_\mu \mu^2$, *and*
(ii) $\|\mathbb{E}\hat{\nabla}_F f(\boldsymbol{x})\|^2 \leq \varepsilon_F \|\nabla_F f(\boldsymbol{x})\|^2 + \varepsilon_{F^c} \|\nabla_{F^c} f(x)\|^2 + \varepsilon_{abs} \mu^2$
Then, the estimator $\hat{\nabla} f(x)$ also verifies, for arbitrary $q \geq 1$:
(a) $\|\mathbb{E}\hat{\nabla}_F f(\boldsymbol{x}) - \nabla_F f(\boldsymbol{x})\|^2 \leq \varepsilon_\mu \mu^2$
(b) $\mathbb{E}\left\|\hat{\nabla}_F f(\boldsymbol{x})\right\|^2 \leq \left(\frac{\varepsilon_F}{q} + 2\right) \|\nabla_F f(\boldsymbol{x})\|^2 + \frac{\varepsilon_{F^c}}{q} \|\nabla_{F^c} f(\boldsymbol{x})\|^2 + \left(\frac{\varepsilon_{abs}}{q} + 2\varepsilon_\mu\right) \mu^2$

Proof. Let us denote by $\hat{\nabla} f(\boldsymbol{x}; (\boldsymbol{u}_i)_{i=1}^q)$ the gradient estimate from (3.1) along the i.i.d. sampled directions $(\boldsymbol{u}_i)_{i=1}^q$ (we simplify it into $\hat{\nabla} f(\boldsymbol{x}; \boldsymbol{u})$ if there is only one direction $\boldsymbol{u}$). We can first see that, since the random directions $\boldsymbol{u}_i$ are independent identically distributed (i.i.d.) we have:

$$\mathbb{E}\hat{\nabla} f(\boldsymbol{x}; (\boldsymbol{u}_i)_{i=1}^q) = \mathbb{E}\frac{1}{q}\sum_{i=1}^q \hat{\nabla} f(\boldsymbol{x}; \boldsymbol{u}_i) = \frac{1}{q}\sum_{i=1}^q \mathbb{E}\hat{\nabla} f(\boldsymbol{x}; \boldsymbol{u}_1) = \mathbb{E}\hat{\nabla} f(\boldsymbol{x}; \boldsymbol{u}_1)$$

This proves A.3.3 (a). Let us now turn to A.3.3 (b). We have:

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\nabla}_F f(\boldsymbol{x};(\boldsymbol{u}_i)_{i=1}^q)\right\|^2\right] &= \mathbb{E}\left\|\frac{1}{q}\sum_{i=1}^q \hat{\nabla}_F f(\boldsymbol{x};\boldsymbol{u}_i)\right\|^2 \\
&= \frac{1}{q^2}\mathbb{E}\left(\sum_{i=1}^q \hat{\nabla}_F f(\boldsymbol{x};\boldsymbol{u}_i)\right)^\top\left(\sum_{i=1}^q \hat{\nabla}_F f(\boldsymbol{x};\boldsymbol{u}_i)\right) \\
&= \frac{1}{q^2}\sum_{i=1}^q\sum_{j=1}^q \mathbb{E}\left[\hat{\nabla}_F f(\boldsymbol{x};\boldsymbol{u}_i)^\top\hat{\nabla}_F f(\boldsymbol{x};\boldsymbol{u}_j)\right] \\
&\overset{(a)}{=} \frac{1}{q^2}\left[q\mathbb{E}\|\hat{\nabla}_F f(\boldsymbol{x};\boldsymbol{u}_1)\|^2 + \sum_{i=1}^q\sum_{j=1(j\neq i)}^q (\mathbb{E}\hat{\nabla}_F f(\boldsymbol{x};\boldsymbol{u}_i))^\top(\mathbb{E}\hat{\nabla}_F f(\boldsymbol{x};\boldsymbol{u}_j))\right] \\
&= \frac{1}{q^2}\left[q\mathbb{E}||\hat{\nabla}_F f(\boldsymbol{x};\boldsymbol{u}_1)\|^2 + q(q-1)\|\mathbb{E}\hat{\nabla}_F f(\boldsymbol{x};\boldsymbol{u}_1)||^2\right] \\
&\overset{(b)}{\leq} \frac{1}{q^2}\left[q\left[\varepsilon_F\|\nabla_F f(\boldsymbol{x})\|^2 + \varepsilon_{F^c}\|\nabla_{F^c} f(\boldsymbol{x})\|^2 + \varepsilon_{abs}\mu^2\right] + q\left(q-1\right)\left\|\mathbb{E}\hat{\nabla}_F f(\boldsymbol{x};\boldsymbol{u}_1)\right\|^2\right]
\end{aligned} \tag{A.18}$$

Where (a) comes from the fact that the random directions are i.i.d. and (b) comes from assumptions (i) and (ii) of the current Lemma (Lemma A.3.3). Assumption (ii) also allows to bound the last term above in the following way:

$$\begin{aligned}
\|\mathbb{E}\hat{\nabla}_F f(\boldsymbol{x};\boldsymbol{u}_1)\|^2 &\leq 2\|\nabla_F f(\boldsymbol{x};\boldsymbol{u}_1) - \mathbb{E}\hat{\nabla}_F f(\boldsymbol{x};\boldsymbol{u}_1)\|^2 + 2\|\nabla_F f(\boldsymbol{x};\boldsymbol{u}_1)\|^2 \\
&\leq 2\varepsilon_\mu\mu^2 + 2\|\nabla_F f(\boldsymbol{x};\boldsymbol{u}_1)\|^2
\end{aligned} \tag{A.19}$$

Plugging (A.19) into (A.18), we obtain:

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\nabla}_F f(\boldsymbol{x})\right\|^2\right] &\leq \frac{1}{q}\left[\varepsilon_F + 2(q-1)\right]\|\nabla_F f(\boldsymbol{x})\|^2 + \frac{\varepsilon_{F^c}}{q}\|\nabla_{F^c} f(\boldsymbol{x})\|^2 \\
&\quad + \frac{1}{q}\left[\varepsilon_{abs}\mu^2 + 2\left(q-1\right)\varepsilon_\mu\mu^2\right] \\
&\leq \left(\frac{\varepsilon_F}{q} + 2\right)\|\nabla_F f(\boldsymbol{x})\|^2 + \frac{\varepsilon_{F^c}}{q}\|\nabla_{F^c} f(\boldsymbol{x})\|^2 + \left(\frac{\varepsilon_{abs}}{q} + 2\varepsilon_\mu\right)\mu^2
\end{aligned}$$

□

A.3.3 Proof of Proposition 1

Proof. Proposition 1 (a) and (b) follow by plugging the values of ε_F, ε_{F^c}, ε_{abs} and ε_μ from Lemma A.3.1 and Lemma A.3.2 into Lemma A.3.3. Proposition (c) follows from the inequality $\|\boldsymbol{a}+\boldsymbol{b}\|^2 \leq 2\|\boldsymbol{a}\|^2 + 2\|\boldsymbol{b}\|^2$, for $\boldsymbol{a}$ and $\boldsymbol{b}$ in $\mathbb{R}^p$ with $p \in \mathbb{N}^*$. □

A.4 Proofs of section 3.1.2

A.4.1 Proof of Theorem 1

Proof. We will combine the proof from [96] and [66], using ideas of the proof of Theorem 8 from Nesterov to deal with zeroth order gradient approximations, and ideas from the proof of [96] (Theorem 2 and 5, Lemma 19), to deal with the hard thresholding operation in the convergence rate. Let us call η an arbitrary learning rate, that will be fixed later in the proof. Let us call F the following support $F = F^{(t-1)} \cup F^{(t)} \cup \text{supp}(\boldsymbol{x}^*)$, with $F^{(t)} = \text{supp}(\boldsymbol{x}^t)$. We have, for a given random direction $\boldsymbol{u}$ and function noise $\boldsymbol{\xi}$, at a given timestep t of SZOHT:

$$\begin{aligned}\|\boldsymbol{x}^t - \boldsymbol{x}^* - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) + \eta\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 = \|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 - 2\eta\langle \boldsymbol{x}^t - \boldsymbol{x}^*, \hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\rangle \\ + \eta^2\|\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2\end{aligned}$$

Taking the expectation with respect to $\boldsymbol{\xi}$ and to the possible random directions $\boldsymbol{u}_1, ..., \boldsymbol{u}_q$ (that we denote with a simple $\boldsymbol{u}$, abusing notations) at step t, we get:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\boldsymbol{x}^t - \boldsymbol{x}^* - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) + \eta\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&= \|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 - 2\eta\langle \boldsymbol{x}^t - \boldsymbol{x}^*, \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}[\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)]\rangle + \eta^2\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&= \|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 - 2\eta\langle \boldsymbol{x}^t - \boldsymbol{x}^*, \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}[\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)]\rangle \\
&\quad - 2\eta\langle \boldsymbol{x}^t - \boldsymbol{x}^*, \mathbb{E}_{\boldsymbol{\xi}}[\mathbb{E}_{\boldsymbol{u}}\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t)]\rangle + \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\eta^2\|\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&= \|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 - 2\eta\langle \boldsymbol{x}^t - \boldsymbol{x}^*, \nabla_F f(\boldsymbol{x}^t) - \nabla_F f(\boldsymbol{x}^*)\rangle \\
&\quad - 2\eta\langle \sqrt{\eta}L_{s'}\left(\boldsymbol{x}^t - \boldsymbol{x}^*\right), \frac{1}{\sqrt{\eta}L_{s'}}(\mathbb{E}_{\boldsymbol{\xi}}\mathbb{E}_{\boldsymbol{u}}[\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t))]\rangle \\
&\quad + \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\eta^2\|\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&\overset{(a)}{\leq} \|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 - 2\eta\langle \boldsymbol{x}^t - \boldsymbol{x}^*, \nabla_F f(\boldsymbol{x}^t) - \nabla_F f(\boldsymbol{x}^*)\rangle + \eta^2 L_{s'}^2\|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 \\
&\quad + \frac{1}{L_{s'}^2}\mathbb{E}_{\boldsymbol{\xi}}\|\mathbb{E}_{\boldsymbol{u}}\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t))\|^2 + \eta^2\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \qquad \text{(A.20)}
\end{aligned}$$

Where (a) follows from the inequality $2\langle \boldsymbol{u}, \boldsymbol{v}\rangle \leq \|\boldsymbol{u}\|^2 + \|\boldsymbol{v}\|^2$ for any $(\boldsymbol{u}, \boldsymbol{v}) \in (\mathbb{R}^d)^2$. From Proposition 1 (b), since almost each $f_{\boldsymbol{\xi}}$ is $(L_{s'}, s')$-RSS (hence also $(L_{s'}, s_2)$-RSS), we know that for the ε_F, ε_{F^c} and ε_{abs} defined in Proposition 1 (b), we have for almost all $\boldsymbol{\xi}$: $\mathbb{E}_{\boldsymbol{u}}\|\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t)\|^2 \leq \varepsilon_F\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t)\|^2 + \varepsilon_{F^c}\|\nabla_{F^c} f_{\boldsymbol{\xi}}(\boldsymbol{x}^t)\|^2 + \varepsilon_{\text{abs}}\mu^2$. This allows to develop the last term of (A.20) into the following:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 &\leq 2\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t)\|^2 + 2\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&\leq 2\varepsilon_F\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t)\|^2 + 2\varepsilon_{F^c}\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_{F^c} f_{\boldsymbol{\xi}}(\boldsymbol{x}^t)\|^2 \\
&\quad + 2\varepsilon_{\text{abs}}\mu^2 + 2\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&\leq 2\varepsilon_F\left[2\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 + 2\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2\right] \\
&\quad + 2\varepsilon_{F^c}\left[2\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_{F^c} f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_{F^c} f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 + 2\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_{F^c} f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2\right] \\
&\quad + 2\varepsilon_{\text{abs}}\mu^2 + 2\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2
\end{aligned}$$

Just like the proof in [96], we will express our result in terms of the infinity norm of $\nabla f(\boldsymbol{x}^*)$. For that, we will plug above the two following inequalites: Same as their proof of Lemma 19, we have $\|\nabla_F f(\boldsymbol{x}^*)\| \leq \|\nabla_s f(\boldsymbol{x}^*)\|$ (that is because we will have equality if the sets in the definition of F, namely $F^{(t-1)}$, $F^{(t)}$ and $\text{supp}(\boldsymbol{x}^*)$, are disjoints (because their cardinality is respectively k, k and k^*), but they may intersect). And we also have $\|\nabla_s f(\boldsymbol{x}^*)\|_2^2 \leq s\|\nabla f(\boldsymbol{x}^*)\|_\infty^2$ (by definition of the ℓ_2 norm and of the ℓ_∞ norm). Similarly, we also have: $\|\nabla_{F^c} f(\boldsymbol{x}^*)\|_2^2 \leq (d-k)\|\nabla f(\boldsymbol{x}^*)\|_\infty^2$, since $|F^c| \leq d-k$.

Therefore, we obtain:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&\leq 4\varepsilon_F \mathbb{E}_{\boldsymbol{\xi}}\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 + 4\varepsilon_{F^c}\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_{F^c} f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&\quad + ((4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k))\mathbb{E}_{\boldsymbol{\xi}}\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|_\infty^2 + 2\varepsilon_{\text{abs}}\mu^2 \\
&\overset{(a)}{\leq} 4\varepsilon_F \mathbb{E}_{\boldsymbol{\xi}}\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 + ((4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k))\mathbb{E}_{\boldsymbol{\xi}}\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|_\infty^2 + 2\varepsilon_{\text{abs}}\mu^2
\end{aligned}$$

Where (a) follows by observing in Proposition 1 (b) that $\varepsilon_{F^c} \leq \varepsilon_F$, and using the definition of the Euclidean norm. Let us plug the above into (A.20), and use the fact that, from Proposition 1 (a), since each $f_{\boldsymbol{\xi}}$ is $(L_{s'}, s' := \max(s_2, s))$-RSS, it is also $(L_{s'}, s_2)$-RSS, so for the ε_μ from Proposition 1 (a), we have, for almost any given $\boldsymbol{\xi}$: $\|\mathbb{E}_{\boldsymbol{u}}\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t))\|^2 \leq \varepsilon_\mu\mu^2$, and let us also use the fact that since each $f_{\boldsymbol{\xi}}$ is $(L_{s'}, \max(s_2, s))$-RSS , it is also $(L_{s'}, |F|)$-RSS (since $|F| \leq s$) which gives that for almost any $\boldsymbol{\xi}$: $f_{\boldsymbol{\xi}}$: $\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \leq L_{s'}^2\|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2$, to finally obtain:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\boldsymbol{x}^t - \boldsymbol{x}^* - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) + \eta\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&\quad \leq (1 + \eta^2 L_{s'}^2 + 4\varepsilon_F\eta^2 L_{s'}^2)\|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 - 2\eta\langle \boldsymbol{x}^t - \boldsymbol{x}^*, \mathbb{E}_{\boldsymbol{\xi}}[\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)]\rangle \\
&\qquad + \frac{\varepsilon_\mu}{L_{s'}^2}\mu + 2\eta^2\varepsilon_{\text{abs}}\mu^2 + \eta^2((4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k))\mathbb{E}_{\boldsymbol{\xi}}\|\nabla f(\boldsymbol{x}^*)\|_\infty^2 \\
&\quad = (1 + \eta^2 L_{s'}^2 + 4\varepsilon_F\eta^2 L_{s'}^2)\|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 - 2\eta\langle \boldsymbol{x}^t - \boldsymbol{x}^*, \nabla f(\boldsymbol{x}^t) - \nabla f(\boldsymbol{x}^*)\rangle \\
&\qquad + \frac{\varepsilon_\mu}{L_{s'}^2}\mu + 2\eta^2\varepsilon_{\text{abs}}\mu^2 + \eta^2((4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k))\mathbb{E}_{\boldsymbol{\xi}}\|\nabla f(\boldsymbol{x}^*)\|_\infty^2
\end{aligned}$$

Since f is (ν_s, s)-RSC, it is also $(\nu_s, |F|)$-RSC, since $|F| \leq 2k + k^* \leq s$, therefore, we have: $\langle \boldsymbol{x}^t - \boldsymbol{x}^*, \nabla f(\boldsymbol{x}^t) - \nabla f(\boldsymbol{x}^*)\rangle \geq \nu_s\|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2$ (this can be proven by adding together the definition of (ν_s, s)-RSC written respectively at $\boldsymbol{x} = \boldsymbol{x}^t$,$\boldsymbol{y} = \boldsymbol{x}^*$, and at $\boldsymbol{x} = \boldsymbol{x}^*$,$\boldsymbol{y} = \boldsymbol{x}^t$). Plugging this into the above:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\boldsymbol{x}^t - \boldsymbol{x}^* - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) + \eta\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&\quad \leq \left(1 - 2\eta\nu_s + (4\varepsilon_F + 1)\, L_{s'}^2\eta^2\right)\|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 \\
&\qquad + \frac{\varepsilon_\mu}{L_{s'}^2}\mu^2 + 2\eta^2\varepsilon_{\text{abs}}\mu^2 + \eta^2((4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k))\mathbb{E}_{\boldsymbol{\xi}}\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|_\infty^2
\end{aligned}$$

The value of η that minimizes the left term in η is equal to $\frac{\nu_s}{(4\varepsilon_F+1)L_{s'}^2}$ (because the optimum of the quadratic function $ax^2 + bx + c$ is attained in $-\frac{b}{2a}$ and its value is $-\frac{b^2}{4a} + c$). Let us

choose it, that is, we fix $\eta = \frac{\nu_s}{(4\varepsilon_F+1)L_{s'}^2}$. Let us now define the following ρ:

$$\rho^2 = 1 - \frac{4\nu_s^2}{4(4\varepsilon_F+1)L_{s'}^2} = 1 - \frac{\nu_s^2}{(4\varepsilon_F+1)L_{s'}^2}$$

We therefore have:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\boldsymbol{x}^t - \boldsymbol{x}^* - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) + \eta\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&\quad \leq \rho^2\|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 + \frac{\varepsilon_\mu}{L_{s'}^2}\mu^2 + 2\eta^2\varepsilon_{\text{abs}}\mu^2 + \eta^2((4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k))\mathbb{E}_{\boldsymbol{\xi}}\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|_\infty^2
\end{aligned}$$

We can now use the fact that for all $(a,b) \in (\mathbb{R}_+)^2 : \sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, as well as Jensen's inequality, to obtain:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\boldsymbol{x}^t - \boldsymbol{x}^* - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) + \eta\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\| \\
&\quad \leq \rho\|\boldsymbol{x}^t - \boldsymbol{x}^*\| + \frac{\sqrt{\varepsilon_\mu}}{L_{s'}}\mu^2 + \eta\sqrt{2\varepsilon_{\text{abs}}\mu^2} + \eta\sqrt{((4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k)))\mathbb{E}_{\boldsymbol{\xi}}\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|_\infty^2}
\end{aligned}$$

We can now formulate a first decrease-rate type of result, before the hard thresholding operation, as follows, using for η the value previously defined, and with:

$$\boldsymbol{y}^t := \boldsymbol{x}^t - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}\left(\boldsymbol{x}^t\right) \tag{A.21}$$

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\boldsymbol{y}^t - \boldsymbol{x}^*\| &= \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\left\|\boldsymbol{x}^t - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}\left(\boldsymbol{x}^t\right) - \boldsymbol{x}^*\right\| \\
&\leq \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\left\|\boldsymbol{x}^t - \boldsymbol{x}^* - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}\left(\boldsymbol{x}^t\right) + \eta\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\right\| + \eta\mathbb{E}_{\boldsymbol{\xi}}\left\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\right\| \\
&= \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\left\|\boldsymbol{x}^t - \boldsymbol{x}^* - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}\left(\boldsymbol{x}^t\right) + \eta\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\right\| + \eta\mathbb{E}_{\boldsymbol{\xi}}\sqrt{\left\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\right\|^2} \\
&\leq \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\left\|\boldsymbol{x}^t - \boldsymbol{x}^* - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}\left(\boldsymbol{x}^t\right) + \eta\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\right\| + \eta\sqrt{\mathbb{E}_{\boldsymbol{\xi}}\left\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\right\|^2} \\
&\leq \rho\left\|\boldsymbol{x}^t - \boldsymbol{x}^*\right\| + \eta(\sqrt{((4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k))\mathbb{E}_{\boldsymbol{\xi}}\left\|\nabla f(\boldsymbol{x}^*)\right\|_\infty^2} \\
&\quad + \sqrt{s}\sqrt{\mathbb{E}_{\boldsymbol{\xi}}\left\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\right\|_\infty^2}) + \frac{\sqrt{\varepsilon_\mu}}{L_{s'}}\mu^2 + \eta\sqrt{2\varepsilon_{\text{abs}}\mu^2} \\
&= \rho\left\|\boldsymbol{x}^t - \boldsymbol{x}^*\right\| + \eta(\sqrt{(4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k)} + \sqrt{s})\sqrt{\mathbb{E}_{\boldsymbol{\xi}}\left\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\right\|_\infty^2} \\
&\quad + \frac{\sqrt{\varepsilon_\mu}}{L_{s'}}\mu + \eta\sqrt{2\varepsilon_{\text{abs}}\mu^2} \\
&\overset{(a)}{\leq} \rho\left\|\boldsymbol{x}^t - \boldsymbol{x}^*\right\| + \eta(\sqrt{(4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k)} + \sqrt{s})\sigma \\
&\quad + \frac{\sqrt{\varepsilon_\mu}}{L_{s'}}\mu + \eta\sqrt{2\varepsilon_{\text{abs}}\mu^2} \\
&\leq \rho\left\|\boldsymbol{x}^t - \boldsymbol{x}^*\right\| + \eta(\sqrt{(4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k)} + \sqrt{s})\sigma \\
&\quad + \frac{\sqrt{\varepsilon_\mu}}{L_{s'}}\mu + \eta\sqrt{2\varepsilon_{\text{abs}}\mu^2}
\end{aligned} \tag{A.22}$$

Where (a) follows from the σ-FGN assumption. We now consider $\boldsymbol{x}^{t+1}$, that is, the best-k-sparse approximation of $\boldsymbol{z}^t := \boldsymbol{x}^t - \eta\hat{\nabla} f_{\boldsymbol{\xi}}(\boldsymbol{x}^t)$ from the hard thresholding operation in SZOHT. We can notice that $\boldsymbol{x}^t_F = \boldsymbol{x}^t$ (because $\text{supp}(\boldsymbol{x}^t) = F^{(t)} \subset F$), which gives $\boldsymbol{y}^t = \boldsymbol{z}^t_F$. Since $F^{(t+1)} \subset F$, the coordinates of the top k magnitude components of $\boldsymbol{z}^t$ are in F, so they are also those of the top k magnitude components of $\boldsymbol{z}^t_F = \boldsymbol{y}^t$. Therefore, $\boldsymbol{x}^{t+1}$ is also the best k-sparse approximation of $\boldsymbol{y}^t$. Therefore, using Corollary A.2.1, we obtain:

$$\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\| \leq \gamma \|\boldsymbol{y}^t - \boldsymbol{x}^*\|$$

with:

$$\gamma := \sqrt{1 + \left(k^*/k + \sqrt{(4 + k^*/k)\, k^*/k}\right)/2} \tag{A.23}$$

Where $k^* = \|\boldsymbol{x}^*\|_0$. Plugging this into (A.22) gives:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\| \leq \gamma\rho \left\|\boldsymbol{x}^t - \boldsymbol{x}^*\right\| + \gamma\eta(\sqrt{(4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k)}) + \sqrt{s})\sigma \\ + \gamma\frac{\sqrt{\varepsilon_\mu}}{L_{s'}}\mu + \eta\sqrt{2\varepsilon_{\text{abs}}}\mu\end{aligned}$$

This will allow us to obtain the following final result:

$$\begin{aligned}\mathbb{E}\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\| \leq \gamma\rho \left\|\boldsymbol{x}^t - \boldsymbol{x}^*\right\| + \gamma \underbrace{\eta\left(\sqrt{(4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k)} + \sqrt{s}\right)}_{:=a}\sigma \\ + \gamma \underbrace{\left(\frac{\sqrt{\varepsilon_\mu}}{L_{s'}} + \eta\sqrt{2\varepsilon_{\text{abs}}}\right)}_{:=b}\mu\end{aligned} \tag{A.24}$$

with $\eta = \frac{\nu_s}{(4\varepsilon_F+1)L^2_{s'}}$ and $\rho^2 = 1 - \frac{2\nu_s^2}{(4\varepsilon_F+1)L^2_{s'}}$. We need to have $\rho\gamma < 1$ in order to have a contraction at each step. Let us suppose that $k \geq \rho^2 k^*/(1-\rho^2)^2$: we will show that this value for k allows to verify that condition on $\rho\gamma$. That implies $\frac{k^*}{k} \leq \frac{(1-\rho^2)^2}{\rho^2}$. We then have, from the definition of γ in (A.23):

$$\begin{aligned}\gamma^2 &\leq 1 + \left(\frac{(1-\rho^2)^2}{\rho^2} + \sqrt{\left(4 + \frac{(1-\rho^2)^2}{\rho^2}\right)\frac{(1-\rho^2)^2}{\rho^2}}\right)\frac{1}{2} \\ &= 1 + \left(\frac{(1-\rho^2)^2}{\rho^2} + \sqrt{\left(\frac{4\rho^2 + 1 + \rho^4 - 2\rho^2}{\rho^2}\right)\frac{(1-\rho^2)^2}{\rho^2}}\right)\frac{1}{2} \\ &= 1 + \left(\frac{(1-\rho^2)^2}{\rho^2} + \sqrt{\frac{(1+\rho^2)^2(1-\rho^2)^2}{\rho^4}}\right)\frac{1}{2} \\ &= 1 + \left(\frac{(1-\rho^2)^2}{\rho^2} + \frac{(1+\rho^2)(1-\rho^2)}{\rho^2}\right)\frac{1}{2} = 1 + \left(\frac{(1-\rho^2)(1-\rho^2+1+\rho^2)}{\rho^2}\right)\frac{1}{2} \\ &= 1 + \frac{(1-\rho^2)}{\rho^2} = \frac{1}{\rho^2}\end{aligned} \tag{A.25}$$

Therefore, we indeed have $\rho\gamma \leq 1$ when choosing $k \geq \rho^2 k^*/(1-\rho^2)^2$.

Unrolling inequality (A.24) through time, we then have, at iteration $t+1$, and denoting by $\boldsymbol{\xi}^{t+1}$ the noise drawn at time step $t+1$ and $\boldsymbol{u}^{t+1}$ the random directions $\boldsymbol{u}_1, ..., \boldsymbol{u}_q$ chosen at time step $t+1$, from the law of total expectations:

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\| &= \mathbb{E}_{\boldsymbol{\xi}^t, \boldsymbol{u}^t, .., \boldsymbol{\xi}^1, \boldsymbol{u}^1} \mathbb{E}_{\boldsymbol{\xi}^{t+1}, \boldsymbol{u}^{t+1} | \boldsymbol{\xi}^t, \boldsymbol{u}^t, .., \boldsymbol{\xi}^1, \boldsymbol{u}^1} \|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\| \\
&\leq \mathbb{E}_{\boldsymbol{\xi}^t, \boldsymbol{u}^t, .., \boldsymbol{\xi}^1, \boldsymbol{u}^1} [\gamma\rho \|\boldsymbol{x}^t - \boldsymbol{x}^*\| + \gamma a\sigma + \gamma b\mu] \\
&= \gamma\rho \mathbb{E}_{\boldsymbol{\xi}^t, \boldsymbol{u}^t, .., \boldsymbol{\xi}^1, \boldsymbol{u}^1} [\|\boldsymbol{x}^t - \boldsymbol{x}^*\|] + \gamma a\sigma + \gamma b\mu \\
&\leq (\gamma\rho)^2 \mathbb{E}_{\boldsymbol{\xi}^{t-1}, \boldsymbol{u}^{t-1}, .., \boldsymbol{\xi}^1, \boldsymbol{u}^1} [\|\boldsymbol{x}^{t-1} - \boldsymbol{x}^*\|] + (\gamma\rho)^2 a\sigma \\
&\quad + \gamma a\sigma + (\gamma\rho)^2 b\mu + \gamma b\mu \\
&\leq (\gamma\rho)^{t+1} \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\| + \left(\sum_{i=0}^{t} (\gamma\rho)^i \right) \gamma a\sigma + \left(\sum_{i=0}^{t} (\gamma\rho)^i \right) \gamma b\mu \\
&= (\gamma\rho)^{t+1} \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\| + \frac{1-(\gamma\rho)^t}{1-\gamma\rho} \gamma a\sigma + \frac{1-(\gamma\rho)^t}{1-\gamma\rho} \gamma b\mu \\
&\leq (\gamma\rho)^{t+1} \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\| + \frac{1}{1-\gamma\rho} \gamma a\sigma + \frac{1}{1-\gamma\rho} \gamma b\mu
\end{aligned}$$

Where the last inequality follows from the fact that $\rho\gamma < 1$. □

A.4.2 Proof of Lemma 1

Below, we prove that there is a minimal value for q:

Indeed, we need the ZOHT to converge so we need $\rho\gamma < 1$. From the expressions of ρ and γ We have $\rho = \rho(q,k)$, and $\gamma = \gamma(k)$. We recall those expressions below:

$\gamma = \sqrt{1 + \left(k^*/k + \sqrt{(4 + k^*/k)\, k^*/k}\right)/2}$

$\rho^2 = 1 - \frac{\nu_s^2}{(4\varepsilon_F+1)L_{s'}^2} = 1 - \frac{1}{(4\varepsilon_F+1)\kappa^2}$ with $\kappa = \frac{L_{s'}}{\nu_s}$.

with: $\varepsilon_F = \frac{2d}{q(s_2+2)} \left(\frac{(s-1)(s_2-1)}{d-1} + 3 \right) + 2$, with $s = 2k + k^*$ (we consider the smallest s possible from Theorem 1)
So therefore:

$$\begin{aligned}
\rho^2 &= 1 - \frac{1}{\left[\frac{8d}{q(s_2+2)} \left(\frac{(s-1)(s_2-1)}{d-1} + 3 \right) + 9 \right] \kappa^2} \\
&= 1 - \frac{1}{\left[\frac{8d}{q(s_2+2)} \left(\frac{(2k+k^*-1)(s_2-1)}{d-1} + 3 \right) + 9 \right] \kappa^2}
\end{aligned}$$

Let us define $a := \frac{16d\kappa^2(s_2-1)}{q(s_2+2)(d-1)}$ and $b := \kappa^2 \left[\frac{8d}{q(s_2+2)} \left[\frac{(s_2-1)(k^*-1)}{d-1} + 3 \right] + 9 \right]$

We then have:

$$\rho^2 = 1 - \frac{1}{ak+b}$$

To ensure convergence, we need to have $\rho\gamma < 1$, therefore the first condition that we need to verify is $k \geq \rho^2 k^*/(1-\rho^2)^2$.

Which means we need:

$$k \geq \frac{\left(1 - \frac{1}{ak+b}\right) k^*}{\left(\frac{1}{ak+b}\right)^2} \tag{A.26}$$

$$k \geq \left[(ak+b)^2 - (ak+b)\right] k^* \tag{A.27}$$

$$k \geq k^* \left[a^2k^2 + 2abk + b^2 - ak - b\right] \tag{A.28}$$

$$0 \geq k^* a^2 k^2 + \left(2ab - \frac{1}{k^*} - a\right) k^* k^2 + \left(b^2 - b\right) k^* \tag{A.29}$$

If we want that there exist a k such that this is true, we need (since $k^* \geq 0$):

$$\Delta \geq 0$$

with:

$$\begin{aligned}
\Delta &:= k^{*2}(2ab - \frac{1}{k^*} - a)^2 - 4k^{*^2}a^2\left(b^2 - b\right) \\
&= k^{*^2}\left(4a^2b^2 + \left(\frac{1}{k^*} + a\right)^2 - 4ab\left(\frac{1}{k^*} + a\right)\right) - 4k^{*2}a^2\left(b^2 - b\right) \\
&= k^{*2}\left[4a^2b^2 + \frac{1}{k^{*^2}} + a^2 + \frac{2a}{k^*} - \frac{4ab}{k^*} - 4a^2b - 4a^2b^2 + 4a^2b\right] \\
&= 1 + a^2k^{*2} + 2ak^* - 4abk^*
\end{aligned}$$

$$\Delta \geq 0 \Rightarrow 1 + a^2k^{*2} + 2ak^* \geq 4abk^* \tag{A.30}$$

Let us express a and b in terms of q, as:

$$a = \frac{A}{q} \quad \text{with} \quad A = \frac{16d\kappa^2\left(s_2 - 1\right)}{\left(s_2 + 2\right)\left(d - 1\right)} \tag{A.31}$$

$$b = \frac{B}{q} + C \quad \text{with} \quad B = \kappa^2\left[\frac{8d}{(s_2+2)}\left(\frac{(s_2-1)\,(k^*-1)}{d-1} + 3\right)\right] \tag{A.32}$$

$$\text{and with } C = 9\kappa^2 \tag{A.33}$$

So plugging in (A.30), what we need is:

$$\begin{aligned}
1 + \frac{A^2}{q^2}k^{*^2} + 2\frac{A}{q}k^* &\geq 4\frac{A}{q}\left(\frac{B}{q} + C\right)k^* \\
q^2 + A^2k^{*2} + 2Ak^*q &\geq 4ABk^* + 4CAqk^* \\
q^2 + q\left(2Ak^* - 4CAk^*\right) + A^2k^{*2} - 4ABk^* &\geq 0
\end{aligned}$$

To ensure that, we need to compute Δ', defined as:

$$\begin{aligned}\Delta' &:= (2Ak^* - 4CAk^*)^2 - 4\left(A^2k^{*2} - 4ABk^*\right) \\ &= 4A^2k^{*2} + 16C^2A^2k^{*2} - 16CA^2k^{*2} - 4A^2k^{*2} + 16ABk^* \\ &= 16CA^2k^{*2}(C-1) + 16ABk^* = 16Ak^*\left[k^*C(C-1)A + B\right]\end{aligned}$$

We now have:

$$C = 9\kappa^2 \Rightarrow C \geq 1 \Rightarrow \Delta' \geq 0$$

Therefore, there is a minimal value for q, and it is:

$$q \geq q_{\min}$$

With:

$$q_{\min} = \frac{-\left(2Ak^* - 4CAk^*\right) + \sqrt{16CA^2k^{*2}(C-1) + 16ABk^*}}{2} \tag{A.34}$$

$$= \frac{2Ak^*\left(2C-1\right) + \sqrt{16A^2k^{*2}\left[C(C-1) + \frac{B}{Ak^*}\right]}}{2} \tag{A.35}$$

where we assume that $s_2 > 1$ (so that $A > 0$), and with, as we recall: $A = \frac{16d\kappa^2(s_2-1)}{(s_2+2)(d-1)}$ and $B = \frac{8\kappa^2 d}{s_2+2}(\frac{(s_2-1)(k^*-1)}{d-1} + 3)$
So, if $s_2 > 1$: $\frac{B}{Ak^*} = \frac{1}{2} - \frac{1}{2k^*} + \frac{3}{2}\frac{d-1}{s_2-1}$
Therefore: $q_{\min} = Ak^*\left[2C - 1 + 2\sqrt{C(C-1) + \frac{1}{2} - \frac{1}{2k^*} + \frac{3}{2}\frac{d-1}{s_2-1}}\right]$
with $C = 9\kappa^2$, which reads: $q_{\min} = \frac{16d(s_2-1)k^*\kappa^2}{(s_2+2)(d-1)}\left[18\kappa - 1 + 2\sqrt{9\kappa(9\kappa-1) + \frac{1}{2} - \frac{1}{2k^*} + \frac{3}{2}\frac{d-1}{s_2-1}}\right]$
Now, if $s_2 = 1$, we have $A = 0$, so therefore, from (A.34), $q_{\min} = 0$, so there is no condition necessary condition on q so that there exist k such that: $k \geq \rho^2 k^*/(1-\rho^2)^2$. We may therefore think that it is possible to take $q = 1$ in that case. However, there is another condition on k that should also be enforced, which is that $k \leq d$ (as explained in the next Remark in the main manuscript). And in that $s_2 = 1$ case, if we take $q = 1$, we have $a = 0$, and $b = \kappa^2[8d + 9]$ (from (A.31) and (A.32)). Therefore, enforcing the condition $k \geq [(ak+b)^2 - (ak+b)] = b(b-1)$ will necessarily violate the condition $k \leq d$. Therefore, there is also a minimal value for q in this case, though it is for a different reason than the conflict between ρ and γ.

A.4.3 Proof of Corollary 1

Proof. We first restrict the result of Theorem 1 to a particular q. By inspection of Proposition 1 (b), we choose q such that the part of ε_F that depends on q becomes 1: we believe this will allow to better understand the dependence between variables in our convergence rate result, although other choices of q are possible. Therefore, we choose:

$$q' := \frac{2d}{s_2+2}\left(\frac{(s-1)(s_2-1)}{d-1} + 3\right) \tag{A.36}$$

so that we obtain: $\varepsilon'_F := 1 + 2 = 3$ (from Proposition 1 (b)), which also implies :

$$\eta' := \frac{\nu_s}{(4\varepsilon'_F + 1)L_{s'}^2} = \frac{\nu_s}{13L_{s'}^2}$$

and:

$$\rho'^2 := 1 - \frac{2\nu_s^2}{(4\varepsilon'_F + 1)L_{s'}^2} = 1 - \frac{2\nu_s^2}{13L_{s'}^2} \tag{A.37}$$

Now, regarding the value of q, we also note that any value of random directions $q'' \geq q'$ can be taken too, since the bound in Proposition 1 (b) would then still be verified for ε'_F (that is, we would still have $\mathbb{E}\|\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x})\|^2 \leq \varepsilon'_F \|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x})\|^2 + \varepsilon'_{F^c} \|\nabla_{F^c} f_{\boldsymbol{\xi}}(\boldsymbol{x})\|^2 + \varepsilon_{abs}\mu^2$) (with ε'_{F^c} the value of ε_{F^c} for $q = q'$).
Therefore, we will choose a value q'' so that our result is simpler. First, notice that $s \leq d \implies 1 - \frac{1}{s} \leq 1 - \frac{1}{d} \implies \frac{s-1}{s} \leq \frac{d-1}{d} \implies \frac{s-1}{d-1} \leq \frac{s}{d}$. Therefore, if we take $q \geq 2s + 6\frac{d}{s_2}$, we will also have $q \geq \frac{2d}{s_2+2}\left(\frac{(s-1)(s_2-1)}{d-1} + 3\right) = q'$.

Let us now impose a lower bound on k that is slightly (twice) bigger than the lower bound from Theorem 1. As will become clear below, this allows us to have a $\rho\gamma$ enough bounded away from 1, which guarantees a reasonable constant in the $\mathcal{O}$ notation for the query complexity (see the end of the proof). Let us therefore take:

$$k \geq 2k^* \frac{\rho^2}{(1-\rho^2)^2} \tag{A.38}$$

and plug the value of ρ above into the expression:

$$k \geq 2k^* \frac{\rho'^2}{(1-\rho'^2)^2} \iff k \geq 2k^* \frac{1 - \frac{2\nu_s^2}{13L_{s'}^2}}{(\frac{2\nu_s^2}{13L_{s'}^2})^2} \iff k \geq 2k^* \left(\left(\frac{13L_{s'}^2}{2\nu_s^2}\right)^2 - \frac{13L_{s'}^2}{2\nu_s^2}\right)$$
$$\iff k \geq 2k^* (\frac{13}{2}\kappa^2)(\frac{13}{2}\kappa^2 - 1)$$

With κ denoting $\frac{L_{s'}}{\nu_s}$. Therefore, if we take:

$$k \geq (86\kappa^4 - 12\kappa^2)k^*$$

we will indeed verify the formula above $k \geq 2k^*(\frac{13}{2}\kappa^2)(\frac{13}{2}\kappa^2 - 1)$.
We now turn to describing the query complexity of the algorithm: To ensure that $(\gamma\rho)^t \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\| \leq \varepsilon$, we need:

$$t \geq \frac{1}{\log\frac{1}{\gamma\rho}} \log(\frac{1}{\varepsilon}) \log(\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|) \tag{A.39}$$

with $\gamma\rho$ belonging to the interval $(0, 1)$. Let us compute more precisely an upper bound to $\rho\gamma$ in this case, to show that it is reasonably enough bounded away from 1: Taking k as

described in (A.38), and plugging that value into the expression of γ from Theorem 1, we obtain:

$$\begin{aligned}\gamma^2 &= 1 + \left(\frac{(1-\rho^2)^2}{2\rho^2} + \sqrt{\left(4 + \frac{(1-\rho^2)^2}{2\rho^2}\right)\frac{(1-\rho^2)^2}{2\rho^2}}\right)/2 \\ &\leq 1 + \frac{1}{\sqrt{2}}\left(\frac{(1-\rho^2)^2}{\rho^2} + \sqrt{\left(4 + \frac{(1-\rho^2)^2}{\rho^2}\right)\frac{(1-\rho^2)^2}{\rho^2}}\right)/2 \\ &\overset{(a)}{=} 1 + \frac{1}{\sqrt{2}}\frac{1-\rho^2}{\rho^2}\end{aligned}$$

Where the simplification in (a) above follows similarly to (A.25). Therefore, in that case, we have:

$$\begin{aligned}\rho^2\gamma^2 &\leq \rho^2 + \frac{1}{\sqrt{2}}(1-\rho^2) = \frac{1}{\sqrt{2}} + \rho^2(1 - \frac{1}{\sqrt{2}}) \\ &= \frac{1}{\sqrt{2}} + (1 - \frac{2}{13\kappa^2})(1 - \frac{1}{\sqrt{2}}) = 1 - \frac{2(1-\frac{1}{\sqrt{2}})}{13\kappa^2} \overset{(a)}{\leq} 1 - \frac{1}{26\kappa^2}\end{aligned}$$

Where (a) follows because $(1 - \frac{1}{\sqrt{2}}) \approx 0.29 \geq 1/4$ Therefore:

$$\frac{1}{(\rho\gamma)^2} \geq \frac{1}{1 - \frac{1}{26\kappa^2}} \tag{A.40}$$

Given that $\log(\frac{1}{1-x}) \geq x$ for all $x \in [0, 1)$, we have:

$$\log\left(\frac{1}{(\rho\gamma)^2}\right) \geq \frac{1}{26\kappa^2}$$

Therefore:

$$\frac{1}{\log(\frac{1}{\rho\gamma})} = \frac{2}{\log(\frac{1}{(\rho\gamma)^2})} \leq 52\kappa^2$$

Therefore, plugging this into (A.39), we obtain that with $t \geq 52\kappa^2 \log(\frac{1}{\varepsilon}) \log(\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|) = \mathcal{O}(\kappa^2 \log(\frac{1}{\varepsilon}))$ iterations, we can get $(\gamma\rho)^t\|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \varepsilon$.

To obtain the query complexity (QC), we therefore just need to multiply the number of iterations by the number of queries per iteration $q = 2s + 6\frac{d}{s_2}$: to ensure $(\gamma\rho)^t\|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \varepsilon$, we need to query the zeroth-order oracle at least the following number of times: $(2s + 6\frac{d}{s_2})52\kappa^2 \log(\frac{1}{\varepsilon}) \log(\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|) = \mathcal{O}((k + \frac{d}{s_2})\kappa \log(\frac{1}{\varepsilon}))$, since $s = 2k + k^*$.

A.4.4 Proof of Corollary 2

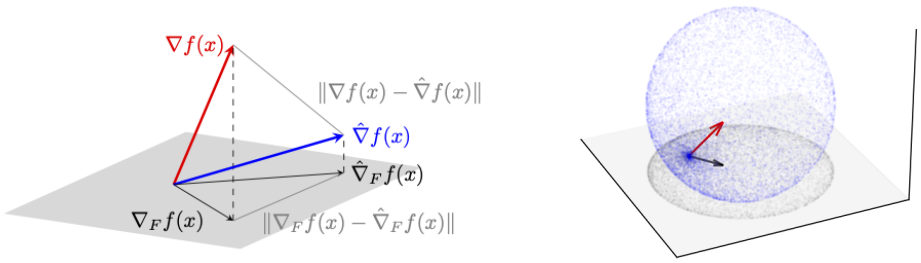
Almost all $f_{\boldsymbol{\xi}}$ are L-smooth, which is equivalent to saying that they are (L, d)-RSS. So we can directly plug $s_2 = d$ in equation (A.36), which gives a necessary value for q of:

$$q = \frac{2d}{d+2}(s+2) \tag{A.41}$$

Since any value of q larger than the one in (A.41) is valid, we choose $q \geq 2(s+2)(\geq \frac{2d}{d+2}(s+2))$ for simplicity. The query complexity is obtained similarly as in the proof of Corollary 1 above, with that new value for q (the number of iterations needed is unchanged from the proof of Corollary 1), only the query complexity q per iteration changes), which means we need to query the zeroth-order oracle the following number of times: $2(s+2)52\kappa^2 \log(\frac{1}{\varepsilon}) \log(\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|) = \mathcal{O}(k\kappa \log(\frac{1}{\varepsilon}))$ □

A.5 Projection of the gradient estimator onto a sparse support

Below we plot the true gradient $\nabla f(x)$ and its estimator $\hat{\nabla} f(x)$ (for $q = 1$), as well as their respective projections $\nabla_F f(x)$ and $\hat{\nabla}_F f(x)$, with $F = \{0, 1\}$ (i.e. F is the hyperplane $z = 0$), for n_{dir} random directions. In Figure A.1(b), due to the large number of random directions, we plot them as points not vectors. For simplicity, the figure is plotted for $\mu \to 0$, and $s_2 = d$. We can see that even though gradient estimates $\hat{\nabla} f(x)$ are poor estimates of $\nabla f(x)$, $\hat{\nabla}_F f(x)$ is a better estimate of $\nabla_F f(x)$.



(a) $n_{\text{dir}} = 1$ (b) $n_{\text{dir}} = 10^6$

Figure A.1: $\nabla f(x)$ and $\hat{\nabla} f(x)$ and their projections $\nabla_F f(x)$ and $\hat{\nabla}_F f(x)$ onto F

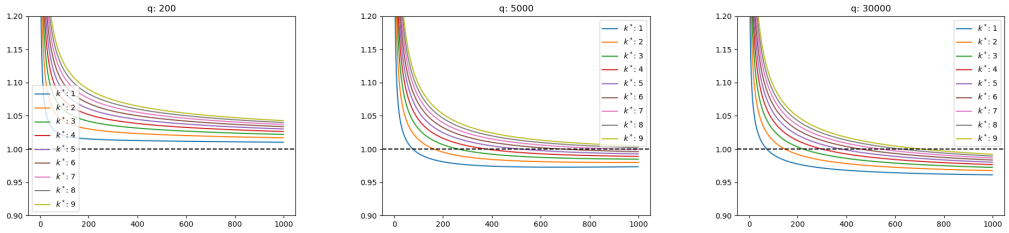
Remark 5. *An interesting fact that can be observed in Figure A.1(b) above is that when $\mu \to 0$ and $s_2 = d$, the ZO gradient estimates belong to a sphere. This comes from the fact that, in that case, the ZO estimate using the random direction $\boldsymbol{u}$ is actually a directional derivative (scaled by d): $\hat{\nabla} f(\boldsymbol{x}) = d\langle \nabla f(\boldsymbol{x}), \boldsymbol{u} \rangle \boldsymbol{u}$, for which we have :*

$$\begin{aligned} \|\hat{\nabla} f(\boldsymbol{x}) - \frac{d}{2}\nabla f(\boldsymbol{x})\|^2 &= d^2(\langle \nabla f(\boldsymbol{x}), \boldsymbol{u} \rangle)^2 \langle \boldsymbol{u}, \boldsymbol{u} \rangle + \frac{d^2}{4}\|\nabla f(\boldsymbol{x})\|^2 \\ &\quad - d^2 \langle \nabla f(\boldsymbol{x}), \boldsymbol{u} \rangle \langle \boldsymbol{u}, \nabla f(\boldsymbol{x}) \rangle \\ &= \frac{d^2}{4}\|\nabla f(\boldsymbol{x})\|^2 \end{aligned}$$

(since $\|\boldsymbol{u}\| = 1$). That is, gradient estimates belong to a sphere of center $\frac{d}{2}\nabla f(\boldsymbol{x})$ and radius $\frac{d}{2}\|\nabla f(\boldsymbol{x})\|$. However, the distribution of $\hat{\nabla} f(\boldsymbol{x})$ is not uniform on that sphere: it is more concentrated around $\mathbf{0}$ as we can observe in Figure A.1(b).

A.6 Value of $\rho\gamma$ depending on q and k^*

In this section, we further illustrate the importance on the value of q as discussed in Lemma 1, by showing in Figure A.2 that if q is too small, then there does not exist any k that verifies the condition $k \geq \frac{k^*\rho^2}{(1-\rho^2)^2}$, no matter how small is k^* (i.e., even if $k^* = 1$). However, if q is large enough, then there exist some k^* such that this condition is true. To generate the curves below, we simply use the formulas for $\gamma = \gamma(k, k^*)$ and $\rho = \rho(s, q)$ with $s = 2k + k^*$ from Theorem 1, and with $d = 30000$ and $s_2 = d$.



(a) $q = 200$ (b) $q = 5000$ (c) $q = 30000$

Figure A.2: $\rho\gamma$ (y axis) as a function of k (x axis) for several values of q and k^*

A.7 Dimension independence/weak-dependence

In this section, we show the dependence of SZOHT on the dimension. To that end, we consider minimizing the following synthetic problem:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{s.t.} \quad \|\boldsymbol{x}\|_0 \leq k$$

with $k = 500$, and f chosen as: $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2$, with $\boldsymbol{y}_i = 0$ if $i < d - k^*$ and $\boldsymbol{y}_i = \frac{1}{(k^*-(d-i))}$ if $i > d - k^*$ with $k^* = 5$. In other words, the k^* last components of $\boldsymbol{y}$ are regularly spaced from $1/k^*$ to 1: in a way, this simulates the recovery of a k^*-sparse vector $\boldsymbol{y}$ by observing only the squared deviation of some queries $\boldsymbol{x}$. In that case, we can easily check that f verifies the following properties:

- f is L-smooth with $L = 1$, as well as $(L_{s'}, s')$-RSS for any s' such that $1 \leq s' \leq d$, with $L_{s'} = 1$, and (ν_s, s)-RSC with $s = 2k + k^*$ and $\nu_s = 1$ (so $\kappa = \frac{L}{\nu_s} = \frac{L_{s'}}{\nu_s} = 1$)
- $\boldsymbol{y} = \boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad s.t. \quad \|\boldsymbol{x}\|_0 \leq k^*$
- $f(\boldsymbol{y}) = f(\boldsymbol{x}^*) = 0$
- $\nabla f(\boldsymbol{y}) = \boldsymbol{0}$ so f is σ-FGN with $\sigma = 0$

We also note that the above setting of k and k^* verifies $k \geq (86\kappa^4 - 12\kappa^2)k^*$ (since $\kappa = 1$). Finally, we initialize $\boldsymbol{x}^0$ such that ${\boldsymbol{x}^0}_i = 1/d$ if $d - k^* \geq i$ and 0 otherwise. We choose this initialization and not $\boldsymbol{x}^0 = \mathbf{0}$, just to ensure that $\nabla f(\boldsymbol{x}^0)_i \neq 0$ for any i: this way the optimization is really done over all d variables, not just the k^* last ones. In addition, this initialization ensures that $\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|$ is constant no matter the d, which makes the convergence curves comparable. We consider several settings of s_2 to showcase the dependence on the dimension below.

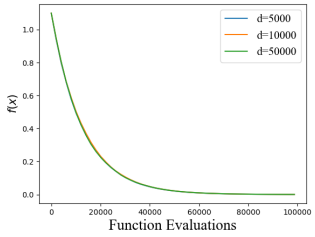
Dimension Independence

- $s_2 = d$: As from Corollary 2, we take $q = 2(s+2)$ with $s = 2k + k^*$ (i.e. $q = 2014$). We choose $\mu = 1e-8$, to have the smallest possible system error due to zeroth-order approximations. As we can see in Figure A.3, all curves are superimposed, which shows that the query complexity is indeed dimension independent, as described by Corollary 2
- $s_2 = \mathcal{O}(\frac{d}{k})$ (We choose $s_2 = \lfloor \frac{d}{k} \rfloor$): As from Corollary 1, we take $q = 2s + 6\frac{d}{s_2}$ with $s = 2k + k^*$. In that case, from Corollary 1, the query complexity will still be $\mathcal{O}(k)$ (i.e. dimension independent), as a sum of two $\mathcal{O}(k)$ terms, although larger than in the case $s_2 = d$ above (since the constant from the $\mathcal{O}$ notation in Corollary 1 will be larger here). We can observe that this is indeed the case in Figure A.4.

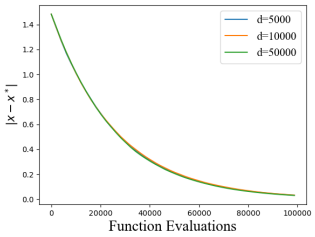
Dimension weak-dependence We now turn to the case where s_2 is fixed. We choose q as in Corollary 1 ($q = 2s + 6\frac{d}{s_2}$ with $s = 2k + k^*$): the query now depends on d in that case, as predicted by Corollary 1, which can indeed be observed in Figure A.5.

A.8 Additional results on adversarial attacks

In this section, we provide additional results for the adversarial attacks problem in 3.1.3, in Figure A.7. The parameters we used for SZOHT to generate that table are the same as in 3.1.3, except for MNIST, for which we choose $k = 20$, $q = 10$, and $s_2 = 10$, and for ImageNet, for which we choose $k = 100000, s_2 = 20000$ and $q = 100$. As we can see, SZOHT allows to obtain sparse attacks, contrary to the other algorithms, and with a smaller ℓ_2 distance and a larger success rate, using less iterations: this shows that SZOHT allows to enforce sparsity, and efficiently exploits that sparsity in order to have a lower query complexity than vanilla sparsity constrained ZO algorithms.

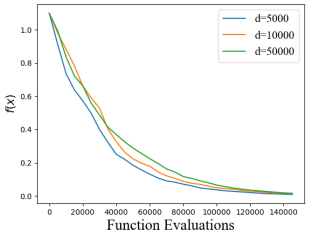


(a) $f(\boldsymbol{x})$

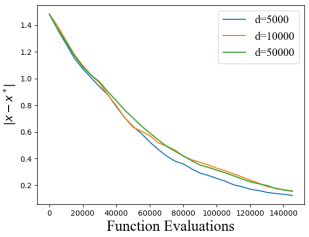


(b) $\|\boldsymbol{x} - \boldsymbol{x}^*\|$

Figure A.3: $s_2 = d$

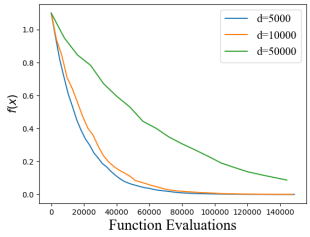


(a) $f(\boldsymbol{x})$

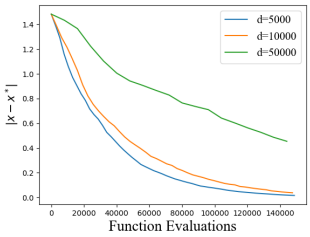


(b) $\|\boldsymbol{x} - \boldsymbol{x}^*\|$

Figure A.4: $s_2 = \lfloor \frac{d}{k} \rfloor$



(a) $f(\boldsymbol{x})$



(b) $\|\boldsymbol{x} - \boldsymbol{x}^*\|$

Figure A.5: $s_2 = 50$

Figure A.6: Dependence on the dimensionality of the query complexity

Method	ASR	ℓ_0 dist.	ℓ_2 dist.	Iter
RSPGF	78%	100%	10.9	67
ZORO	75%	100%	15.1	550
ZSCG	79%	100%	10.3	252
SZOHT	79%	2.5%	8.5	36

(a) MNIST

Method	ASR	ℓ_0 dist.	ℓ_2 dist.	Iter
RSPGF	83%	100%	4.1	326
ZORO	86%	100%	62.9	592
ZSCG	86%	100%	8.4	126
SZOHT	91%	1.9%	2.6	26

(b) CIFAR

Method	ASR	ℓ_0 dist.	ℓ_2 dist.	Iter.
RSPGF	91%	100%	19.9	137
ZORO	90%	100%	111.9	674
ZSCG	76%	100%	111.3	277
SZOHT	95%	37.3%	10.5	61

(c) ImageNet

Figure A.7: Summary of results on adversarial attacks

Appendix B

Appendix: k-support norm

B.1 Notations and definitions

First, we describe some of the notations that will be used in this Appendix. $[\boldsymbol{v}]_S$ denotes the restriction of a vector $\boldsymbol{v}$ to the support S, $[\boldsymbol{v}]_i$ denotes its i-th component, $\boldsymbol{M}^\top$ denotes the transpose of a matrix $\boldsymbol{M}$, and $\boldsymbol{M}^\dagger$ denotes the Moore-Penrose pseudo-inverse of $\boldsymbol{M}$ [37]. $\boldsymbol{I}_{r\times r}$ denotes the identity matrix in $\mathbb{R}^{r,r}$. $[d]$ denotes the set $\{1, ..., d\}$, and $\binom{[d]}{k}$ denotes the set of all the sets of k elements from $\{1, ..., d\}$. $\bar{S}$ denotes the complement in $[d]$ of a support S, that is, all the integers from $[d]$ that are not in S. ∂f denotes the *subgradient* of a function f [76]. $\text{conv}(\mathcal{A})$ denotes the convex hull of a set of vectors $\mathcal{A} \subset \mathbb{R}^d$ (that is, the set of all the convex combinations of elements of $\mathcal{A}$). We then introduce the following definitions:

Definition 4 (Legendre-Fenchel dual [76])**.** *For any function $f : \mathbb{R}^d \to \mathbb{R} \cup \{-\infty, +\infty\}$, the function $f^* : \mathbb{R}^d \to \mathbb{R}$ defined by*

$$f^*(\boldsymbol{y}) := \sup_{\boldsymbol{w}} \{\langle \boldsymbol{y}, \boldsymbol{w}\rangle - f(\boldsymbol{w})\}$$

is the Fenchel conjugate or dual to f.

Definition 5 (hard-thresholding operator [14])**.** *We define the hard-thresholding operator for all $\boldsymbol{z} \in \mathbb{R}^d$ as the set $\pi_{HT}(\boldsymbol{z}) \subset \mathbb{R}^d$ below:*

$$\pi_{HT}(\boldsymbol{z}) := \arg \min_{\boldsymbol{w}\in\mathbb{R}^d \ s.t. \|\boldsymbol{w}\|_0 \le k} \|\boldsymbol{w} - \boldsymbol{z}\|_2^2$$

Remark 6. *$\pi_{HT}(\boldsymbol{z})$ keeps the k-largest values of $\boldsymbol{z}$ in magnitude: but if there is a tie between some values, several solutions exist to the problem above, and the set $\pi_{HT}(\boldsymbol{z})$ is not a singleton.*

Example 3. *With $k = 1$: $\pi_{HT}((2, 1)) = \{(2, 0)\}$ and $\pi_{HT}((2, 2)) = \{(2, 0), (0, 2)\}$*

Definition 6 (top-k norm)**.** *We define the following top-k norm $\|\cdot\|_{(k)}$, for all $\boldsymbol{w} \in \mathbb{R}^d$:*

$$\|\boldsymbol{w}\|_{(k)} = \|\pi^*_{HT}(\boldsymbol{w})\|_2$$

Where $\pi^*_{HT}(\boldsymbol{w})$ denotes any element from $\pi_{HT}(\boldsymbol{w})$ (since they all have the same norm). In other words, $\|\boldsymbol{w}\|_{(k)}$ is the ℓ_2 norm of the top-k elements from $\boldsymbol{w}$.

B.2 Recall on the conditions of recovery with l1 regularization

In this section, we briefly recall the conditions for sparse recovery with ℓ_1 norm regularization from [40], and why they are equivalent to Assumption 7. The authors of [40] proved in their Theorem 4.7 that this Condition 4.3 is a necessary and sufficient condition for achieving a linear rate of convergence for Tikhonov regularization with a priori parameter choice. We present below that original Condition 4.3:.

Assumption 8.

1. $\boldsymbol{w}^*$ *solves the equation* $\boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}$
2. *Strong source condition: There exist some* $\boldsymbol{\lambda} \in \mathbb{R}^n$ *such that :*

$$\boldsymbol{X}^\top \boldsymbol{\lambda} \in \partial \|\cdot\|_1(\boldsymbol{w}^*) \quad \textit{and} \quad |\langle \boldsymbol{x}_i, \boldsymbol{\lambda}\rangle| < 1 \quad \textit{for} \quad i \notin \textit{supp}(\boldsymbol{w}^*)$$

   where $\textit{supp}(\boldsymbol{w}^*)$ *is the support of* $\boldsymbol{w}^*$ *(that is, the set of the coordinates of its nonzero elements)*

3. *Restricted injectivity: The restricted mapping* $\boldsymbol{X}_{\textit{supp}(\boldsymbol{w}^*)}$ *is injective.*

We now show that this Assumption 8 is equivalent to the following assumption:

Assumption 9.

1. $\boldsymbol{w}^*$ *solves the equation* $\boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}$
2. *Restricted injectivity: The restricted mapping* $\boldsymbol{X}_{\textit{supp}(\boldsymbol{w}^*)}$ *is injective.*
3. $\boldsymbol{w}^*$ *verifies:*

$$\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \textit{sgn}(\boldsymbol{w}_S^*)\rangle| < 1 \qquad \textit{with } S = \textit{supp}(\boldsymbol{w}^*)$$

Lemma 2. *Assumptions 8 and 9 are equivalent.*

Proof. **Assume 8.** Since it is a known property of the ℓ_1 norm that: $[\partial\|\cdot\|_1(\boldsymbol{w}^*)]_i = [\operatorname{sgn}(\boldsymbol{w}_S^*)]_i$ if $i \in S$, and $[\partial\|\cdot\|_1(\boldsymbol{w}^*)]_i = [-1, 1]$ if $i \notin S$, we obtain that, for some $\boldsymbol{\lambda} \in \mathbb{R}^n$, and $S = \operatorname{supp}(\boldsymbol{w}^*)$, we have:

$$\begin{aligned}
&\{\forall i \in \operatorname{supp}(\boldsymbol{w}^*) : \boldsymbol{X}^\top \boldsymbol{\lambda} = \operatorname{sgn}(\boldsymbol{w}_S^*) \quad \text{and} \quad \forall i \notin \operatorname{supp}(\boldsymbol{w}^*) : \quad |\langle \boldsymbol{x}_i, \boldsymbol{\lambda}\rangle| < 1\} \\
&\implies \{\boldsymbol{X}_S^\top \boldsymbol{\lambda} = \operatorname{sgn}(\boldsymbol{w}_S^*) \quad \text{and} \quad \max_{\ell \in \bar{S}} |\langle \boldsymbol{x}_\ell, \boldsymbol{\lambda}\rangle| < 1\} \\
&\overset{(a)}{\implies} \{\boldsymbol{\lambda} = (\boldsymbol{X}_S^\dagger)^\top \operatorname{sgn}(\boldsymbol{w}_S^*) \quad \text{and} \quad \max_{\ell \in \bar{S}} |\langle \boldsymbol{x}_\ell, \boldsymbol{\lambda}\rangle| < 1\} \\
&\implies \max_{\ell \in \bar{S}} |\langle \boldsymbol{x}_\ell, (\boldsymbol{X}_S^\dagger)^\top \operatorname{sgn}(\boldsymbol{w}_S^*)\rangle| < 1 \implies \max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \operatorname{sgn}(\boldsymbol{w}_S^*)\rangle| < 1
\end{aligned}$$

Where (a) follows from the fact that $\boldsymbol{X}_S$ is injective (hence full column rank, which implies $\boldsymbol{X}_S^\dagger \boldsymbol{X}_S = \boldsymbol{I}$ hence $\boldsymbol{X}_S^\top (\boldsymbol{X}_S^\dagger)^\top = \boldsymbol{I}$).

Therefore, this implies 9.

Now, **assume 9**, and take $\boldsymbol{\lambda} = (\boldsymbol{X}_S^\dagger)^\top \text{sign}(\boldsymbol{w}_S^*)$.

Then, since $\boldsymbol{X}_S$ is injective, we have: $\boldsymbol{X}_S^\top (\boldsymbol{X}_S^\dagger)^\top \text{sign}(\boldsymbol{w}_S^*) = (\boldsymbol{X}_S^\dagger \boldsymbol{X}_S)^\top \text{sign}(\boldsymbol{w}_S^*) = \text{sign}(\boldsymbol{w}_S^*) = [\partial \|\cdot\|_1(\boldsymbol{w}^*)]_S$, and additionally, from condition 3 in 9:

$$\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \text{sgn}(\boldsymbol{w}_S^*)\rangle| < 1 \implies \max_{\ell \in \bar{S}} |\langle \boldsymbol{x}_\ell, (\boldsymbol{X}_S^\dagger)^\top \text{sgn}(\boldsymbol{w}_S^*)\rangle| < 1 \implies \max_{\ell \in \bar{S}} |\langle \boldsymbol{x}_\ell, \boldsymbol{\lambda}\rangle| < 1$$

Therefore, (since that last inequality also implies that for all $i \notin S, \langle \boldsymbol{x}_\ell, \boldsymbol{\lambda}\rangle \in [-1, 1] = [\partial \|\cdot\|_1(\boldsymbol{w}^*)]_i$), we finally have that this $\boldsymbol{\lambda}$ verifies the existence conditions from 8.

□

B.3 Proof of Theorem 2

Proof of Theorem 2. Theorem 2 follows by combining Lemma 3 with Theorem 4 from [57]: in particular, when plugging from Lemma 3 the value (denoted by $\boldsymbol{\lambda}^*$ in Lemma 3 (2)) of the solution of the dual problem of (B.1), (denoted by $\boldsymbol{v}^*$ in Theorem 4) we obtain:

$$\|\boldsymbol{v}^*\| = \|(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*\|$$

□

Lemma 3. *Under Assumptions 5 and 6 , we have, with* $\boldsymbol{\lambda}^* := -(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*$*:*

(1) $-\boldsymbol{X}^\top \boldsymbol{\lambda}^* \in \partial R(\boldsymbol{w}^*)$

(2) $\boldsymbol{\lambda}^*$ *is solution to the dual problem of the noiseless problem below:*

$$\begin{aligned} (I_{ks}\text{-noiseless}): \quad & \min_{\boldsymbol{w}} R(\boldsymbol{w}) \\ s.t. \quad & \boldsymbol{X}\boldsymbol{w} = \boldsymbol{y} \end{aligned} \tag{B.1}$$

Proof. **Proof of (1):**

We start by re-writing the condition $-\boldsymbol{X}^\top \boldsymbol{\lambda} \in \partial R(\boldsymbol{w}^*)$ (for any given $\boldsymbol{\lambda}$) into a form easier to check:

First, recall that $R(\boldsymbol{w}) = \frac{1-\alpha}{2}\|\boldsymbol{w}\|_k^{sp2} + \frac{\alpha}{2}{\|\boldsymbol{w}\|_2}^2$. We then have, for any $\boldsymbol{\lambda} \in \mathbb{R}^n$:

$$\{-\boldsymbol{X}^\top \boldsymbol{\lambda} \in \partial R(\boldsymbol{w}^*)\} \iff \{(1-\alpha)\partial(\frac{1}{2}\|\cdot\|_k^{sp2})(\boldsymbol{w}^*) \ni -\boldsymbol{X}^\top \boldsymbol{\lambda} - \alpha \boldsymbol{w}^*\} \tag{B.2}$$

$$\overset{(a)}{\iff} \{(1-\alpha)\boldsymbol{w}^* \in \partial(\frac{1}{2}\|\cdot\|_{(k)}^2)(-\boldsymbol{X}^\top \boldsymbol{\lambda} - \alpha \boldsymbol{w}^*)\} \tag{B.3}$$

$$\overset{(b)}{\iff} \{(1-\alpha)\boldsymbol{w}^* \in \text{conv}(\pi_{HT}(-\boldsymbol{X}^\top \boldsymbol{\lambda} - \alpha \boldsymbol{w}^*))\} \tag{B.4}$$

Where (a) follows from Proposition 2 and Corollary 3, and (b) from Lemma 4.

Let us now define $\boldsymbol{\lambda}^* := -(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*$. We then have:

$$\mathrm{conv}(\pi_{HT}(-\boldsymbol{X}^\top \boldsymbol{\lambda}^* - \alpha \boldsymbol{w}^*)) = \mathrm{conv}(\pi_{HT}(\boldsymbol{X}^\top (\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* - \alpha \boldsymbol{w}^*)) \tag{B.5}$$

We now use the fact that :

(A) $\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^* \rangle| < \min_{j \in S} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_j, \boldsymbol{w}_S^* \rangle|$

(B) $0 < \alpha < \frac{\min_{j \in S} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_j, \boldsymbol{w}_S^* \rangle| - \max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^* \rangle|}{\|\boldsymbol{w}^*\|_\infty}$ (from the choice of α described in Theorem 2)

Which implies, for all $i \in S$, that:

$$\begin{aligned}
[|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* - \alpha \boldsymbol{w}^*|]_i &\overset{(a)}{\geq} [|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*| - |\alpha \boldsymbol{w}^*|]_i \\
&= [|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*|]_i - \alpha [|\boldsymbol{w}^*|]_i \\
&\geq [|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*|]_i - \alpha \|\boldsymbol{w}^*\|_\infty \\
&\overset{(b)}{>} [|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*|]_i - \min_{j \in S} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_j, \boldsymbol{w}_S^* \rangle| + \max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^* \rangle| \\
&\geq \max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^* \rangle| \\
&= \max_{\ell \in \bar{S}} [|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*|]_\ell \\
&\overset{(c)}{=} \max_{\ell \in \bar{S}} [|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* - \alpha \boldsymbol{w}^*|]_\ell
\end{aligned}$$

Where (a) follows from the reverse triangle inequality, (b) follows from (B), and (c) follows from the fact that the support of $\boldsymbol{w}^*$ is S (so: $\forall j \in \bar{S} : \ \boldsymbol{w}_j^* = 0$).

Therefore, for all $i \in S, \ell \in \bar{S}$:

$$[|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* - \alpha \boldsymbol{w}^*|]_i > [|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* - \alpha \boldsymbol{w}^*|]_\ell$$

This allows us to simplify (B.5), given that the hard-thresholding operation selects the top k-components of a vector (in absolute value) (the conv operation disappears here because since the inequality above is strict, there are no "ties" when computing the top-k components (in absolute value)):

Therefore, for all $i \in [d]$:

$$[\text{conv}\left(\pi_{HT}(-\boldsymbol{X}^\top\boldsymbol{\lambda}^* - \alpha\boldsymbol{w}^*)\right)]_i = \begin{cases} \langle \boldsymbol{x}_i, (\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* \rangle - \alpha \boldsymbol{w}_i^* & \text{if} \quad i \in S \\ 0 \quad \text{if} \quad i \in \bar{S} \end{cases} \tag{B.6}$$

$$= \begin{cases} \langle \boldsymbol{x}_i, (\boldsymbol{X}_S^\top)^\dagger \boldsymbol{X}_S^\dagger \boldsymbol{y} \rangle - \alpha \boldsymbol{w}_i^* & \text{if} \quad i \in S \\ 0 \quad \text{if} \quad i \in \bar{S} \end{cases} \tag{B.7}$$

$$\overset{(a)}{=} \begin{cases} [\boldsymbol{X}_S^\dagger \boldsymbol{y}]_i - \alpha \boldsymbol{w}_i^* & \text{if} \quad i \in S \\ 0 \quad \text{if} \quad i \in \bar{S} \end{cases} \tag{B.8}$$

$$\overset{(b)}{=} \begin{cases} \boldsymbol{w}_i^* - \alpha \boldsymbol{w}_i^* & \text{if} \quad i \in S \\ 0 \quad \text{if} \quad i \in \bar{S} \end{cases} \tag{B.9}$$

$$= \begin{cases} (1-\alpha)\boldsymbol{w}_i^* & \text{if} \quad i \in S \\ 0 \quad \text{if} \quad i \in \bar{S} \end{cases} \tag{B.10}$$

Where (a) follows from the following property of the pseudo-inverse for a matrix $\boldsymbol{M}$, applied to $\boldsymbol{M} = \boldsymbol{X}_S^\top$: $\boldsymbol{M}\boldsymbol{M}^\dagger(\boldsymbol{M}^\top)^\dagger = (\boldsymbol{M}^\top)^\dagger$.

(This property can be understood using the Singular Value Decomposition (SVD) expression for the pseudo-inverse [37]: with $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$, we have: $\boldsymbol{M}\boldsymbol{M}^\dagger(\boldsymbol{M}^\top)^\dagger = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top\boldsymbol{V}\boldsymbol{D}^{-1}\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{V}^\top = \boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{V}^\top = (\boldsymbol{M}^\top)^\dagger$), and (b) follows from the fact that $\boldsymbol{w}^*$ is the min ℓ_2 norm solution on its support S (as we assumed in Assumption 5), so $\boldsymbol{X}_S^\dagger \boldsymbol{y} = \boldsymbol{w}_S^*$ (III, 2, Corr. 3, [12], [72]).

Therefore, aggregating (B.10) for all indices, we finally obtain:

$$\text{conv}\left(\pi_{HT}(-\boldsymbol{X}^\top\boldsymbol{\lambda}^* - \alpha\boldsymbol{w}^*)\right) = (1-\alpha)\boldsymbol{w}^*$$

That is, $\boldsymbol{\lambda}^*$ verifies (B.4).

So to sum up, under Assumptions 6 and 5, we have that, for $\boldsymbol{\lambda}^* := -(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*$: $-\boldsymbol{X}^\top\boldsymbol{\lambda}^* \in \partial R(\boldsymbol{w}^*)$.

Note: In addition, since (B.4) is equivalent to (B.2), plugging that value of $\boldsymbol{\lambda}^*$ into (B.2) we also have:

$$(1-\alpha)\partial(\frac{1}{2}\|\cdot\|_k^{sp2})(\boldsymbol{w}^*) \ni \boldsymbol{X}^\top(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* - \alpha\boldsymbol{w}^* \tag{B.11}$$

(This latter equation will be useful in the proof of (2) below)

Proof of (2):

We now turn to proving the second part (i.e. (2)) of Lemma 3.

As described in [57], the dual problem of (B.1) can be written as (see e.g. Definition 15.19 in [8]):

$$\min_{\boldsymbol{v}} R^*(-\boldsymbol{X}^\top\boldsymbol{v}) + \langle \boldsymbol{y}, \boldsymbol{v} \rangle \tag{B.12}$$

where R^* denotes the Fenchel Dual of R (see Definition 4).

Let us define, for all $\boldsymbol{v} \in \mathbb{R}^n$: $f(\boldsymbol{v}) = R^*(-\boldsymbol{X}^\top \boldsymbol{v})$

The first order optimality condition of problem (B.12) can be written as:

$$\partial f(\boldsymbol{v}) + \boldsymbol{y} \ni \boldsymbol{0}$$

Which is equivalent to:

$$-\partial f(\boldsymbol{v}) \ni \boldsymbol{y}$$

Therefore, if we find $\boldsymbol{v}$ such that the expression above is verified, then that $\boldsymbol{v}$ is solution of (B.12).

Now, from Theorem 23.9 in [76], we have that: $-\boldsymbol{X}\partial R^*(-\boldsymbol{X}^\top \boldsymbol{v}) \subset \partial f(\boldsymbol{v})$ (that is, the subgradient verifies a similar chain rule as the usual gradient, in one direction of inclusion).

Note now that since R is α-strongly convex (due to the squared ℓ_2 norm term), R^* is differentiable and α-smooth [46] and therefore, its gradient is well defined, so we can rewrite ∂R^* into ∇R^* (the subgradient is a singleton).

Now, take $\boldsymbol{v}^* := -(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*$.

Let us compute $\nabla R^*(-\boldsymbol{X}^\top \boldsymbol{v}^*) = \nabla R^*(\boldsymbol{X}^\top (\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*)$.

Let us denote $\boldsymbol{z} := \nabla R^*(\boldsymbol{X}^\top (\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*)$. From 2, we have the following equivalences:

$$\begin{aligned}
&\boldsymbol{X}^\top (\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* \in \partial R(\boldsymbol{z}) \\
&\iff \boldsymbol{X}^\top (\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* \in (1-\alpha)\partial(\frac{1}{2}\|\cdot\|_k^{sp2})(\boldsymbol{z}) + \alpha \boldsymbol{z} \\
&\iff \boldsymbol{X}^\top (\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* - \alpha \boldsymbol{z} \in (1-\alpha)\partial(\frac{1}{2}\|\cdot\|_k^{sp2})(\boldsymbol{z}) \qquad \text{(B.13)}
\end{aligned}$$

Now, we know from (B.11) that taking $\boldsymbol{z} := \boldsymbol{w}^*$ satisfies expression (B.13). Therefore: $\nabla R^*(-\boldsymbol{X}^\top \boldsymbol{v}^*) = \boldsymbol{w}^*$

Now, we can see that the proof is complete, since we know from Assumption 5 that $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^*$. So using the above, we have:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^* = \boldsymbol{X}\nabla R^*(-\boldsymbol{X}^\top \boldsymbol{v}^*) \in \{\boldsymbol{X}\nabla R^*(-\boldsymbol{X}^\top \boldsymbol{v}^*)\} = \boldsymbol{X}\partial R^*(-\boldsymbol{X}^\top \boldsymbol{v}^*) \subset -\partial f(\boldsymbol{v}^*)$$

So to sum up, we have that: $\boldsymbol{y} \in -\partial f(\boldsymbol{v}^*)$, which means that $\boldsymbol{v}^* = -(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*$ is solution of the dual problem of (B.1).

□

Theorem 4 ([57]). *Let $\delta \in]0, 1]$ and let $(\hat{\boldsymbol{w}}_t)_{t\in\mathbb{N}}$ be the sequence generated by ADGD (cf. [57]). Assume that there exists $\boldsymbol{\lambda} \in \mathbb{R}^n$ such that $-\boldsymbol{X}^T\boldsymbol{\lambda} \in \partial R(\boldsymbol{w}^*)$. Set $a = 4\|\boldsymbol{X}\|^{-1}$ and $b = 2\|\boldsymbol{X}\|\|\boldsymbol{v}^*\|/\alpha$, where $\boldsymbol{v}^*$ is a solution of the dual problem of* (B.1). *Then, for every $t \geq 2$,*

$$\|\hat{\boldsymbol{w}}_t - \boldsymbol{w}^*\| \leq at\delta + bt^{-1}. \qquad \text{(B.14)}$$

In particular, choosing $t_\delta = \lceil c\delta^{-1/2} \rceil$ for some $c > 0$,

$$\|\hat{\boldsymbol{w}}_t - \boldsymbol{w}^*\| \leq \left[a(c+1) + bc^{-1}\right]\delta^{1/2}. \qquad \text{(B.15)}$$

Proof. Proof in [57] □

B.4 Useful Results

Here we present some lemmas and theorems that are used in the proofs above:

Theorem 5 (Corollary 4.3.2, [6]). *Let $f_1, ..., f_m$ be m convex functions from $\mathbb{R}^d$ to $\mathbb{R}$ and define*

$$f := \max\{f_1, ..., f_m\}.$$

Denoting by $I(\boldsymbol{w}) := \{i : f_i(\boldsymbol{w}) = f(\boldsymbol{w})\}$ the active index-set, we have:

$$\partial f(\boldsymbol{w}) = conv(\cup \partial f_i(\boldsymbol{w}) : i \in I(\boldsymbol{w}))$$

Proof. Proof in [6]. □

Lemma 4 (Subgradient of the half-squared top-k norm). *Let n be the (half-squared) top k-norm: $n(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2_{(k)}$. We have:*

$$\partial n(\boldsymbol{w}) = conv(\pi_{HT}(\boldsymbol{w}))$$

Proof. Let us denote each possible supports of k coordinates from $\binom{d}{k}$ by $\mathcal{I}_i$ for $i = 1, ..., \binom{k}{d}$ The top-k norm can be written as follows:

$$n(\boldsymbol{w}) = \max_i n_i(x) = \max\{n_1(\boldsymbol{w}), ..., n_{\binom{k}{d}}(\boldsymbol{w})\}$$

where each $n_i = \frac{1}{2}\|\boldsymbol{w}_{\mathcal{I}_i}\|_2^2$, with $\boldsymbol{w}_{\mathcal{I}_i}$ the thresholding of $\boldsymbol{w}$ with all coordinates not in $\mathcal{I}_i$ set to 0. Let us denote, for a given $\boldsymbol{w} \in \mathbb{R}^d$, $\Pi(\boldsymbol{w}) \subset \binom{d}{k}$ to be the set of *supports* such that for any $j \in \Pi(\boldsymbol{w})$: $n_j(\boldsymbol{w}) = n(\boldsymbol{w})$. In other words, $\Pi(\boldsymbol{w})$ denotes the *active index set* described in Theorem 5. Those supports are those which select the top-k components of $\boldsymbol{w}$ in absolute value (several choices are possible). In other words:

$$\pi_{HT}(\boldsymbol{w}) = \{\boldsymbol{w}_{\mathcal{I}_j} : j \in \Pi(\boldsymbol{w})\}$$

Now, we know that for all $i \in \binom{k}{d}$, n_i is differentiable, since n_i is simply the half squared ℓ_2 norm of the thresholding of $\boldsymbol{w}$ on a fixed support $\mathcal{I}_i$. Since it is differentiable, its subgradient is thus a singleton composed of its gradient: $\partial n_i(\boldsymbol{w}) = \{\nabla n_i(\boldsymbol{w})\} = \{\boldsymbol{w}_{\mathcal{I}_i}\}$.

Therefore, from Theorem 5, we have:

$$\partial f(\boldsymbol{w}) = \text{conv}(\nabla f_i(\boldsymbol{w}) : i \in \Pi(\boldsymbol{w})) = \text{conv}(\boldsymbol{w}_{\mathcal{I}_j} : j \in \Pi(x)) = \text{conv}(\pi_{HT}(\boldsymbol{w}))$$

□

Proposition 2 (Proposition 11.3, [76]). *For any proper, lsc, convex function f, denote by f^* its Fenchel dual defined above in 4. One has $\partial f^* = (\partial f)^{-1}$ and $\partial f = (\partial f^*)^{-1}$.*

Proof. Proof in [76]. □

Lemma 5 (Fenchel conjugate of a half squared norm [15] Example 3.27, p. 94)**.** *Consider the function* $f(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2$*, where* $\|\cdot\|$ *is a norm, with dual norm* $\|\cdot\|_*$*. Its Fenchel conjugate is* $f^*(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_*^2$*.*

Proof. Proof in [15]. □

Lemma 6 (Dual of the k-support norm, [3], 2.1)**.** *Denote by* $(\|\cdot\|)_*$ *the dual norm of a norm* $\|\cdot\|$*. The top-k norm (see Definition 6) is the dual norm of the k-support :*

$$(\|\cdot\|_k^{sp})_* = \|\cdot\|_{(k)}$$

Corollary 3.

$$(\frac{1}{2}\|\cdot\|_k^{sp2})^* = \frac{1}{2}\|\cdot\|_{(k)}^2$$

Proof. Corollary 3 follows from Lemmas 5 and 6. □

B.5 Proximal operator of the k-support norm

In this section, we describe the method that we use to compute the proximal operator of the half-squared k-support norm, as is described in Algorithm 1 from [59]. In our code (available in the supplementary material), we use an existing implementation from the `modopt` package [30]. Note that Algorithm 1 from [59] was originally described in a more general formulation, from which the algorithm described below can be obtained by fixing $a = 0$, $b = 1$, and $c = k$ (we refer the reader to [59] for more details on what variables a, b, and c refer to).

Algorithm 4: ([59], Algorithm 1) Computation of $\boldsymbol{x} = \text{prox}_{\frac{\lambda}{2}\|\cdot\|_{(k)}^2}(\boldsymbol{w})$, with: $\forall \alpha \in \mathbb{R}_+ : S(\alpha) := \sum_{i=1}^d \min(1, \max(0, \alpha|w_i| - \lambda))$.

Initialization: *parameter:* λ. **1.** Sort points $\{\alpha^i\}_{i=1}^{2d} = \left\{\frac{\lambda}{|w_j|}, \frac{1+\lambda}{|w_j|}\right\}_{j=1}^{d}$ such that $\alpha^i \leq \alpha^{i+1}$.
2. Identify points α^i and α^{i+1} such that $S(\alpha^i) \leq k$ and $S(\alpha^{i+1}) \geq k$ by binary search.
3. Find α^* between α^i and α^{i+1} such that $S(\alpha^*) = k$ by linear interpolation.
4. Compute $\theta_i(\alpha^*) := \min(1, \max(0, \alpha^*|w_i| - \lambda))$ for $i = 1 \ldots, d$.
5. Return $x_i = \frac{\theta_i w_i}{\theta_i + \lambda}$ for $i = 1 \ldots, d$.

B.6 Experiment with a Correlated Design Matrix

In this section, we describe a simple linear regression setting with a correlated design matrix, i.e. where the matrix $\boldsymbol{X}$ is formed by n i.i.d. samples from a correlated d-dimensional

Gaussian random variable $\{X_1, .., X_d\}$ of zero mean and unit variance, such that:

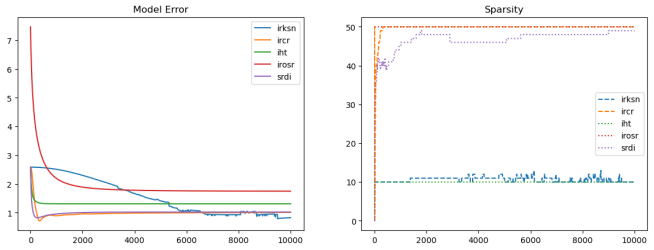
$$\forall i \in \{1, \dots, d\} : \mathbb{E}[X_i] = 0, \mathbb{E}[X_i^2] = 1;$$

$$\forall (i, j) \in \{1, \dots, d\}^2, i \neq j : \mathbb{E}[X_i X_j] = \rho^{|i-j|},$$

with $\rho = 0.2$. The generated dataset contains $n = 100$ samples of $d = 50$ features. Additionally, the true $\boldsymbol{w}^*$ is a 10-sparse vector (with a randomly drawn support), and $\boldsymbol{y}$ is obtained with a noise vector $\boldsymbol{\epsilon}$ created from i.i.d. samples from a random normal distribution, and rescaled to enforce a given signal to noise ratio (SNR), as follows:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\epsilon}$$

such that the signal to noise ratio is $\text{snr} = \frac{\|\boldsymbol{X}\boldsymbol{w}^*\|}{\|\boldsymbol{\epsilon}\|}$ (we fix it to snr := 3.). Such a dataset is generated using the `make_correlated_data` function from the `benchopt` package [61]. In Figure B.1 below, we run the iterative methods from Table 2.2 (IRKSN, IRCR, IROSR and SRDI) (as well as IHT for comparison), and measure the recovery error $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|$ as well as the sparsity $\|\hat{\boldsymbol{w}}\|_0$ of the iterates. For the parameter k in IRKSN and IHT, we set it to 10 (i.e. the true sparsity of $\boldsymbol{w}^*$). As we can see, IRKSN is the only algorithm that allows to achieve comparable or lower model error than other algorithms, while staying very close to the true sparsity of the true model.

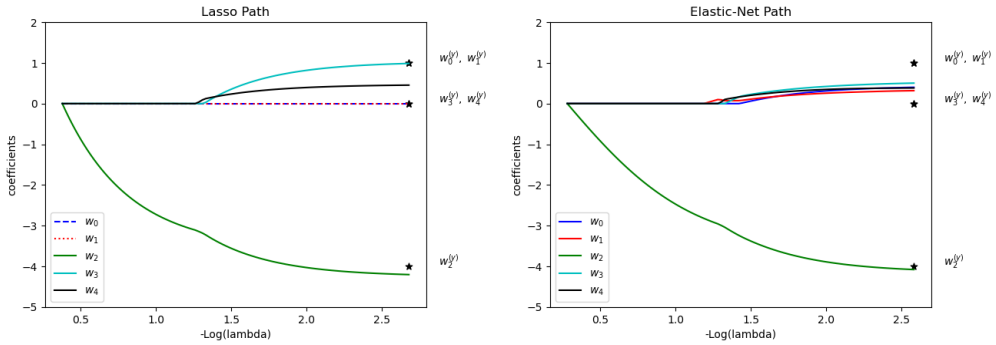


(a) Model error $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|$ (b) Model sparsity $\|\hat{\boldsymbol{w}}\|_0$

Figure B.1: Iterative regularization methods on a toy experiment with correlated design matrix.

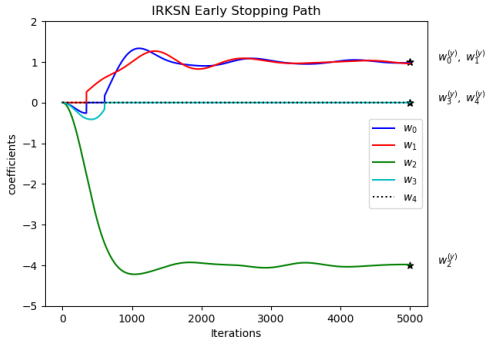
B.7 Path of IRKSN vs Lasso vs ElasticNet

In this section, we plot in Figure B.2 the path of ElasticNet (with an ℓ_1 ratio of 0.8, i.e. its penalty is $\lambda(0.8\|\cdot\|_1 + 0.2\|\cdot\|_2^2)$), in addition to the plot of the Lasso path and the IRKSN path, from section 3.2.2. As we can see, the ElasticNet, as the Lasso, cannot recover the true sparse vector.



(a) Lasso path

(b) ElasticNet path



(c) IRKSN path

Figure B.2: Comparison of the path of IRKSN with Lasso and Elasticnet

B.8 Details on the implementation of algorithms

In this section, we present additional details on the experiments from Section 3.2.3. First, for all the algorithms, we added a preprocessing step that centers and standardizes each column on the trainset (i.e. substract its mean and divides it by its standard deviation), and that removes columns that have 0 variance (i.e. column containing the same, replicated value). We later use this learned transformation on the validation set and the test set. In addition, we fit the intercept b of the linear regression separately, as is common in sparse linear regression, by centering the target $\boldsymbol{y}$ before training, and then using the below formula for the intercept:

$$b = \bar{y} - \langle \bar{\boldsymbol{X}}, \hat{\boldsymbol{w}} \rangle$$

Where $\bar{y}$ is the average of the target vector $\boldsymbol{y}$, $\hat{\boldsymbol{w}}$ is the final estimated model on the train set (fitted with a centered target $\boldsymbol{y} - \bar{y}$), and $\bar{\boldsymbol{X}}$ is the column-wise average of the (preprocessed) training data matrix $\boldsymbol{X}$. The prediction of a new preprocessed data sample $\boldsymbol{x}'_i$ is then $\hat{y}_i := \langle \hat{\boldsymbol{w}}, \boldsymbol{x}'_i \rangle + b$.

We recode most algorithms from scratch in `numpy` [41], except for the Lasso, ElasticNet, and OMP, for which we use the `scikit-learn` [71] implementation. For the implementation of the proximal operator of the (half-squared) k-support norm (used in IRKSN and KSN penalized), we use the existing implementation from the `modopt` package [30], that is based

on the efficient algorithm described in [59]. Below we present the grid-search parameters for each algorithms, that allowed them to achieve a good performance consistently on all datasets from Table 3.1. For all iterative regularization algorithms (i.e. SRDI, IROSR, IRCR, and IRKSN), we monitor the validation MSE every 5 iterations, and choose the stopping time as the iteration number with the best MSE. For IHT we also compute an early stopping time based on the validation MSE at each iteration: indeed, IHT is a non-convex algorithm and therefore the last iterate is not necessarily the best one (decrease is not ensured at each iteration). We run each iterative algorithm that we reimplemented (IHT, KSN penalty, SRDI, IROSR, IRCR, IRKSN) with a maximum number of iterations of 500. Finally, we release our code in the Supplementary Material.

IHT [14] We search k (the number of components kept at each iterations) in an evenly spaced interval from 1 to d containing 5 values, and search the learning rate η in $[0.0001, 0.001, 0.01, 0.1, 1.]$.

Lasso [83] We use the implementation `lasso_path` from `scikit-learn` [71], with its default parameters, which automatically choses the path of λ based on a data criterion.

ElasticNet [102] We use the implementation `enet_path` from `scikit-learn` [71], which similarly as above, automatically chooses the path of λ based on a data criterion. In addition, we choose the recommended values $[.1, .5, .7, .9, .95, .99, 1]$ of `ElasticNetCV` for the relative weight of the ℓ_1 penalty.

KSN penalty [3] We choose the strenght of the k-support norm penalty λ in $[0.1, 1.]$, the k (from the k-support norm) in an evenly spaced interval from 1 to d containing 5 values, and we found that simply setting the constant L from [3] (which is the inverse of the learning rate) to $1e6$ achieves consistently good results across all datasets.

OMP [85] We use the implementation from `scikit-learn` [71], and we search k in an evenly spaced interval from 1 to $\min(n, d)$ (indeed, OMP needs k not to be bigger than $\min(n, d)$) containing 5 values. **SRDI** [68] We search for the parameters κ and α from [68], respectively in the intervals $[0.0001, 0.001, 0.01, 0.1, 1.]$ and $[0.0001, 0.001, 0.01, 0.1, 1.]$. **IROSR** [89] We search for the parameters η and α respectively in $[0.0001, 0.001, 0.01, 0.1, 1.]$ and $[0.0001, 0.001, 0.01, 0.1, 1.]$.

IRCR [60] For IRSR, we found that setting τ and σ to $\frac{0.9}{\sqrt{2\|\boldsymbol{X}\|^2}}$ (in order to verify the condition of equation (6) in [60]) consistently performs well on all datasets.

IRKSN (ours) For IRKSN, we search α (from Algorithm 2) in $[0.0001, 0.001, 0.01, 0.1, 1, 10]$, and k (from the k-support norm), in an evenly spaced interval from 1 to d containing 5 values. For the `RHEE2006` dataset, we found that the hyperparameters need to be tuned slighty more to attain comparable performance with other algorithms: the reported performance is for $\alpha = 0.6$, $k = 33$, ran for 1,000 iterations.