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

by

William de Vazelhes

PhD 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
Assistant Professor, Department of Machine Learning,
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)

External Examiner: Dr. Xiao-Tong Yuan
Professor, School of Intelligence Science and Technology,
Nanjing University

Internal Member: Dr. Chih-Jen Lin
Affiliated Professor, Department of Machine Learning,
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)

Internal Member: Dr. Karthik Nandakumar
Associate Professor, Department of Machine Learning,
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)

Internal Member: Dr. Zhiqiang Xu
Assistant Professor, Department of Machine Learning,
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

The primary contribution of this thesis is to analyze several new extensions of the Iterative Hard-Thresholding (IHT) algorithm. We first focus on analyzing a zeroth-order extension of IHT, zeroth-order hard-thresholding (ZOHT): in particular, we analyze the conflict between the error of the zeroth-order gradient estimator, and the expansivity of the hard-thresholding operator. We prove global convergence guarantees in the restricted strongly convex (RSC) and restricted smooth (RSS) setting, for such algorithm. We then analyze the convergence of variance-reduction variants of the ZOHT algorithm (in the RSC and RSS settings), and analyze how the conditions on the number of random directions are improved. We then propose a variant of the original proof of convergence of ZOHT in the non-convex and discontinuous setting, useful for instance in a reinforcement learning setting. Then, we analyze a generalization of IHT which can tackle additional convex constraints verifying mild assumptions, in the zeroth-order and first-order (stochastic and deterministic) settings: when doing so, we also revisits previous proofs of convergence in risk for IHT, providing simpler proofs for existing results, removing the original system error present in the first proof of ZOHT, and extending the convergence result of all those algorithms to the case with extra constraints. Finally, we analyze an algorithm for sparse recovery, IRKSN (Iterative Regularization with k-Support Norm), inspired by a dual perspective on IHT, and show that its provides different conditions for recovery than IHT and usual sparse recovery algorithms based on the ℓ_1 norm, therefore providing a useful complement to those algorithms for sparse recovery.

In memory of my grandmother Josèphe.
To my wife Mouna and my daughter Ayla.

"'Tis a lesson you should heed–
Try again;
If at first you don't succeed,
Try again."

– William Hickson

Acknowledgements

First and foremost, I would like to thank my supervisor, Bin Gu, for his everlasting support and guidance. He helped me, advised me, and encouraged me during the thesis; thanks to him I could accomplish more than I thought I could. I would also like to express my gratitude to Xiao-Tong Yuan for sharing with me his enthusiasm in exploring proof techniques for hard-thresholding algorithms. I also extend my gratitude to the other members of the jury, Chih-Jen Lin, Karthik Nandakumar, and Zhiqiang Xu and to the professors with whom I have co-authored papers, with special thanks to Huan Xiong.

I would also like to thank all the professors at MBZUAI for their amazing lectures, in particular Martin Takáč for his optimization course, our Department Chair Kun Zhang, our Provost Timothy Baldwin, our President Eric Xing, as well as all the staff of MBZUAI and all the persons involved in making this place so special and caring.

I would also like to thank my friends, colleagues, students and researchers who I met during my PhD, especially Hilal, Bhaskar, Velibor, Srinivas, Haiyan, Dilshod, Chengqian, Zhenhao, Boyang, Zhengqing, Shunxing, Hualin, Wanli, Xinzhe, Xiaohan, Huimin, Anas, Klea, Nhat, The Phong, Steven, Sahal, Bakhita, Hashmat, Asma, Faris, Naif, Abdelrahman, Sultan, Sulaiman, Numan, Ashraf, Hanan, Shahina, Salem, Brahim, Mijd, Nada, Xiangrui, Alex, Nicolas, Six, Sasha, Abdurakhmon, and many more.

I also thank all the colleagues, professors, and friends, whom I was lucky to cross paths with, earlier on in my journey before starting a PhD: Romain, Nicolas, Merwan, Matthieu, Philippe, Thibault, Jean-Cyril, Thomas C., Thomas M., Timothée, Victor, Clément, Aurélien, Nathalie, Marc, Rémi, Pascal, Mikaela, Carlos, Mathieu, Mahsa, César, Mariana, Onkar, Brij, Arijus, Igor, Moitree, Pradipta, Géraud, Thibault, Tanh, Michal, Gaël, Alexandre, Olivier, Guillaume, Loïc, Roman, Nicolas, Joan, Joris, Matei, Pierre, Septimia, Abdellatif, Charles, Alexis, Paul, Maxime, and many more.

Aside from work, I am also grateful for my friends from France, the UAE, and beyond, with whom I can always unwind and recharge.

I am also deeply grateful to my family for their everlasting support.

Finally, I am eternally grateful to my amazing wife Mouna for her loving support and for standing by my side throughout the years.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivations

In machine learning, ensuring sparsity of the learned model is essential for numerous reasons such as computational (to reduce the size of the model and improve the speed and memory cost of a model), or statistical (for instance for variable selection to potentially increase the quality of predictions, or for model recovery or interpretability). The Iterative Hard-Thresholding (IHT) [20] is a fundamental algorithm to achieve sparse learning, which possesses several desirable properties. Indeed, it allows a practitioner to enforce a fixed sparsity k for all the iterates of the algorithms (as well as the returned solution), without requiring to tune any hyperparameter as is the case for instance with several methods such as the Lasso [139]. However, so far, such an algorithm still leaves several open questions which we seek to address in this thesis.

First, IHT is a (non-convex) projected gradient descent algorithm, and as such, requires access to the gradient of the objective function. However, in some cases, such gradient may be inaccessible, for computational reasons (it can be too costly to obtain, such as in certain graphical modeling tasks [146]), or for privacy reasons (the objective function considered may be computed partially on a private remote location, inaccessible to the optimizer, which arises for instance when the dataset is private as in distributed learning [66, 161], or in black-box adversarial attacks when the model is private [95]). In such cases, one can resort to zeroth-order methods, which only requires input and outputs of the considered function. However, is the zeroth-order estimator of the gradient compatible with hard-thresholding, and if so, how to tune the various parameters of such an algorithm to ensure convergence ? Also, dependence on the dimension was shown to be unavoidable for strongly convex and smooth zeroth-order optimization problems [77]: is it avoidable for zeroth-order hard-thresholding algorithms (which are non-convex algorithms) ?

Additionally, zeroth-order algorithms in the convex and general non-convex case are often combined with variance reduction techniques [68, 72, 79] when the function has a finite-sum structure (as in empirical risk minimization in machine learning), in order to obtain a fast convergence (similar to the one from full-batch optimization methods), with the low computational cost of stochastic (mini-batch) methods. Can we apply such a technique to the zeroth-order hard-thresholding algorithm above in order to make them more applicable ? Furthermore, zeroth-order optimization is tightly related to evolution strategies in reinforcement learning [129]. As such, it would be important to prove convergence of zeroth-order hard-thresholding (ZOHT) in such a case, in order to use ZOHT for reinforcement learning problems. However, in a reinforcement setting, the cost function is usally discontinuous and non-convex. Therefore, can ZOHT converge in such non-convex and discontinuous setting, and how to tune its parameters to ensure such convergence ?

Then, in several applications such as in portfolio optimization, one may require additional constraints in addition to the sparsity constraints. Indeed, in such portfolio optimization example, one may seek to enforce a total budget constraint on the investments, which can be enforced through an extra ℓ_1 constraint, as in [138]. As another example, in sparse non-negative matrix factorization, when estimating the hidden components, one may seek to enforce both a norm constraint and a sparsity constraint [73]. In the convex case, it is known that the intersection of two convex sets is also convex, and as such, combining constraints can easily be done as long as one can project onto the combined constraints: convergence of the optimization procedure will then be ensured under standard convex optimization assumptions. But in the non-convex sparse case, which algorithm can successfully ensure both the sparse and the extra constraints ? Under which conditions will convergence be ensured, in the deterministic, stochastic (first-order), and zeroth-order settings ?

Finally, IHT is known to suffer from restrictive applicability conditions: in the compressed sensing and sparse recovery literature, such conditions are ensured by the Restricted Isometry Property [33], which is known to be unrealistic in many high-dimensional optimization settings [76]. Therefore, could we derive alternative algorithms than IHT, in particular in the compressed sensing setting, which would ensure recovery under different, hence complementary conditions ?

1.2 Background

In this section, we introduce the basic notions that will use in the thesis.

1.2.1 Convex Optimization

Although our work is mostly about hard-thresholding algorithms, which are non-convex optimization algorithms as they optimize over a non-convex constraint (the ℓ_0 pseudo-ball), we will provide guarantees that are global (using the restricted strong convexity and restricted smoothness assumption which we will define below), and deriving such guarantees heavily build on tools and techniques from the convex optimization literature. Below, we provide some usual definitions from the convex optimization literature, which are the most common assumptions used to prove convergence of first-order and zeroth-order methods, namely convexity, strong convexity and smoothness. For simplicity of exposition, we consider such definitions in the case where f is differentiable, although the notions of convexity and strong convexity can be defined even if f is not differentiable. These notions can be found in [110].

Definition 1 (Convexity)**.** *A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be convex if for all $(\boldsymbol{x}, \boldsymbol{y}) \in (\mathbb{R}^d)^2$:*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle. \tag{1.1}$$

Such a property of functions is at the same time very commonly encountered in many usual settings in machine learning, such as in linear regression and logistic regression, and is also very powerful, as when f is convex, it is possible to prove the convergence of first order (such as (sub)-gradient descent) methods to solve the following problem:

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x}). \tag{1.2}$$

Furthermore, under some additional assumptions, faster convergence of gradient descent methods can be proven. We describe below such conditions:

Definition 2 (ν-strong convexity)**.** *A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be ν-strongly convex if for all $(\boldsymbol{x}, \boldsymbol{y}) \in (\mathbb{R}^d)^2$:*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\nu}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2 \tag{1.3}$$

Such a property ensures that the *curvature* of the function is bounded from below. This is useful amongst other things to ensure that the minimum of f over $\mathbb{R}^d$ is unique, and to prove fast convergence of first-order method. We now turn to another useful property of functions, used in the field of convex optimization, but also in non-convex optimization:

Definition 3 (L-smoothness)**.** *A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be L-smooth, if it is differentiable, and there exist a generic constant L such that for all $(\boldsymbol{x}, \boldsymbol{y}) \in (\mathbb{R}^d)^2$:*

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L \|\boldsymbol{x} - \boldsymbol{y}\| \tag{1.4}$$

Such a definition ensures that the function f is *well-behaved* enough, more precisely, that the gradient does not change too abruptly. If one knows that f is convex, then an alternative definition of smoothness is as follows:

Definition 4 (L-smoothness (alt.))**.** *A convex function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be L-smooth, if it is differentiable, and there exist a generic constant L such that for all $(\boldsymbol{x}, \boldsymbol{y}) \in (\mathbb{R}^d)^2$:*

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{L}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2 \tag{1.5}$$

1.2.2 Stochastic Optimization

In many cases, one actually does not directly observe $f(\boldsymbol{x})$ for some $\boldsymbol{x}$, but rather, one observes a noisy version of $\boldsymbol{x}$. One of such cases occurs for instance when f can be expressed as an expectation over a random variable $\boldsymbol{\xi}$, as below:

$$f(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{\xi}} f(\boldsymbol{x}, \boldsymbol{\xi}) \tag{1.6}$$

For simplicity, in such case we may denote $f_{\boldsymbol{\xi}}(\boldsymbol{x}) := f(\boldsymbol{x}, \boldsymbol{\xi})$ for all $\boldsymbol{x} \in \mathbb{R}^d$. In such a case, one may needs some assumptions on the variance of the gradient, at a particular point (usually the optimum of f, $\boldsymbol{x}^*$):

Definition 5 (σ^2-FGN [64], Assumption 2.3 (Finite Gradient Noise))**.** *f is said to have σ-finite gradient noise at $\boldsymbol{x}^*$ if for almost any $\boldsymbol{\xi}$, $f_{\boldsymbol{\xi}}$ is differentiable and the gradient noise $\sigma = \sigma(f, \boldsymbol{\xi})$ defined below is finite:*

$$\sigma^2 = \mathbb{E}_{\boldsymbol{\xi}}[\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|_2^2] \tag{1.7}$$

1.2.2.1 Finite-Sum Optimization

In machine learning, one often minimizes a function which can be expressed as an average of n terms, which is also called Empirical Risk Minimization. More precisely, the function f to be minimized can be expressed as follows, for all $\boldsymbol{w} \in \mathbb{R}^d$:

$$f(\boldsymbol{w}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{w}), \tag{1.8}$$

where each $f_i(\boldsymbol{w})$ is often of the form $g(\boldsymbol{w}, \boldsymbol{x}_i, y_i)$ for some function g (which is for instance the logistic loss), and some samples $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ and labels $y_1, ..., y_n$. Such case is therefore a special case of the stochastic case above, where $\boldsymbol{\xi}$ is actually the random index of each sample, which follows a uniform distribution with probability $\frac{1}{n}$ for each index.

1.2.3 Zeroth-Order Optimization

In several settings, one actually cannot access the gradient $\nabla f(\boldsymbol{x})$, for instance if the function values are obtained through a long recurrent process which would make backpropagation too costly [11, 134], or if computing $\nabla R(\boldsymbol{w})$ is too expensive such as in certain graphical modeling tasks [146], or if the dataset is private as in distributed learning [66, 161], or if the model is private as in black-box adversarial attacks [95]. In such a case, one can take an existing first order method, but replace the gradient by an approximation of it, which is based on the finite approximation of the gradient. Some review of such methods can be found for instance in [18] and [93], a first introduction of such methods using gaussian random smoothing can be found in [113], and a first appearance of such methods can be found in [136]. In this paragraph, we present one of such methods, described for instance in [57], which approximates the gradient using q random directions $\boldsymbol{u}_1, ..., \boldsymbol{u}_q$ sampled independently and uniformly at random along the unit sphere in $\mathbb{R}^d$, and with the following approximation of the gradient $\hat{\nabla} f(\boldsymbol{x})$:

$$\hat{\nabla} f(\boldsymbol{x}) := \frac{d}{q} \sum_{i=1}^{q} \frac{f(\boldsymbol{x} + \mu \boldsymbol{u}_i) - f(\boldsymbol{x})}{\mu} \boldsymbol{u}_i, \tag{1.9}$$

where μ is a *smoothing radius*, which should be taken as small as possible, as much as is allowed by the machine precision such that it will not introduce numerical errors. As will be discussed in more details later, such an estimator of the gradient is actually biased, with a bias growing with μ. Additionally, in order to reduce its variance, one may sample more random directions q, and in this thesis we will study the impact of such choice of q in the setting of hard-thresholding zeroth-order algorithms.

1.2.4 Constrained Convex Optimization

We now make a first step towards the main topic of this thesis, by now describing the constrained convex optimization problem. The results in this section can be found in [110]. Constrained convex optimization problems can generally be formulated as follows:

$$\min_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x}), \tag{1.10}$$

where $\mathcal{C}$ is a convex constraint set. We give below the definition of a convex set, which essentially states that any point in between two points of the set must also belong to the set:

Definition 6 (Convex Set)**.** *A set $\mathcal{C} \subseteq \mathbb{R}^d$ is convex if, for all $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{C}^2$, and for all $\lambda \in [0, 1]$: $\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y} \in \mathcal{C}$.*

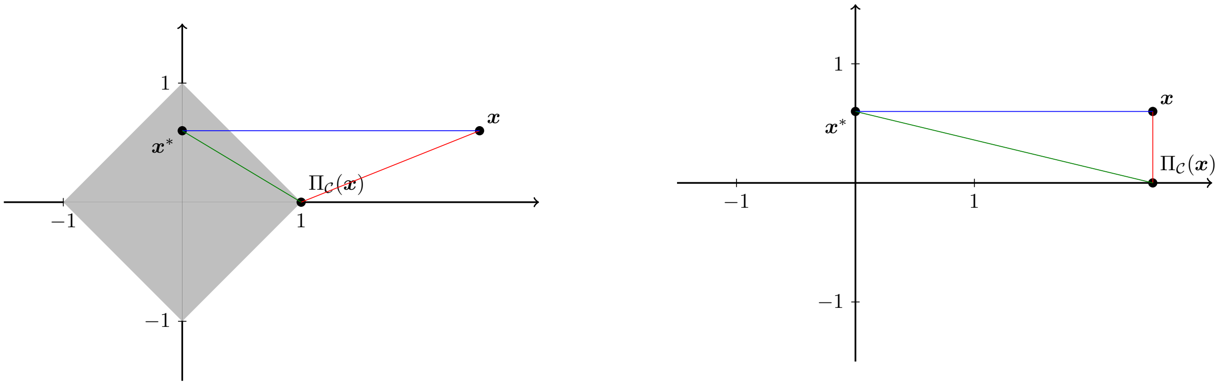
Proofs in constrained convex optimization usually build on the following properties of projection onto convex sets, where we denote by $\Pi_{\mathcal{C}}(\boldsymbol{x})$ the projection of $\boldsymbol{x}$ onto $\mathcal{C}$.

1.2.4.1 Non-Expansivity and Three-Point Lemma

Lemma 1 (Non-expansivity)**.**

$$\forall (\boldsymbol{x}, \boldsymbol{x}^*) \in \mathbb{R}^d \times \mathcal{C} : \; \|\boldsymbol{x} - \boldsymbol{x}^*\| \geq \|\Pi_{\mathcal{C}}(\boldsymbol{x}) - \boldsymbol{x}^*\| \tag{1.11}$$

This property essentially states that when one projects onto $\mathcal{C}$, one gets closer to any given point $\boldsymbol{x}^* \in \mathcal{C}$ from the constraint. Actually, one can even be more precise on the amount by which one gets closer to $\boldsymbol{x}^*$, by using a stronger version of the non-expansivity lemma. Such lemma is also sometimes called the three-point lemma when used in a general Bregman divergence form to prove convergence of mirror descent for smooth functions



(a): Projection onto the ℓ_1 unit ball.

(b): Projection onto the ℓ_0 unit pseudo-ball.

Figure 1.1: For projection onto the ℓ_1 ball, we have $\|\boldsymbol{x} - \boldsymbol{x}^*\|^2 \geq \|\boldsymbol{x} - \Pi_{\mathcal{C}}(\boldsymbol{x})\|^2 + \|\Pi_{\mathcal{C}}(\boldsymbol{x}) - \boldsymbol{x}^*\|^2$ (three-point lemma), but this is not true if $\mathcal{C}$ is the ℓ_0 pseudo-ball. In that case, even the weaker non-expansivity property ($\|\boldsymbol{x} - \boldsymbol{x}^*\| \geq \|\Pi_{\mathcal{C}}(\boldsymbol{x}) - \boldsymbol{x}^*\|$) is not verified in general.

in [27]. It is indeed sometimes necessary to use such lemma rather than the non-expansivity of the hard-thresholding operator, in some specific proofs [1], and we present it below.

Lemma 2 (Convex Three-Point Lemma)**.**

$$\forall (\boldsymbol{x}, \boldsymbol{x}^*) \in \mathbb{R}^d \times \mathcal{C} : \ \|\boldsymbol{x} - \boldsymbol{x}^*\|^2 \geq \|\boldsymbol{x} - \Pi_{\mathcal{C}}(\boldsymbol{x})\|^2 + \|\Pi_{\mathcal{C}}(\boldsymbol{x}) - \boldsymbol{x}^*\|^2 \tag{1.12}$$

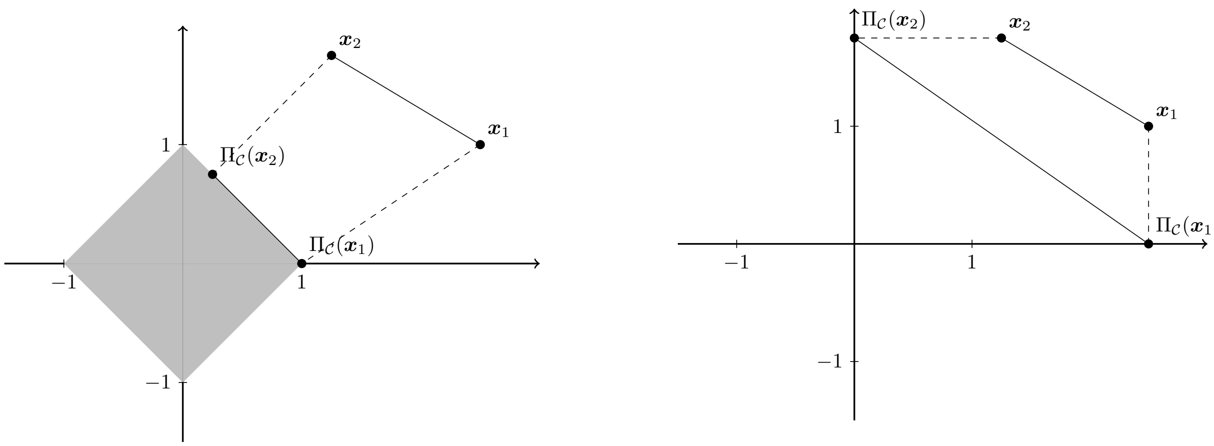
However, such two properties above are not verified if the set $\mathcal{C}$ is not convex, as we illustrate on Figure 1.1, where we plot the projection operator onto the ℓ_1 unit ball, as well as the projection operator onto the ℓ_0 pseudo-ball [2] of radius k, also known as the *hard-thresholding operator* [108], which keeps the k-largest values (in magnitude) of a given vector (if there are ties between components, one may break ties randomly or based on lexicographical order of the component index). As we can observe in Figure 1.1 (b), the projection of $\boldsymbol{x}$ onto the ℓ_0 unit ball is actually *further away* from $\boldsymbol{x}^*$, hence the non-expansivity is not verified (and consequently the three-point lemma is not verified either). As we will discuss in Chapter 2 (resp. Chapter 3), it is however possible to keep some of the techniques from convex optimization, and to replace the non-expansivity of projection onto a convex set (resp. the convex three-point lemma), by a modified version which is valid for projection onto the ℓ_0 pseudo-ball.

1.2.4.2 Contractivity

We now turn to describe another property of projection, namely, the contractivity.

[1] In particular, the proof for constrained optimization in the smooth case, cf. Proof of Theorem 3.4 here: https://raw.githubusercontent.com/epfml/OptML_course/master/lecture_notes/lecture-notes.pdf.

[2] The ℓ_0 pseudo-ball of radius k denotes the set $\mathcal{C} = \{\boldsymbol{x} \in \mathbb{R}^d : \ \|\boldsymbol{x}\|_0 \leq k\}$, where $\|\cdot\|_0$ is the ℓ_0 pseudo-norm, denoting the number of non-zero components of a vector.



(a): Projection onto the ℓ_1 unit ball.

(b): Projection onto the ℓ_0 unit pseudo-ball.

Figure 1.2: For projection onto the ℓ_1 ball, we have $\|\Pi_{\mathcal{C}}(\boldsymbol{x}_1) - \Pi_{\mathcal{C}}(\boldsymbol{x}_2)\| \leq \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$ (contractivity), but this is not true if $\mathcal{C}$ is the ℓ_0 pseudo-ball.

Lemma 3 (Contractivity of projection onto a convex set)**.**

$$\forall(\boldsymbol{x}_1, \boldsymbol{x}_2) \in (\mathbb{R}^d)^2 : \ \|\Pi_{\mathcal{C}}(\boldsymbol{x}_1) - \Pi_{\mathcal{C}}(\boldsymbol{x}_2)\| \leq \|\boldsymbol{x}_1 - \boldsymbol{x}_2\| \tag{1.13}$$

Such property essentially states that the distance between the projections of two points is smaller than the original distance between those two points. We illustrate such a property in Figure 1.2. As we can see there too, such a property is not verified for projection onto the ℓ_0 pseudo-ball (hard-thresholding operator). Such property can also be read as the Lipschitz smoothness of a *potential function*, which gradient is the projection operator. As such, it is also related to the strong convexity of the Fenchel dual of such potential, as we will discuss in Section 4.1. We introduce the Fenchel dual in Section 1.2.6.3 below, and will explore in more details this view of the projection operator as some gradient of a potential function, by analyzing the Dual Averaging algorithm, and finding some ways to deal with this departure from convex projection, in particular in Section 4.1.

1.2.5 Hard-Thresholding Algorithm

In this thesis, we consider mostly the following constrained optimization problem over the ℓ_0 pseudo-ball:

$$\min_{\boldsymbol{x} \text{ s.t. } \|\boldsymbol{x}\|_0 \leq k} f(\boldsymbol{x}) \tag{1.14}$$

We now discuss the main algorithm which we consider in this thesis, which is the hard-thresholding algorithm, and which goal is to solve the problem above (approximately, since the problem above is NP-hard as we will discuss). An early appearance of such algorithm can be found in [20]. It is a projected gradient descent algorithm, where the projection is onto the ℓ_0 pseudo-ball of radius k, which constitutes the hard-thresholding operator denoted $\mathcal{H}_k$ below. It consists in keeping only the k largest components of a vector (in absolute value), and setting all other components to 0. Note that if ties are present, any method to break them (e.g. lexicographically) is admissible. We describe it below:

```
Algorithm 1: Hard-Thresholding
  Initialization: $\boldsymbol{x}_0$
  for $t = 0, ..., T-1$ do
      $\boldsymbol{x}_{t+1} := \mathcal{H}_k(\boldsymbol{x}_t - \eta \nabla f(\boldsymbol{x}_t))$
  end
  Output : $\hat{\boldsymbol{x}_T} :=$ e.g. $\boldsymbol{x}_T$ or $\arg\min_{\boldsymbol{x} \in \{\boldsymbol{x}_i\}_{i=1}^T} f(\boldsymbol{x})$
```

As will be discussed later, in this thesis, we will consider proofs which provide approximate global guarantees of convergence to such a problem above. Therefore, we will need some variants of the usual smoothness and strong convexity assumptions, which are tailored to the structure of the problem.

Definition 7 ((ν_s, s)-RSC, [76,88,96,108,116,133,157])**.** *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 $\|\boldsymbol{x} - \boldsymbol{y}\|_0 \leq s$:*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\nu_s}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2 \tag{1.15}$$

Definition 8 ((L_s, s)-RSS, [116,133])**.** *A function f 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 $\|\boldsymbol{x} - \boldsymbol{y}\|_0 \leq s$:*

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L_s \|\boldsymbol{x} - \boldsymbol{y}\| \tag{1.16}$$

Essentially, such definitions are similar to the ones from convex optimization, except that they only need to be enforced on a subset of the considered space.

1.2.6 Non-smooth Optimization

In Chapter 4, we will need some additional tools from non-smooth convex optimization, which we introduce below. Such definitions can be found in [110] and [121].

1.2.6.1 Subgradient

Definition 9. *Consider a convex function $f : \mathcal{C} \subseteq \mathbb{R}^d \to \mathbb{R}$. We say that $\boldsymbol{g}$ is a subgradient of f at $\boldsymbol{x}$, if for all $\boldsymbol{y} \in \mathcal{C}$:*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{g}, \boldsymbol{y} - \boldsymbol{x} \rangle \tag{1.17}$$

The set of all subgradients of f at $\boldsymbol{x}$ is called the subdifferential of f at $\boldsymbol{x}$ and is denoted $\partial f(\boldsymbol{x})$.

In other words, the notion of subgradient allows to define the notion of *tangent plane(s)* to a curve of a function, even when the gradient is not defined at a particular point.

1.2.6.2 Proximal Operator

Another useful notion when dealing with non-smooth convex functions f is the notion of proximal operator. Such an operator is also a generalization of the projection onto a convex set, and is defined as follows:

Definition 10 (Proximal Operator)**.** *Given a function $f : \mathbb{R}^d \to \mathbb{R}$, the proximal operator of f at $\boldsymbol{x} \in \mathbb{R}^d$ is defined by:*

$$prox_f(\boldsymbol{x}) = \inf_{\boldsymbol{y}} f(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 \tag{1.18}$$

One can see for instance that for a convex set $\mathcal{C}$, denoting by $\mathbf{1}_{\mathcal{C}}$ the indicator function of that set (i.e. the function $\mathbf{1}_{\mathcal{C}} : \boldsymbol{x} \to \begin{cases} 0 \text{ if } \boldsymbol{x} \in \mathcal{C} \\ +\infty \text{ otherwise} \end{cases}$), the projection operator onto $\mathcal{C}$ is actually the proximal operator of that indicator function, that is, we have:

$$\mathrm{prox}_{\mathbf{1}_{\mathcal{C}}} = \Pi_{\mathcal{C}} \tag{1.19}$$

1.2.6.3 Fenchel Duality

In Chapter 4, we will make use of the notion of Fenchel duality (also called convex duality). Such notion leads, amongst other things, to powerful theorems, which allow to easily derive closed form for several proximal operators or subgradient of functions for instance, using known form for the Fenchel dual (also called convex dual) of such functions and for the proximal operator or subgradient of such dual function, and using theorems relating the proximal operator or subgradient of a function and its dual.

Definition 11 (Fenchel dual)**.** *Given a function $f : \mathbb{R}^d \to \mathbb{R}$, the Fenchel dual of f, denoted f^*, is the following function, such that for all $\boldsymbol{y} \in \mathbb{R}^d$:*

$$f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x} \in \mathbb{R}^d} \langle \boldsymbol{x}, \boldsymbol{y} \rangle - f(\boldsymbol{x}) \tag{1.20}$$

1.3 Bibliographic Notes

Our thesis is based on the works described below (* denotes equal contribution):

- *Zeroth-Order Hard-Thresholding: Gradient Error vs. Expansivity* [46], by William de Vazelhes, Hualin Zhang, Huimin Wu, Xiao-Tong Yuan, Bin Gu. Published in NeurIPS 2022. Included in Chapter 2.

- *New Insight of Variance Reduce in Zero-Order Hard-Thresholding: Mitigating Gradient Error and Expansivity Contradictions* [159], by Xinzhe Yuan, William de Vazelhes, Bin Gu, Huan Xiong. Published in ICLR 2023. Included in Chapter 2.

- *Hard-Thresholding Meets Evolution Strategies in Reinforcement Learning*, by Chengqian Gao*, William de Vazelhes*, Hualin Zhang, Zhiqiang Qu, Bin Gu. Accepted at IJCAI 2024. Included in Chapter 2.
- *Optimization over sparse restricted convex sets via two steps projection* [45], by William de Vazelhes, Xiaotong Yuan, Bin Gu. A preliminary version, had been submitted to ICLR, and is available on OpenReview, and an updated version is currently under review at COLT 2024. Included in Chapter 3.
- *Iterative Regularization with k-support Norm: An Important Complement to Sparse Recovery* [44], by William de Vazelhes, Bhaskar Mukhoty, Xiao-Tong Yuan, Bin Gu. Published in AAAI 2024. Included in Chapter 4.

1.4 Outline

In **Chapter 2**, we analyze our first algorithm, Zeroth-Order Hard-Thresholding (ZOHT), which is an adaptation of the Iterative Hard Thresholding algorithm to the zeroth-order case, i.e. where the true (stochastic) gradient is replaced by a zeroth-order approximation of it. We analyze the conflict between the error of such a zeroth-order estimator on one side, and the expansivity of the hard-thresholding operator on the other side, and analyze how to tune the parameters of the algorithm (number of random directions q, sparsity k and step-size η), in order to ensure convergence. Importantly, our results show that under standard assumptions, the query complexity of ZOHT is dimension independent, which is a very important property in the zeroth-order literature. Our results are based on novel bounds on the error of the zeroth-order estimator *restricted to a given sparse support*, which we obtain using properties of integrals on slices of spheres. Such result on the *support restricted* error of the zeroth-order gradient estimator will also be at the core of most of our results on zeroth-order hard-thresholding variants. Finally, we illustrate the applicability of ZOHT on several use cases: a portfolio optimization task, as well as a black-box adversarial attacks task. Then, in the last two sections of Chapter 2, we analyze several variance reduction variants of ZOHT, such as the ones in [68, 72, 79, 115], i.e. SVRG, SAGA, SARAH, and q-SAGA. We analyze in particular the effect of variance reduction on the previous conflict between zeroth-order and expansivity of the hard-thresholding operator, and in particular how variance reduction allows to reduce stringent requirements on the number of random directions of the zeroth-order estimator. We illustrate such algorithms on several use cases including portfolio optimization and black-box adversarial attacks. Then, we consider the original ZOHT algorithm, but in the case where the function to be optimized is discontinuous (but bounded) and non-convex, as is the case in general in reinforcement learning settings. We show that a carefully tuned zeroth-order hard-thresholding is guaranteed to converge to a stationary point of the smoothed objective in such case, and provide, up to our knowledge, the first explicit convergence rate for such problem, using constants of the problem (which we believe are of independent interest for the evolutionary strategies community in particular).

In **Chapter 3**, we consider a variant of IHT, using a two-step projection operator, which can tackle extra convex constraints such that projection of a sparse vector onto them is *support preserving.* We provide a global convergence guarantee in terms of function value for such an algorithm (similar to existing global convergence guarantees for IHT, i.e. for a relaxed sparsity level since the problem is NP-hard in general), which exhibit a novel compromise between sub-optimality gap and sparsity relaxation specific to these new constraints, and balanced in our bounds through a parameter ρ. To derive such results, we develop, using tools from [90], a non-convex variant of the usual three-point lemma classically used to prove convergence of projected gradient descent in the smooth and convex setting. We first provide such a three-point lemma in a vanilla flavor to characterize the behaviour of the hard-thresholding operator itself, which allows us as a byproduct to significantly simplify existing proofs of convergence of IHT in the deterministic and stochastic setting, and to come up with a novel zeroth-order hard-thresholding algorithm with exponentially increasing random directions, which can get rid (as we prove thanks to our new framework) of the system error of ZOHT. We also provide a variant of the three-point lemma which incorporates the extra constraints, and which we use to obtain the (first, up to our knowledge) global convergence guarantees in the setting with extra constraints.

Finally, in **Chapter 4**, we consider an algorithm for sparse recovery which is inspired from a dual perspective on IHT. More precisely, such an algorithm is an iterative regularization algorithm following the framework from [99], but where the usual regularization based on the ℓ_1 norm is replaced by a regularizer based on the k-support norm. We provide sufficient conditions for sparse recovery with such an algorithm, and show how they differ from, hence are complementary from, the ones for recovery with ℓ_1 norm. We obtain our results by finding the corresponding first order conditions for recovery, through the derivation of the subgradient of the (squared) top-k norm. We illustrate the applicability of our algorithm on toy synthetic experiments as well as real-life experiments including an fMRI decoding task.

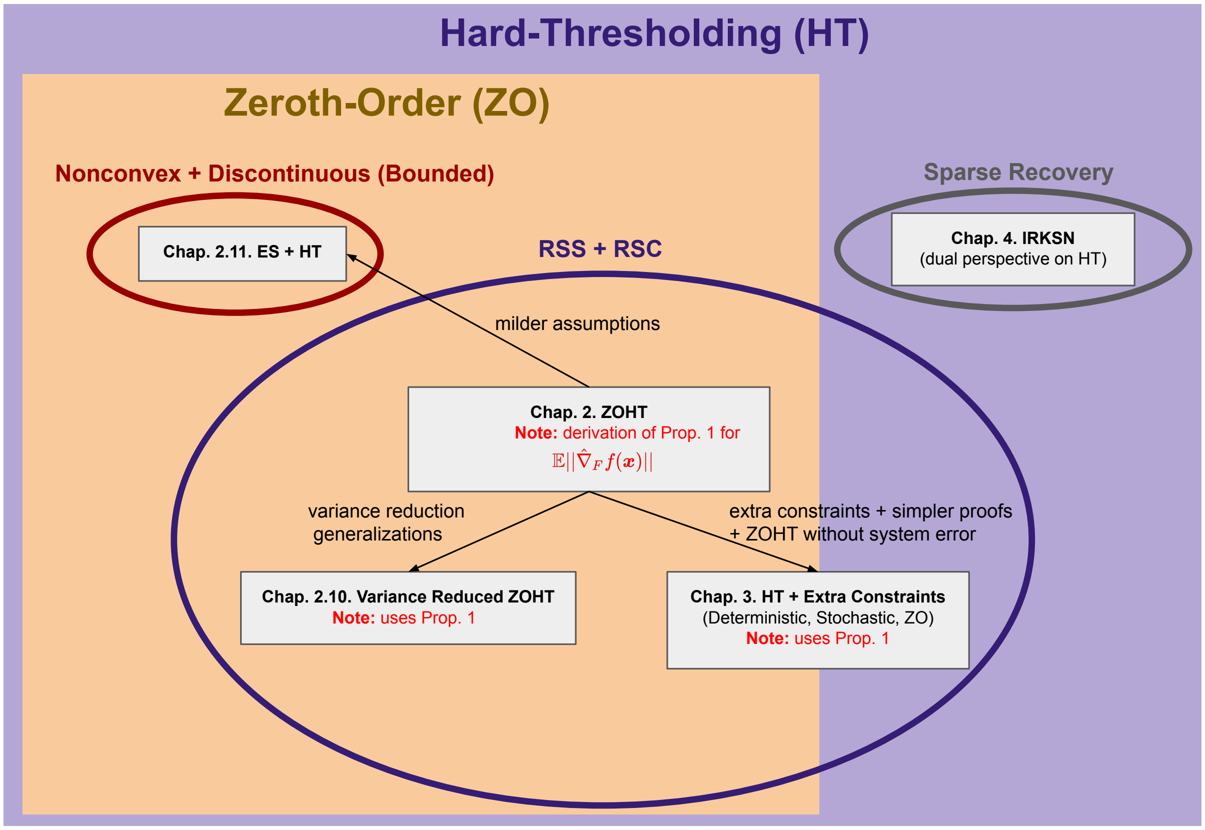

Hard-Thresholding (HT)
Zeroth-Order (ZO)
Nonconvex + Discontinuous (Bounded)
Sparse Recovery
Chap. 2.11. ES + HT
RSS + RSC
Chap. 4. IRKSN
(dual perspective on HT)
milder assumptions
Chap. 2. ZOHT
Note: derivation of Prop. 1 for
$\mathbb{E}||\hat{\nabla}_F f(\boldsymbol{x})||$
variance reduction
generalizations
extra constraints + simpler proofs
+ ZOHT without system error
Chap. 2.10. Variance Reduced ZOHT
Note: uses Prop. 1
Chap. 3. HT + Extra Constraints
(Deterministic, Stochastic, ZO)
Note: uses Prop. 1

Figure 1.3: Organization of the thesis. ES: Evolutionary Strategies, IRKSN: Iterative Regularization with k-Support Norm.

1.5 Notations

Throughout this thesis, we will denote vectors in bold letters, and we will use the following notations:

- $\nabla f(\boldsymbol{x})$ denotes the gradient of f at $\boldsymbol{x}$.
- $\boldsymbol{u}_i$ 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 (pseudo-)norm (which is not a proper norm).
- $\|\cdot\|$ (or $\|\cdot\|_2$) denote the ℓ_2 norm.
- $\|\cdot\|_p$ denotes the ℓ_p norm for $p \in [1, +\infty)$.
- $\|\cdot\|_\infty$ denotes the maximum absolute component of a vector.
- $\boldsymbol{x}_1, .., \boldsymbol{x}_n \overset{i.i.d}{\sim} \mathcal{P}$ denotes that we draw n i.i.d. samples $\boldsymbol{x}_1, .., \boldsymbol{x}_n$ from the distribution $\mathcal{P}$.
- $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}}$ (or simply $\mathbb{E}_{\boldsymbol{x}}$ if there is no possible confusion) denotes the expectation of $\boldsymbol{x}$ which follows the distribution $\mathcal{P}$.
- $[d]$ denotes the set of all integers between 1 and d: $\{1, .., d\}$.
- $\mathrm{supp}(\boldsymbol{x})$ denotes the support of a vector $\boldsymbol{x}$, that is the set of its non-zero coordinates.
- $|F|$ denotes the cardinality (number of elements) of a 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 $\mathcal{S}^d(R) = \{\boldsymbol{u} \in \mathbb{R}^d : \|\boldsymbol{u}\|_2 = R\}$.
- $\mathcal{U}(\mathcal{S}^d)$ denotes the uniform distribution on that unit sphere.
- $\beta(d)$ denotes the surface area of the unit d-sphere defined above.
- $\mathcal{S}^d_S$ denotes a set that we call the restricted d-sphere on S, described as: $\{\boldsymbol{u}_S : \boldsymbol{u} \in \{\boldsymbol{v} \in \mathbb{R}^d : \|\boldsymbol{v}_S\|_2 = 1\}\}$, that is the set of unit vectors supported by S.
- $\mathcal{U}(\mathcal{S}^d_S)$ denotes the uniform distribution on that restricted sphere above.
- $\boldsymbol{u}_F$ (resp. $\nabla_F f(\boldsymbol{x})$) denotes 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 collection 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).

- $\Pi_\Gamma(\boldsymbol{w})$ denotes the Euclidean projection of $\boldsymbol{w}$ onto a set Γ, i.e. $\Pi_\Gamma(\boldsymbol{w}) \in \arg\min_{\boldsymbol{z} \in \Gamma} \|\boldsymbol{w} - \boldsymbol{z}\|_2$.

- $\mathcal{B}_0(k)$ denotes the ℓ_0 pseudo-ball of radius k, i.e. $\mathcal{B}_0(k) = \{\boldsymbol{w} \in \mathbb{R}^d : \|\boldsymbol{w}\|_0 \leq k\}$.

- $\mathcal{H}_k$ denotes the Euclidean projection onto $\mathcal{B}_0(k)$, also known as the hard-thresholding operator (which keeps the k largest (in magnitude) components of a vector, and sets the others to 0 (if there are ties, we can break them e.g. lexicographically)).

- $\bar{\Pi}_\Gamma^k$ denotes the Two-step projection of sparsity k onto the set Γ, i.e. $\bar{\Pi}_\Gamma^k(\cdot) = \Pi_\Gamma(\mathcal{H}_k(\cdot))$.

- $|S|$ denotes the number of elements of a set $S \subseteq [d]$ (cardinality).

Chapter 2

Zeroth-Order Hard-Thresholding

This chapter is based on the paper [46].

2.1 Introduction

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 [28, 109, 125, 158]. We formulate the problem as follows:

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

where $f(\cdot, \boldsymbol{\xi}) : \mathbb{R}^d \rightarrow \mathbb{R}$ 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 [76, 116, 157] 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 ([140, 143]) 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 [76], but has been later on extended to the stochastic setting by [116], which developed a stochastic gradient descent (SGD) version of hard thresholding (StoIHT), and further more with [163], [133] and [88], 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

Table 2.1: Complexity of sparsity-enforcing algorithms. We give the query complexity for a precision ε, up to the system error (see section 2.4). 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 [116]	RSS, RSC	$\mathcal{O}(\kappa \log(\frac{1}{\varepsilon}))$	$\mathcal{O}(\kappa \log(\frac{1}{\varepsilon}))$
ZO/ℓ_1	RSPGF [59]	smooth[3]	$\mathcal{O}(\frac{d}{\varepsilon^2})$	—
ZO/ℓ_1	ZSCG[2][7]	convex, smooth	$\mathcal{O}(\frac{d}{\varepsilon^2})$	—
ZO/ℓ_1	ZORO [31]	s-sparse gradient, weakly sparse hessian, smooth[3], RSC$_{\text{bis}}$[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 [31] is different from ours and that of [116], hence the bis subscript.

[2] We refer to the modified version of ZSCG (Algorithm 3 in [7]).

[3] RSPGF and ZORO minimize $f(x) + \lambda \|x\|_1$: only f needs to be smooth.

modeling tasks [146], 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 [132], black-box adversarial attacks [38, 39, 142], or reinforcement learning [41, 98, 129]. To tackle those problems, ZO optimization methods have been developed [114]. Those 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 deal with a convex constraints set, and can therefore be used to solve the ℓ_1 convex relaxation of problem equation 2.1. To that end, [59], and [31] introduce proximal ZO algorithms, [95] introduce a ZO projected gradient algorithm and [7] introduce a ZO conditional gradient [86] 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 [31] 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 [7], 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 [31] 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.

To fill this gap, in this paper, we focus on the ℓ_0 constrained black-box stochastic optimization problems, and 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) [116, 133], 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 2.3.1) 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 [7, 31, 59].

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 [77] 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 [7, 30, 31, 31, 61, 77, 91, 135, 149], 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 the main contributions of our paper 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.

2.2 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 equation 2.1 defined in the introduction, for some target sparsity k^* which could be smaller than k. To derive our result, we will need the following assumptions on f.

Assumption 1 ((ν_s, s)-RSC, [76, 88, 96, 108, 116, 133, 157])**.** *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 $\|\boldsymbol{x} - \boldsymbol{y}\|_0 \leq s$:*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\nu_s}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 \tag{2.2}$$

Assumption 2 ((L_s, s)-RSS, [116, 133])**.** *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 $\|\boldsymbol{x} - \boldsymbol{y}\|_0 \leq s$:*

$$\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}) - \nabla f_{\boldsymbol{\xi}}(\boldsymbol{y})\| \leq L_s \|\boldsymbol{x} - \boldsymbol{y}\| \tag{2.3}$$

Assumption 3 (σ^2-FGN [64], 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:*

$$\sigma^2 = \mathbb{E}_{\boldsymbol{\xi}}[\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|_\infty^2] \tag{2.4}$$

Remark 1. *Even though the original version of [64] 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 [163]. We also note that in [64], $\boldsymbol{x}^*$ denotes an unconstrained minimum when in our case it denotes the constrained minimum for some sparsity k^*.*

For Corollary 2, 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 $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^d$:*

$$\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}) - \nabla f_{\boldsymbol{\xi}}(\boldsymbol{y})\| \leq L \|\boldsymbol{x} - \boldsymbol{y}\| \tag{2.5}$$

2.3 Algorithm

2.3.1 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 [94]. More formally, our gradient estimator is described below:

$$\hat{\nabla} f_{\boldsymbol{\xi}}(\boldsymbol{x}) = \frac{d}{q\mu} \sum_{i=1}^{q} \left(f_{\boldsymbol{\xi}}(\boldsymbol{x} + \mu \boldsymbol{u}_i) - f_{\boldsymbol{\xi}}(\boldsymbol{x}) \right) \boldsymbol{u}_i \tag{2.6}$$

where each random direction $\boldsymbol{u}_i$ is a unit vector sampled uniformly from the set $\mathcal{S}^d_{s_2} := \{\boldsymbol{u} \in \mathbb{R}^d : \|\boldsymbol{u}\|_0 \leq s_2, \|\boldsymbol{u}\| = 1\}$. 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 2) and then by sampling a random unit vector $\boldsymbol{u}$ supported on that support S, that is, uniformly sampled from the set $\mathcal{S}^d_S := \{\boldsymbol{u} \in \mathbb{R}^d : \boldsymbol{u}_{[d]-S} = \boldsymbol{0}, \|\boldsymbol{u}\| = 1\}$, (denoted as $\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d_S)$ in algorithm 2). The original uniform smoothing technique on the sphere is described in more detail in [57]. 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* [57].

- If $1 \leq s_2 \leq d$, our estimator is similar to the Random Block-Coordinate gradient estimator from [89], 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 [30,31]: 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 [157]), 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 2.2 in Section 2.6. 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 Section 2.5.3) Let us consider any support $F \subset [d]$ of size s ($|F| = s$). For the Z0 gradient estimator in equation 2.6, 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):

(a) $\|\mathbb{E}\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x})\|^2 \leq \varepsilon_\mu \mu^2$

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

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

$$\begin{gathered} \textit{with} \quad \varepsilon_\mu = L_{s_2}^2 sd, \quad \varepsilon_F = \frac{2d}{q(s_2+2)}\left(\frac{(s-1)(s_2-1)}{d-1} + 3\right) + 2, \\ \varepsilon_{F^c} = \frac{2d}{q(s_2+2)}\left(\frac{s(s_2-1)}{d-1}\right) \ \textit{and} \ \varepsilon_{abs} = \frac{2dL_{s_2}^2 ss_2}{q}\left(\frac{(s-1)(s_2-1)}{d-1} + 1\right) + L_{s_2}^2 sd \end{gathered} \tag{2.7}$$

2.3.2 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 2.

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 [133], 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 [133]. 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 [116]) , 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 [7] (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 [30]. Furthermore, as described in the Introduction, contrary to [7], we provide the analysis of our algorithm without any gradient sparsity assumption.

The key challenge arising in our analysis is described in Figure 2.1: the hard-thresholding operator being expansive [133], 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.

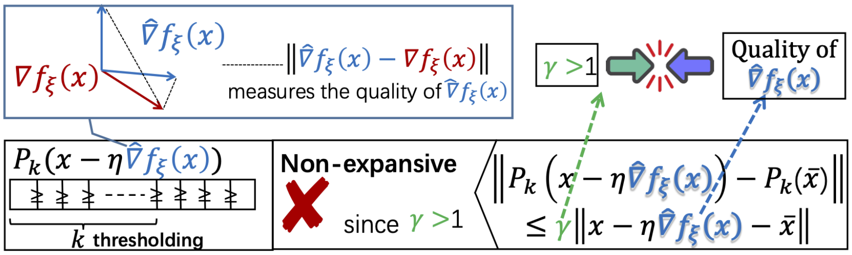

$\hat{\nabla} f_\xi(x)$
$\nabla f_\xi(x)$
$\|\hat{\nabla} f_\xi(x) - \nabla f_\xi(x)\|$
measures the quality of $\hat{\nabla} f_\xi(x)$
$\gamma > 1$
Quality of $\hat{\nabla} f_\xi(x)$
$P_k(x - \eta \hat{\nabla} f_\xi(x))$
k thresholding
Non-expansive
since $\gamma > 1$
$\left\|P_k\left(x - \eta \hat{\nabla} f_\xi(x)\right) - P_k(\bar{x})\right\|$
$\le \gamma \|x - \eta \hat{\nabla} f_\xi(x) - \bar{x}\|$

Figure 2.1: Conflict between the hard-thresholding operator and the zeroth-order estimate.

Algorithm 2: Stochastic Zeroth-Order Hard-Thresholding (SZOHT)

```
Initialization: Learning rate $\eta$, 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 $S$:
          $\boldsymbol{u}_i \sim \mathcal{U}\left(\mathcal{S}^d_S\right)$
        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 $\boldsymbol{x}^t = \tilde{\boldsymbol{x}}^t_k$ as the truncation of $\tilde{\boldsymbol{x}}^t$ with top $k$ entries preserved;
end
```

2.4 Convergence analysis

In this section, we provide the convergence analysis of SZOHT, using the assumptions from section 2.2, 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 Section 2.5.4) 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 $\frac{d-k^*}{2} \geq k \geq \rho^2 k^*/(1-\rho^2)^2$, with ρ defined as below. Suppose that we run SZOHT with random supports of size s_2,*

q random directions, a learning rate of $\eta = \frac{\nu_s}{(4\varepsilon_F+1)L^2_{s'}}$, 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:

$$\mathbb{E}\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\| \leq (\gamma\rho)^t \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\| + \left(\frac{\gamma a}{1-\gamma\rho}\right)\sigma + \left(\frac{\gamma b}{1-\gamma\rho}\right)\mu \tag{2.8}$$

$$\text{with} \quad a = \eta\left(\sqrt{(4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k)} + \sqrt{s}\right),\ b = \left(\frac{\sqrt{\varepsilon_\mu}}{L_{s'}} + \eta\sqrt{2\varepsilon_{abs}}\right),$$
$$\rho^2 = 1 - \frac{\nu_s^2}{(4\varepsilon_F+1)L^2_{s'}},\ \text{and } \gamma = \sqrt{1 + \left(k^*/k + \sqrt{(4+k^*/k)\,k^*/k}\right)/2} \tag{2.9}$$

and $\varepsilon_F, \varepsilon_{abs}$, and ε_μ are defined in equation 2.7.

Remark 2 (System error)**.** *The format of our result is similar to the ones in [157] and [116], 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 [124, 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 [116, 133]), 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 [133], 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 [116, 163], it is generally dealt with either by ensuring some conditions on the function f [116], forming bigger batches of samples [163], 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 [163], 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 [133]), and the error from the zeroth-order estimate, in Figure 2.1. Therefore, it is even more crucial to tune precisely our remaining degree of freedom at hand which is q. More precisely, a minimal value of q is always necessary to ensure convergence in our setting, contrary to most ZO setting (in which taking even $q = 1$ can work, as long as other constants like η are well chosen, see for instance [92, Corollary 3]). The remark below gives some necessary conditions on q to illustrate that fact.

Remark 4 (Some necessary condition on q, proof in 2.5.5). *Let $k^* \in \mathbb{N}^*$ and assume, that k is such that $k > \rho^2 k^*/(1-\rho^2)^2$ (which ensures that $\rho\gamma < 1$), and that $k \leq \frac{d-k^*}{2}$. These conditions imply the following necessary (but not sufficient) condition on q:*

- ***if*** *$s_2 > 1$: $q \geq \frac{16d(s_2-1)k^*\kappa^2}{(s_2+2)(d-1)}\left[18\kappa^2 - 1 + 2\sqrt{9\kappa^2(9\kappa^2-1) + \frac{1}{2} - \frac{1}{2k^*} + \frac{3}{2}\frac{d-1}{k^*(s_2-1)}}\right]$*
- ***if*** *$s_2 = 1$: $q \geq \frac{8\kappa^2 d}{\sqrt{\frac{d}{k^*}}+1}$*

Remark 4 is just a warning that usual rules from ZO do not apply to SZOHT, but it does not say how to choose q to ensure convergence: for that we would need some sufficient conditions on q for Theorem 1 to apply. We give such conditions in the next section.

2.4.1 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 sufficient 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 Section 2.5.6). *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:*

$$\mathbb{E}\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\| \leq (\gamma\rho)^t \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\| + \left(\frac{\gamma a}{1-\gamma\rho}\right)\sigma + \left(\frac{\gamma b}{1-\gamma\rho}\right)\mu \tag{2.10}$$

with a, b and γ are defined in equation 2.9, 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 Section 2.5.7)**.** *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:*

$$\mathbb{E}\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\| \leq (\gamma\rho)^t \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\| + \left(\frac{\gamma a}{1-\gamma\rho}\right)\sigma + \left(\frac{\gamma b}{1-\gamma\rho}\right)\mu \tag{2.11}$$

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 [157]). 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 [59].

2.5 Proofs of the Main Results

2.5.1 Auxilliary Lemmas

Lemma 2.5.1 ([137] (10))**.** *Let $\boldsymbol{p} \in \mathbb{N}^d$, and denote $p := \sum_{i=1}^d \boldsymbol{p}_i$, we have:*

$$\int_{\mathcal{S}^d} \prod_{i=1}^{d} \left(\boldsymbol{u}_i^2\right)^{\boldsymbol{p}_i} d\boldsymbol{u} = 2\frac{\prod_{i=1}^n \Gamma(\boldsymbol{p}_i + 1/2)}{\Gamma(p + d/2)} \tag{2.12}$$

Proof. The proof is given in [137]. □

Lemma 2.5.2. *Let F be a subset of $[d]$, of size s, with $(s, d) \in \mathbb{N}_*^2$. We have the following:*

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\|\boldsymbol{u}_F\| \leq \sqrt{\frac{s}{d}} \tag{2.13}$$

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\|\boldsymbol{u}_F\|^2 = \frac{s}{d} \tag{2.14}$$

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\|\boldsymbol{u}_F\|^4 = \frac{(s+2)s}{(d+2)d} \tag{2.15}$$

Proof. We start by proving equation 2.14. Decomposing the norm onto every component, we get:

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\|\boldsymbol{u}_F\|^2 = \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\sum_{i\in F}\boldsymbol{u}_i^2 = \sum_{i\in F}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\boldsymbol{u}_i^2 \tag{2.16}$$

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

$$\forall i \in [d]: \qquad \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\boldsymbol{u}_i^2 = \frac{1}{d}\sum_{i=1}^{d}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\boldsymbol{u}_i^2 \tag{2.17}$$

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

$$\sum_{i=1}^{d}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\boldsymbol{u}_i^2 = \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\sum_{i=1}^{d}\boldsymbol{u}_i^2 = \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\|\boldsymbol{u}\|^2 = \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}1 = 1 \tag{2.18}$$

Therefore, combining equation 2.17 and equation 2.18:

$$\forall i \in [d]: \qquad \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\boldsymbol{u}_i^2 = \frac{1}{d} \tag{2.19}$$

Plugging this into equation 2.16, we get equation 2.14:

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\|\boldsymbol{u}_F\|^2 = \frac{s}{d} \tag{2.20}$$

Using Jensen's inequality, equation 2.13 follows from equation 2.14. Let us now prove equation 2.15. By definition of the expectation for a uniform distribution on the unit sphere:

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\|\boldsymbol{u}_F\|^4 = \frac{1}{\beta(d)}\int_{\mathcal{S}^d}\|\boldsymbol{u}_F\|^4 d\boldsymbol{u} \tag{2.21}$$

We further develop the integral as follows:

$$\begin{aligned}\int_{\mathcal{S}^d}\|\boldsymbol{u}_F\|^4 d\boldsymbol{u} &= \int_{\mathcal{S}^d}(\|\boldsymbol{u}_F\|^2)^2 d\boldsymbol{u} = \int_{\mathcal{S}^d}(\sum_{i\in F}\boldsymbol{u}_i^4 + \sum_{(i,j)\in F, j\neq i}\boldsymbol{u}_i^2\boldsymbol{u}_j^2)d\boldsymbol{u} \\ &= s\int_{\mathcal{S}^d}\boldsymbol{u}_1^4 d\boldsymbol{u} + 2\binom{s}{2}\int_{\mathcal{S}^d}\boldsymbol{u}_1^2\boldsymbol{u}_2^2 d\boldsymbol{u} \quad \text{(by symmetry)} \end{aligned} \tag{2.22}$$

Using Lemma 2.5.1 in the expression above, with $\boldsymbol{p}^{(a)} := (2, 0, ..., 0)$, and $\boldsymbol{p}^{(b)} := (1, 1, 0, ..., 0)$, we obtain:

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

$$\overset{(a)}{=} \frac{6s\sqrt{\pi}^d}{(d+2)d\Gamma(d/2)} + \frac{2s(s-1)\sqrt{\pi}^d}{(d+2)d\Gamma(d/2)} = \frac{2(s+2)s\sqrt{\pi}^d}{(d+2)d\Gamma(d/2)} \tag{2.23}$$

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

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\|\boldsymbol{u}_F\|^4 = \frac{1}{\beta(d)}\int_{S^d}\|\boldsymbol{u}_F\|^4 d\boldsymbol{u} \overset{(b)}{=} \frac{s+2}{d+2}\frac{s}{d} \tag{2.24}$$

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 2.5.3 ([57], Lemma 7.3.b)**.**

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\boldsymbol{u}\boldsymbol{u}^T = \frac{1}{d}\boldsymbol{I} \tag{2.25}$$

Proof. The proof is given in [57]. □

Lemma 2.5.4 ([133], Theorem 1; [157], 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 [133]. □

Corollary 2.5.1. *With the notations and variables above in Lemma 2.5.4, we also have the following, simpler bound, from [157]:*

$$\|\boldsymbol{b}_k - \boldsymbol{a}\| \leq \gamma\|\boldsymbol{b} - \boldsymbol{a}\| \tag{2.26}$$

with

$$\gamma = \sqrt{1 + \left(\bar{k}/k + \sqrt{\left(4 + \bar{k}/k\right)\bar{k}/k}\right)/2} \tag{2.27}$$

Proof. There are two possibilities for β in Lemma 2.5.4: 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:

$$d-k \leq \bar{k} \implies d-\bar{k} \leq k \implies \frac{k-\bar{k}}{d-\bar{k}} \geq \frac{k-\bar{k}}{k} \implies 1-\frac{k-\bar{k}}{d-\bar{k}} \leq 1-\frac{k-\bar{k}}{k} \implies \frac{d-k}{d-\bar{k}} \leq \frac{\bar{k}}{k} \tag{2.28}$$

Therefore, in both cases, $\beta \leq \frac{\bar{k}}{k}$, which, plugging into Lemma 2.5.4, gives Corollary 2.5.1. □

2.5.2 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 2.5.2.1 the error of the gradient estimate if we sample only one direction ($q = 1$). Then, in section 2.5.2.2, we show how sampling q directions reduces the error of the gradient estimator, producing the results of Proposition 1.

2.5.2.1 One Direction Estimator

Throughout all this section, we assume that $q = 1$ for the gradient estimator $\hat{\nabla} f(x)$ defined in equation 2.6.

Expected Deviation From The Mean.

Lemma 2.5.5. *For any (L_{s_2}, s_2)-RSS function f, using the gradient estimator $\hat{\nabla} f(x)$ defined in equation 2.6 with $q = 1$, we have, for any support $F \in [d]$, with $|F| = s$:*

$$\left\| \mathbb{E}\left[\hat{\nabla}_F f(\boldsymbol{x})\right] - \nabla_F f(\boldsymbol{x}) \right\|^2 \leq \varepsilon_\mu \mu^2 \tag{2.29}$$

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

Proof. From the definition of the gradient estimator in equation 2.6:

$$\|\mathbb{E}[\hat{\nabla}_F f(\boldsymbol{x})] - \nabla_F f(\boldsymbol{x})\| = \left\| \mathbb{E} d \frac{f(\boldsymbol{x} + \mu \boldsymbol{u}) - f(\boldsymbol{x})}{\mu} \boldsymbol{u}_F - \nabla_F f(\boldsymbol{x}) \right\| \tag{2.30}$$

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:

$$\mathbb{E}\boldsymbol{u}\boldsymbol{u}^T = \mathbb{E}_{S \sim \mathrm{ss}_2 d} \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}_S^d)} \boldsymbol{u}\boldsymbol{u}^T \overset{(a)}{=} \mathbb{E}_{S \sim \mathrm{ss}_2 d} \frac{1}{s_2} \boldsymbol{I}_S = \frac{1}{s_2} \mathbb{E}_{S \sim \mathrm{ss}_2 d} \boldsymbol{I}_S \overset{(b)}{=} \frac{1}{s_2} \frac{s_2}{d} \boldsymbol{I} = \frac{1}{d} \boldsymbol{I} \tag{2.31}$$

Where for (a) comes from applying Lemma 2.5.3 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:

$$\|\mathbb{E}[\hat{\nabla}_F f(\boldsymbol{x})] - \nabla_F f(\boldsymbol{x})\| = d \|\mathbb{E}\langle \nabla f(\boldsymbol{x} + c\boldsymbol{u}) - \nabla f(\boldsymbol{x}), \boldsymbol{u} \rangle \boldsymbol{u}_F\|$$

$$\leq d\mathbb{E}\|\boldsymbol{u}_F\boldsymbol{u}^T[\nabla f(\boldsymbol{x}+c\boldsymbol{u})-\nabla f(\boldsymbol{x})]\| \tag{2.32}$$

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:

$$\|\mathbb{E}[\hat{\nabla}_F f(\boldsymbol{x})]-\nabla_F f(\boldsymbol{x})\|\leq\mathbb{E}_{S\sim\mathrm{ss}_2d}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d_S)}\|\boldsymbol{u}_F\|\|\boldsymbol{u}\|\|\nabla_S f(\boldsymbol{x}+c\boldsymbol{u})-\nabla_S f(\boldsymbol{x})\| \tag{2.33}$$

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:

$$\begin{aligned}\|\mathbb{E}[\hat{\nabla}_F f(\boldsymbol{x})]-\nabla_F f(\boldsymbol{x})\| &\leq \mathbb{E}_S\mathbb{E}_{\boldsymbol{u}}dL_{s_2}\mu\|\boldsymbol{u}_F\|\|\boldsymbol{u}\|\|\boldsymbol{u}\|=\mathbb{E}_S\mathbb{E}_{\boldsymbol{u}}dL_{s_2}\mu\|\boldsymbol{u}_F\|\|\boldsymbol{u}\|^2\\ &=\mathbb{E}_S\mathbb{E}_{\boldsymbol{u}}dL_{s_2}\mu\|\boldsymbol{u}_F\|\overset{(a)}{\leq}dL_{s_2}\mu\mathbb{E}_S\mathbb{E}_{\boldsymbol{u}}\sqrt{\frac{|S\cap F|}{s_2}}\\ &\overset{(b)}{\leq}dL_{s_2}\mu\sqrt{\mathbb{E}_S\frac{|S\cap F|}{s_2}}=dL_{s_2}\mu\sqrt{\mathbb{E}_k\mathbb{E}_{S||S\cap F|=k}\frac{k}{s_2}}\\ &=dL_{s_2}\mu\sqrt{\frac{ss_2}{ds_2}}=L_{s_2}\mu\sqrt{sd}\end{aligned} \tag{2.34}$$

Where (a) follows from Lemma 2.5.2, restricted to the support S, and (b) □

Expected Norm.

Lemma 2.5.6. *For any (L_{s_2},s_2)-RSS function f, using the gradient estimator $\hat{\nabla}f(x)$ defined in equation 2.6 with $q=1$, we have, for any support $F\in[d]$, with $|F|=s$:*

$$\mathbb{E}\|\hat{\nabla}_F f(\boldsymbol{x})\|^2=\varepsilon_F\|\nabla_F f(\boldsymbol{x})\|^2+\varepsilon_{F^c}\|\nabla_{F^c}f(\boldsymbol{x})\|^2+\varepsilon_{abs}\mu^2 \tag{2.35}$$

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) $\varepsilon_{abs}=2dL_s^2ss_2\left(\frac{(s-1)(s_2-1)}{d-1}+1\right)$

Proof.

$$\begin{aligned}\mathbb{E}\|\hat{\nabla}_F f(\boldsymbol{x})\|^2&=\mathbb{E}\left\|d\frac{f(\boldsymbol{x}+\mu\boldsymbol{u})-f(\boldsymbol{x})}{\mu}\boldsymbol{u}_F\right\|^2\\ &=\mathbb{E}\frac{d^2}{\mu^2}|f(\boldsymbol{x}+\mu\boldsymbol{u})-f(\boldsymbol{x})|^2\|\boldsymbol{u}_F\|^2\\ &=\frac{d^2}{\mu^2}\mathbb{E}[f(\boldsymbol{x}+\mu\boldsymbol{u})-f(\boldsymbol{x})-\langle\nabla f(\boldsymbol{x}),\mu\boldsymbol{u}\rangle+\langle\nabla f(\boldsymbol{x}),\mu\boldsymbol{u})]^2\|\boldsymbol{u}_F\|^2\end{aligned} \tag{2.36}$$

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

$$f(\boldsymbol{x} + \mu\boldsymbol{u}) - f(\boldsymbol{x}) = \langle \nabla f(\boldsymbol{x} + c), \mu\boldsymbol{u} \rangle \tag{2.37}$$

Therefore, plugging this in the above:

$$\begin{aligned}
\mathbb{E}\|\hat{\nabla}_F f(\boldsymbol{x})\|^2 &\leq d^2\mathbb{E}[\langle \nabla f(\boldsymbol{x} + c\boldsymbol{u}) - \nabla f(\boldsymbol{x}), \boldsymbol{u}\rangle + \langle \nabla f(\boldsymbol{x}), \boldsymbol{u}\rangle]^2 \|\boldsymbol{u}_F\|^2 \\
&\overset{(a)}{\leq} d^2\mathbb{E}\left[2\langle \nabla f(\boldsymbol{x} + c\boldsymbol{u}) - \nabla f(\boldsymbol{x}), \boldsymbol{u}\rangle^2 \|\boldsymbol{u}_F\|^2 + \langle \nabla f(\boldsymbol{x}), \boldsymbol{u}\rangle^2 \|\boldsymbol{u}_F\|^2\right] \\
&\leq 2d^2\mathbb{E}[\|\nabla f(\boldsymbol{x} + c\boldsymbol{u}) - \nabla f(\boldsymbol{x})\|^2 \|\boldsymbol{u}\|^2 \|\boldsymbol{u}_F\|^2 + \langle \nabla f(\boldsymbol{x}), \boldsymbol{u}\rangle^2 \|\boldsymbol{u}_F\|^2] \\
&\overset{\leq}{(b)} 2d^2\mathbb{E}[L_s^2\mu^2\|\boldsymbol{u}\|^2\|\boldsymbol{u}\|^2\|\boldsymbol{u}_F\|^2 + \langle \nabla f(\boldsymbol{x}), \boldsymbol{u}\rangle^2 \|\boldsymbol{u}_F\|^2] \\
&\overset{(c)}{=} 2d^2\mathbb{E}[L_s^2\mu^2\|\boldsymbol{u}_F\|^2 + \langle \nabla f(\boldsymbol{x}), \boldsymbol{u}\rangle^2 \|\boldsymbol{u}_F\|^2] \\
&= 2d^2[L_s^2\mu^2\mathbb{E}\|\boldsymbol{u}_F\|^2 + \nabla f(\boldsymbol{x})^T \left(\mathbb{E}\boldsymbol{u}\boldsymbol{u}^T\|\boldsymbol{u}_F\|^2\right) \nabla f(\boldsymbol{x})] \\
&= 2d^2[L_{s_2}^2\mu^2\mathbb{E}\|\boldsymbol{u}_F\|^2 + \nabla f(\boldsymbol{x})^T (\mathbb{E}_{S\sim \mathrm{ss}_2 d}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\boldsymbol{u}\boldsymbol{u}^T\|\boldsymbol{u}_F\|^2)\nabla f(\boldsymbol{x})] \\
&\overset{(d)}{=} 2d^2[L_{s_2}^2\mu^2\mathbb{E}\|\boldsymbol{u}_F\|^2 + \mathbb{E}_{S\sim \mathrm{ss}_2 d}[\nabla f(\boldsymbol{x})^T (\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\boldsymbol{u}\boldsymbol{u}^T\|\boldsymbol{u}_F\|^2)\nabla f(\boldsymbol{x})]] \quad (2.38)
\end{aligned}$$

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:

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\boldsymbol{u}_{i_1}^2\|\boldsymbol{u}_F\|^2 = ... = \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\boldsymbol{u}_{i_s}^2\|\boldsymbol{u}_F\|^2 \tag{2.39}$$

Therefore, for all $i \in F$:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\boldsymbol{u}_i^2\|\boldsymbol{u}_F\|^2 &= \frac{1}{|S \cap F|}\sum_{j=1}^{s}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\boldsymbol{u}_{i_j}^2\|\boldsymbol{u}_F\|^2 \\
&= \frac{1}{|S \cap F|}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\sum_{j=1}^{s}\boldsymbol{u}_{i_j}^2\|\boldsymbol{u}_F\|^2 = \frac{1}{|S \cap F|}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\|\boldsymbol{u}_F\|^4 \qquad (2.40)
\end{aligned}$$

By definition of the restricted d-sphere on F (see section 4.6.1), 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 equation 2.40, but only their cardinality, equation 2.40 can be rewritten using a simpler expectation over a unit $|S|$-sphere as follows :

$$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}_S^d)}\|\boldsymbol{u}_F\|^4 = \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^{|S|})}\|\boldsymbol{u}_{[|S\cap F|]}\|^4 \tag{2.41}$$

Using Lemma 2.5.2 to get a closed form expression of the expected value above, we further obtain:

$$\forall i \in F : \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(\mathcal{S}^d)}\boldsymbol{u}_i^2\|\boldsymbol{u}_F\|^2 = \frac{1}{|S \cap F|}\frac{|S \cap F|(|S \cap F| + 2)}{d(d + 2)} = \frac{|S \cap F| + 2}{d(d + 2)} \tag{2.42}$$

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

$$\mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d_S)} \boldsymbol{u}_{i_j}^2 \|\boldsymbol{u}_F\|^2 = ... = \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d_S)} \boldsymbol{u}_{i_j}^2 \|\boldsymbol{u}_F\|^2 \tag{2.43}$$

Therefore, for all $i \notin F$:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d_S)} \boldsymbol{u}_i^2 \|\boldsymbol{u}_F\|^2 &= \frac{1}{d-s} \sum_{j=1}^{d-s} \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d_S)} \boldsymbol{u}_{i_j}^2 \|\boldsymbol{u}_F\|^2 = \frac{1}{d-s} \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d_S)} \sum_{j=1}^{d-s} \boldsymbol{u}_{i_j}^2 \|\boldsymbol{u}_F\|^2 \\
&\overset{(a)}{=} \frac{1}{d-s} \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d_S)} (\|\boldsymbol{u}\|^2 - \|\boldsymbol{u}_F\|^2) \|\boldsymbol{u}_F\|^2 \\
&\overset{(b)}{=} \frac{1}{d-s} (\mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d_S)} \|\boldsymbol{u}_F\|^2 - \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d_S)} \|\boldsymbol{u}\|^4)
\end{aligned} \tag{2.44}$$

Where (a) follows from the Pythagorean theorem and (b) follows from $\|\boldsymbol{u}\| = 1$. Similarly as before, rewriting those expected values and using Lemma 2.5.2, we obtain:

$$\forall i \notin F : \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d)} \boldsymbol{u}_i^2 \|\boldsymbol{u}_F\|^2 = \frac{1}{d - |S \cap F|} \frac{|S \cap F|(d + 2 - (|S \cap F| + 2))}{d(d+2)} = \frac{|S \cap F|}{d(d+2)} \tag{2.45}$$

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

$$\mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d_S)} \boldsymbol{u}_i \boldsymbol{u}_j \|\boldsymbol{u}_F\|^2 = \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d_S)} (-\boldsymbol{u}_i) \boldsymbol{u}_j \|\boldsymbol{u}_F\|^2 = -\mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d_S)} \boldsymbol{u}_i \boldsymbol{u}_j \|\boldsymbol{u}_F\|^2 \tag{2.46}$$

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

$$\mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d_S)} \boldsymbol{u}_i \boldsymbol{u}_j \|\boldsymbol{u}_F\|^2 = 0 \tag{2.47}$$

Therefore, combining equation 2.42, equation 2.45 and equation 2.47, we obtain:

$$\mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d_S)} \boldsymbol{u} \boldsymbol{u}^T \|\boldsymbol{u}_F\|^2 = \begin{bmatrix} a_1 & & & \\ & a_2 & & \\ & & \ddots & \\ & & & a_d \end{bmatrix} \tag{2.48}$$

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 equation 2.38, we obtain:

$$\begin{aligned}
A &:= \mathbb{E}_{S \sim \mathrm{ss}_2 d} [\nabla f(\boldsymbol{x})^T \left(\mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(\mathcal{S}^d_S)} \boldsymbol{u} \boldsymbol{u}^T \|\boldsymbol{u}_F\|^2 \right) \nabla f(\boldsymbol{x})] \\
&= \mathbb{E}_{S \sim \mathrm{ss}_2 d} \left[\frac{|S \cap F| + 2}{s_2(s_2 + 2)} \|\nabla_{S \cap F} f(\boldsymbol{x})\|^2 + \frac{|S \cap F|}{s_2(s_2 + 2)} \|\nabla_{S \setminus (S \cap F)} f(\boldsymbol{x})\|^2 \right] \\
&= \frac{1}{s_2(s_2 + 2)} \left[\mathbb{E}_{S \sim \mathrm{ss}_2 d} \left[|S \cap F| \; \|\nabla_{F \cap S} f(\boldsymbol{x})\|^2 \right] \right. \\
&\quad \left. + 2 \mathbb{E}_{S \sim \mathrm{ss}_2 d} \left[\|\nabla_{F \cap S} f(\boldsymbol{x})\|^2 + |S \cap F| \; \|\nabla_{S \setminus (S \cap F)} f(\boldsymbol{x})\|^2 \right] \right]
\end{aligned} \tag{2.49}$$

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 $\mathbb{E}_{S||S\cap F|=k}[\cdot] = \mathbb{E}_{S\sim ss_2d||S\cap F|=k}[\cdot]$. Therefore, rewriting equation 2.49 using the law of total expectation, we obtain:

$$\begin{aligned} A = & \frac{1}{s_2(s_2+2)} \left[\mathbb{E}_k \mathbb{E}_{S||S\cap F|=k} k \|\nabla_{S\cap F} f(\boldsymbol{x})\|^2 + 2\mathbb{E}_k \mathbb{E}_{S||S\cap F|=k} \|\nabla_{S\cap F} f(\boldsymbol{x})\|^2 \right. \\ & \left. + \mathbb{E}_k \mathbb{E}_{S||S\cap F|=k} k \|\nabla_{S\setminus(S\cap F)} f(\boldsymbol{x})\|^2 \right] \\ = & \frac{1}{s_2(s_2+2)} \left[\mathbb{E}_k k \mathbb{E}_{S||S\cap F|=k} \|\nabla_{S\cap F} f(\boldsymbol{x})\|^2 + 2\mathbb{E}_S \mathbb{E}_{S||S\cap F|=k} \|\nabla_{S\cap F} f(\boldsymbol{x})\|^2 \right. \\ & \left. + \mathbb{E}_k k \mathbb{E}_{S||S\cap F|=k} \|\nabla_{S\setminus(S\cap F)} f(\boldsymbol{x})\|^2 \right] \end{aligned} \tag{2.50}$$

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:

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

$$= \frac{s}{k}\|\nabla_F f(\boldsymbol{x})\|^2 \tag{2.51}$$

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 equation 2.51 we have, for the second expectation:

$$\mathbb{E}_{S||S\cap F|=k}\|\nabla_{S\setminus(S\cap F)} f(\boldsymbol{x})\|^2 = \frac{s_2 - k}{d-s}\|\nabla_{F^c} f(\boldsymbol{x})\|^2 \tag{2.52}$$

Therefore, plugging equation 2.51 and equation 2.52 into equation 2.50

$$\begin{aligned} A &= \frac{1}{s_2(s_2+2)}\left[\mathbb{E}_k k\frac{k}{s}\|\nabla_F f(\boldsymbol{x})\|^2 + 2\mathbb{E}_k\frac{k}{s}\|\nabla_F f(\boldsymbol{x})\|^2 + \mathbb{E}_k k\frac{s_2-k}{d-s}\|\nabla_{F^c} f(\boldsymbol{x})\|^2\right] \\ &= \frac{1}{s_2(s_2+2)}\left[\frac{1}{s}\|\nabla_F f(\boldsymbol{x})\|^2\left[\mathbb{E}_k k^2 + 2\mathbb{E}_k k\right] + \|\nabla_{F^c} f(\boldsymbol{x})\|^2\left[\frac{s_2}{d-s}\left(\mathbb{E}_k k\right) - \frac{1}{d-s}\mathbb{E}_k k^2\right]\right] \end{aligned} \tag{2.53}$$

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 [147], 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:

$$\begin{aligned} \mathbb{E}_k k^2 &= Var(k) + (\mathbb{E}_k[k])^2 \overset{(a)}{=} \frac{ss_2}{d}\frac{d-s}{d}\frac{d-s_2}{d-1} + \left(\frac{ss_2}{d}\right)^2 = \frac{ss_2}{d}\left(\frac{d-s}{d}\frac{d-s_2}{d-1} + \frac{ss_2}{d}\right) \\ &= \frac{ss_2}{d}\left(\frac{d^2 - sd - s_2 d + ss_2 + ss_2 d - ss_2}{d(d-1)}\right) = \frac{ss_2}{d}\left(\frac{d - s - s_2 + ss_2}{d-1}\right) \\ &= \frac{ss_2}{d}\left(\frac{(s-1)(s_2-1)}{d-1} + 1\right) \end{aligned} \tag{2.54}$$

Where (a) follows by the closed form for the variance of a hypergeometric variable given in [147]. Therefore, plugging in into equation 2.53:

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

$$+\|\nabla_{F^c} f(\boldsymbol{x})\|^2 \left[\frac{s}{(d-s)}\left(s_2 - \left(\frac{(s-1)(s_2-1)}{d-1}+1\right)\right)\right]\Bigg] \tag{2.55}$$

Let us simplify the rightmost term:

$$\begin{aligned}\frac{s}{(d-s)}\left(s_2 - \left(\frac{(s-1)(s_2-1)}{d-1}+1\right)\right) &= \frac{s(s_2-1)}{d-s}\left[1-\frac{s-1}{d-1}\right]\\ &= \frac{s(s_2-1)}{(d-s)}\left[\frac{d-s}{d-1}\right] = \frac{s(s_2-1)}{d-1}\end{aligned} \tag{2.56}$$

Plugging it back into equation 2.55:

$$\begin{aligned}&\mathbb{E}_S \nabla f(\boldsymbol{x})^T \left(\mathbb{E}_{\mathcal{U}_S|S} \boldsymbol{u}\boldsymbol{u}^T \|\boldsymbol{u}_F\|^2\right) \nabla f(\boldsymbol{x})\\ &= \frac{1}{d(s_2+2)}\left[\|\nabla_F f(\boldsymbol{x})\|^2\left(\frac{(s-1)(s_2-1)}{d-1}+3\right) + \|\nabla_{F^c} f(\boldsymbol{x})\|^2\left(\frac{s(s_2-1)}{d-1}\right)\right]\end{aligned} \tag{2.57}$$

Finally, plugging this back into equation 2.38:

$$\begin{aligned}\mathbb{E}\|\hat{\nabla}_F f(\boldsymbol{x})\|^2 &= 2d^2\left[L_{s_2}^2\mu^2\mathbb{E}\|\boldsymbol{u}_F\|^2 + \nabla f(\boldsymbol{x})^T\left(\mathbb{E}\boldsymbol{u}\boldsymbol{u}^T\|\boldsymbol{u}_F\|^2\right)\nabla f(\boldsymbol{x})\right]\\ &= 2d^2\left[L_{s_2}^2\mu^2\mathbb{E}_k\mathbb{E}_{\mathbf{u}||S\cap F|=k}\|\boldsymbol{u}_F\|^2 + \nabla f(\boldsymbol{x})^T\left(\mathbb{E}\boldsymbol{u}\boldsymbol{u}^T\|\boldsymbol{u}_F\|^2\right)\nabla f(\boldsymbol{x})\right]\\ &= 2d^2\left[L_{s_2}^2\mu^2\mathbb{E}_k k^2 + \nabla f(\boldsymbol{x})^T\left(\mathbb{E}\boldsymbol{u}\boldsymbol{u}^T\|\boldsymbol{u}_F\|^2\right)\nabla f(\boldsymbol{x})\right]\\ &= d2L_{s_2}^2\mu^2 s s_2\left(\frac{(s-1)(s_2-1)}{d-1}+1\right)\\ &\quad+\frac{2d}{(s_2+2)}\left[\|\nabla_F f(\boldsymbol{x})\|^2\left(\frac{(s-1)(s_2-1)}{d-1}+3\right)\right.\\ &\quad\left.+\|\nabla_{F^c} f(\boldsymbol{x})\|^2\left(\frac{s(s_2-1)}{d-1}\right)\right]\end{aligned} \tag{2.58}$$

□

2.5.2.2 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 [95], however we make sure that it works for our random support gradient estimator, and with our new expression in 2.5.6, 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 2.5.5 and 2.5.6 into Lemma 2.5.7 below.

Lemma 2.5.7. *For any (L_{s_2}, s_2)-RSS function f, we use the gradient estimator $\hat{\nabla} f(x)$ defined in equation 2.6 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) $\mathbb{E}\left\|\hat{\nabla}_F f(\boldsymbol{x})\right\|^2 \leq \left(\frac{\varepsilon_F}{q} + 2\right)\|\nabla_F f(\boldsymbol{x})\|^2 + \frac{\varepsilon_{Fc}}{q}\|\nabla_{F^c} f(\boldsymbol{x})\|^2 + \left(\frac{\varepsilon_{abs}}{q} + 2\varepsilon_\mu\right)\mu^2$

Proof. Let us denote by $\hat{\nabla} f(\boldsymbol{x}; (\boldsymbol{u}_i)_{i=1}^q)$ the gradient estimate from equation 2.6 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:

$$\mathbb{E}\hat{\nabla} f(\boldsymbol{x}; (\boldsymbol{u}_i)_{i=1}^q) = \mathbb{E}\frac{1}{q}\sum_{i=1}^q \hat{\nabla} f(\boldsymbol{x}; \boldsymbol{u}_i) = \frac{1}{q}\sum_{i=1}^q \mathbb{E}\hat{\nabla} f(\boldsymbol{x}; \boldsymbol{u}_1) = \mathbb{E}\hat{\nabla} f(\boldsymbol{x}; \boldsymbol{u}_1) \tag{2.59}$$

This proves 2.5.7 (a). Let us now turn to 2.5.7 (b). We have:

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\nabla}_F f(\boldsymbol{x}; (\boldsymbol{u}_i)_{i=1}^q)\right\|^2\right] &= \mathbb{E}\left\|\frac{1}{q}\sum_{i=1}^q \hat{\nabla}_F f(\boldsymbol{x}; \boldsymbol{u}_i)\right\|^2 \\
&= \frac{1}{q^2}\mathbb{E}\left(\sum_{i=1}^q \hat{\nabla}_F f(\boldsymbol{x}; \boldsymbol{u}_i)\right)^\top \left(\sum_{i=1}^q \hat{\nabla}_F f(\boldsymbol{x}; \boldsymbol{u}_i)\right) \\
&= \frac{1}{q^2}\sum_{i=1}^q\sum_{j=1}^q \mathbb{E}\left[\hat{\nabla}_F f(\boldsymbol{x}; \boldsymbol{u}_i)^\top \hat{\nabla}_F f(\boldsymbol{x}; \boldsymbol{u}_j)\right] \\
&\overset{(a)}{=} \frac{1}{q^2}\left[q\mathbb{E}\|\hat{\nabla}_F f(\boldsymbol{x}; \boldsymbol{u}_1)\|^2 + \sum_{i=1}^q \sum_{j=1 (j\neq i)}^q (\mathbb{E}\hat{\nabla}_F f(\boldsymbol{x}; \boldsymbol{u}_i))^\top(\mathbb{E}\hat{\nabla}_F f(\boldsymbol{x}; \boldsymbol{u}_j))\right] \\
&= \frac{1}{q^2}\left[q\mathbb{E}\|\hat{\nabla}_F f(\boldsymbol{x}; \boldsymbol{u}_1)\|^2 + q(q-1)\|\mathbb{E}\hat{\nabla}_F f(\boldsymbol{x}; \boldsymbol{u}_1)\|^2\right] \\
&\overset{(b)}{\leq} \frac{1}{q^2}\left[q\left[\varepsilon_F\|\nabla_F f(\boldsymbol{x})\|^2 + \varepsilon_{F^c}\|\nabla_{F^c} f(\boldsymbol{x})\|^2 + \varepsilon_{abs}\mu^2\right] + q\,(q-1)\left\|\mathbb{E}\hat{\nabla}_F f(\boldsymbol{x}; \boldsymbol{u}_1)\right\|^2\right]
\end{aligned} \tag{2.60}$$

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 2.5.7). Assumption (ii) also allows to bound the last term above in the following way:

$$\begin{aligned}
\|\mathbb{E}\hat{\nabla}_F f(\boldsymbol{x}; \boldsymbol{u}_1)\|^2 &\leq 2\|\nabla_F f(\boldsymbol{x}; \boldsymbol{u}_1) - \mathbb{E}\hat{\nabla}_F f(\boldsymbol{x}; \boldsymbol{u}_1)\|^2 + 2\|\nabla_F f(\boldsymbol{x}; \boldsymbol{u}_1)\|^2 \\
&\leq 2\varepsilon_\mu\mu^2 + 2\|\nabla_F f(\boldsymbol{x}; \boldsymbol{u}_1)\|^2
\end{aligned} \tag{2.61}$$

Plugging equation 2.61 into equation 2.60, we obtain:

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

$$
\begin{aligned}
&+\frac{1}{q}\left[\varepsilon_{abs}\mu^2+2\left(q-1\right)\varepsilon_{\mu}\mu^2\right] \\
&\leq\left(\frac{\varepsilon_F}{q}+2\right)\|\nabla_F f(\boldsymbol{x})\|^2+\frac{\varepsilon_{F^c}}{q}\|\nabla_{F^c} f(\boldsymbol{x})\|^2+\left(\frac{\varepsilon_{abs}}{q}+2\varepsilon_{\mu}\right)\mu^2 \qquad (2.62)
\end{aligned}
$$

□

2.5.3 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 2.5.5 and Lemma 2.5.6 into Lemma 2.5.7. 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 $p \in \mathbb{N}^*$. □

2.5.4 Proof of Theorem 1

Proof. We will combine the proof from [157] and [114], using ideas of the proof of Theorem 8 from Nesterov to deal with zeroth order gradient approximations, and ideas from the proof of [157] (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:

$$
\begin{aligned}
\|\boldsymbol{x}^t-\boldsymbol{x}^*-\eta\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t)+\eta\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 &= \|\boldsymbol{x}^t-\boldsymbol{x}^*\|^2-2\eta\langle\boldsymbol{x}^t-\boldsymbol{x}^*,\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t)-\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\rangle \\
&\quad+\eta^2\|\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t)-\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \qquad (2.63)
\end{aligned}
$$

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 \\
&\quad+\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^2L_{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 (2.64)
\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 equation 2.64 into the following:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi}.\boldsymbol{u}}\|\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 &\leq 2\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t)\|^2 + 2\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&\leq 2\varepsilon_F \mathbb{E}_{\boldsymbol{\xi}}\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t)\|^2 + 2\varepsilon_{F^c}\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_{F^c} f_{\boldsymbol{\xi}}(\boldsymbol{x}^t)\|^2 \\
&\quad + 2\varepsilon_{\text{abs}}\mu^2 + 2\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&\leq 2\varepsilon_F\left[2\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 + 2\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2\right] \\
&\quad + 2\varepsilon_{F^c}\Big[2\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_{F^c} f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_{F^c} f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&\quad + 2\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_{F^c} f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2\Big] + 2\varepsilon_{\text{abs}}\mu^2 + 2\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \qquad (2.65)
\end{aligned}$$

Just like the proof in [157], 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:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&\leq 4\varepsilon_F \mathbb{E}_{\boldsymbol{\xi}}\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 + 4\varepsilon_{F^c}\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_{F^c} f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&\quad + ((4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k))\mathbb{E}_{\boldsymbol{\xi}}\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|_\infty^2 + 2\varepsilon_{\text{abs}}\mu^2 \\
&\overset{(a)}{\leq} 4\varepsilon_F \mathbb{E}_{\boldsymbol{\xi}}\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) - \nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 + ((4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k))\mathbb{E}_{\boldsymbol{\xi}}\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|_\infty^2 + 2\varepsilon_{\text{abs}}\mu^2 \\
&\hfill (2.66)
\end{aligned}$$

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 equation 2.64, 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:

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

$$+ \frac{\varepsilon_\mu}{L_{s'}^2}\mu + 2\eta^2\varepsilon_{\text{abs}}\mu^2 + \eta^2((4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k))\mathbb{E}_{\boldsymbol{\xi}}\|\nabla f(\boldsymbol{x}^*)\|_\infty^2 \tag{2.67}$$

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:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\boldsymbol{x}^t - \boldsymbol{x}^* - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) + \eta\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \\
&\quad \leq \left(1 - 2\eta\nu_s + (4\varepsilon_F + 1)L_{s'}^2\eta^2\right)\|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 \\
&\qquad + \frac{\varepsilon_\mu}{L_{s'}^2}\mu^2 + 2\eta^2\varepsilon_{\text{abs}}\mu^2 + \eta^2((4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k))\mathbb{E}_{\boldsymbol{\xi}}\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|_\infty^2
\end{aligned} \tag{2.68}$$

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:

$$\rho^2 = 1 - \frac{4\nu_s^2}{4(4\varepsilon_F + 1)L_{s'}^2} = 1 - \frac{\nu_s^2}{(4\varepsilon_F + 1)L_{s'}^2} \tag{2.69}$$

We therefore have:

$$\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\boldsymbol{x}^t - \boldsymbol{x}^* - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) + \eta\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2 \tag{2.70}$$

$$\leq \rho^2\|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 + \frac{\varepsilon_\mu}{L_{s'}^2}\mu^2 + 2\eta^2\varepsilon_{\text{abs}}\mu^2 + \eta^2((4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k))\mathbb{E}_{\boldsymbol{\xi}}\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|_\infty^2 \tag{2.71}$$

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:

$$\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\boldsymbol{x}^t - \boldsymbol{x}^* - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^t) + \eta\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\| \tag{2.72}$$

$$\leq \rho\|\boldsymbol{x}^t - \boldsymbol{x}^*\| + \frac{\sqrt{\varepsilon_\mu}}{L_{s'}}\mu^2 + \eta\sqrt{2\varepsilon_{\text{abs}}\mu^2} + \eta\sqrt{((4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k)))\mathbb{E}_{\boldsymbol{\xi}}\|\nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|_\infty^2} \tag{2.73}$$

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:

$$\boldsymbol{y}^t := \boldsymbol{x}^t - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}\left(\boldsymbol{x}^t\right) \tag{2.74}$$

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\|\boldsymbol{y}^t - \boldsymbol{x}^*\| &= \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\left\|\boldsymbol{x}^t - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}\left(\boldsymbol{x}^t\right) - \boldsymbol{x}^*\right\| \\
&\leq \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\left\|\boldsymbol{x}^t - \boldsymbol{x}^* - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}\left(\boldsymbol{x}^t\right) + \eta\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\right\| + \eta\mathbb{E}_{\boldsymbol{\xi}}\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\| \\
&= \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\left\|\boldsymbol{x}^t - \boldsymbol{x}^* - \eta\hat{\nabla}_F f_{\boldsymbol{\xi}}\left(\boldsymbol{x}^t\right) + \eta\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\right\| + \eta\mathbb{E}_{\boldsymbol{\xi}}\sqrt{\|\nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*)\|^2}
\end{aligned}$$

$$
\begin{aligned}
&\leq \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}} \left\| \boldsymbol{x}^t - \boldsymbol{x}^* - \eta \hat{\nabla}_F f_{\boldsymbol{\xi}} \left(\boldsymbol{x}^t\right) + \eta \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*) \right\| + \eta \sqrt{\mathbb{E}_{\boldsymbol{\xi}} \left\| \nabla_F f_{\boldsymbol{\xi}}(\boldsymbol{x}^*) \right\|^2} \\
&\leq \rho \left\| \boldsymbol{x}^t - \boldsymbol{x}^* \right\| + \eta(\sqrt{((4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k))\mathbb{E}_{\boldsymbol{\xi}} \left\| \nabla f(\boldsymbol{x}^*) \right\|_\infty^2} \\
&\quad + \sqrt{s}\sqrt{\mathbb{E}_{\boldsymbol{\xi}} \left\| \nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*) \right\|_\infty^2}) + \frac{\sqrt{\varepsilon_\mu}}{L_{s'}}\mu^2 + \eta\sqrt{2\varepsilon_{\text{abs}}\mu^2} \\
&= \rho \left\| \boldsymbol{x}^t - \boldsymbol{x}^* \right\| + \eta(\sqrt{(4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k)} + \sqrt{s}) \sqrt{\mathbb{E}_{\boldsymbol{\xi}} \left\| \nabla f_{\boldsymbol{\xi}}(\boldsymbol{x}^*) \right\|_\infty^2} \\
&\quad + \frac{\sqrt{\varepsilon_\mu}}{L_{s'}}\mu + \eta\sqrt{2\varepsilon_{\text{abs}}\mu^2} \\
&\overset{(a)}{\leq} \rho \left\| \boldsymbol{x}^t - \boldsymbol{x}^* \right\| + \eta(\sqrt{(4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k)} + \sqrt{s})\sigma \\
&\quad + \frac{\sqrt{\varepsilon_\mu}}{L_{s'}}\mu + \eta\sqrt{2\varepsilon_{\text{abs}}\mu^2} \\
&\leq \rho \left\| \boldsymbol{x}^t - \boldsymbol{x}^* \right\| + \eta(\sqrt{(4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k)} + \sqrt{s})\sigma \\
&\quad + \frac{\sqrt{\varepsilon_\mu}}{L_{s'}}\mu + \eta\sqrt{2\varepsilon_{\text{abs}}\mu^2} \qquad (2.75)
\end{aligned}
$$

Where (a) follows from the σ-FGN assumption. We now consider $\boldsymbol{x}^{t+1}$, that is, the best-k-sparse approximation of $\boldsymbol{z}^t := \boldsymbol{x}^t - \eta \hat{\nabla} f_{\boldsymbol{\xi}}\left(\boldsymbol{x}^t\right)$ from the hard thresholding operation in SZOHT. We can notice that $\boldsymbol{x}_F^t = \boldsymbol{x}^t$ (because $\text{supp}(\boldsymbol{x}^t) = F^{(t)} \subset F$), which gives $\boldsymbol{y}^t = \boldsymbol{z}_F^t$. 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 $\boldsymbol{z}_F^t = \boldsymbol{y}^t$. Therefore, $\boldsymbol{x}^{t+1}$ is also the best k-sparse approximation of $\boldsymbol{y}^t$. Therefore, using Corollary 2.5.1, we obtain:

$$
\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\| \leq \gamma \|\boldsymbol{y}^t - \boldsymbol{x}^*\| \qquad (2.76)
$$

with:

$$
\gamma := \sqrt{1 + \left(k^*/k + \sqrt{\left(4 + k^*/k\right) k^*/k}\right)/2} \qquad (2.77)
$$

Where $k^* = \|\boldsymbol{x}^*\|_0$. Plugging this into equation 2.75 gives:

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}} \|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\| \leq \gamma\rho \left\| \boldsymbol{x}^t - \boldsymbol{x}^* \right\| + \gamma\eta(\sqrt{(4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k)}) + \sqrt{s})\sigma \qquad (2.78) \\
+ \gamma\frac{\sqrt{\varepsilon_\mu}}{L_{s'}}\mu + \eta\sqrt{2\varepsilon_{\text{abs}}}\mu \qquad (2.79)
\end{aligned}
$$

This will allow us to obtain the following final result:

$$
\begin{aligned}
\mathbb{E}\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\| \leq \gamma\rho \left\| \boldsymbol{x}^t - \boldsymbol{x}^* \right\| + \gamma \underbrace{\eta \left(\sqrt{(4\varepsilon_F s + 2) + \varepsilon_{F^c}(d-k)} + \sqrt{s} \right)}_{:=a} \sigma \\
+ \gamma \underbrace{\left(\frac{\sqrt{\varepsilon_\mu}}{L_{s'}} + \eta\sqrt{2\varepsilon_{\text{abs}}} \right)}_{:=b} \mu \qquad (2.80)
\end{aligned}
$$

with $\eta = \frac{\nu_s}{(4\varepsilon_F + 1)L_{s'}^2}$ and $\rho^2 = 1 - \frac{2\nu_s^2}{(4\varepsilon_F + 1)L_{s'}^2}$. 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 equation 2.77:

$$\begin{aligned}
\gamma^2 &\leq 1 + \left(\frac{(1-\rho^2)^2}{\rho^2} + \sqrt{\left(4 + \frac{(1-\rho^2)^2}{\rho^2}\right)\frac{(1-\rho^2)^2}{\rho^2}}\right)\frac{1}{2} \\
&= 1 + \left(\frac{(1-\rho^2)^2}{\rho^2} + \sqrt{\left(\frac{4\rho^2 + 1 + \rho^4 - 2\rho^2}{\rho^2}\right)\frac{(1-\rho^2)^2}{\rho^2}}\right)\frac{1}{2} \\
&= 1 + \left(\frac{(1-\rho^2)^2}{\rho^2} + \sqrt{\frac{(1+\rho^2)^2(1-\rho^2)^2}{\rho^4}}\right)\frac{1}{2} \\
&= 1 + \left(\frac{(1-\rho^2)^2}{\rho^2} + \frac{(1+\rho^2)(1-\rho^2)}{\rho^2}\right)\frac{1}{2} = 1 + \left(\frac{(1-\rho^2)(1-\rho^2+1+\rho^2)}{\rho^2}\right)\frac{1}{2} \\
&= 1 + \frac{(1-\rho^2)}{\rho^2} = \frac{1}{\rho^2}
\end{aligned} \tag{2.81}$$

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

Unrolling inequality equation 2.80 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:

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\| &= \mathbb{E}_{\boldsymbol{\xi}^t,\boldsymbol{u}^t,..,\boldsymbol{\xi}^1,\boldsymbol{u}^1}\mathbb{E}_{\boldsymbol{\xi}^{t+1},\boldsymbol{u}^{t+1}|\boldsymbol{\xi}^t,\boldsymbol{u}^t,..,\boldsymbol{\xi}^1,\boldsymbol{u}^1}\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\| \\
&\leq \mathbb{E}_{\boldsymbol{\xi}^t,\boldsymbol{u}^t,..,\boldsymbol{\xi}^1,\boldsymbol{u}^1}[\gamma\rho\|\boldsymbol{x}^t - \boldsymbol{x}^*\| + \gamma a\sigma + \gamma b\mu] \\
&= \gamma\rho\mathbb{E}_{\boldsymbol{\xi}^t,\boldsymbol{u}^t,..,\boldsymbol{\xi}^1,\boldsymbol{u}^1}[\|\boldsymbol{x}^t - \boldsymbol{x}^*\|] + \gamma a\sigma + \gamma b\mu \\
&\leq (\gamma\rho)^2\mathbb{E}_{\boldsymbol{\xi}^{t-1},\boldsymbol{u}^{t-1},..,\boldsymbol{\xi}^1,\boldsymbol{u}^1}[\|\boldsymbol{x}^{t-1} - \boldsymbol{x}^*\|] + (\gamma\rho)^2 a\sigma \\
&\quad + \gamma a\sigma + (\gamma\rho)^2 b\mu + \gamma b\mu \\
&\leq (\gamma\rho)^{t+1}\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\| + \left(\sum_{i=0}^{t}(\gamma\rho)^i\right)\gamma a\sigma + \left(\sum_{i=0}^{t}(\gamma\rho)^i\right)\gamma b\mu \\
&= (\gamma\rho)^{t+1}\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\| + \frac{1-(\gamma\rho)^t}{1-\gamma\rho}\gamma a\sigma + \frac{1-(\gamma\rho)^t}{1-\gamma\rho}\gamma b\mu \\
&\leq (\gamma\rho)^{t+1}\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\| + \frac{1}{1-\gamma\rho}\gamma a\sigma + \frac{1}{1-\gamma\rho}\gamma b\mu
\end{aligned} \tag{2.82}$$

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

2.5.5 Proof of Remark 4

Proof. We show below that, due to the complex impact of q and k on the convergence analysis in our ZO + HT (hard-thresholding) setting (compared to ZO only), q cannot be taken as small as we want here (in particular we can never take $q = 1$, which is different from classical ZO algorithms such as [92, Corollary 3]), if we want Theorem 1 to apply with $\rho\gamma < 1$. In other words, there is a necessary (but not sufficient) minimal (i.e. > 1) value for q.

A necessary condition for Theorem 1 to describe convergence of SZOHT is that $\rho\gamma < 1$. 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:

$$\begin{aligned}\rho^2 &= 1 - \frac{1}{\left[\frac{8d}{q(s_2+2)}\left(\frac{(s-1)(s_2-1)}{d-1} + 3\right) + 9\right]\kappa^2}\\ &= 1 - \frac{1}{\left[\frac{8d}{q(s_2+2)}\left(\frac{(2k+k^*-1)(s_2-1)}{d-1} + 3\right) + 9\right]\kappa^2}\end{aligned} \tag{2.83}$$

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:

$$\rho^2 = 1 - \frac{1}{ak+b} \tag{2.84}$$

To ensure convergence, we need to have $\rho\gamma < 1$, therefore (following the same derivation as in equation 2.81) a necessary condition that we need to verify is $k \geq \rho^2 k^*/(1-\rho^2)^2$.

Which means we need:

$$\begin{aligned}&k \geq \frac{\left(1 - \frac{1}{ak+b}\right)k^*}{\left(\frac{1}{ak+b}\right)^2}\\ &k \geq \left[(ak+b)^2 - (ak+b)\right]k^*\\ &k \geq k^*\left[a^2k^2 + 2abk + b^2 - ak - b\right]\\ &0 \geq k^*a^2k^2 + \left(2ab - \frac{1}{k^*} - a\right)k^*k + \left(b^2 - b\right)k^*\end{aligned} \tag{2.85}$$

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

$$\Delta \geq 0 \tag{2.86}$$

with:

$$\Delta := k^{*2}(2ab - \frac{1}{k^*} - a)^2 - 4k^{*^2}a^2\left(b^2 - b\right)$$

$$= k^{*^2}\left(4a^2b^2 + \left(\frac{1}{k^*} + a\right)^2 - 4ab\left(\frac{1}{k^*} + a\right)\right) - 4k^{*2}a^2\left(b^2 - b\right) \tag{2.87}$$

$$\begin{aligned} &= k^{*^2}\left[4a^2b^2 + \frac{1}{k^{*^2}} + a^2 + \frac{2a}{k^*} - \frac{4ab}{k^*} - 4a^2b - 4a^2b^2 + 4a^2b\right] \\ &= 1 + a^2k^{*2} + 2ak^* - 4abk^* \end{aligned} \tag{2.88}$$

$$\Delta \geq 0 \Rightarrow 1 + a^2k^{*2} + 2ak^* \geq 4abk^* \tag{2.89}$$

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

$$a = \frac{A}{q} \quad \text{with} \quad A = \frac{16d\kappa^2\left(s_2 - 1\right)}{\left(s_2 + 2\right)\left(d - 1\right)} \tag{2.90}$$

$$b = \frac{B}{q} + C \quad \text{with} \quad B = \kappa^2\left[\frac{8d}{\left(s_2 + 2\right)}\left(\frac{\left(s_2 - 1\right)\left(k^* - 1\right)}{d - 1} + 3\right)\right] \tag{2.91}$$

$$\text{and with } C = 9\kappa^2 \tag{2.92}$$

So plugging in equation 2.89, what we need is:

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

$$q^2 + q\left(2Ak^* - 4CAk^*\right) + A^2k^{*2} - 4ABk^* \geq 0 \tag{2.93}$$

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

$$\begin{aligned} \Delta' &:= \left(2Ak^* - 4CAk^*\right)^2 - 4\left(A^2k^{*2} - 4ABk^*\right) \\ &= 4A^2k^{*2} + 16C^2A^2k^{*2} - 16CA^2k^{*2} - 4A^2k^{*2} + 16ABk^* \\ &= 16CA^2k^{*2}(C - 1) + 16ABk^* = 16Ak^*\left[k^*C(C - 1)A + B\right] \end{aligned} \tag{2.94}$$

We now have:

$$C = 9\kappa^2 \Rightarrow C \geq 1 \Rightarrow \Delta' \geq 0 \tag{2.95}$$

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

$$q \geq q_{\min} \tag{2.96}$$

With:

$$\begin{aligned} q_{\min} &= \frac{-\left(2Ak^* - 4CAk^*\right) + \sqrt{16CA^2k^{*2}(C - 1) + 16ABk^*}}{2} \\ &= \frac{2Ak^*\left(2C - 1\right) + \sqrt{16A^2k^{*2}\left[C(C - 1) + \frac{B}{Ak^*}\right]}}{2} \end{aligned} \tag{2.97}$$

Case $s_2 > 1$: Assuming $s_2 > 1$ gives $A > 0$, and since $A = \frac{16d\kappa^2(s_2-1)}{(s_2+2)(d-1)}$ and $B = \frac{8\kappa^2 d}{s_2+2}\left(\frac{(s_2-1)(k^*-1)}{d-1} + 3\right)$

This gives: $\frac{B}{Ak^*} = \frac{1}{2} - \frac{1}{2k^*} + \frac{3}{2}\frac{d-1}{k^*(s_2-1)}$
Therefore: $q_{\min} = Ak^*\left[2C - 1 + 2\sqrt{C(C-1) + \frac{1}{2} - \frac{1}{2k^*} + \frac{3}{2}\frac{d-1}{k^*(s_2-1)}}\right]$
with $C = 9\kappa^2$, which reads:

$$q_{\min} = \frac{16d(s_2-1)k^*\kappa^2}{(s_2+2)\,(d-1)}\left[18\kappa^2 - 1 + 2\sqrt{9\kappa^2(9\kappa^2-1) + \frac{1}{2} - \frac{1}{2k^*} + \frac{3}{2}\frac{d-1}{k^*(s_2-1)}}\right] \tag{2.98}$$

Case $s_2 = 1$: In the case $s_2 = 1$, we have $A = 0$, so therefore, from equation 2.97, $q_{\min} = 0$, so the necessary condition on q as above so that there exist k such that: $k \geq \rho^2 k^*/(1-\rho^2)^2$ does not apply here. We may therefore think that it may be possible to take $q = 1$ in that case. However, there is another condition on k that should also be enforced, which is that $k \leq d$ (since we cannot keep more components than d). And in that $s_2 = 1$ case, we have $a = 0$, and $b = \kappa^2[8\frac{d}{q} + 9]$ (from equation 2.90 and equation 2.91). Now, enforcing the condition $k \geq k^*[(ak+b)^2 - (ak+b)] = k^*b(b-1)$ leads to the following chain of implications (i.e. each downstream assertion is a necessary condition for the upstream assertion):

$$\begin{aligned}
\frac{k}{k^*} \geq b(b-1) \quad \text{and} \quad k \leq d \implies & \frac{d}{k^*} \geq (b-1)^2 \\
\implies & \sqrt{\frac{d}{k^*}} + 1 \geq b \implies \sqrt{\frac{d}{k^*}} + 1 \geq \frac{B}{q} + C \\
\implies & \sqrt{\frac{d}{k^*}} + 1 - C \geq \frac{B}{q} \\
\implies & q \geq \frac{B}{\sqrt{\frac{d}{k^*}} + 1 - C} \quad \text{and} \quad C - \sqrt{\frac{d}{k^*}} + 1 > 0 \\
\implies & q \geq \frac{B}{\sqrt{\frac{d}{k^*}} + 1} \implies q \geq \frac{8\kappa^2 d}{\sqrt{\frac{d}{k^*}} + 1}
\end{aligned} \tag{2.99}$$

Where the last inequality follows from the expression of B in equation 2.91 when $s_2 = 1$.

So the right hand side in equation 2.99 is also a minimal necessary value for q in this case, though for a different reason than in the case $s_2 > 1$.

□

2.5.6 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:

$$q' := \frac{2d}{s_2+2}\left(\frac{(s-1)(s_2-1)}{d-1}+3\right) \tag{2.100}$$

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

$$\eta' := \frac{\nu_s}{(4\varepsilon'_F+1)L_{s'}^2} = \frac{\nu_s}{13L_{s'}^2} \tag{2.101}$$

and:

$$\rho'^2 := 1 - \frac{2\nu_s^2}{(4\varepsilon'_F+1)L_{s'}^2} = 1 - \frac{2\nu_s^2}{13L_{s'}^2} \tag{2.102}$$

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:

$$k \geq 2k^* \frac{\rho^2}{(1-\rho^2)^2} \tag{2.103}$$

and plug the value of ρ above into the expression:

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

$$\iff k \geq 2k^*(\frac{13}{2}\kappa^2)(\frac{13}{2}\kappa^2-1) \tag{2.104}$$

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

$$k \geq (86\kappa^4 - 12\kappa^2)k^* \tag{2.105}$$

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:

$$t \geq \frac{1}{\log\frac{1}{\gamma\rho}} \log(\frac{1}{\varepsilon}) \log(\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|) \tag{2.106}$$

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 equation 2.103, and plugging that value into the expression of γ from Theorem 1, we obtain:

$$\gamma^2 = 1 + \left(\frac{(1-\rho^2)^2}{2\rho^2} + \sqrt{\left(4 + \frac{(1-\rho^2)^2}{2\rho^2}\right) \frac{(1-\rho^2)^2}{2\rho^2}} \right) /2 \tag{2.107}$$

$$\leq 1 + \frac{1}{\sqrt{2}} \left(\frac{(1-\rho^2)^2}{\rho^2} + \sqrt{\left(4 + \frac{(1-\rho^2)^2}{\rho^2}\right) \frac{(1-\rho^2)^2}{\rho^2}} \right) /2 \tag{2.108}$$

$$\overset{(a)}{=} 1 + \frac{1}{\sqrt{2}} \frac{1-\rho^2}{\rho^2} \tag{2.109}$$

Where the simplification in (a) above follows similarly to equation 2.81. Therefore, in that case, we have:

$$\rho^2\gamma^2 \leq \rho^2 + \frac{1}{\sqrt{2}}(1-\rho^2) = \frac{1}{\sqrt{2}} + \rho^2(1 - \frac{1}{\sqrt{2}})$$

$$= \frac{1}{\sqrt{2}} + (1 - \frac{2}{13\kappa^2})(1 - \frac{1}{\sqrt{2}}) = 1 - \frac{2(1-\frac{1}{\sqrt{2}})}{13\kappa^2} \overset{(a)}{\leq} 1 - \frac{1}{26\kappa^2} \tag{2.110}$$

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

$$\frac{1}{(\rho\gamma)^2} \geq \frac{1}{1 - \frac{1}{26\kappa^2}} \tag{2.111}$$

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

$$\log\left(\frac{1}{(\rho\gamma)^2}\right) \geq \frac{1}{26\kappa^2} \tag{2.112}$$

Therefore:

$$\frac{1}{\log(\frac{1}{\rho\gamma})} = \frac{2}{\log(\frac{1}{(\rho\gamma)^2})} \leq 52\kappa^2 \tag{2.113}$$

Therefore, plugging this into equation 2.106, 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 $s = 2k + k^*$.

2.5.7 Proof of Corollary 2

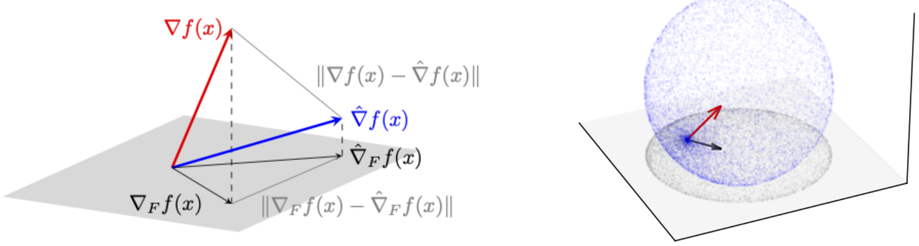
Almost all $f_{\boldsymbol{\xi}}$ are L-smooth, which is equivalent to saying that they are (L, d)-RSS. So we can directly plug $s_2 = d$ in equation equation 2.100, which gives a necessary value for q of:

$$q = \frac{2d}{d+2}(s+2) \tag{2.114}$$

Since any value of q larger than the one in equation 2.114 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: $2(s+2)52\kappa^2 \log(\frac{1}{\varepsilon}) \log(\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|) = \mathcal{O}(k\kappa \log(\frac{1}{\varepsilon}))$ □

2.6 Visualization: Projection of the Gradient Estimator onto a Sparse Support

Below we plot the true gradient $\nabla f(\boldsymbol{x})$ and its estimator $\hat{\nabla} f(\boldsymbol{x})$ (for $q = 1$), as well as their respective projections $\nabla_F f(\boldsymbol{x})$ and $\hat{\nabla}_F f(\boldsymbol{x})$, with $F = \{0, 1\}$ (i.e. F is the hyperplane $z = 0$), for n_{dir} random directions. In Figure 2.2(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 $\nabla_F f(\boldsymbol{x})$.


$\nabla f(x)$
$\|\nabla f(x) - \hat{\nabla} f(x)\|$
$\hat{\nabla} f(x)$
$\hat{\nabla}_F f(x)$
$\nabla_F f(x)$
$\|\nabla_F f(x) - \hat{\nabla}_F f(x)\|$

(a) $n_{\text{dir}} = 1$

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

Figure 2.2: $\nabla f(x)$ and $\hat{\nabla} f(x)$ and their projections $\nabla_F f(\boldsymbol{x})$ and $\hat{\nabla}_F f(\boldsymbol{x})$ onto F.

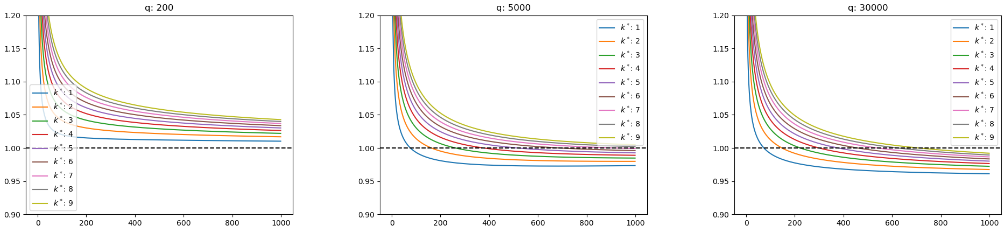
Remark 5. *An interesting fact that can be observed in Figure 2.2(b) above is that when $\mu \to 0$ and $s_2 = d$, the ZO gradient estimates belong to a sphere. This comes from the fact that, in that case, the ZO estimate using the random direction $\boldsymbol{u}$ is actually a directional derivative (scaled by d): $\hat{\nabla} f(\boldsymbol{x}) = d\langle \nabla f(\boldsymbol{x}), \boldsymbol{u}\rangle \boldsymbol{u}$, for which we have :*

$$\begin{aligned}\|\hat{\nabla} f(\boldsymbol{x}) - \frac{d}{2}\nabla f(\boldsymbol{x})\|^2 &= d^2(\langle \nabla f(\boldsymbol{x}), \boldsymbol{u}\rangle)^2 \langle \boldsymbol{u}, \boldsymbol{u}\rangle + \frac{d^2}{4}\|\nabla f(\boldsymbol{x})\|^2 \\ &\quad - d^2 \langle \nabla f(\boldsymbol{x}), \boldsymbol{u}\rangle \langle \boldsymbol{u}, \nabla f(\boldsymbol{x})\rangle \\ &= \frac{d^2}{4}\|\nabla f(\boldsymbol{x})\|^2 \end{aligned} \tag{2.115}$$

(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 2.2(b).

2.7 Parameters Relations: 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 Remark 4, by showing in Figure 2.3 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 2.3: $\rho\gamma$ (y axis) as a function of k (x axis) for several values of q and k^*.

2.8 Experiments

2.8.1 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:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{s.t.} \quad \|\boldsymbol{x}\|_0 \leq k \tag{2.116}$$

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 $\kappa = \frac{L}{\nu_s} = \frac{L_{s'}}{\nu_s} = 1$)
- $\boldsymbol{y} = \boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad s.t. \quad \|\boldsymbol{x}\|_0 \leq k^*$
- $f(\boldsymbol{y}) = f(\boldsymbol{x}^*) = 0$

- $\nabla f(\boldsymbol{y}) = \mathbf{0}$ 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.

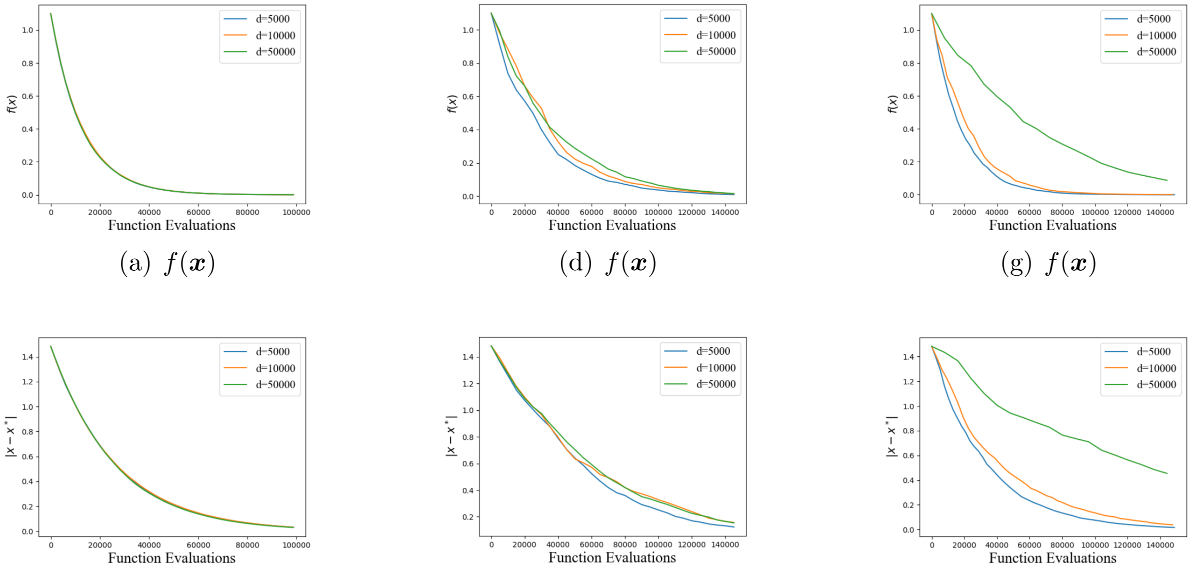
- $s_2 = d$: As from Corollary 2, we take $q = 2(s + 2)$ with $s = 2k + k^*$ (i.e. $q = 2014$). We choose $\mu = 1e - 8$, to have the smallest possible system error due to zeroth-order approximations. As we can see in Figure 2.3(c), all curves are superimposed, which shows that the query complexity is 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 2.3(f).

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 2.3(i).

2.8.2 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 2.4. 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 2.6, for six values of q. We can observe on Figure 2.6(b) that the smaller the q, the less $f(\boldsymbol{x})$ can descend. Interestingly, we can also see on Figure 2.6(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.



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

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

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

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

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

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

(c): $s_2 = d$

(f): $s_2 = \lfloor \frac{d}{k} \rfloor$

(i): $s_2 = 50$

Figure 2.4: Dependence on the dimensionality of the query complexity.

2.8.3 Real Data Experiments

2.8.3.1 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:

1. **ZSCG** [7] is a Frank-Wolfe ZO algorithm, for which we consider an ℓ_1 ball constraint.
2. **RSPGF** [59] is a proximal ZO algorithm, for which we consider an ℓ_1 penalty.
3. **ZORO** [31] 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.

2.8.3.2 Applications

We compare the algorithms above on two tasks: a sparse asset risk management task from [36], and an adversarial attack task [38] with a sparsity constraint.

Sparse Asset Risk Management. We consider the portfolio management task and dataset from [36], similarly to [31]. 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:

$$\min_{x \in \mathbb{R}^d} \frac{\boldsymbol{x}^\top \boldsymbol{C} \boldsymbol{x}}{2\left(\sum_{i=1}^d \boldsymbol{x}_i\right)^2} + \lambda \left(\min \left\{ \frac{\sum_{i=1}^d \boldsymbol{m}_i \boldsymbol{x}_i}{\sum_{i=1}^d \boldsymbol{x}_i} - r, 0 \right\} \right)^2 \quad \text{s.t.} \quad \|\boldsymbol{x}\|_0 \leq k \tag{2.117}$$

We use three datasets: port3, port4 and port5 from the OR-library [12], 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 2.7.

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 [38], 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 2.8. All experiments are conducted in the workstation with four NVIDIA RTX A6000 GPUs, and take about one day to run.

We also provide additional results for the adversarial attacks problem in Figure 2.5. The parameters we used for SZOHT to generate that table are the same as in 2.8.3.2, 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.

2.8.3.3 Results and Discussion

We can observe from Figures 2.7 and 2.8 that the performance of SZOHT is comparable or better than the other algorithms. This can be explained by the fact that SZOHT has a

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

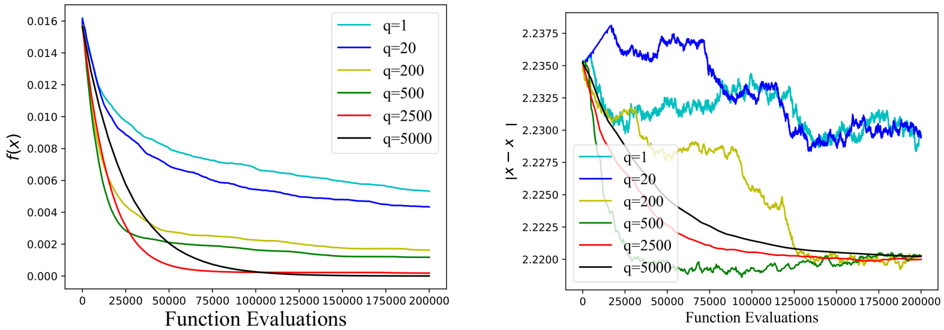
(c) ImageNet

Figure 2.5: Summary of results on adversarial attacks.

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 [58], when proximal algorithms get a linear rate [13, 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.

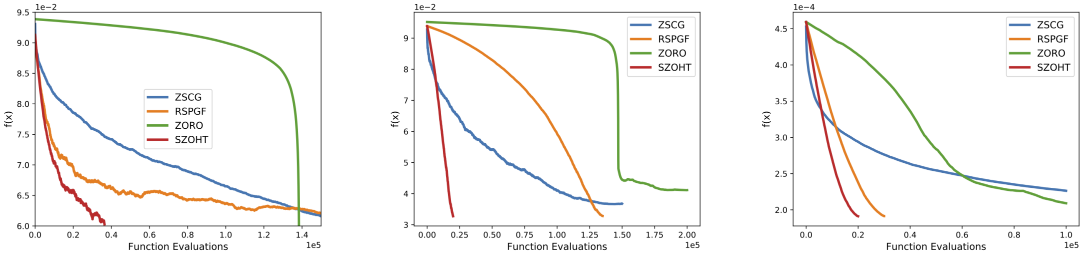
2.9 Conclusion

In this paper, we proposed a new algorithm, SZOHT, for sparse zeroth-order optimization. We gave its convergence analysis and showed that it is dimension independent in the smooth case, and weak dimension-dependent in the RSS case. We further verified experimentally the efficiency of SZOHT in several settings. Moreover, throughout the paper, we showed how the condition number of f as well as the gradient error have an important impact on the convergence of SZOHT. As such, it would be interesting to study whether we can improve the query complexity by regularizing f, by using an adaptive learning rate or acceleration methods, or by using recent variance reduction techniques. Finally, it would also be interesting to extend this work to a broader family of sparse structures, such as low-rank approximations or graph sparsity. We leave this for future work.



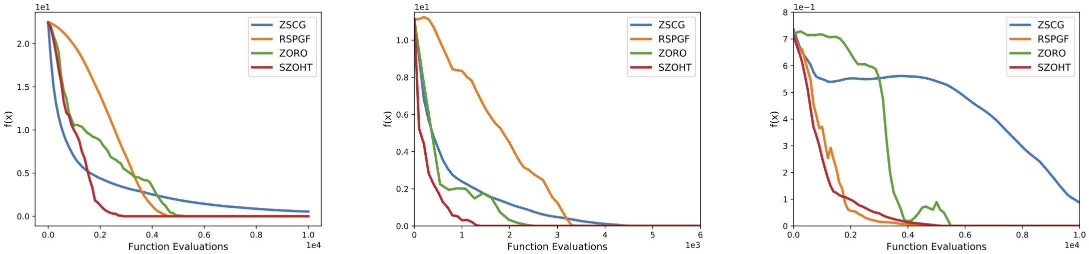
(a) $f(\boldsymbol{x})$ (b) $\|\boldsymbol{x} - \boldsymbol{x}^*\|$

Figure 2.6: Sensitivity analysis.



(a) port3 (b) port4 (c) port5

Figure 2.7: $f(\boldsymbol{x})$ vs. # queries (asset management).



(a) MNIST (b) CIFAR (c) Imagenet

Figure 2.8: $f(\boldsymbol{x})$ vs. # queries (adversarial attack).

2.10 ZOHT Extension: Variance Reduction

2.10.1 Introduction

So far, we have analyzed an algorithm for zeroth-order hard-thresholding, and shown that in order to compensate for the potential expansivity of the hard-thresholding operator, one needs to take a specific number of random directions at each step. In this section, we discuss how variance reduction algorithms ([47, 68, 72, 79, 88, 115]) can improve such number of random directions. The paper [159] proposes several variance reduction methods for zeroth-order hard-thresholding, in a unified way. For more details on those algorithms, as well as their theoretical analysis, we refer the reader to [159]. In this section, we provide the experimental comparison of such algorithms, which is our main contribution in the paper [159]. We formulate the minimization problem to be solved as follows:

$$\min_{\theta \in \mathbb{R}^d} \mathcal{F}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{\theta}), \quad s.t.\ \|\boldsymbol{\theta}\|_0 \leq k, \tag{2.118}$$

where, $\mathcal{F}(\boldsymbol{\theta})$ is the empirical risk. $\|\boldsymbol{\theta}\|_0$ represents the number of non-zero directions, and d is the dimension of $\boldsymbol{\theta}$. Below we give the original formulation of such algorithms, from [159], and where the zeroth-order gradient estimate is the one from the section above, i.e. for any $i \in [n]$:

$$\hat{\nabla} f_i(\boldsymbol{\theta}) = \frac{d}{q\mu}\sum_{j=1}^{q}\left(f_i(\boldsymbol{\theta} + \mu \boldsymbol{u}_j) - f_i(\boldsymbol{\theta})\right)\boldsymbol{u}_j, \tag{2.119}$$

with q the number of random directions, and $\boldsymbol{u}_1, ..., \boldsymbol{u}_n$ sampled as per Section 2.3.1. For the pM-SZHT algorithm (Algorithm 3 below), a key step is to select at each iteration a random index set $J \subseteq [n]$ of memory locations to update according to:

$$\forall j \in [n]: \ \hat{\boldsymbol{a}}_j^+ := \begin{cases} \hat{\nabla} f_j(\boldsymbol{\theta}), & \text{if } j \in J \\ \hat{\boldsymbol{a}}_j, & \text{otherwise} \end{cases} ,$$

in such a way that any j has the same probability p/n of being updated[1], and where p is the number of directions updated at each time (see [72]). The distribution over sets J is determined according to the specific algorithm one would wish to implement. For example, if the probability to sampled a set J, which we denote $\boldsymbol{P}\{J\}$, is as follows: $\boldsymbol{P}\{J\} = \begin{cases} 1/\binom{n}{p} \text{ if } |J| = p \\ 0 \text{ otherwise} \end{cases}$, then we obtain the p-SAGA-ZHT algorithm. Additionally, we denote SAGA-SZHT the p-SAGA-SZHT algorithm when $p = 1$.

[1] Originally, p-Memorization is called q-Memorization. We changed it to p to avoid conflicts with the q standing for the number of random directions in zeroth-order.

Algorithm 3: Stochastic Variance Reduced Zeroth-Order Hard-Thresholding with p-Memorization (pM-SZHT)

```
Initialization: Learning rate $\eta$, maximum number of iterations $T$, initial point $\boldsymbol{\theta}^{(0)}$,
 number of random directions $q$, and number of coordinates to keep at each iteration
 $k$.
Output : $\boldsymbol{\theta}^{(T)}$.
for $r = 1$ to $T$ do
    Update $\hat{\boldsymbol{a}}^{(r-1)}$
    Sample $i_r$ uniformly at random from $\{1, 2, \ldots, n\}$
    Compute $\hat{\boldsymbol{g}}^{(r-1)}(\boldsymbol{\theta}^{(r-1)}) = \hat{\nabla} f_{i_r}(\boldsymbol{\theta}^{(r-1)}) - \hat{\boldsymbol{a}}_{i_r}^{(r-1)} + \frac{1}{n}\sum_{j=1}^{n} \hat{\boldsymbol{a}}_j^{(r-1)}$
    Compute $\boldsymbol{\theta}^{(r)} = \mathcal{H}_k(\boldsymbol{\theta}^{(r-1)} - \eta \boldsymbol{g}^{(r-1)}(\boldsymbol{\theta}^{(r-1)}))$
end
```

Algorithm 4: Stochastic variance reduced zeroth-order Hard-Thresholding (VR-SZHT)

```
Initialization: Learning rate $\eta$, maximum number of iterations $T$, initial point $\boldsymbol{\theta}^{(0)}$,
 SVRG update frequency $m$, number of random directions $q$, and number of
 coordinates to keep at each iteration $k$.
Output : $\boldsymbol{\theta}^{T}$.
for $r = 1$ to $T$ do
    $\boldsymbol{\theta}^{(0)} = \boldsymbol{\theta}^{r-1}$;
    $\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n} \hat{\nabla} f_i(\boldsymbol{\theta}^{(0)})$;
    for $t = 0, 1, \ldots, m-1$ do
        Randomly sample $i_t$ uniformly at random from $\{1, 2, \ldots, n\}$;
        Compute ZO estimate $\hat{\nabla} f_{i_t}(\boldsymbol{\theta}^{(r)})$, $\hat{\nabla} f_{i_t}(\boldsymbol{\theta}^{(0)})$;
        $\bar{\boldsymbol{\theta}}^{(r+1)} = \boldsymbol{\theta}^{(r)} - \eta(\hat{\nabla} f_{i_t}(\boldsymbol{\theta}^{(r)}) - \hat{\nabla} f_{i_t}(\boldsymbol{\theta}^{(0)}) + \hat{\boldsymbol{\mu}})$;
        $\boldsymbol{\theta}^{(r+1)} = \mathcal{H}_k(\bar{\boldsymbol{\theta}}^{(r+1)})$;
    end
    $\boldsymbol{\theta}^{r} = \boldsymbol{\theta}^{(r+1)}$, random $t' \in [m-1]$;
end
```

Algorithm 5: Stochastic variance reduced zeroth-order Hard-Thresholding with SARAH (SARAH-SZHT)

```
Input   : Learning rate $\eta$, maximum number of iterations $T$, initial point $\boldsymbol{\theta}^{(0)}$,
          number of random directions $q$, maximum error $\varepsilon$, and number of
          coordinates to keep at each iteration $k$.
Output : $\boldsymbol{\theta}^T$.
Initialization: Initialize $\boldsymbol{\theta}^0 = \widetilde{\boldsymbol{\theta}}^{(r-1)}$ and $g^{(0)} = \frac{1}{n}\sum_{i=1}^{n} \hat{\nabla} f_i(\boldsymbol{\theta}_0)$. for $r = 1, \dots, T$
 do
    $\boldsymbol{\theta}^0 = \widetilde{\boldsymbol{\theta}}^{(r-1)}$;
    $\boldsymbol{g}^{(0)} = \frac{1}{n}\sum_{i=1}^{n} \hat{\nabla} f_i(\boldsymbol{\theta}_0)$;
    $\boldsymbol{\theta}^{(1)} = \boldsymbol{\theta}^{(0)} - \eta g^{(0)}$;
    for $t = 1, ..., m-1$ do
        Sample $i_t$ uniformly at random from $[n]$;
        $\hat{\boldsymbol{g}}^{(t)} = \hat{\nabla} f_{i_t}(\boldsymbol{\theta}^{(t)}) - \hat{\nabla} f_{i_t}(\boldsymbol{\theta}^{(t-1)}) + \hat{\boldsymbol{g}}^{(t-1)}$;
        $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \boldsymbol{g}^{(t)}$;
    end
    Set $\widetilde{\boldsymbol{\theta}}^{(r)} = \boldsymbol{\theta}^{(d)}$ with $d$ chosen uniformly at random from $\{0, 1, \dots, m\}$;
end
```

2.10.2 Experiments

We compare the performance of VR-SZHT, SAGA-SZHT, and SARAH-SZHT with that of the following algorithms, in terms of IZO (iterative zeroth-order oracle, i.e. number of calls to f_i) and NHT (number of hard-thresholding operations):

- SZOHT [46]: the vanilla stochastic ZO hard-thresholding algorithm from the section above.
- FGZOHT: the full gradient version of SZOHT.

2.10.2.1 Ridge Regression

We first consider the following ridge regression problem, where the functions f_i are defined as follows: $f_i(\boldsymbol{\theta}) = (\boldsymbol{x}_i^\top \boldsymbol{\theta} - y_i)^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2$, where λ is some regularization parameter. We generate each $\boldsymbol{x}_i$ randomly from a unit norm ball in $\mathbb{R}^d$, and a true random model $\boldsymbol{\theta}^*$ from a normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{d\times d})$. Each y_i is defined as $y_i = \boldsymbol{x}_i^\top \boldsymbol{\theta}^*$. We set the constants of the problem as such: $n = 10, d = 5, \lambda = 0.5$. Before training, we preprocess each column by subtracting its mean and dividing it by its empirical standard deviation. We run each algorithm with $k = 3, q = 200, \mu = 10^{-4}, s_2 = d = 5$, and for the variance reduced algorithms, we choose $m = 10$. For all algorithms, the learning rate η is found through grid-search in $\{0.005, 0.01, 0.05, 0.1, 0.5\}$: we choose the learning rates giving the lowest function value (averaged over several runs) at the end of training. We stop each algorithm once its number of IZO reaches 80,000. All curves are averaged over 3 runs, and we plot their mean and standard deviation in Figure 2.9. As we can observe, SZOHT converges to higher function values than other algorithms: this illustrates the advantage of the variance reduction techniques, which can allow to attain smaller function values than plain SZOHT, but at a cheaper cost than FGZOHT.

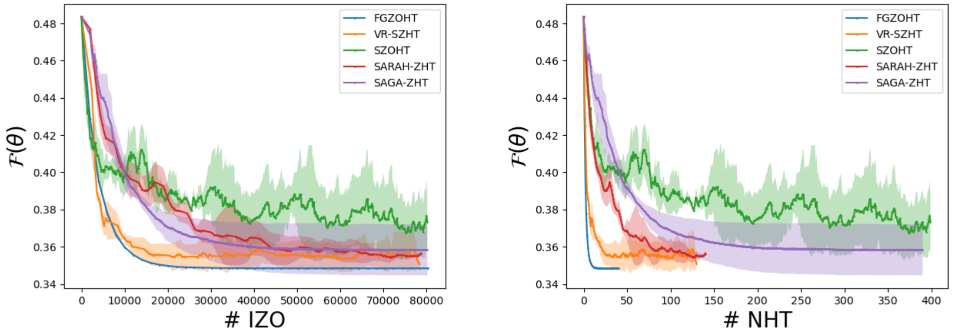


Figure 2.9: #IZO and #NHT on the ridge regression task.

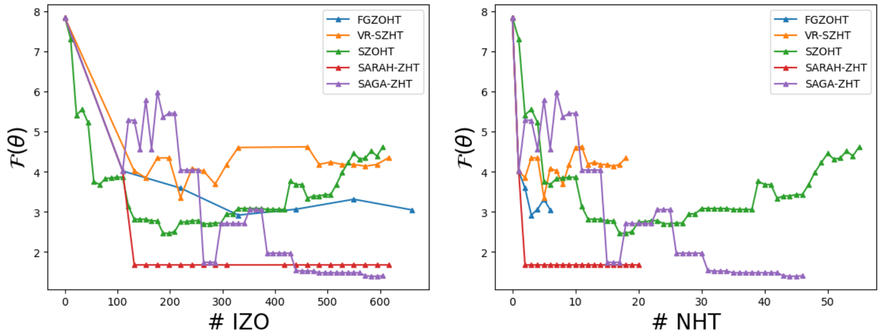
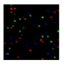
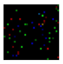
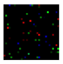
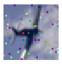
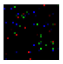
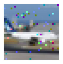
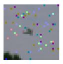
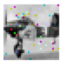
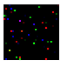
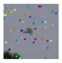
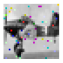


Figure 2.10: #IZO and #NHT on the few pixels adv. attacks (CIFAR-10), for the original class 'airplane'.

2.10.2.2 Few Pixels Universal Adversarial Attacks

Finally, we consider a few-pixel universal adversarial attacks problem. Let some classifier be trained on a dataset of images. We assume that it can only be accessed as a black box, i.e. it only returns the log probabilities of each estimated class, given an input image. This is a typical real-life scenario, where for instance the model can only be accessed through a provider's API. We seek to find a single perturbation $\boldsymbol{\theta} \in \mathbb{R}^d$, to apply to several images at once, (we denote those images by $\boldsymbol{x}_i$, $i = \{1, \ldots, n\}$, and their true label as y_i) to make the predicted class for those images different than their true class. Further discussion on universal perturbations can be found in [50]. In addition, we seek an adversarial perturbation that is sparse, to preserve as much as possible the original image. As is usual in black-box adversarial attacks, we maximize the following Carlini-Wagner

Table 2.2: Comparison of universal adversarial attacks on $n = 10$ images from the CIFAR-10 test-set, from the 'airplane' class. For each algorithm, the leftmost image is the sparse adversarial perturbation applied to each image in the row. ('auto' stands for 'automobile', and 'plane' for 'airplane')

Image ID	3	27	44	90	97	98	111	116	125	153
Original										
FGZOHT 	 plane	 plane	 plane	 **ship**	 **deer**	 plane	 plane	 plane	 **ship**	 **truck**
SZOHT 	 plane	 plane	 plane	 plane	 **deer**	 plane	 **bird**	 **bird**	 **ship**	 **truck**
VR-SZHT 	 plane	 plane	 **auto**	 plane	 **ship**	 plane	 plane	 plane	 **ship**	 **truck**
SAGA-SZHT 	 plane	 **frog**	 plane	 plane	 **deer**	 plane	 plane	 plane	 **ship**	 **ship**
SARAH-SZHT 	 plane	 plane	 **auto**	 **ship**	 **ship**	 plane	 **bird**	 **plane**	 **frog**	 **truck**

loss [34,38], which encourages the prediction from the model to be different from the true class:

$$f_i(\boldsymbol{\theta}) = \max\{F_{y_i}(\text{clip}(\boldsymbol{x}_i + \boldsymbol{\theta})) - \max_{j \neq y_i} F_j(\text{clip}(\boldsymbol{x}_i + \boldsymbol{\theta})), 0\}, \tag{2.120}$$

where $\boldsymbol{x}_i$ is the original i-th image (rescaled to have values in $[-0.5, 0.5]$), of true class y_i, clip denotes the clipping operation into $[-0.5, 0.5]$, $\boldsymbol{\theta}$ is the universal perturbation that we seek to optimize, and each function F_k outputs the log-probability of image $\boldsymbol{x}_i$ being of class k as predicted by the model, for $k \in \{1, .., K\}$, with K the number of classes (similarly to [38,74,95]). Similarly to [95] (Section A.11), we evaluate the algorithm on a dataset of $n = 10$ images from the test-set of the CIFAR-10 dataset [83], of dimensionality $32 \times 32 \times 3 = 3,072$, from the same class 'airplane', which we display in Table 2.2. We take as model F a fixed neural network, already trained on the train-set of CIFAR-10, obtained from the supplementary material of [46]. We set $k = 60$, $\mu = 0.001$, $q = 10$, $s_2 = d = 3,072$, and the number of inner iterations of the variance reduced algorithms to $m = 10$. We check at each iteration the number of IZO, and we stop training if it exceeds 600. Finally, for each algorithm, we grid-search the learning rate η in $\{0.001, 0.005, 0.01, 0.05\}$. The best learning rates (giving the curve which obtained the smallest minimum function value), are respectively: FGZOHT: 0.05, SZOHT: 0.005, VR-SZHT: 0.01, SAGA-SZHT: 0.05, SARAH-SZHT: 0.05. Our experiments are conducted on a workstation of 128 CPU cores. The training curves are presented in Figure 2.10: SAGA-SZHT obtains the lowest function value at the end of the training, followed by SARAH-SZHT. In terms of attack success rate, SARAH-SZHT presents the highest success rate, as it has successfully attacked 7/10 images. We provide further results, on 3 more classes ('ship', 'bird', and 'dog') in the Section, which demonstrate even further the advantage of variance reduction methods in our setting.

2.10.2.3 Extra Experiments on Synthetic Ridge Regression

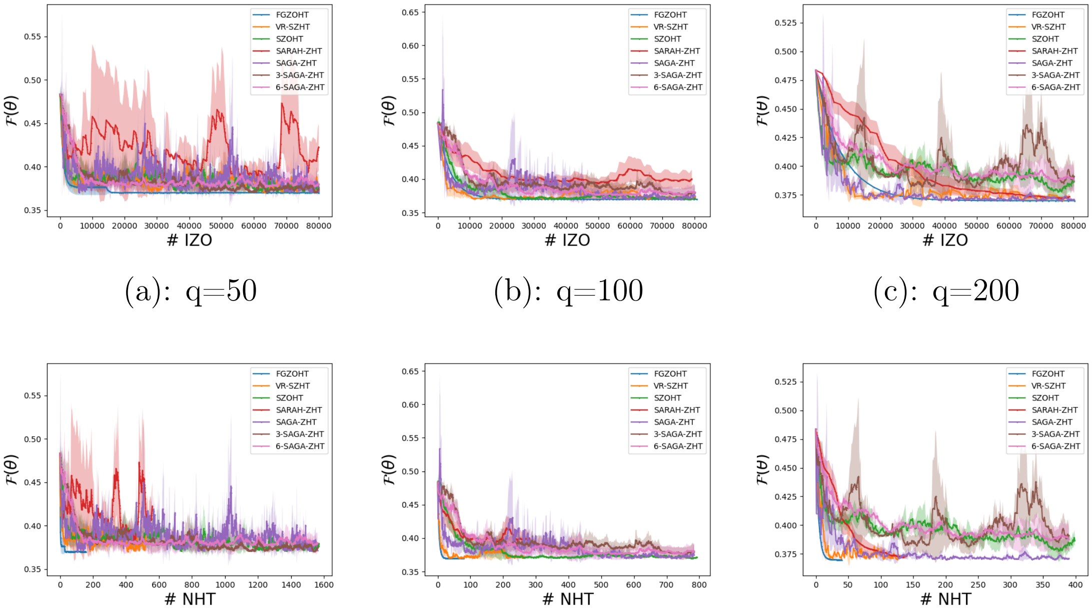
In this section we provide additional curves for the first problem described in section 2.10.2, which we recall here for sake of completeness. We consider a ridge regression problem, where each function f_i is defined as follows:

$$f_i(\boldsymbol{\theta}) = (\boldsymbol{x}_i^\top \boldsymbol{\theta} - y_i)^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2, \tag{2.121}$$

where λ is some regularization parameter.

Experimental Setting. First, as in the main paper, we consider a synthetic dataset: we generate each $\boldsymbol{x}_i$ randomly from a unit norm ball in $\mathbb{R}^d$, and a true random model $\boldsymbol{\theta}^*$ from a normal distribution $\mathcal{N}(0, I_{d\times d})$. Each y_i is defined as $y_i = \boldsymbol{x}_i^\top \boldsymbol{\theta}^*$. We set the constants of the problem as such: $n = 10, d = 5, \lambda = 0.5$. Before training, we pre-process each column by subtracting its mean and dividing it by its empirical standard deviation. We run each algorithm with $\mu = 10^{-4}, s_2 = d = 5$, and for the variance reduced algorithms, we choose $m = 10$. For all algorithms, the learning rate η is found through grid-search in $\{0.005, 0.01, 0.05, 0.1, 0.5\}$, and we keep the η giving the lowest function value (averaged over 3 runs) at the end of training. We stop each algorithm once its number of IZO reaches 80,000. We plot in Figures 2.11, 2.11(g), and 2.12 the mean and standard deviation of the curves for a value of $k = 2$, 3, and 4 respectively.

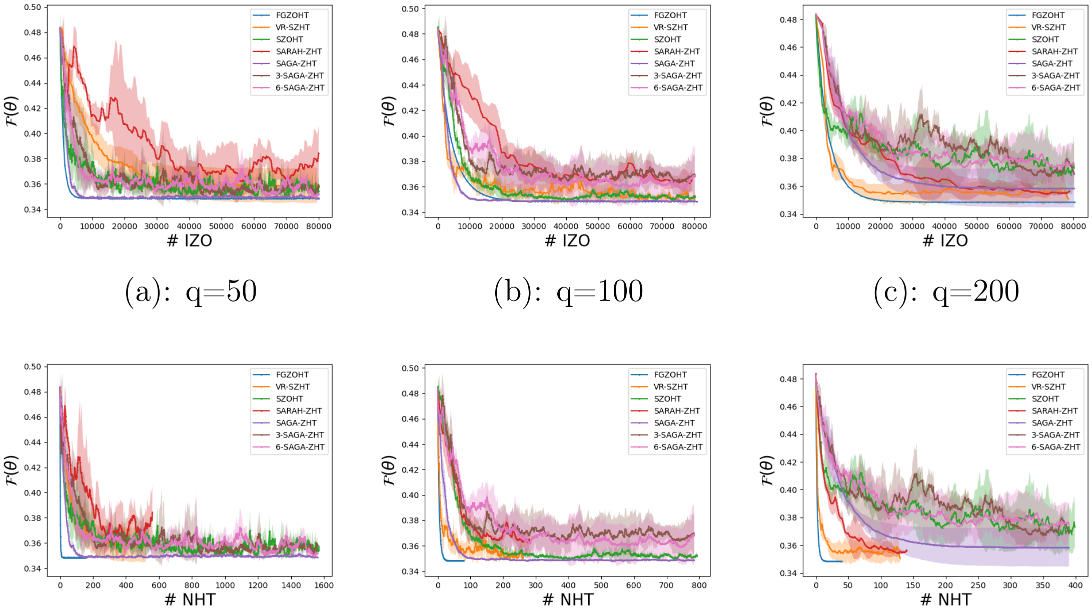
Results and Discussion. We can observe the several phenomena on the Figures 2.11, 2.11(g), and 2.12. First, we can observe that for larger k, the algorithms converge to lower function values (which is natural because optimization is then over a larger set), but also, the algorithms are more stable (for example, SARAH-SZHT converges more easily with $k = 4$ than with $k = 2$), which is due to the hard-thresholding operator being more non-expansive. Then, although a larger number of random directions q may slow down the query complexity (IZO), we observe that it can also stabilize some algorithms that would otherwise be less unstable, such as SARAH-SZHT (which converges better for $q = 200$ than for $q = 50$).



(a): q=50 (b): q=100 (c): q=200

(d): q=50 (e): q=100 (f): q=200

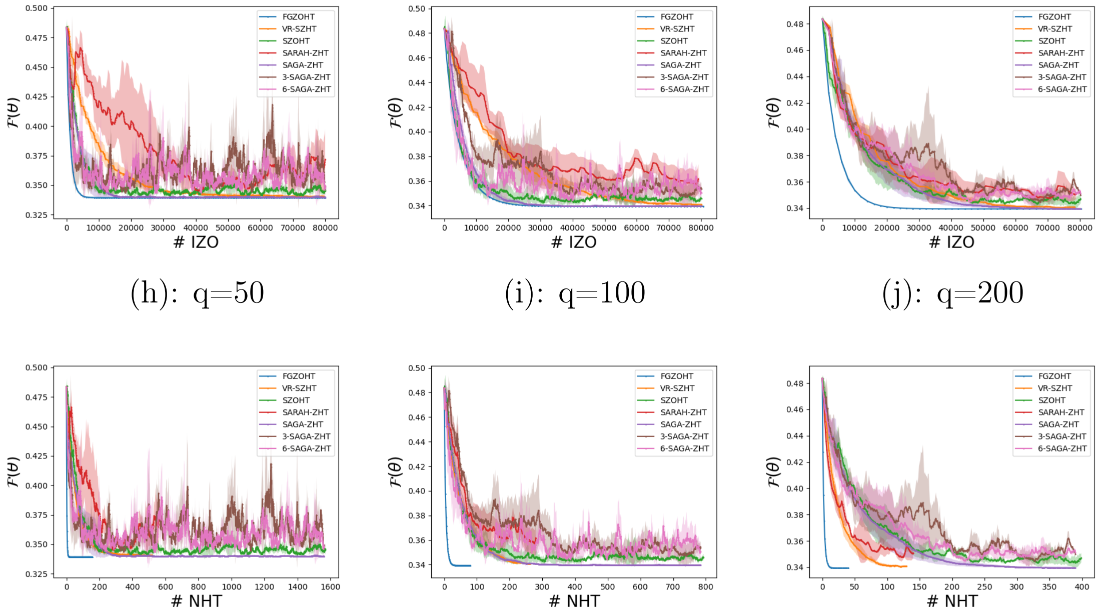
Figure 2.11: #IZO (up) and #NHT (down) on the ridge regression task, synthetic example (k=2).



(a): q=50 (b): q=100 (c): q=200

(d): q=50 (e): q=100 (f): q=200

(g): #IZO (up) and #NHT (down) on the ridge regression task, synthetic example (k=3).



(h): q=50 (i): q=100 (j): q=200

(k): q=50 (l): q=100 (m): q=200

Figure 2.12: #IZO (up) and #NHT (down) on the ridge regression task, synthetic example (k=4).

2.10.2.4 Real Data Ridge Regression

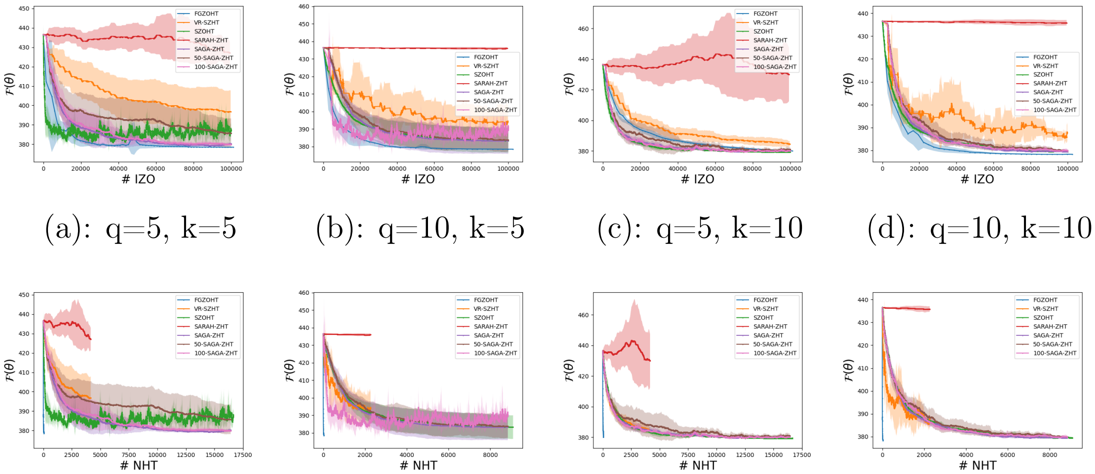
Experimental Setting. Second, we now compare the above algorithms, for the same ridge regression problem as above, but on the following open source real-life datasets (obtained from OpenML [144]), of which a summary is presented in Table 2.3. We take $\lambda = 0.5$. Similarly as above, before training, we pre-process each column by subtracting its mean and dividing it by its empirical standard deviation. We run each algorithm with $\mu = 10^{-4}, s_2 = d$ (where d depends on the dataset), and for the variance reduced algorithms with an inner and outer loop (VR-SZHT and SARAH-SZHT), we choose $m = \lfloor \frac{1}{n}2 \rfloor$. For all algorithms, the learning rate η is found through grid-search in $\{10^{-i},\ i \in \{1, ..., 7\}\}$, and we choose the one giving the lowest function value (averaged over 5 runs) at the end of training. We stop each algorithm once its number of IZO reaches 100,000. We plot the optimization curves (averaged over the 5 runs) for several values of q and k, to study their impact on the convergence.

Results and Discussion. We present our results in Figures 2.13 and 2.14. Those results are consistent with preliminary results on the synthetic dataset from Section 2.10.2.3, namely, that overall, although taking a larger q may worsen the IZO complexity, it can help some algorithms to converge more smoothly, by reducing the error of the zeroth-order estimator. Additionally, we can observe that taking a larger k often helps to achieve smoother convergence. Finally, consistently across experiments, we observe that SARAH-SZHT has difficulties converging: this seems to indicate that SARAH-SZHT may be highly

Table 2.3: Datasets used in the comparison. *Reference:* [51], *Source:* [144], downloaded with `scikit-learn` [122].

Dataset	d	n
BODYFAT$^{(2)}$	14	252
AUTO-PRICE$^{(3)}$	15	159

impacted by the errors introduced by the zeroth-order estimator. SARAH-SZHT could potentially be improved by a more careful choice of the number of inner iterations, and/or by running SARAH+, which is an adaptive version of SARAH [116], which we leave for future work.



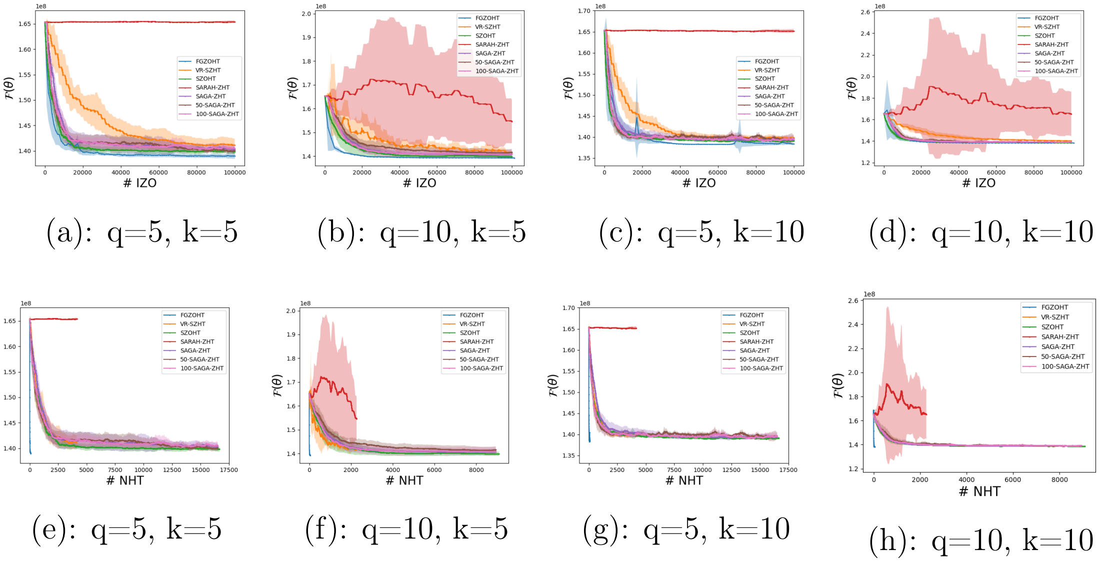
(a): q=5, k=5 (b): q=10, k=5 (c): q=5, k=10 (d): q=10, k=10

(e): q=5, k=5 (f): q=10, k=5 (g): q=5, k=10 (h): q=10, k=10

Figure 2.13: #IZO (up) and #NHT (down) on the ridge regression task, `bodyfat` dataset.

2.10.2.5 Additional Results for Universal Adversarial Attacks

In this section , we provide additional results for the universal adversarial attacks setting from our Experiments Section, for the 3 additional classes: 'ship', 'bird', and 'dog'. As we can observe in Figures 2.16 , 2.15, and 2.17 below respectively, in most of such cases, there is a variance-reduced algorithm which can achieve better performance than the vanilla zeroth-order hard-thresholding algorithms, (for instance, SARAH-ZHT in Figure 2.15, and SAGA-ZHT in Figure 2.16) which demonstrates the applicability of such algorithms. Correspondingly, this can also be verified by observing the misclassification success in Table 2.5, 2.4 and 2.6: even if a smaller value for the cost does not necessarily imply a strictly higher attack success rate, still, overall, more successful universal attacks also have a higher success rate of attack.



(a): q=5, k=5 (b): q=10, k=5 (c): q=5, k=10 (d): q=10, k=10

(e): q=5, k=5 (f): q=10, k=5 (g): q=5, k=10 (h): q=10, k=10

Figure 2.14: #IZO (up) and #NHT (down) on the ridge regression task, `autoprice` dataset.

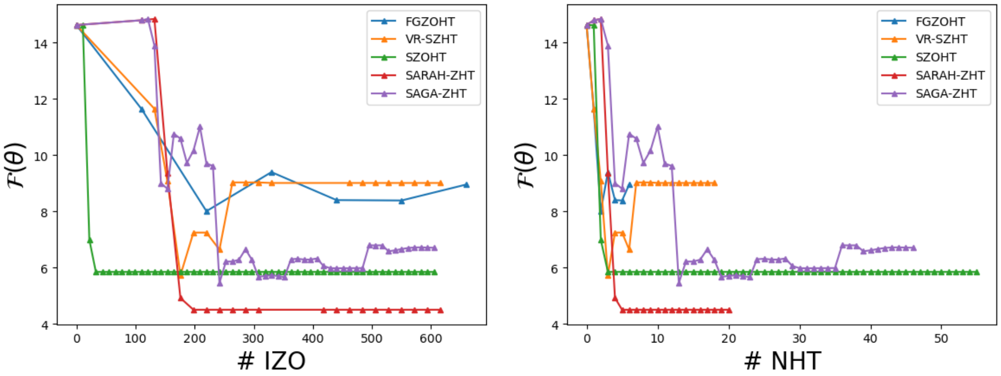
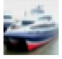
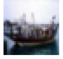
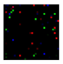
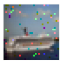
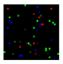
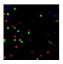
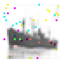
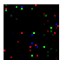
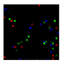


Figure 2.15: #IZO and #NHT on the few pixels adversarial attacks task (CIFAR-10), for the original class 'ship'.

Table 2.4: Comparison of universal adversarial attacks on $n = 10$ images from the CIFAR-10 test-set, from the 'ship' class. For each algorithm, the leftmost image is the sparse adversarial perturbation applied to each image in the row. ('auto' stands for 'automobile', and 'plane' for 'airplane')

Image ID	1	15	18	51	54	55	72	73	79	80
Original										
FGZOHT 	 ship	 **frog**	 ship	 ship	 ship	 ship	 ship	 ship	 ship	 ship
SZOHT 	 ship	 **frog**	 ship	 **deer**	 ship	 **deer**	 ship	 ship	 ship	 **plane**
VR-SZHT 	 ship	 **frog**	 **truck**	 **plane**	 ship	 ship	 ship	 ship	 ship	 **auto**
SAGA-SZHT 	 ship	 **frog**	 **truck**	 ship	 **plane**	 **deer**	 ship	 ship	 ship	 ship
SARAH-SZHT 	 ship	 **frog**	 **auto**	 **auto**	 ship	 ship	 ship	 ship	 ship	 ship

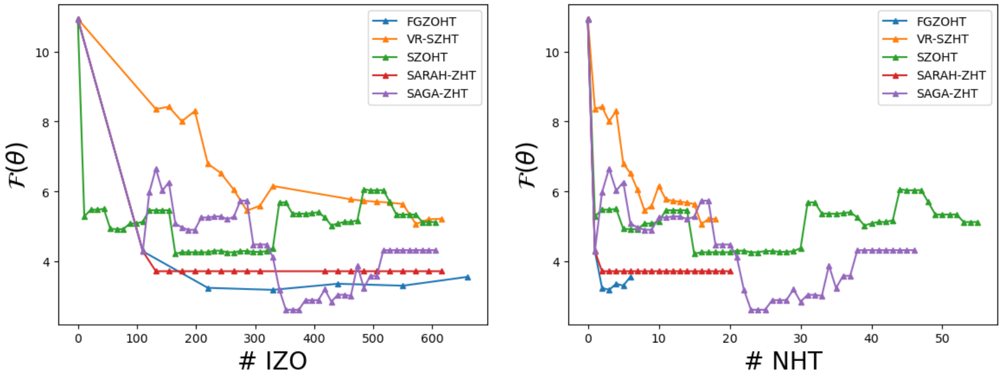
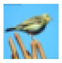
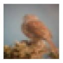
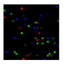
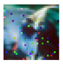
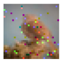
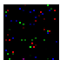
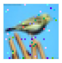
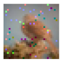
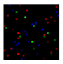
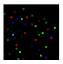
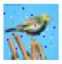
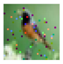
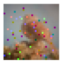
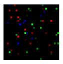
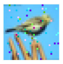
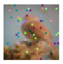


Figure 2.16: #IZO and #NHT on the few pixels adversarial attacks task (CIFAR-10), for the original class 'bird'.

Table 2.5: Comparison of universal adversarial attacks on $n = 10$ images from the CIFAR-10 test-set, from the 'bird' class. For each algorithm, the leftmost image is the sparse adversarial perturbation applied to each image in the row.

Image ID	65	67	70	75	86	113	123	129	138	149
Original										
FGZOHT 	 **frog**	 **bird**	 **deer**	 **dog**	 **deer**	 bird	 bird	 **deer**	 **frog**	 bird
SZOHT 	 **frog**	 bird	 bird	 bird	 bird	 bird	 bird	 **ship**	 bird	 bird
VR-SZHT 	 **frog**	 bird	 bird	 bird	 bird	 **frog**	 bird	 **ship**	 bird	 bird
SAGA-SZHT 	 **frog**	 bird	 **deer**	 **dog**	 bird	 bird	 **cat**	 **deer**	 **frog**	 bird
SARAH-SZHT 	 **frog**	 bird	 **frog**	 bird	 bird	 **frog**	 bird	 bird	 **frog**	 bird

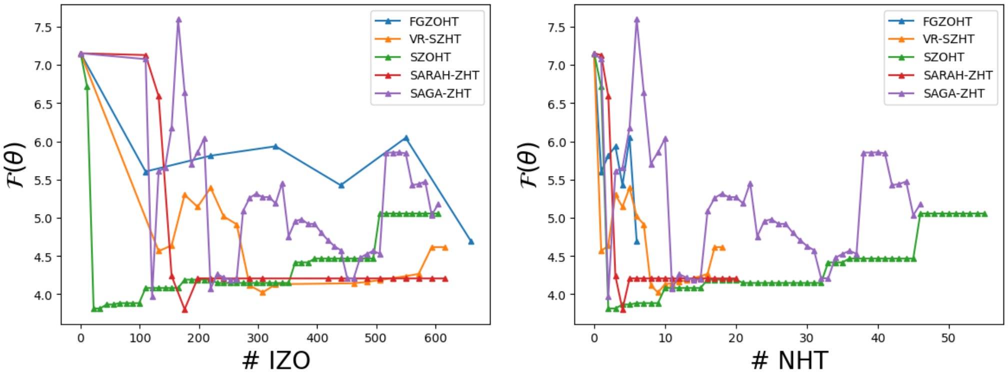
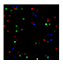
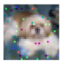
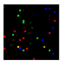
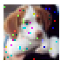
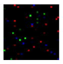
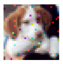
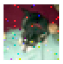
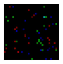
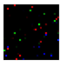
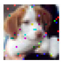


Figure 2.17: #IZO and #NHT on the few pixels adversarial attacks task (CIFAR-10), for the original class 'dog'.

Table 2.6: Comparison of universal adversarial attacks on $n = 10$ images from the CIFAR-10 test-set, from the 'dog' class. For each algorithm, the leftmost image is the sparse adversarial perturbation applied to each image in the row.

Image ID	12	16	31	33	39	42	101	128	141	148
Original										
FGZOHT										
										
	bird	dog	**deer**	dog	dog	**cat**	**cat**	**cat**	dog	dog
SZOHT										
										
	frog	dog	**bird**	dog	dog	**cat**	dog	**cat**	dog	dog
VR-SZHT										
										
	dog	dog	**bird**	dog	dog	**horse**	dog	dog	dog	dog
SAGA-SZHT										
										
	dog	dog	dog	dog	dog	**horse**	dog	**cat**	dog	dog
SARAH-SZHT										
										
	deer	dog	dog	**frog**	dog	**cat**	**frog**	dog	dog	dog

2.11 ZOHT Extension: Discontinuous and Non-convex Case

We now turn to a variant of ZOHT which still does zeroth-order hard-thresholding (although the random directions are sampled from a Gaussian distribution), but which this time considers another setting for the convergence proofs: the setting of bounded, discontinuous, and non-convex functions. This section is based on our co-authored paper *Hard-Thresholding Meets Evolution Strategies in Reinforcement Learning*, currently under review at IJCAI 2024. In this section, we mostly describe the theoretical part from the paper, which corresponds to our main contribution amongst the collaboration. As mentioned, we will analyze the convergence of a hard-thresholding zeroth-order algorithm, in the case where the function to be optimized is a potentially discontinuous and non-convex function. Such case can occur for instance in a reinforcement learning setting. In such a case, we will prove local convergence of our algorithm. In this section, we will use a slightly different terminology, closer to the reinforcement learning literature, and call the optimization algorithm which optimizes a cumulative reward based on a zeroth-order estimator of the gradient, an *Evolutionary Strategy* (ES).

2.11.1 Preliminaries

Markov Decision Process. We are concerned with the reinforcement learning problem, where our objective is to optimize a policy, denoted by $\pi_{\boldsymbol{\theta}}$, parameterized by $\boldsymbol{\theta}$. This policy is defined over a Markov decision process represented by $M = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, d_0, \mathcal{R}, \gamma \rangle$. For each episode, the initial state $\boldsymbol{s}_0$ is sampled from the distribution d_0. At each time step t, given an observation $\boldsymbol{s_t} \in \mathcal{S}$, the policy determines an action $\boldsymbol{a_t} \in \mathcal{A}$, which then results in an immediate reward $r(\boldsymbol{s_t}, \boldsymbol{a_t}) \in \mathcal{R}$. Subsequently, the system transitions to a new observation $\boldsymbol{s_{t+1}}$ in accordance with the dynamics $\mathcal{T}$. The resulting trajectory can be presented as $\boldsymbol{\tau} = \{(\boldsymbol{s}_t, \boldsymbol{a}_t, r, \boldsymbol{s}_{t+1})\}$. In order to balance the trade-off between immediate rewards and long-term rewards, the discount factor γ is introduced.

2.11.2 Objective Function

Our objective is to maximize the *fitness score* achieved by the policy while minimizing the *impact* induced from the task-irrelevant features. We hypothesize that employing a sparse policy can effectively manage these redundant observations.

Fitness Score. The performance of the policy can be quantified by the *fitness function*, which is defined as the expected sum of rewards over its rollout trajectories:

$$F(\boldsymbol{\theta}) := \mathbb{E}_{\tau \sim d_0, \pi_{\boldsymbol{\theta}}, \mathcal{T}} f_\tau(\boldsymbol{\theta}), \text{ with } f_\tau(\boldsymbol{\theta}) := \sum_{t=0}^{|\tau|} r(\boldsymbol{s_t}, \boldsymbol{a_t}) \tag{2.122}$$

It's important to note that, in this context, the discount factor γ is set to 1. This is in contrast to traditional RL settings where it often assumes values such as 0.99 or 0.9. Another characteristic is that the fitness function can be discontinuous *w.r.t.* the policy parameters due to the randomness in environments and the complex reward function.

ℓ_0-Constrained Optimization. We propose mitigating the impact of task-irrelevant features through a sparse policy, under the premise that sparsity can effectively filter out irrelevant information present in inputs. Formally, our objective is to improve a policy while also constraining its complexity, i.e., the ℓ_0 constrained optimization, with $\|\cdot\|_0$ denotes the ℓ_0 (pseudo-)norm (number of non-zero components of a vector):

$$\max_{\boldsymbol{\theta}} F(\boldsymbol{\theta}) \quad s.t. \quad \|\boldsymbol{\theta}\|_0 \leq k \tag{2.123}$$

Why the ℓ_0 Constraint? In our context, where only a small subset of observations is task-relevant, irrelevant features can significantly degrade performance. ℓ_0-constrained optimization directly enforces a constraint on the ℓ_0 norm of the learned parameter vector, ensuring the sparsity of the resulting model, alluring for feature selection tasks. Unlike ℓ_1-constrained optimization, which promotes sparsity but does not guarantee exact zero values, ℓ_0-constrained optimization offers precise control over sparsity by allowing certain model parameters to be set exactly to zero. This capability not only enhances model interpretability but also makes it well-suited for our setting, i.e., decision-making with irrelevant observations.

2.11.3 Our Proposal: NESHT

We introduce NESHT, a solution for decision-making problems involving both task-relevant and irrelevant features. While NES and the Hard-Thresholding operator are not novel concepts individually, their compatibility when used together may raise questions. To be self-contained, we now provide brief descriptions of each.

NES We employ the competitive NES algorithm, to optimize the policy, with the following gradient estimator:

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I)} F(\boldsymbol{\theta} + \sigma \boldsymbol{\epsilon}) = \frac{1}{\sigma} \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d \times d})} F(\boldsymbol{\theta} + \sigma \boldsymbol{\epsilon}) \boldsymbol{\epsilon}, \tag{2.124}$$

where $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d \times d})$ denotes the unit-variance centered Gaussian in $\mathbb{R}^d$. In NES, the gradient is approximated through sampling and serves as an approximation, bypassing challenges with non-differentiable functions or exploding gradients. For the derivation about Equation (2.124), please refer to Section 2.11.5.1.

Hard-Thresholding operator. To achieve the ℓ_0-constrained optimization described in Equation (2.123), we introduce the hard-thresholding operator into NES. It truncates the parameter vector, retaining only k components with the most significant absolute magnitudes, represented as $\text{trunc}(\boldsymbol{\theta}, k)$, or, more succinctly, as $\text{trunc}(\boldsymbol{\theta})$. While incorporating HT into NES is straightforward, the compatibility between HT and NES remains an open question.

Compatibility concerns. To establish the convergence of NESHT, it is essential to demonstrate the convergence of the hard-thresholding algorithm for non-convex and discontinuous F, with a gradient estimated as in equation 2.124 via the NES algorithm. In the literature, [152] proved the convergence of stochastic algorithms in the case of non-convex objective functions F, for a non-convex proximal term which can be taken as the indicator function of the set of all k-sparse vectors (i.e. the ℓ_0 pseudo-ball). This proof of convergence applies to stochastic hard-thresholding algorithms. However, their analysis assumes Lipschitz-smoothness of F and considers a general stochastic estimator of the

Algorithm 6: NES with Hard-Thresholding

```
Input : $\alpha$ - Learning rate, $\boldsymbol{\theta_0}$ - Initial policy parameters in $\mathbb{R}^d$, $n$ - Population size,
        $N$ - Number of rollouts collected for each agent, $\sigma$ - Noise standard
        deviation, $k$ - Number of parameters to be kept.
for $t = 0, 1, 2, ...T - 1$ do
    for $i = 1, \ ..., \ n$ do
        Sample a Gaussian perturbation $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d\times d})$ ;
        for $j = 1, \ ..., \ N$ do
            Sample a rollout $\tau_j^{\boldsymbol{\epsilon}_i}$;
            Compute returns $f_{\tau_j^{\boldsymbol{\epsilon}_i}}(\boldsymbol{\theta}_t + \sigma\boldsymbol{\epsilon_i})$;
        end
    end
    Set $\boldsymbol{\theta}_{t+\frac{1}{2}} \leftarrow \boldsymbol{\theta}_t + \frac{\alpha}{nN\sigma}\sum_{i=1}^{n}\sum_{j=1}^{N} f\tau_j^{\boldsymbol{\epsilon}_i}(\boldsymbol{\theta}_t + \sigma\boldsymbol{\epsilon_i})\boldsymbol{\epsilon}_i$;
    Truncate the parameters: $\boldsymbol{\theta}_{t+1} \leftarrow \text{trunc}(\boldsymbol{\theta}_{t+\frac{1}{2}}, k)$;
end
```

gradient. Therefore, it does not account for the specific errors introduced by the gradient estimator from equation 2.124. More recently, the work of [103], analyzes the convergence of zeroth-order methods (similar to evolutionary strategies) for a Lipschitz-continuous and non-convex function F. However, in our case, F is discontinuous in general. Thus, to the best of our knowledge, the convergence of evolutionary strategies in such setting remains an open question. In the next section, we address this question by demonstrating that, under mild assumptions, proper convergence of Algorithm 6 is guaranteed.

2.11.4 Convergence Analysis

The integration of NES with HT is detailed in Algorithm 6, where the hard-thresholding operator is applied to the learned parameters after each update. In this section, we provide a proof of convergence for NES combined with Hard-Thresholding, i.e., our NESHT, addressing the compatibility concern. Additionally, we would like to highlight that our analysis can also cover the case where no hard-thresholding operator is used (it only suffices to take the proximal term r in our proof of Theorem 2 in Section 2.11.5.4 to be the constant zero): to our knowledge, such a proof of convergence for NES for general discontinuous functions F (which correspond to a realistic reinforcement learning setting) is the first in the literature, and we hope that such a result, as well as the subsequent remarks and discussions on the influence of each parameter on the convergence rate (bound on the expected reward B, dimension d, etc.) can be of interest to the NES community.

2.11.4.1 Assumptions

To proceed with the proof of convergence of NESHT, we will need the following assumptions below.

Assumption 5 (Boundedness of F)**.** *The fitness function F is bounded on its domain, that is, there exists a universal constant $B > 0$ such that:*

$$\forall \boldsymbol{\theta} \in \mathbb{R}^d : |F(\boldsymbol{\theta})| \leq B \tag{2.125}$$

Remark 6. *$F(\boldsymbol{\theta})$ represents the expected rewards obtained by executing policy $\pi_{\boldsymbol{\theta}}$. The boundedness assumption is typically reasonable since immediate rewards do not tend to infinity, and evaluation trajectories always have finite lengths. Importantly, this assumption remains valid even when dealing with task-irrelevant features.*

Additionally, we will need the following assumption on the variance of the cumulative reward, for a given parameter vector $\boldsymbol{\theta}$.

Assumption 6 (Bounded variance of f_τ)**.** *We posit the existence of a universal constant $C > 0$ such that the variance of the cumulative reward for any $\boldsymbol{\theta} \in \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_{\frac{1}{2}}, ..., \boldsymbol{\theta}_{T-\frac{1}{2}}, \boldsymbol{\theta}_T\}$ is bounded by C, i.e.:*

$$\mathbb{E}_\tau \left[|f_\tau(\boldsymbol{\theta}) - F(\boldsymbol{\theta})|^2 \right] \leq C.$$

Remark 7. *Assumption 6 reflects the inherent randomness from both the policy, whether it is deterministic or stochastic, and the environment, which introduces randomness through factors such as the dynamics $\mathcal{T}$, the reward function $r(s, a)$, and the initial distribution of states d_0. Also, please note that if the reward and the episode length are limited, as is usually the case in RL, then Assumptions 5 and 6 are satisfied. An observant reader may notice that the inclusion of task-irrelevant features unavoidably leads to an increase in the constant C due to the introduction of randomness. As we will see later, this increase hampers the convergence of NES algorithms.*

2.11.4.2 Smoothness

Since F can be discontinuous in general, maximizing F directly is impossible with evolutionary strategies. For instance if F is Dirac-like, such as $F(\boldsymbol{\theta}) = \begin{cases} 1 \text{ if } \boldsymbol{\theta} = \mathbf{0} \\ 0 \text{ otherwise} \end{cases}$, the probability (for a given $\boldsymbol{\theta}$), to successfully sample an $\boldsymbol{\epsilon}$ such that $F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon}) = 1$ is actually zero, which means the parameters will be updated with probability zero. However, we can instead analyze the convergence of a smoothed version of F, F_σ, defined below:

$$F_\sigma(\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d \times d})} F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})$$

Note that F_σ converges towards F for small σ in terms of *eh-convergence*, as described in Theorem 3.2 from [156]. The first step, to derive the convergence rate of our algorithm with F_σ, is to prove that F_σ is smooth, and to derive its smoothness constant, which we then use in a proof framework similar to [152].

Lemma 4. *Under Assumption 5, F_σ is Lipschitz-smooth (i.e. its gradient is Lipschitz-continuous), with a smoothness constant $L = \frac{(d+1)B}{\sigma^2}$, that is, such L verifies:*

$$\forall \boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in (\mathbb{R}^d)^2 : \|\nabla F_\sigma(\boldsymbol{\theta}_1) - \nabla F_\sigma(\boldsymbol{\theta}_2)\| \leq L\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\| \tag{2.126}$$

Proof. Proof in Section 2.11.5.3. □

For discontinuous functions F, the fact that F_σ is smooth was already known before in the literature (see e.g. [52]). However, such works did not provide an explicit formula for the smoothness constant L. Here, for the first time in the literature (to the best of our knowledge), using the boundedness assumption on F, we could derive an explicit formula for the smoothness constant L.

One can therefore see that L is proportional to both the bound of the fitness function, B, and the dimension of the policy parameters, d, while being inversely proportional to the variance σ^2. In Section 4.4, we will observe the role of such smoothness constant L: the smaller it is, the faster the NES algorithm will converge.

2.11.4.3 Error of the gradient estimator

We now consider the gradient estimator with a general population of n random perturbations, and a number of rollouts of N for each perturbation. More precisely, assume that we sample n random directions $\{\boldsymbol{\epsilon}_i\}_{i=1}^n := \{\boldsymbol{\epsilon}_1, ..., \boldsymbol{\epsilon}_n\}$ independently and identically distributed, and that for each of these random directions $\boldsymbol{\epsilon}_i$, we sample we sample N rollouts $\{\tau_j^{\boldsymbol{\epsilon}_i}\}_{j=1}^N := \{\tau_1^{\boldsymbol{\epsilon}_i}, .., \tau_N^{\boldsymbol{\epsilon}_i}\}$ independently and identically distributed, to obtain a final collection of rollouts $\{\{\tau_j^{\boldsymbol{\epsilon}_i}\}_{j=1}^N\}_{i=1}^n$, and to get $N \times n$ gradient estimators $\hat{g}_{\sigma,\boldsymbol{\epsilon}_i,\tau_j^{\boldsymbol{\epsilon}_i}}, (i,j) \in [n] \times [N]$ defined below:

$$\hat{g}_{\sigma,\boldsymbol{\epsilon}_i,\tau_j^{\boldsymbol{\epsilon}_i}}(\boldsymbol{\theta}) := \frac{1}{\sigma} f_{\tau_j^{\boldsymbol{\epsilon}_i}}(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon}_i)\boldsymbol{\epsilon}_i \tag{2.127}$$

which we aggregate in the following estimator:

$$\bar{g}_{\sigma,\{\boldsymbol{\epsilon}_i\}_{i=1}^n,\{\{\tau_j^{\boldsymbol{\epsilon}_i}\}_{j=1}^N\}_{i=1}^n}(\boldsymbol{\theta}) := \frac{1}{nN}\sum_{i=1}^n\sum_{j=1}^N \hat{g}_{\sigma,\boldsymbol{\epsilon}_i,\tau_j^{\boldsymbol{\epsilon}_i}}(\boldsymbol{\theta}) \tag{2.128}$$

Lemma 5. *Under Assumptions 5 and 6, the estimator above is an unbiased estimate of the gradient of the smoothed function F, and its variance is bounded, more precisely, for any $\boldsymbol{\theta} \in \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_{\frac{1}{2}}, ..., \boldsymbol{\theta}_{T-\frac{1}{2}}, \boldsymbol{\theta}_T\}$:*

$$\mathbb{E}\bar{g}_{\sigma,\{\boldsymbol{\epsilon}_i\}_{i=1}^n,\{\{\tau_j^{\boldsymbol{\epsilon}_i}\}_{j=1}^N\}_{i=1}^n}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} F_\sigma(\boldsymbol{\theta}) \tag{2.129}$$

$$\mathbb{E}\|\bar{g}_{\sigma,\{\boldsymbol{\epsilon}_i\}_{i=1}^n,\{\{\tau_j^{\boldsymbol{\epsilon}_i}\}_{j=1}^N\}_{i=1}^n}(\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}} F_\sigma(\boldsymbol{\theta})\|^2 \leq \frac{Cd}{N\sigma^2} + \frac{dB^2}{n\sigma^2} \tag{2.130}$$

Proof. See Section 2.11.5.2. We begin by examining the unbiasedness (using a standard proof) and variance (using a novel proof up to our knowledge) of the gradient estimator for a single perturbation, i.e., $\hat{g}_{\sigma,\boldsymbol{\epsilon}_i,\tau_j^{\boldsymbol{\epsilon}_i}}$. We then generalize our results to account for multiple perturbations (n) and rollouts (N). □

Advantages of NESHT: Reduction in Constant C. We present here a formal explanation for the superiority of NESHT over NES in the lens of constant C. Thanks to hard-thresholding, along training, $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_{t+\frac{1}{2}}$ remain in the space of k-sparse vectors (up to small perturbations $\sigma\boldsymbol{\epsilon}$), whereas they could live anywhere in $\mathbb{R}^d$ in the case of NES. Based on the hypothesis that the hard-thresholding operation effectively selects relevant features, NESHT can successfully mitigate the impact of irrelevant features and reduces the value of C. To illustrate this, one can consider the following scenario.

Example 1. *Consider a one-step decision-making experiment, with linear policy, and fitness score given as: $f_\tau(\boldsymbol{\theta}) := \boldsymbol{x}^\top(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$, where $\boldsymbol{\theta}^*$ is a k-sparse vector, with $S \subseteq [d]$ being the set of coordinates of its non-zero components, i.e., the relevant features. In addition, $\boldsymbol{x}$ is the input state, which we assume follows a normal distribution $\mathcal{N}(\mathbf{0}, \sigma \boldsymbol{I}_{d\times d})$ for $\sigma > 0$ ($\boldsymbol{I}_{d\times d}$ denoting the identity). We then have, for any bounded policy $\boldsymbol{\theta} \in [-1,1]^d$:*

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x}}|f_\tau(\boldsymbol{\theta}) - F(\boldsymbol{\theta})|^2 &= \mathbb{E}_{\boldsymbol{x}}(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top \boldsymbol{x}\boldsymbol{x}^\top(\boldsymbol{\theta} - \boldsymbol{\theta}^*) \\ &= (\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top \sigma^2 \boldsymbol{I}_{d\times d}(\boldsymbol{\theta} - \boldsymbol{\theta}^*) = \sigma^2\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2 \end{aligned} \tag{2.131}$$

Therefore, if there are many irrelevant components present (i.e. $|[d] \setminus S|$ is large), the episode-wise variance of f_τ (and its bound C) will be higher when $\boldsymbol{\theta}$ is dense (proportionally to σ^2). As established in Lemma 5, the proper convergence of NESHT depends on this variance. The application of a hard-thresholding operator explicitly filters out some of the noisy features, introducing a bias that steers the policy towards making decisions exclusively based on sparse observations. This reduces the variance and ensures better convergence to the optimal policy.

In practical terms, given a fixed interaction budget for n and N, the variance of the gradient estimator may be too high for vanilla NES, causing it to fail to converge to the optimal policy. However, with the reduced variance of the gradient estimator in NESHT, as described above, convergence of the parameters $\boldsymbol{\theta}$ to a stationary point of the fitness function F can be successfully ensured, as stated in Theorem 2.

2.11.4.4 Convergence Rate

Equipped with Lemmas 4 and 5, we can now prove the convergence of Algorithm 6, following for the most part the framework of [152] for stochastic gradient descent with a non-convex function and a non-convex non-smooth proximal term, but plugging into it our novel bounds for (i) the smoothness constant of F_σ and (ii) the variance of the gradient estimator $\bar{g}_{\sigma,\{\boldsymbol{\epsilon}_i\}_{i=1}^n,\{\{\tau_j^{\boldsymbol{\epsilon}_i}\}_{j=1}^N\}_{i=1}^n}(\boldsymbol{\theta})$, under our specific assumption of boundedness of F. Because of such non-convex and non-smooth optimization problem, convergence is proven in terms of the expected distance of the Fréchet sub-differential $\hat{\partial}(-F_\sigma(\boldsymbol{\theta}) + \mathbb{1}_{\ell_0(k)}(\boldsymbol{\theta}_T))$ to zero [128], where $\mathbb{1}_{\ell_0(k)}$ denotes the indicator function of the ℓ_0 constraint, i.e. $\mathbb{1}_{\ell_0(k)}(\boldsymbol{\theta}) = \begin{cases} 0 \text{ if } \boldsymbol{\theta} \text{ is } k\text{-sparse} \\ +\infty \text{ otherwise} \end{cases}$. Note that this is the standard way to define stationary points for non-smooth regularizers (such as sparsity constraints) (see e.g. Thm. 2 in [152] or Thm. 3 in [48]).

Theorem 2. *Under Assumption 5 and 6, run Algorithm 6, with* $\alpha = \frac{c}{L}\left(0 < c < \frac{1}{2}\right)$*, a number of iterations* $T = 2c_2 B / \left(\alpha\epsilon^2\right)$ *and* $N \geq \frac{4c_1 dC}{\sigma^2\epsilon^2}$ *and* $n \geq \frac{4c_1 dB^2}{\sigma^2\epsilon^2}$ *for* $t = 0, \ldots, T-1$*, then the output* $\boldsymbol{\theta}_T$ *of Algorithm 6 satisfies*

$$\mathbb{E}\left[\operatorname{dist}\left(\mathbf{0}, \hat{\partial}\left(-F_\sigma\left(\boldsymbol{\theta}_T\right) + \mathbb{1}_{\ell_0(k)}(\boldsymbol{\theta}_T)\right)\right)\right] \leq \epsilon,$$

where $c_1 = \frac{2c(1-2c)+2}{c(1-2c)}$*, and* $c_2 = \frac{12-8c}{1-2c}$*, and where* $\operatorname{dist}(\boldsymbol{z}, S)$ *is the distance of a set* S *to a point* $\boldsymbol{z}$*, defined as the minimal Euclidean distance of any point in* S *to* $\boldsymbol{z}$*. In particular in order to have* $\mathbb{E}\left[\operatorname{dist}\left(0, \hat{\partial}(-F_\sigma(\boldsymbol{\theta}) + \mathbb{1}_{\ell_0(k)}(\boldsymbol{\theta}_T))\right)\right] \leq \epsilon$*, that is, in order to ensure convergence to a stationary point, it suffices to set* $T = O\left(1/\epsilon^2\right)$*.*

Proof. Proof in Section 2.11.5.4. □

Remark 8. *As per Theorem 2, we can see that a large smoothing radius* σ *will ease convergence, as it allows one to evaluate fewer random perturbations and rollouts. However, the counterpart is that the function optimized* F_σ *may be further away from the true function* F*.*

Remark 9 (Overall complexity)**.** *From Theorem 2, to ensure convergence to a stationary point up to tolerance* ϵ*, we need to take* $N = O(\frac{dC}{\sigma^2\epsilon^2})$*,* $n = O(\frac{dB^2}{\sigma^2\epsilon^2})$ *, and* $T = O(\frac{B}{\alpha\epsilon^2}) \overset{(a)}{=} O(\frac{BL}{\epsilon^2}) \overset{(b)}{=} O(\frac{B^2 d}{\epsilon^2\sigma^2})$*, where (a) follows from the definition of* α *from Theorem 2, and (b) follows from Lemma 4. Therefore, the overall number of episodes needed to ensure convergence is* $TNn = O(\frac{d^3 B^4 C}{\sigma^6\epsilon^6})$*. Note however that if one has access to a massively parallel device able to run in parallel* Nn *simulations, which is very common in RL settings (e.g. as in [130]), the time complexity of the whole optimization process is simply* $T = O(\frac{B^2 d}{\epsilon^2\sigma^2})$*.*

2.11.5 Proofs of the Main Results

2.11.5.1 NES Gradients

OpenAI NES [130] approximates the gradient with the following estimator:

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_{d\times d})}F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon}) = \frac{1}{\sigma}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_{d\times d})}\{F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon}\}, \tag{2.132}$$

with $\boldsymbol{\theta}$ for the learned policy's parameters, $\boldsymbol{\epsilon}$ for the Gaussian noise on the parameter vector, and σ to control the standard deviation.

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_{d\times d})}F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon}) &= \nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\theta},\sigma^2 I)}F(\boldsymbol{x}) \\
&= \nabla_{\boldsymbol{\theta}}\int_{\boldsymbol{x}} P(\boldsymbol{x}|\boldsymbol{\theta}, \sigma^2 I)F(\boldsymbol{x})d\boldsymbol{x} \\
&= \int_{\boldsymbol{x}} \nabla_{\boldsymbol{\theta}}P(\boldsymbol{x}|\boldsymbol{\theta}, \sigma^2 I)F(\boldsymbol{x})d\boldsymbol{x} \qquad \text{(Leibniz integral rule)} \\
&= \int_{\boldsymbol{x}} P(\boldsymbol{x}|\boldsymbol{\theta}, \sigma^2 I)\nabla_{\boldsymbol{\theta}}\log\left[P(\boldsymbol{x}|\boldsymbol{\theta}, \sigma^2 I)\right]F(\boldsymbol{x})d\boldsymbol{x} \qquad \Big(\text{log derivative trick}\Big)
\end{aligned}$$

$$= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\theta},\sigma^2 I)}\big\{F(\boldsymbol{x})\nabla_\theta\big(-\frac{\|\boldsymbol{x}-\boldsymbol{\theta}\|_2^2}{2\sigma^2}\big)\big\} \qquad \text{(Gaussian P.D.F.)}$$

$$= \frac{1}{\sigma}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_{d\times d})}\big\{F(\boldsymbol{\theta}+\sigma\boldsymbol{\epsilon})\cdot\boldsymbol{\epsilon}\big\} \quad \big(\boldsymbol{x}\leftarrow\boldsymbol{\theta}+\sigma\boldsymbol{\epsilon}\big) \tag{2.133}$$

2.11.5.2 Proof of Lemma 5

Bias and Variance of the Single Perturbation, Full Expected Policy Gradient Estimator. Before deriving the error of the full gradient estimator (i.e. averaged over both the n perturbations, and the N rollouts), we first provide the bias and variance of the gradient estimator for a single random perturbation $\boldsymbol{\epsilon}$, defined below as:

$$\hat{g}_{\sigma,\boldsymbol{\epsilon}}(\boldsymbol{\theta}) = \frac{1}{\sigma}F(\boldsymbol{\theta}+\sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon}. \tag{2.134}$$

Bias: We first start by deriving the bias of such estimator.

Lemma 6.

$$\mathbb{E}_{\tau,\boldsymbol{\epsilon}}[\hat{g}_{\sigma,\boldsymbol{\epsilon}}(\boldsymbol{\theta})] = \nabla_{\boldsymbol{\theta}}F_\sigma(\boldsymbol{\theta}) \tag{2.135}$$

Proof. We proceed as in [52].

Let us denote the following d-dimensional isotropic Normal distribution:

$$\phi(\boldsymbol{\epsilon}) = \frac{1}{(2\pi)^{d/2}}e^{-\frac{\|\boldsymbol{\epsilon}\|^2}{2}}. \tag{2.136}$$

Note that we have:

$$\nabla\phi(\boldsymbol{\epsilon}) = -\boldsymbol{\epsilon}\frac{1}{(2\pi)^{d/2}}e^{-\frac{\|\boldsymbol{\epsilon}\|^2}{2}} = -\boldsymbol{\epsilon}\phi(\boldsymbol{\epsilon}). \tag{2.137}$$

Therefore:

$$\begin{aligned}
\nabla F_\sigma(\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}}\mathbb{E}F(\boldsymbol{\theta}+\sigma\boldsymbol{\epsilon}) \\
&= \nabla_{\boldsymbol{\theta}}\int_{\mathbb{R}^d}F(\boldsymbol{\theta}+\sigma\boldsymbol{\epsilon})\phi(\boldsymbol{\epsilon})d\boldsymbol{\epsilon} \\
&\overset{(a)}{=} \frac{1}{\sigma^d}\nabla_{\boldsymbol{\theta}}\int_{\mathbb{R}^d}F(\boldsymbol{\epsilon}')\phi\left(\frac{\boldsymbol{\epsilon}'-\boldsymbol{\theta}}{\sigma}\right)d\boldsymbol{\epsilon}' \\
&= \frac{1}{\sigma^d}\nabla_{\boldsymbol{\theta}}\int_{\mathbb{R}^d}F(\boldsymbol{\epsilon})\phi\left(\frac{\boldsymbol{\epsilon}-\boldsymbol{\theta}}{\sigma}\right)d\boldsymbol{\epsilon} \\
&\overset{(b)}{=} \frac{1}{\sigma^d}\int_{\mathbb{R}^d}F(\boldsymbol{\epsilon})\nabla_{\boldsymbol{\theta}}\left[\phi\left(\frac{\boldsymbol{\epsilon}-\boldsymbol{\theta}}{\sigma}\right)\right]d\boldsymbol{\epsilon}
\end{aligned} \tag{2.138}$$

$$= \frac{1}{\sigma^d}\int_{\mathbb{R}^d} F(\boldsymbol{\epsilon})\left(-\frac{1}{\sigma}\right)\left(-\frac{\boldsymbol{\epsilon}-\boldsymbol{\theta}}{\sigma}\right)\phi\left(\frac{\boldsymbol{\epsilon}-\boldsymbol{\theta}}{\sigma}\right)d\boldsymbol{\epsilon}$$

$$\overset{(c)}{=} \int_{\mathbb{R}^d} F(\sigma\boldsymbol{\epsilon}'+\boldsymbol{\theta})\left(\frac{1}{\sigma}\right)\boldsymbol{\epsilon}'\phi\left(\boldsymbol{\epsilon}'\right)d\boldsymbol{\epsilon}'$$

$$= \mathbb{E}_{\boldsymbol{\epsilon}}\frac{1}{\sigma}F(\boldsymbol{\theta}+\sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon} = \mathbb{E}\hat{g}_{\sigma,\boldsymbol{\epsilon}}(\boldsymbol{\theta})$$

Where in (a), we do the change of variable $\boldsymbol{\epsilon}' = \boldsymbol{\theta}+\sigma\boldsymbol{\epsilon}$, in (b) we exchange integral and differentiation (as per Leibniz integral rule), which is possible in our case since F is bounded per Assumption 5 (see [52] (24) for instance), and in (c) we use the reverse change of variable as before ($\boldsymbol{\epsilon}' = \frac{\boldsymbol{\epsilon}-\boldsymbol{\theta}}{\sigma}$, so $\boldsymbol{\epsilon} = \sigma\boldsymbol{\epsilon}'+\boldsymbol{\theta}$).

□

Variance: We now proceed with deriving the variance of such estimator: such result, expressed in terms of the bound B on F, is, up to our knowledge, novel.

Lemma 7. *Under Assumption 5, we have:*

$$\mathbb{E}_{\tau,\boldsymbol{\epsilon}}\|\hat{g}_{\sigma,\boldsymbol{\epsilon}}(\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}}F_\sigma(\boldsymbol{\theta})\|^2 \leq \frac{dB^2}{\sigma^2} \tag{2.139}$$

Proof. With $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{I}_{d\times d})$, we have:

$$\mathbb{E}\|\boldsymbol{\epsilon}\|^2 = \mathbb{E}\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon} = \mathbb{E}\sum_{i=1}^d u_i^2 = \sum_{i=1}^d \mathbb{E}u_i^2 = d. \tag{2.140}$$

Using the definition of $\hat{g}_{\sigma,\boldsymbol{\epsilon}}$ and Assumption 5, we have that:

$$\|\hat{g}_{\sigma,\boldsymbol{\epsilon}}\|^2 \leq \frac{B^2}{\sigma^2}\|\boldsymbol{\epsilon}\|^2 \tag{2.141}$$

Therefore:

$$\mathbb{E}\|\hat{g}_{\sigma,\boldsymbol{\epsilon}}\|^2 \leq \frac{B^2}{\sigma^2}\mathbb{E}\|\boldsymbol{\epsilon}\|^2 = \frac{B^2}{\sigma^2}d. \tag{2.142}$$

We now use the bias-variance decomposition (in norm) (for a random variable X, $\mathbb{E}\|X-\mathbb{E}X\|^2 = \mathbb{E}\|X\|^2 - \|\mathbb{E}X\|^2$):

For any $\boldsymbol{\theta} \in \mathbb{R}^d$:

$$\mathbb{E}\|\hat{g}_{\sigma,\boldsymbol{\epsilon}}(\boldsymbol{\theta}) - \nabla F_\sigma(\boldsymbol{\theta})\|^2 = \mathbb{E}\|\hat{g}_{\sigma,\boldsymbol{\epsilon}}(\boldsymbol{\theta})\|^2 - \|\nabla F_\sigma(\boldsymbol{\theta})\|^2 \leq \mathbb{E}\|\hat{g}_{\sigma,\boldsymbol{\epsilon}}(\boldsymbol{\theta})\|^2 = \frac{dB^2}{\sigma^2} \tag{2.143}$$

□

Bias and Variance of the Single Perturbation, Single Rollout Policy Gradient Estimator. We now proceed with proving the bias and variance of the policy gradient estimator for a single perturbation $\boldsymbol{\epsilon}$, and a single rollout τ, defined below as:

$$\hat{g}_{\sigma,\boldsymbol{\epsilon},\tau}(\boldsymbol{\theta}) := \frac{1}{\sigma} f_{\tau}(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon} \tag{2.144}$$

where the rollout τ is sampled for a given $\boldsymbol{\epsilon}$ (i.e. one first samples some $\boldsymbol{\epsilon}$ to obtain a policy parameterized by $\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon}$, and then, one samples a rollout τ from that policy).

Lemma 8 (Bias)**.** *The gradient estimator is unbiased:*

$$\mathbb{E}_{\boldsymbol{\epsilon},\tau}\hat{g}_{\sigma,\boldsymbol{\epsilon},\tau}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} F_{\sigma}(\boldsymbol{\theta}) \tag{2.145}$$

Proof. By the law of total probabilities, and using Assumption 6 we have:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\epsilon},\tau}\hat{g}_{\sigma,\boldsymbol{\epsilon},\tau}(\boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{\epsilon}}\mathbb{E}_{\tau|\boldsymbol{\epsilon}}\hat{g}_{\sigma,\boldsymbol{\epsilon},\tau}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\epsilon}}\mathbb{E}_{\tau|\boldsymbol{\epsilon}}\frac{1}{\sigma}f_{\tau}(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon} \\ &= \mathbb{E}_{\boldsymbol{\epsilon}}\frac{1}{\sigma}\left(\mathbb{E}_{\tau|\boldsymbol{\epsilon}}f_{\tau}(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\right)\boldsymbol{\epsilon} \\ &= \mathbb{E}_{\boldsymbol{\epsilon}}\frac{1}{\sigma}F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon} = \nabla_{\boldsymbol{\theta}}F_{\sigma}(\boldsymbol{\theta}) \end{aligned} \tag{2.146}$$

Where the last equality follows from Lemma 6. □

Lemma 9 (Variance)**.** *Assume that Assumption 5 is verified, as well as Assumption 6. We have, for any $\boldsymbol{\theta} \in \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_{\frac{1}{2}}, ..., \boldsymbol{\theta}_{T-\frac{1}{2}}, \boldsymbol{\theta}_T\}$:*

$$\mathbb{E}_{\tau,\boldsymbol{\epsilon}}\|\hat{g}_{\sigma,\boldsymbol{\epsilon},\tau}(\boldsymbol{\theta}) - \nabla F_{\sigma}(\boldsymbol{\theta})\|^2 \leq \frac{Cd}{\sigma^2} + \frac{dB^2}{\sigma^2} \tag{2.147}$$

Proof. For simplicity, let us fix $\boldsymbol{\theta} \in \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_{\frac{1}{2}}, ..., \boldsymbol{\theta}_{T-\frac{1}{2}}, \boldsymbol{\theta}_T\}$ and denote $\hat{g}_{\sigma,\boldsymbol{\epsilon},\tau} := \hat{g}_{\sigma,\boldsymbol{\epsilon},\tau}(\boldsymbol{\theta})$.

$$\begin{aligned}&\mathbb{E}_{\tau,\boldsymbol{\epsilon}}\|\hat{g}_{\sigma,\boldsymbol{\epsilon},\tau} - \nabla F_{\sigma}(\boldsymbol{\theta})\|^2 \qquad (2.148)\\ =&\mathbb{E}_{\tau,\boldsymbol{\epsilon}}\|\hat{g}_{\sigma,\boldsymbol{\epsilon},\tau} - \frac{1}{\sigma}F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon}\|^2 + \mathbb{E}_{\tau,\boldsymbol{\epsilon}}\|\frac{1}{\sigma}F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon} - \nabla F_{\sigma}(\boldsymbol{\theta})\|^2 \\ &+ 2\mathbb{E}_{\tau,\boldsymbol{\epsilon}}\langle\hat{g}_{\sigma,\boldsymbol{\epsilon},\tau} - \frac{1}{\sigma}F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon}, \frac{1}{\sigma}F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon} - \nabla F_{\sigma}(\boldsymbol{\theta})\rangle \\ =&\mathbb{E}_{\tau,\boldsymbol{\epsilon}}\|\frac{1}{\sigma}\left(f_{\tau}(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon} - F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon}\right)\|^2 + \mathbb{E}_{\tau,\boldsymbol{\epsilon}}\|\frac{1}{\sigma}F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon} - \nabla F_{\sigma}(\boldsymbol{\theta})\|^2 \\ &+ 2\mathbb{E}_{\tau,\boldsymbol{\epsilon}}\langle\frac{1}{\sigma}f_{\tau}(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon} - \frac{1}{\sigma}F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon}, \frac{1}{\sigma}F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon} - \nabla F_{\sigma}(\boldsymbol{\theta})\rangle \\ =&\frac{1}{\sigma^2}\mathbb{E}_{\tau,\boldsymbol{\epsilon}}|f_{\tau}(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon}) - F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})|\|\boldsymbol{\epsilon}\|^2 + \mathbb{E}_{\tau,\boldsymbol{\epsilon}}\|\frac{1}{\sigma}F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon} - \nabla F_{\sigma}(\boldsymbol{\theta})\|^2 \\ &+ 2\mathbb{E}_{\boldsymbol{\epsilon}}\langle\mathbb{E}_{\tau|\boldsymbol{\epsilon}}\frac{1}{\sigma}f_{\tau}(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon} - \frac{1}{\sigma}F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon}, \frac{1}{\sigma}F(\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon} - \nabla F_{\sigma}(\boldsymbol{\theta})\rangle\end{aligned}$$

$$\overset{(a)}{=}\frac{1}{\sigma^2}\mathbb{E}_{\tau,\boldsymbol{\epsilon}}|f_\tau(\boldsymbol{\theta}+\sigma\boldsymbol{\epsilon})-F(\boldsymbol{\theta}+\sigma\boldsymbol{\epsilon})|\|\boldsymbol{\epsilon}\|^2+\mathbb{E}_{\tau,\boldsymbol{\epsilon}}\|\frac{1}{\sigma}F(\boldsymbol{\theta}+\sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon}-\nabla F_\sigma(\boldsymbol{\theta})\|^2$$
$$\overset{(b)}{\leq}\frac{1}{\sigma^2}C\mathbb{E}_{\tau,\boldsymbol{\epsilon}}\|\boldsymbol{\epsilon}\|^2+\mathbb{E}_{\tau,\boldsymbol{\epsilon}}\|\frac{1}{\sigma}F(\boldsymbol{\theta}+\sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon}-\nabla F_\sigma(\boldsymbol{\theta})\|^2$$
$$\overset{(c)}{\leq}\frac{Cd}{\sigma^2}+\frac{dB^2}{\sigma^2} \tag{2.149}$$

Where (a) follows from Lemma 8 (which implies $\mathbb{E}_{\tau|\boldsymbol{\epsilon}}\frac{1}{\sigma}f_\tau(\boldsymbol{\theta}+\sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon}-\frac{1}{\sigma}F(\boldsymbol{\theta}+\sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon}=0$), (b) follows from Assumption 6, and (c) follows from Lemma 7.

□

Proof of Lemma 5: Bias and Variance of the Averaged Gradient Estimator. We can now finally proceed with proving the bias and variance of the full gradient estimator, which is the averaging of the above single random perturbation and rollout gradient estimator, over several random perturbations $\boldsymbol{\epsilon}$ and rollouts τ. We recall Lemma 5 in its full form, including the necessary notations, in Lemma 10 below:

Lemma 10 (i.e. Lemma 5 from subsection 2.11.4.3)**.** *Assume that we sample n random directions $\{\boldsymbol{\epsilon}_i\}_{i=1}^n := \{\boldsymbol{\epsilon}_1, ..., \boldsymbol{\epsilon}_n\}$ independently and identically distributed, and that for each of those random directions $\boldsymbol{\epsilon}_i$, we sample we sample N rollouts $\{\tau_j^{\boldsymbol{\epsilon}_i}\}_{j=1}^N := \{\tau_1^{\boldsymbol{\epsilon}_i}, .., \tau_N^{\boldsymbol{\epsilon}_i}\}$ independently and identically distributed, to obtain a final collection of rollouts $\{\{\tau_j^{\boldsymbol{\epsilon}_i}\}_{j=1}^N\}_{i=1}^n$, and to get $N\times n$ gradient estimators $\hat{g}_{\sigma,\boldsymbol{\epsilon}_i,\tau_j^{\boldsymbol{\epsilon}_i}}$, $(i,j)\in[n]\times[N]$, and to obtain the following estimator, for any $\boldsymbol{\theta}\in\{\boldsymbol{\theta}_0,\boldsymbol{\theta}_{\frac{1}{2}},...,\boldsymbol{\theta}_{T-\frac{1}{2}},\boldsymbol{\theta}_T\}$:*

$$\bar{g}_{\sigma,\{\boldsymbol{\epsilon}_i\}_{i=1}^n,\{\{\tau_j^{\boldsymbol{\epsilon}_i}\}_{j=1}^N\}_{i=1}^n}(\boldsymbol{\theta})=\frac{1}{nN}\sum_{i=1}^n\sum_{j=1}^N\hat{g}_{\sigma,\boldsymbol{\epsilon}_i,\tau_j^{\boldsymbol{\epsilon}_i}}(\boldsymbol{\theta}) \tag{2.150}$$

Then , we have:

$$\bar{g}_{\sigma,\{\boldsymbol{\epsilon}_i\}_{i=1}^n,\{\{\tau_j^{\boldsymbol{\epsilon}_i}\}_{j=1}^N\}_{i=1}^n}(\boldsymbol{\theta})=\nabla_{\boldsymbol{\theta}}F_\sigma(\boldsymbol{\theta}) \tag{2.151}$$

and:

$$\mathbb{E}\|\bar{g}_{\sigma,\{\boldsymbol{\epsilon}_i\}_{i=1}^n,\{\{\tau_j^{\boldsymbol{\epsilon}_i}\}_{j=1}^N\}_{i=1}^n}(\boldsymbol{\theta})-\nabla_{\boldsymbol{\theta}}F_\sigma(\boldsymbol{\theta})\|^2\leq\frac{Cd}{N\sigma^2}+\frac{dB^2}{n\sigma^2} \tag{2.152}$$

Proof. The unbiasedness follows from the linearity of expectation and the proof of Lemmas 8, and the variance follows from the fact that using m i.i.d. samples of a random variable X (for some integer m) divides the variance of the sample mean of X by m, in the previous proof of 9. □

2.11.5.3 Proof of Lemma 4

Such proof is, up to our knowledge, novel, and uses the bound B on F to derive a bound on the Hessian $\nabla^2 F_\sigma(\boldsymbol{\theta})$.

Proof. We have, with $\phi(\boldsymbol{\epsilon}) = \frac{1}{(2\pi)^{d/2}} e^{-\frac{\|\boldsymbol{\epsilon}\|^2}{2}}$ (cf. Equation 2.138):

$$\nabla_{\boldsymbol{\theta}} F_\sigma(\boldsymbol{\theta}) = \frac{1}{\sigma^d} \int_{\mathbb{R}^d} F(\boldsymbol{\epsilon}) \nabla_{\boldsymbol{\theta}} \left[\phi \left(\frac{\boldsymbol{\epsilon} - \boldsymbol{\theta}}{\sigma} \right) \right] d\boldsymbol{\epsilon} \tag{2.153}$$

And we also have, from Equation (2.137), and with $\frac{\partial}{\partial \boldsymbol{\epsilon}}$ denoting the partial derivative with respect to $\boldsymbol{\epsilon}$, and denoting for simplicity ϕ'' the Hessian of ϕ:

$$\phi''(\boldsymbol{\epsilon}) = \frac{\partial}{\partial \boldsymbol{\epsilon}} (-\boldsymbol{\epsilon} \phi(\boldsymbol{\epsilon})) \tag{2.154}$$

Therefore:

$$\phi''(\boldsymbol{\epsilon}) = -\boldsymbol{I} \phi(\boldsymbol{\epsilon}) - \boldsymbol{\epsilon}(-\boldsymbol{\epsilon}^\top \phi(\boldsymbol{\epsilon})) = (\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top - \boldsymbol{I}) \phi(\boldsymbol{\epsilon}) \tag{2.155}$$

In Equation (2.153) above, we can exchange differentiation and integral since the gradient of the Gaussian function, which we denote by $\phi'(\boldsymbol{\theta})$, is continuously differentiable and tends to zero faster than any polynomial of $\boldsymbol{\theta}$, and $F(\boldsymbol{\theta})$ is bounded according to our assumptions (and therefore grows to infinity not faster than a bounded polynomial of $\boldsymbol{\theta}$, cf. [52] p. (20)). Therefore, we obtain: :

$$\begin{aligned}
\nabla^2_{\boldsymbol{\theta}} F_\sigma(\boldsymbol{\theta}) &= \frac{1}{\sigma^d} \int_{\mathbb{R}^d} F(\boldsymbol{\epsilon}) \nabla^2_{\boldsymbol{\theta}} \left[\phi \left(\frac{\boldsymbol{\epsilon} - \boldsymbol{\theta}}{\sigma} \right) \right] d\boldsymbol{\epsilon} \\
&= \frac{1}{\sigma^{d+2}} \int_{\mathbb{R}^d} F(\boldsymbol{\epsilon}) \phi'' \left(\frac{\boldsymbol{\epsilon} - \boldsymbol{\theta}}{\sigma} \right) d\boldsymbol{\epsilon} \\
&= \frac{1}{\sigma^{d+2}} \int_{\mathbb{R}^d} F(\boldsymbol{\epsilon}) \left(\left(\frac{\boldsymbol{\epsilon} - \boldsymbol{\theta}}{\sigma} \right) \left(\frac{\boldsymbol{\epsilon} - \boldsymbol{\theta}}{\sigma} \right)^\top - \boldsymbol{I} \right) \phi \left(\frac{\boldsymbol{\epsilon} - \boldsymbol{\theta}}{\sigma} \right) d\boldsymbol{\epsilon} \\
&= \frac{1}{\sigma^2} \int_{\mathbb{R}^d} F(\boldsymbol{\theta} + \sigma \boldsymbol{\epsilon}) \left(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top - \boldsymbol{I} \right) \phi\left(\boldsymbol{\epsilon}\right) d\boldsymbol{\epsilon}.
\end{aligned} \tag{2.156}$$

Therefore, we have, with $\| \cdot \|_{\mathrm{s}}$ denoting the spectral norm:

$$\|\nabla^2 F_\sigma(\boldsymbol{\theta})\|_s = \| \frac{1}{\sigma^2} \int_{\mathbb{R}^d} F(\boldsymbol{\theta} + \sigma \boldsymbol{\epsilon}) \left(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top - \boldsymbol{I} \right) \phi\left(\boldsymbol{\epsilon}\right) d\boldsymbol{\epsilon} \|_s \tag{2.157}$$

$$\overset{(a)}{\leq} \frac{1}{\sigma^2} \int_{\mathbb{R}^d} \| F(\boldsymbol{\theta} + \sigma \boldsymbol{\epsilon}) \left(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top - \boldsymbol{I} \right) \|_s \phi\left(\boldsymbol{\epsilon}\right) d\boldsymbol{\epsilon} \tag{2.158}$$

$$
\begin{aligned}
&= \frac{1}{\sigma^2}\int_{\mathbb{R}^d} |F(\boldsymbol{\theta}+\sigma\boldsymbol{\epsilon})|\| \left(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top - \boldsymbol{I}\right) \|_s \phi\left(\boldsymbol{\epsilon}\right) d\boldsymbol{\epsilon} && (2.159)\\
&\le \frac{1}{\sigma^2}\int_{\mathbb{R}^d} B\| \left(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top - \boldsymbol{I}\right) \|_s \phi\left(\boldsymbol{\epsilon}\right) d\boldsymbol{\epsilon} && (2.160)\\
&= \frac{B}{\sigma^2}\int_{\mathbb{R}^d} \|\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top - \boldsymbol{I}\|_s \phi\left(\boldsymbol{\epsilon}\right) d\boldsymbol{\epsilon} && (2.161)\\
&\overset{(b)}{\le} \frac{B}{\sigma^2}\int_{\mathbb{R}^d} \left(\|\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\|_s + \|\boldsymbol{I}\|_s\right) \phi\left(\boldsymbol{\epsilon}\right) d\boldsymbol{\epsilon} && (2.162)\\
&\overset{(c)}{=} \frac{B}{\sigma^2}\int_{\mathbb{R}^d} \left(\|\boldsymbol{\epsilon}\|_2^2 + 1\right) \phi\left(\boldsymbol{\epsilon}\right) d\boldsymbol{\epsilon} && (2.163)\\
&= \frac{B}{\sigma^2}\left[\left(\mathbb{E}\|\boldsymbol{\epsilon}\|_2^2\right) + \left(\int_{\mathbb{R}^d} \phi(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon}\right)\right] && (2.164)\\
&= \frac{B}{\sigma^2}\left[d+1\right] && (2.165)\\
&= \frac{(d+1)B}{\sigma^2}. && (2.166)
\end{aligned}
$$

Where (a) follows from Jensen inequality for expectation (and since any norm, including the spectral norm, is convex) , and where (b) follows from the triangular inequality. And where (c) follows from the fact that the spectral norm of $\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top$ is $\|\boldsymbol{\epsilon}\|_2^2$ since the Singular Value Decomposition of $\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top$ is $\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top = \frac{\boldsymbol{\epsilon}}{\|\boldsymbol{\epsilon}\|_2}\|\boldsymbol{\epsilon}\|_2^2\frac{\boldsymbol{\epsilon}^\top}{\|\boldsymbol{\epsilon}\|_2}$ (therefore, the largest singular value of $\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top$, which is the spectral norm by definition, is equal to $\|\boldsymbol{\epsilon}\|_2^2$). We can now use Lemma 1.2.2 in [112] to relate such bound on the Hessian to the smoothness constant of F_σ.

□

2.11.5.4 Proof of Theorem 2: Final convergence rate

We can now use the above results into the general framework from [152], with some additional modifications to adapt their proof to our case of rewards *maximization* (and not function minimization), and to our specific proximal term, which is the indicator function of the ℓ_0 pseudo-ball of radius k (for which the Euclidean projection onto it is the hard-thresholding operator), as well as a few modifications where we use our boundedness assumption on F (Assumption 5) instead of Assumption 1(ii) in [152].

Proof. Let $F_\sigma^-(\boldsymbol{\theta}) := -F_\sigma(\boldsymbol{\theta})$, then we have

$$\max_{\|\boldsymbol{\theta}\|_0 \le k} F_\sigma(\boldsymbol{\theta}) = \min_{\|\boldsymbol{\theta}\|_0 \le k} F_\sigma^-(\boldsymbol{\theta}) \tag{2.167}$$

Note that the nonconvex optimization problem $\min_{\|\boldsymbol{\theta}\|_0 \le k} F_\sigma^-(\boldsymbol{\theta})$ can be reformulated as an alternative nonconvex optimization problem, wherein a nonsmooth, nonconvex indicator function serves as a regularization term:

$$\min_{\|\boldsymbol{\theta}\|_0 \le k} F_\sigma^-(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}\in\mathbb{R}^d} F_\sigma^-(\boldsymbol{\theta}) + r(\boldsymbol{\theta}) \quad \text{where} \quad r(\boldsymbol{\theta}) := \begin{cases} 0, & \text{if } \|\boldsymbol{\theta}\|_0 \le k \\ +\infty, & \text{otherwise} \end{cases}$$

Note that $r(\boldsymbol{\theta})$ is a nonconvex lower-semicontinuous function (cf. the Introduction of [152] for instance).

For simplification, denote by $\boldsymbol{g}_t$ the averaged gradient estimator of $\nabla_{\boldsymbol{\theta}} F_\sigma^-(\boldsymbol{\theta}_t)$ at time step t, i.e.,

$$\boldsymbol{g}_t = -\bar{g}_{\sigma,\{\boldsymbol{\epsilon}_i\}_{i=1}^n,\{\{\tau_j^{\boldsymbol{\epsilon}_i}\}_{j=1}^N\}_{i=1}^n}(\boldsymbol{\theta}_t). \tag{2.168}$$

Then the update rule of $\boldsymbol{\theta}_{t+1}$ is equivalent to

$$\begin{aligned}\boldsymbol{\theta}_{t+1} =& \mathrm{trunc}\left(\boldsymbol{\theta}_t - \alpha \boldsymbol{g}_t\right)\\ \in& \operatorname*{arg\,min}_{\boldsymbol{\theta}\in\mathbb{R}^d}\left\{r(\boldsymbol{\theta}) + \frac{1}{2\alpha}\|\boldsymbol{\theta} - (\boldsymbol{\theta}_t - \alpha\boldsymbol{g}_t)\|^2\right\}\\ =& \operatorname*{arg\,min}_{\boldsymbol{\theta}\in\mathbb{R}^d}\left\{r(\boldsymbol{\theta}) + \langle \boldsymbol{g}_t, \boldsymbol{\theta} - \boldsymbol{\theta}_t\rangle + \frac{1}{2\alpha}\|\boldsymbol{\theta} - \boldsymbol{\theta}_t\|^2\right\}\end{aligned} \tag{2.169}$$

Then we have, with $\hat{\partial}$ denoting the Fréchet derivative (see [48, 152] for more details, in particular the proof of Theorem 2 in [152].):

$$-(\boldsymbol{g}_t + \frac{1}{\alpha}(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t)) \in \hat{\partial} r(\boldsymbol{\theta}_{t+1}), \quad \nabla F_\sigma^-(\boldsymbol{\theta}_{t+1}) - (\boldsymbol{g}_t + \frac{1}{\alpha}(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t)) \in \hat{\partial}(F_\sigma^- + r)(\boldsymbol{\theta}_{t+1}), \tag{2.170}$$

$$r(\boldsymbol{\theta}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\rangle + \frac{1}{2\alpha}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2 \leq r(\boldsymbol{\theta}_t) + \langle \boldsymbol{g}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}_t\rangle + \frac{1}{2\alpha}\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_t\|^2 = r(\boldsymbol{\theta}_t)$$

According to Lemma 4, the spectrum norm of Hessian matrix of F_σ is bounded by $L := \frac{(d+1)B}{\sigma^2}$, which also implies that F_σ^- is L smooth. Then we have,

$$F_\sigma^-(\boldsymbol{\theta}_{t+1}) \leq F_\sigma^-(\boldsymbol{\theta}_t) + \left\langle \nabla F_\sigma^-(\boldsymbol{\theta}_t), \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t \right\rangle + \frac{L}{2}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2. \tag{2.171}$$

Then we have

$$\left\langle \boldsymbol{g}_t - \nabla F_\sigma^-(\boldsymbol{\theta}_t), \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t \right\rangle + \frac{1}{2}(\frac{1}{\alpha} - L)\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2 \leq (F_\sigma^- + r)(\boldsymbol{\theta}_t) - (F_\sigma^- + r)(\boldsymbol{\theta}_{t+1}). \tag{2.172}$$

By Young's inequality,

$$\begin{aligned}\frac{1}{2}(\frac{1}{\alpha} - L)\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2 \leq& (F_\sigma^- + r)(\boldsymbol{\theta}_t) - (F_\sigma^- + r)(\boldsymbol{\theta}_{t+1}) + \frac{1}{2L}\|\boldsymbol{g}_t - \nabla F_\sigma^-(\boldsymbol{\theta}_t)\|^2\\ &+ \frac{L}{2}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2.\end{aligned} \tag{2.173}$$

Summing up the above inequality over time steps $t = 0, \ldots, T-1$, we have

$$\left(\frac{1}{2\alpha} - L\right)\sum_{t=0}^{T-1}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2 \leq (F_\sigma^- + r)(\boldsymbol{\theta}_0) - (F_\sigma^- + r)(\boldsymbol{\theta}_{T-1}) + \frac{1}{2L}\sum_{t=0}^{T-1}\|\boldsymbol{g}_t - \nabla F_\sigma^-(\boldsymbol{\theta}_t)\|^2$$

$$
\begin{aligned}
&\leq (F^-_\sigma + r)(\boldsymbol{\theta}_0) - (F^-_\sigma + r)(\boldsymbol{\theta}_{T-1}) + \frac{1}{2L}\sum_{t=0}^{T-1}\|\boldsymbol{g}_t - \nabla F^-_\sigma(\boldsymbol{\theta}_t)\|^2 \\
&= F^-_\sigma(\boldsymbol{\theta}_0) - F^-_\sigma(\boldsymbol{\theta}_{T-1}) + \frac{1}{2L}\sum_{t=0}^{T-1}\|\boldsymbol{g}_t - \nabla F^-_\sigma(\boldsymbol{\theta}_t)\|^2 \\
&\leq 2B + \frac{1}{2L}\sum_{t=0}^{T-1}\|\boldsymbol{g}_t - \nabla F^-_\sigma(\boldsymbol{\theta}_t)\|^2, \qquad (2.174)
\end{aligned}
$$

where the equality is due to the definition of the indicator function $r(\boldsymbol{\theta})$ and since $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_{T-1}$ are k-sparse, and the last inequality is due to Assumption 5. According to Eq.equation 2.172, we also have:

$$
\begin{aligned}
&\frac{2}{\alpha}\left\langle \boldsymbol{g}_t - \nabla F^-_\sigma(\boldsymbol{\theta}_{t+1}), \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t \right\rangle + \frac{1-\alpha L}{\alpha^2}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2 \\
\leq &\frac{2\left((F^-_\sigma + r)(\boldsymbol{\theta}_t) - (F^-_\sigma + r)(\boldsymbol{\theta}_{t+1})\right)}{\alpha} - \frac{2}{\alpha}\left\langle \nabla F^-_\sigma(\boldsymbol{\theta}_{t+1}) - \nabla F^-_\sigma, \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t \right\rangle. \qquad (2.175)
\end{aligned}
$$

Since $2\left\langle \boldsymbol{g}_t - \nabla F^-_\sigma(\boldsymbol{\theta}_{t+1}), \frac{1}{\alpha}(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t)\right\rangle = \|\boldsymbol{g}_t - \nabla F^-_\sigma(\boldsymbol{\theta}_{t+1}) + \frac{1}{\alpha}(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t)\|^2 - \|\boldsymbol{g}_t - \nabla^-_\sigma(\boldsymbol{\theta}_{t+1})\|^2 - \frac{1}{\alpha^2}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2$, we have :

$$
\begin{aligned}
&\|\boldsymbol{g}_t - \nabla F^-_\sigma(\boldsymbol{\theta}_{t+1}) + \frac{1}{\alpha}(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t)\|^2 \\
\leq &\|\boldsymbol{g}_t - \nabla^-_\sigma(\boldsymbol{\theta}_{t+1})\|^2 + \frac{1}{\alpha^2}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2 - \frac{1-\alpha L}{\alpha^2}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2 \\
&+ \frac{2\left((F^-_\sigma + r)(\boldsymbol{\theta}_t) - (F^-_\sigma + r)(\boldsymbol{\theta}_{t+1})\right)}{\alpha} - \frac{2}{\alpha}\left\langle \nabla F^-_\sigma(\boldsymbol{\theta}_{t+1}) - \nabla F^-_\sigma, \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t \right\rangle \\
\leq &2\|\boldsymbol{g}_t - \nabla F^-_\sigma(\boldsymbol{\theta}_t)\|^2 + 2\|\nabla^-_\sigma(\boldsymbol{\theta}_t) - \nabla^-_\sigma(\boldsymbol{\theta}_{t+1})\|^2 + \frac{L}{\alpha}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2 \\
&+ \frac{2\left((F^-_\sigma + r)(\boldsymbol{\theta}_t) - (F^-_\sigma + r)(\boldsymbol{\theta}_{t+1})\right)}{\alpha} - \frac{2}{\alpha}\left\langle \nabla F^-_\sigma(\boldsymbol{\theta}_{t+1}) - \nabla F^-_\sigma, \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t \right\rangle \\
\leq &2\|\boldsymbol{g}_t - \nabla F^-_\sigma(\boldsymbol{\theta}_t)\|^2 + \frac{2\left((F^-_\sigma + r)(\boldsymbol{\theta}_t) - (F^-_\sigma + r)(\boldsymbol{\theta}_{t+1})\right)}{\alpha} + (2L^2 + \frac{3L}{\alpha})\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2, \\
&\qquad (2.176)
\end{aligned}
$$

where the second inequality is due to Young's inequality and the last inequality is due to the smoothness of the gradient. Summing up the above inequality over time steps $t = 0, \ldots, T-1$, we have :

$$
\begin{aligned}
&\sum_{t=0}^{T-1}\|\boldsymbol{g}_t - \nabla F^-_\sigma(\boldsymbol{\theta}_{t+1}) + \frac{1}{\alpha}(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t)\|^2 \\
\leq &2\sum_{t=0}^{T-1}\|\boldsymbol{g}_t - \nabla F^-_\sigma(\boldsymbol{\theta}_t)\|^2 + \frac{2\left((F^-_\sigma + r)(\boldsymbol{\theta}_0) - (F^-_\sigma + r)(\boldsymbol{\theta}_{T-1})\right)}{\alpha} \\
&+ (2L^2 + \frac{3L}{\alpha})\sum_{t=0}^{T-1}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2
\end{aligned}
$$

$$\leq 2\sum_{t=0}^{T-1}\|\boldsymbol{g}_t - \nabla F_\sigma^-(\boldsymbol{\theta}_t)\|^2 + \frac{4B}{\alpha} + \frac{2}{\alpha^2}\sum_{t=0}^{T-1}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2, \tag{2.177}$$

where the last inequality is due to Assumption 5 and setting $\alpha = \frac{c}{L} < \frac{1}{2L}$. Combing the above inequality with Equation equation 2.174 and Equation equation 2.170, we have :

$$\begin{aligned}
&\mathbb{E}[\mathrm{dist}(0, \hat{\partial}(F_\sigma^- + r)(\boldsymbol{\theta}_T))^2] \\
\leq &\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\boldsymbol{g}_t - \nabla F_\sigma^-(\boldsymbol{\theta}_{t+1}) + \frac{1}{\alpha}(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t)\|^2\right] \\
\leq &\frac{2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{g}_t - \nabla F_\sigma^-(\boldsymbol{\theta}_t)\|^2 + \frac{4B}{T\alpha} + \frac{2}{\alpha^2 T}\sum_{t=0}^{T-1}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2 \\
\leq &\frac{2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{g}_t - \nabla F_\sigma^-(\boldsymbol{\theta}_t)\|^2 + \frac{4B}{T\alpha} + \frac{2}{\alpha^2 T}\left(\frac{4B}{\frac{1}{\alpha} - 2L} + \frac{1}{\frac{L}{\alpha} - 2L^2}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{g}_t - \nabla F_\sigma^-(\boldsymbol{\theta}_t)\|^2\right) \\
= &\frac{2c(1-2c)+2}{c(1-2c)}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{g}_t - \nabla F_\sigma^-(\boldsymbol{\theta}_t)\|^2 + \frac{12-8c}{1-2c}\frac{B}{\alpha T}
\end{aligned} \tag{2.178}$$

We can now plug the result we obtained in Lemma 10 in the result above, to obtain:

$$\mathbb{E}\left[\mathrm{dist}\left(\mathbf{0}, \hat{\partial}\left(-F_\sigma\left(\boldsymbol{\theta}_T\right) + \mathbb{1}_{\ell_0(k)}(\boldsymbol{\theta}_T)\right)\right)^2\right] \leq \epsilon^2,$$

Using Jensen inequality and the fact that the square-root function is concave, we obtain Theorem 2.

□

Chapter 3

Iterative Hard Thresholding over Sparse Support-Preserving Sets

3.1 Introduction

In sparse optimization, directly enforcing sparsity with the ℓ_0 pseudo-norm has several advantages over its convex relaxation counterpart. In compressive sensing for instance [55], one may seek to recover an unknown vector, which sparsity level is known to be at most k. Similarly, in portfolio optimization, due to transaction costs, one may seek to ensure hard constraints on the maximum number of assets invested in [26, 49]. However, in several use cases, one may also seek to enforce additional constraints, such as, for instance, a budget constraint in the case of portfolio optimization, which can be enforced through an extra ℓ_1 constraint, as in [138]. As another example, in sparse non-negative matrix factorization, when estimating the hidden components, one seeks to enforce at the same time a norm constraint and a sparsity constraint [73]. The problem of ℓ_0 empirical risk minimization (ERM) with additional constraints can be formulated as follows, where R is an empirical risk function, $\Gamma \subseteq \mathbb{R}^d$ denotes a convex constraint set, and $\|\cdot\|_0$ denotes the ℓ_0 pseudo-norm (number of non-zero components of a vector):

$$\min_{\boldsymbol{w} \in \mathbb{R}^p} R(\boldsymbol{w}), \quad \text{s.t. } \|\boldsymbol{w}\|_0 \leq k \text{ and } \boldsymbol{w} \in \Gamma. \tag{3.1}$$

In the literature, several algorithms have been developed to address such a problem with mixed constraints, but they typically require the existence of a closed form for the projection onto the mixed constraint, and/or their convergence guarantees are only local, which makes it difficult to estimate the sub-optimality of the output of the algorithm. More precisely, on one hand, some works provide convergence analyses for variants of a (non-convex) projected gradient descent, explicitly for mixed sparse constraints [14, 97, 103, 120], or for general proximal terms (which encompasses our mixed constraints) ([4,22,23,43,56,67,87,152–155]), but such analyses are only local. On the other hand, several existing works on Iterative Hard Thresholding (IHT) provide global guarantees on sub-optimality gap [46,76,88,116,133], but they do not apply to the mixed constraint case we consider. In between the two approaches, one can also find [9] and [90] which, in the deterministic case, give global guarantees for general non-convex constraints or thresholding operators, but which do not provide explicit convergence guarantees for the particular mixed constraint setting that we consider: their rates depend on some constants (the relative concavity or the local concavity constant) for which, up to our knowledge, an explicit form is still unknown for the mixed constraints we consider. We present a more detailed review of related works in Section 3.2, and an overview of them in Table 3.1. To fill this gap, we focus on solving problem 3.1 in the case where Γ belongs to a general family of *support-preserving* sets, which encompasses many usual sets encountered in the literature. As will be described in more detail in Section 3.3, such sets are convex sets for which the projection of a k-sparse vector onto them gets its support preserved, such as for instance ℓ_p norm balls (for $p \geq 1$), or a broader family of *sign–free* convex sets described for instance in [97] and [14].

Adapted to the properties of such constraints, we propose a new variant of IHT, with a two-step projection operator, which, as a first step, identifies the set S of coordinates of the top k components of a given vector and sets the other components to 0 (hard-thresholding), and as a second step projects the resulting vector onto Γ. This two-step projection can offer a simpler alternative to Euclidean projection onto the mixed constraint in the cases where there is a closed form for the latter projection, and handle the cases where there is not. We then provide global sub-optimality guarantees without system error for the objective value, for such an algorithm as well as its stochastic and zeroth-order variants, under the restricted strong-convexity (RSC) and restricted smoothness (RSS) assumptions, in Theorems 4, 7, and 9. Key to our analysis is a novel extension of the three-point lemma to such non-convex setting with mixed constraints, which also allows, as a byproduct, to simplify existing proofs of convergence in objective value for IHT and its variants. In the zeroth-order case, such technique also allows to obtain, up to our knowledge, the first convergence in risk result without system error for a zeroth-order hard-thresholding algorithm. Additionally, our results highlight a compromise between sparsity and sub-optimality gap specific to the additional constraints setting: through a free parameter ρ, one can obtain smaller upper bounds in terms of risk but at the cost of relaxing further the sparsity level of the iterates, or, alternatively, enforce sparser iterates but at the cost of a larger upper bound on the risk.

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

1. We present a variant of IHT to solve hard sparsity problems with additional support-preserving constraints, using a novel two-step projection operator.

2. We describe a novel extension of the three-point lemma to such constraint which allows to simplify existing proofs for IHT and to provide global convergence guarantees in objective value without system error for the algorithm above, in the RSC/RSS setting, highlighting a novel trade-off between sparsity of iterates and sub-optimality gap in such mixed constraints setting.

3. We extend the above algorithm to the stochastic and zeroth-order optimization settings, obtaining similar global convergence guarantees in objective value (without system error) for such mixed constraints setting. In the zeroth-order case, this also provides, up to our knowledge, the first convergence result in objective value without

system error for a zeroth-order hard-thresholding algorithm (with or without extra constraints).

Table 3.1: Comparison of results for Iterative Hard Thresholding with/without additional constraints. [1] $\mathcal{S}$: symmetric convex sets being sign-free or non-negative [97], $\mathcal{A}$: sets verifying Assumption 9. [2] If a paper reports both $\|\boldsymbol{w} - \bar{\boldsymbol{w}}\|$ and $R(\boldsymbol{w}) - R(\bar{\boldsymbol{w}})$, we report only the latter. $\hat{T}$: time index of the $\boldsymbol{w}$ returned by the method (e.g. $\hat{T} = \arg\min_{t \in [T]} R(\boldsymbol{w}_t)$). $\bar{\boldsymbol{w}}$: $\bar{k}$-sparse vector in Γ. Δ: System error (non-vanishing term which depends on the gradient at optimality (e.g. $\mathbb{E}_i \|\nabla R_i(\bar{\boldsymbol{w}})\|$, (see corresponding references))). [4]: $\kappa_s = \frac{L_s}{\nu_s}$ and $\kappa_{s'} = \frac{L_{s'}}{\nu_s}$ (cf. corresponding refs. for defs. of s and s'). [3] SM: Lipschitz-smooth, D: Deterministic. S: Stochastic, Z: Zeroth-Order, L: Lipschitz continuous. [5]: see also Thm. 3, [6]: see also Thm. 6.

Reference	Γ[1]	Convergence[2]	$\boldsymbol{k}$	Setting[3]
[76][5]	$\mathbb{R}^d$	$R(\boldsymbol{w}_{\hat{T}}) \leq R(\bar{\boldsymbol{w}}) + \varepsilon$	$\Omega(\kappa_s^2 \bar{k})$	D, RSS, RSC
[116]	$\mathbb{R}^d$	$\mathbb{E}\|\boldsymbol{w}_{\hat{T}} - \bar{\boldsymbol{w}}\| \leq \varepsilon + \mathcal{O}(\Delta)$	$\Omega(\kappa_s^2 \bar{k})$	S, RSS, RSC
[88]	$\mathbb{R}^d$	$\mathbb{E}R(\boldsymbol{w}_{\hat{T}}) \leq R(\bar{\boldsymbol{w}}) + \varepsilon + \mathcal{O}(\Delta)$	$\Omega(\kappa_s^2 \bar{k})$	S, RSS, RSC
[163][6]	$\mathbb{R}^d$	$\mathbb{E}R(\boldsymbol{w}_{\hat{T}}) \leq R(\bar{\boldsymbol{w}}) + \varepsilon$	$\Omega(\kappa_s^2 \bar{k})$	S, RSS, RSC
[46]	$\mathbb{R}^d$	$\mathbb{E}\|\boldsymbol{w}_{\hat{T}} - \bar{\boldsymbol{w}}\| \leq \varepsilon + \mathcal{O}(\Delta) + \mathcal{O}(\mu)$	$\Omega(\kappa_{s'}^4 \bar{k})$	S, Z, RSS’, RSC
[97], [14]	$\Gamma \in \mathcal{S}$	local convergence	-	D, SM
[103]	ℓ_∞ ball around 0	local convergence	-	S, Z, L
IHT-TSP (Thm. 4)	$\Gamma \in \mathcal{A}$	$R(\boldsymbol{w}_{\hat{T}}) \leq (1 + 2\rho) R(\bar{\boldsymbol{w}}) + \varepsilon$	$\Omega\left(\frac{\kappa_s^2 \bar{k}}{\rho^2}\right)$	D, RSS, RSC
HSG-HT-TSP (Thm. 7)	$\Gamma \in \mathcal{A}$	$\mathbb{E}R(\boldsymbol{w}_{\hat{T}}) \leq (1 + 2\rho) R(\bar{\boldsymbol{w}}) + \varepsilon$	$\Omega\left(\frac{\kappa_s^2 \bar{k}}{\rho^2}\right)$	S, RSS, RSC
HZO-HT (Thm. 8)	$\mathbb{R}^d$	$\mathbb{E}[R(\boldsymbol{w}_{\hat{T}}) - R(\bar{\boldsymbol{w}})] \leq \varepsilon + \mathcal{O}(\mu)$	$\Omega(\kappa_{s'}^2 \bar{k})$	Z, RSS’, RSC
HZO-HT-TSP (Thm. 9)	$\Gamma \in \mathcal{A}$	$\mathbb{E}R(\boldsymbol{w}_{\hat{T}}) \leq (1 + 2\rho) R(\bar{\boldsymbol{w}}) + \varepsilon + \mathcal{O}(\mu)$	$\Omega\left(\frac{\kappa_{s'}^2 \bar{k}}{\rho^2}\right)$	Z, RSS’, RSC

3.2 Related Works

Below we present a more detailed review of the related works.

3.2.1 Local Guarantees for Combined Constraints

Among the works considering optimization over the intersection of the ℓ_0 pseudo-ball of radius k and a set Γ, [103] analyze the convergence of a first-order and zeroth-order stochastic algorithm with a weighted ℓ_0 group norm constraint (which generalizes the ℓ_0 norm), combined with an ℓ_∞ ball constraint. [120] provide a deterministic algorithm which can tackle extra positivity constraints. [97] and [14] analyze the convergence of variants of hard-thresholding in the deterministic case, with extra constraints that are symmetric and sign-free or positive. Other line of works such as [4, 22, 23, 43, 56, 67, 87, 152–155] have a general approach, and analyze the convergence of general proximal algorithms, for composite problems of the form $\min_{\boldsymbol{w}} R(\boldsymbol{w}) + h(\boldsymbol{w})$ where h is a more general non-convex regularizer which can include the ℓ_0 constraint combined with an additional constraint, as long as the closed form for the projection onto the mixed constraint is known (or an approximation of it in the case of [67]). However, all of these works only provide guarantees of convergence towards a critical point, or at best, a local optimum. We provide an overview of those works in Table 3.1. More details about algorithms with local convergence specialized to ℓ_0 optimization can also be found in Table 1 from [42].

3.2.2 Global Guarantees for IHT and RSC Functions

On the other hand, in the case of restricted strongly convex (RSC) and restricted smooth (RSS) functions, existing approximate global guarantees for the IHT algorithm do not apply to problems with such combined constraints. Indeed, several works have considered global convergence guarantees for IHT in various settings: the full gradient (deterministic) setting (IHT [76]), the stochastic setting [88, 116, 133], and the zeroth-order setting [46]. However, they do not address the case where the extra constraint Γ is added to the original sparsity constraint. The works of [9, 90] tackle respectively general non-convex thresholding operators, and general non-convex constraints, in the full gradient (deterministic) setting but however they do not provide explicit convergence rates for the particular type of sets that we consider in this paper: their rates depend on some constants (the relative concavity or the local concavity constant) for which, up to our knowledge, an explicit form is still unknown for the sets we consider.

3.3 Preliminaries

Throughout this paper, we adopt the following notations. For any $\boldsymbol{w} \in \mathbb{R}^d$, $\Pi_\Gamma(\boldsymbol{w})$ denotes a Euclidean projection of $\boldsymbol{w}$ onto Γ, that is $\Pi_\Gamma(\boldsymbol{w}) \in \arg\min_{\boldsymbol{z}\in\Gamma} \|\boldsymbol{w} - \boldsymbol{z}\|_2$, and w_i denotes the i-th component of $\boldsymbol{w}$. $\mathcal{B}_0(k)$ denotes the ℓ_0 pseudo-ball of radius k, i.e. $\mathcal{B}_0(k) = \{\boldsymbol{w} \in \mathbb{R}^d : \|\boldsymbol{w}\|_0 \leq k\}$, with $\|\cdot\|_0$ the ℓ_0 pseudo-norm (i.e. the number of nonzero components of a vector). $\mathcal{H}_k$ denotes the Euclidean projection onto $\mathcal{B}_0(k)$, also known as the hard-thresholding operator (which keeps the k largest (in magnitude) components of a vector, and sets the others to 0 (if there are ties, we can break them e.g. lexicographically)). $\|\cdot\|_p$ denotes the ℓ_p norm for $p \in [1, +\infty)$, and $\|\cdot\|$ the ℓ_2 norm (unless otherwise specified).

$[n]$ denotes the set $\{1, ..., n\}$ for $n \in \mathbb{N}^*$. For any $S \subseteq [d]$, $|S|$ denotes its number of elements. For any $\boldsymbol{w} \in \mathbb{R}^d$, $\text{supp}(\boldsymbol{w})$ denotes its support, i.e. the set of coordinates of its non-zero components. We also introduce below the usual assumptions on R for IHT proofs, i.e. RSC ([76, 88, 96, 108, 116, 133, 157]), and RSS ([76, 88, 157]).

Assumption 7 ((ν_s, s)-RSC)**.** *R is ν_s restricted strongly convex with sparsity parameter s, i.e. it is differentiable, and there exists a generic constant ν_s such that for all $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^d$ with $\|\boldsymbol{x} - \boldsymbol{y}\|_0 \leq s$:*

$$R(\boldsymbol{y}) \geq R(\boldsymbol{x}) + \langle \nabla R(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\nu_s}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2 \tag{3.2}$$

Assumption 8 ((L_s, s)-RSS)**.** *R is L_s restricted smooth with sparsity level s, i.e. it is differentiable, and there exists a generic constant L_s such that for all $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^d$ with $\|\boldsymbol{x} - \boldsymbol{y}\|_0 \leq s$:*

$$R(\boldsymbol{y}) \leq R(\boldsymbol{x}) + \langle \nabla R(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{L_s}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2 \tag{3.3}$$

We then define the notion of support-preserving set that we will use throughout the paper. It essentially requires that projecting any k-sparse vector $\boldsymbol{w}$ onto Γ preserves its support. That is, the convex constraint Γ should be compatible with the sparsity level constraint $\|\boldsymbol{w}\|_0 \leq k$.

Assumption 9 (k-support-preserving set)**.** *$\Gamma \subseteq \mathbb{R}^d$ is k-support-preserving , i.e. it is convex and for any $\boldsymbol{w} \in \mathbb{R}^d$ such that $\|\boldsymbol{w}\|_0 \leq k$, $supp(\Pi_\Gamma(\boldsymbol{w})) \subseteq supp(\boldsymbol{w})$.*

Remark 10. *Below we present some examples of usual sets that also verify Assumption 9 (see Section 3.3.1 for a proof of such statements):*

- *Elementwise decomposable constraints, such as box constraints of the form $\{\boldsymbol{w} \in \mathbb{R}^d : \forall i \in [d], l_i \leq w_i \leq u_i\}$.*
- *Group-wise separable constraints where the constraint on each group is k-support-preserving (such as our constraints in Section 3.8 for the index tracking problem).*
- *Sign-free convex sets [14, 97] (def. in Section 3.3.1), e.g. ℓ_q norm-balls.*

3.3.1 Proof of Remark 10

Before proceeding with the proof of Remark 10, we recall the definition of sign-free convex sets from [97] and [14] below. Essentially, sign-free convex sets are convex sets that are closed by swapping the sign of any coordinate.

Definition 12 ([97], [14])**.** *A convex set Γ is sign-free if for all $\boldsymbol{y} \in \{-1, 1\}^d$ and for all $\boldsymbol{x} \in \Gamma$, $\boldsymbol{x} \odot \boldsymbol{y} \in \Gamma$, where $\odot$ denotes the element-wise vector multiplication (Hadamard product for vectors).*

We now proceed with the proof of Remark 10.

Proof of Remark 10. It is easy to show that any elementwise decomposable constraint such as box constraint is support-preserving (as projection can be done component-wise, independently). Similarly, for group-wise separable constraints where the constraint on each group is k-support-preserving (such as the constraint for the index tracking problem in our Section 3.8), for a k-sparse vector $\boldsymbol{x} \in \mathbb{R}^d$, one can project each group of coordinates independently, and each of such projection will have its support preserved (since each such group of coordinates also contains less than k non-zero elements, i.e. they are k-sparse). Therefore, we analyze in more detail the case of sign-free convex sets. Let Γ be a sign-free convex set, and let $\boldsymbol{x} \in \mathbb{R}^d$ be a k-sparse vector. Define $\boldsymbol{z} = \Pi_\Gamma(\boldsymbol{x})$ and assume that $\text{supp}(\boldsymbol{z}) \nsubseteq \text{supp}(\boldsymbol{x})$. This implies that there exist some non-empty set of coordinates $S \subseteq [d]$, such that for all $i \in S$: $z_i \neq 0$ and $x_i = 0$. Define $\boldsymbol{z}'$ such that $z'_k = \begin{cases} -z_k \text{ if } k \in S \\ z_k \text{ otherwise} \end{cases}$. Since Γ is sign-free, $\boldsymbol{z}' \in \Gamma$. Now, define $\boldsymbol{z}''$ such that $z''_k = \begin{cases} 0 \text{ if } k \in S \\ z_k \text{ if otherwise} \end{cases}$. Since Γ is convex and since $\boldsymbol{z}'' = \frac{1}{2}\boldsymbol{z}' + \frac{1}{2}\boldsymbol{z}$, we have $\boldsymbol{z}'' \in \Gamma$. Now, we have:

$$\begin{aligned} \|\boldsymbol{x} - \boldsymbol{z}''\|_2^2 &= \sum_{k=1}^{d}(x_k - z''_k)^2 = \sum_{k \in [d]\setminus S}(x_k - z_k)^2 \\ &< \sum_{k \in [d]\setminus S}(x_k - z_k)^2 + \sum_{k \in S}(x_k - z_k)^2 = \sum_{k=1}^{d}(x_k - z_k)^2 = \|\boldsymbol{x} - \boldsymbol{z}\|_2^2 \end{aligned} \tag{3.4}$$

Therefore, we encounter a contradiction since we have defined $\boldsymbol{z} = \Pi_\Gamma(\boldsymbol{x})$, and therefore, our assumption $\text{supp}(\boldsymbol{z}) \nsubseteq \text{supp}(\boldsymbol{x})$ is wrong, which means that $\text{supp}(\boldsymbol{z}) \subseteq \text{supp}(\boldsymbol{x})$. □

3.4 Deterministic Case

3.4.1 Algorithm

Two-Step Projection. In all the algorithms of this paper, we will make use of a *two-step projection* operator (TSP), which is different in general from the usual Euclidean projection (EP), in order to obtain, from an arbitrary vector $\boldsymbol{w} \in \mathbb{R}^d$, a vector in $\boldsymbol{w} \in \mathcal{B}_0(k) \cap \Gamma$. We consider such a TSP instead of EP since it enables the derivation a variant of three-point lemma (Lemma 13) which can handle our specific non-convex mixed constraints, and is key to obtaining the convergence analyses we present in Sections 3.4 and 3.6. In addition, the TSP can be more intuitive and efficient to implement than EP (see Section 3.8.2.1 for more discussions about TSP vs

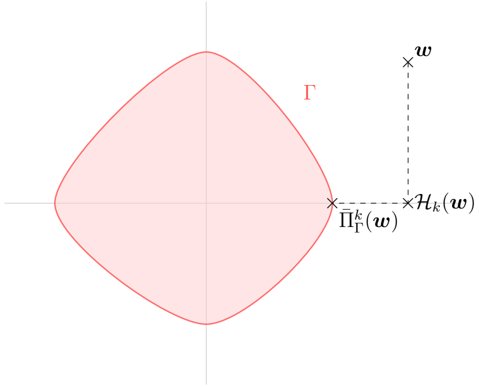

$\boldsymbol{w}$
Γ
$\mathcal{H}_k(\boldsymbol{w})$
$\Pi_\Gamma^k(\boldsymbol{w})$

Figure 3.1: Support-preserving set and two-step projection ($d = 2, k = 1$).

EP). The TSP procedure, which we denote by $\bar{\Pi}^k_\Gamma$, is as follows: we first project $\boldsymbol{w}$ onto $\mathcal{B}_0(k)$ through the hard-thresholding operator $\mathcal{H}_k$, to obtain a k-sparse vector $\boldsymbol{v}_k = \mathcal{H}_k(\boldsymbol{w})$. Then, we project $\boldsymbol{v}_k$ onto Γ , to obtain a final vector $\boldsymbol{w}_S = \Pi_\Gamma(\boldsymbol{v}_k)$, where $S = \text{supp}(\boldsymbol{v}_k)$. Note that consequently, the obtained $\boldsymbol{w}_S$ is not necessarily the EP of $\boldsymbol{w}$ onto $\mathcal{B}_0(k) \cap \Gamma$, that is, we do not necessarily have $\boldsymbol{w}_S = \Pi_{\mathcal{B}_0(k)\cap\Gamma}(\boldsymbol{w})$. However, when Assumption 9 is verified, we have $\boldsymbol{w}_S \in \mathcal{B}_0(k) \cap \Gamma$ (since, because of Assumption 9, $\text{supp}(\boldsymbol{w}_S) \subseteq \text{supp}(\boldsymbol{v}_k)$ and hence $\|\boldsymbol{w}_S\|_0 \leq \|\boldsymbol{v}_k\|_0 \leq k$), therefore each iteration remains feasible in the constraint. We illustrate such a two-step projection on Figure 3.1. We now present our full algorithm in the case where R is a deterministic function without further knowledge of its structure. It is similar to the usual (non-convex) projected gradient descent algorithm, that is, a gradient update step followed by a projection step, except that instead of projecting onto $\Gamma \cap \mathcal{B}_0(k)$ using the Euclidean projection, we obtain a vector $\boldsymbol{w}_k \in \Gamma \cap \mathcal{B}_0(k)$ through the two-step projection method described above. We describe the algorithm in Algorithm 7 below.

Algorithm 7: Deterministic IHT with extra constraints (IHT-TSP)

```
Input: $\boldsymbol{w}_0$: initial value, $\eta$: learning rate, $T$: number of iterations
for $t = 1$ to $T$ do
    $\boldsymbol{w}_t \leftarrow \bar{\Pi}^k_\Gamma(\boldsymbol{w}_{t-1} - \eta \nabla R(\boldsymbol{w}_{t-1}))$;
end
Output: $\boldsymbol{w}_T$
```

Remark 11. *In the case where Γ is a symmetric sign-free convex set (we refer to [97] for the definition of such sets, which include for instance any ℓ_p norm constraint set for $p \in [1, +\infty)$), then the two-step projection is actually the closed form of an Euclidean projection onto the mixed constraint $\Gamma \cap \mathcal{B}_0(k)$ (see Theorem 2.1 from [97]). Therefore, in such cases, Algorithm 7 is identical to a vanilla (non-convex) projected gradient descent algorithm (for which up to now there was still no global convergence guarantees in such a mixed constraints setting in the literature).*

3.4.2 Convergence Analysis

Before proceeding with the convergence analysis, we first present below a variant of the usual three-point lemma from constrained convex optimization, which plays a key role in our proofs. The common three-point lemma for a projection onto a convex set $\mathcal{E}$ relates the distance between a point $\boldsymbol{w} \in \mathbb{R}^d$, its projection $\Pi_\mathcal{E}(\boldsymbol{w})$, and any vector $\bar{\boldsymbol{w}}$ from the set $\mathcal{E}$, through the relation $\|\boldsymbol{w} - \bar{\boldsymbol{w}}\|^2 \geq \|\Pi_\mathcal{E}(\boldsymbol{w}) - \boldsymbol{w}\|^2 + \|\Pi_\mathcal{E}(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2$. Such a three-point lemma is used for instance in a general Bregman divergence form to prove convergence of mirror descent for smooth functions in [27]. Indeed, although proving the convergence of projected gradient descent in the non-smooth case only needs the non-expansivity of projection onto a convex set, the proof for the smooth case usually needs such a three-point lemma, which can be seen as a stronger version of non-expansivity. However, due to the non-convexity of the ℓ_0 pseudo-ball, the convex three-points lemma above does not hold.

Fortunately, building upon Lemma 4.1 from [90], we can obtain a three-point lemma for projection onto the ℓ_0 pseudo-ball.

Lemma 11 (ℓ_0 three-point lemma, proof in Section 3.5.1.2)**.** *Consider $\boldsymbol{w}, \bar{\boldsymbol{w}} \in \mathbb{R}^p$ with $\|\bar{\boldsymbol{w}}\|_0 \leq \bar{k}$. For any $\bar{k} \leq k$, with $\beta := \frac{\bar{k}}{k}$, it holds that:*

$$\|\mathcal{H}_k(\boldsymbol{w}) - \boldsymbol{w}\|^2 \leq \|\boldsymbol{w} - \bar{\boldsymbol{w}}\|^2 - \left(1 - \sqrt{\beta}\right) \|\mathcal{H}_k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2. \tag{3.5}$$

Note that if $k \gg \bar{k}$, β tends to 0, and therefore we approach the usual three-point lemma from convex optimization. This is coherent with the literature on IHT, in which relaxing the sparsity degree (i.e. considering some $k \gg \bar{k}$) is known to make the problem easier to solve (see also Remark 12 below). In addition, the inequality in Lemma 11 is tight with respect to the coefficient $\sqrt{\beta}$, as illustrated by the following lemma.

Lemma 12 (Tightness, proof in Section 3.5.1.3)**.** *Consider an arbitrary pair of integers $(k, \bar{k})$ with $k > \bar{k}$ and an arbitrary scalar $\rho \in (0, 1)$. Then there exist $\boldsymbol{w}$ and $\bar{\boldsymbol{w}}$ with $\|\boldsymbol{w}\|_0 = k$ and $\|\bar{\boldsymbol{w}}\|_0 = \bar{k}$ such that the following holds:*

$$\|\boldsymbol{w} - \mathcal{H}_k(\boldsymbol{w})\|^2 > \|\boldsymbol{w} - \bar{\boldsymbol{w}}\|^2 - \|\mathcal{H}_k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2 + \rho\sqrt{\frac{\bar{k}}{k}}\|\mathcal{H}_k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2. \tag{3.6}$$

Lemma 11 allows us to prove the following rate for convergence in risk of IHT without system error, which appeared first in [76]. Our proof, however, is simpler than the original proof from [76], as we will discuss below.

Theorem 3. *(Equivalent to Thm. 1 from [76], see also Thm. 3.1 from [90]. Proof in Section 3.5.2.1) Assume that $\Gamma = \mathbb{R}^d$. Suppose that Assumption 7 and Assumption 8 hold. Let $s = 2k$. Let $\eta = \frac{1}{L_s}$. Let $\bar{\boldsymbol{w}}$ be an arbitrary $\bar{k}$-sparse vector. Suppose that $k \geq 4\kappa_s^2\bar{k}$ with $\kappa_s := \frac{L_s}{\nu_s}$. Then for any $\varepsilon > 0$, the iterate of IHT satisfies $R(\boldsymbol{w}_t) \leq R(\bar{\boldsymbol{w}}) + \varepsilon$ if*

$$t \geq \left\lceil \frac{2L_s}{\nu_s} \log\left(\frac{(L_s - \nu_s)\|\boldsymbol{w}_0 - \bar{\boldsymbol{w}}\|^2}{2\varepsilon}\right)\right\rceil + 1.$$

Proof Sketch. Using the L_s-RSS of R and some algebraic manipulations, and denoting $\boldsymbol{g}_t = \nabla R(\boldsymbol{w}_t)$ and $\boldsymbol{v}_t := \mathcal{H}_k(\boldsymbol{w}_{t-1} - \frac{1}{L_s}\boldsymbol{g}_{t-1})$ ($= \boldsymbol{w}_t$ when $\Gamma = \mathbb{R}^d$), we have:

$$\begin{aligned}
R(\boldsymbol{v}_t) &\leq R(\boldsymbol{w}_{t-1}) + \frac{L_s}{2}\|\boldsymbol{v}_t - \boldsymbol{w}_{t-1} + \frac{1}{L_s}\boldsymbol{g}_{t-1}\|^2 - \frac{1}{2L_s}\|\boldsymbol{g}_{t-1}\|^2 \\
&\overset{(a)}{\leq} R(\boldsymbol{w}_{t-1}) + \frac{L_s}{2}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1} + \frac{1}{L_s}\boldsymbol{g}_{t-1}\|^2 - \frac{L_s}{2}(1 - \sqrt{\beta})\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2 - \frac{1}{2L_s}\|\boldsymbol{g}_{t-1}\|^2 \\
&\overset{(b)}{\leq} R(\bar{\boldsymbol{w}}) + \frac{L_s - \nu_s}{2}\|\boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\|^2 - \frac{L_s}{2}(1 - \sqrt{\beta})\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2,
\end{aligned} \tag{3.7}$$

where (a) follows from Lemma 11, and in (b) we used the RSC of R with some rearrangements. The proof for Theorem 3 can be concluded with telescopic sum arguments. □

Remark 12 (Necessity of $k = \Omega(\bar{k}\kappa^2)$)**.** *Note that the relaxation of k to $\Omega(\bar{k}\kappa^2)$ in Theorem 3 is unimprovable for IHT, as we detail in Section 3.5.3 with a counter-example, similar to but slightly simpler than the counter-example from Section E.1 in [6]). Therefore, we highlight that all of the following results in this paper will also be expressed in terms of such a relaxed k: this is a fundamental limitation of IHT, and not a limitation of our proof techniques. More details on such relaxation (which is widespread amongst IHT-type algorithms as can be seen in Table 3.1) and how it is a natural way to obtain global guarantees for sparsity enforcing algorithms, can be found in [5, 6, 90].*

Comparison with Previous Proofs. Perhaps the original and most widespread proof framework for convergence in risk of IHT without system error is the one from [76] Theorem 1. Their proof framework is also used for instance in some stochastic extensions of IHT (see Theorem 2 in [163], or Theorems 1 and 2 in [124], even if [124] assume R to have a $\bar{k}$-sparse minimizer which is a strong requirement). The proof from [76] uses specific properties of the hard-thresholding operator to carefully bound the magnitude of the components of $\nabla R(\boldsymbol{w}_t)$ on various sets of coordinates (the support of $\boldsymbol{w}_t$, $\boldsymbol{w}_{t+1}$, and $\bar{\boldsymbol{w}}$, and some intersections and unions of such sets). Using such techniques, however, makes it difficult to derive proofs of IHT in other settings (stochastic, zeroth-order, extra constraints). However, recently, [90] provided a proof of convergence for IHT which avoids such complex considerations about the support sets of the gradient, using their Lemma 4.1 on the *relative concavity* of the hard-thresholding operator. Our work goes in a similar line of work, but we build upon their Lemma 4.1 to prove a three-points lemma for hard-thresholding (our Lemma 11) which allows us to obtain simple proof frameworks also for the stochastic case (retrieving the previous from [163]) and the zeroth-order case (obtaining a new result). But perhaps more importantly, we are able to extend our Lemma 11 to the case with extra constraints Γ verifying Assumption 9 (Lemma 13 below). Such a lemma will allows us to obtain convergence results in the new extra constraints setting that we consider in this paper (providing three new results, in the deterministic, stochastic, and zeroth-order case). It relates together the four points involved in the two step projection ($\boldsymbol{w} \in \mathbb{R}^d$, $\mathcal{H}_k(\boldsymbol{w})$, $\bar{\Pi}_\Gamma^k(\boldsymbol{w})$, and $\bar{\boldsymbol{w}} \in \Gamma \cap \mathcal{B}_0(k)$).

Lemma 13 (Constrained ℓ_0-Three-Point, proof in Section 3.5.1.4)**.** *Suppose that Assumption 9 holds for a set Γ. Consider $\boldsymbol{w}, \bar{\boldsymbol{w}} \in \mathbb{R}^p$ with $\|\bar{\boldsymbol{w}}\|_0 \leq \bar{k}$ and $\bar{\boldsymbol{w}} \in \Gamma$. Then the following holds for any $k > \bar{k}$:*

$$\|\bar{\Pi}_\Gamma^k(\boldsymbol{w}) - \boldsymbol{w}\|^2 \leq \|\boldsymbol{w} - \bar{\boldsymbol{w}}\|^2 - \|\bar{\Pi}_\Gamma^k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2 + \sqrt{\beta}\|\mathcal{H}_k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2, \text{ with } \beta := \frac{\bar{k}}{k}.$$

Equipped with such lemma, we can now present the convergence analysis of Algorithm 7 below, using the assumptions from Section 3.3, and we will describe how the results give rise to a trade-off between the sparsity of the iterates and the tightness of the sub-optimality bound, specific to our mixed constraints setting.

Theorem 4 (Proof in Section 3.5.2.2)**.** *Suppose that Assumption 7, 8, and 9 hold, and that R is non-negative (without loss of generality). Let $s = 2k$, $\eta = \frac{1}{L_s}$, and $\bar{\boldsymbol{w}}$ be an arbitrary*

$\bar{k}$-sparse vector. Let $\rho \in (0, \frac{1}{2}]$ be an arbitrary scalar. Suppose that $k \geq \frac{4(1-\rho)^2 L_s^2}{\rho^2 \nu_s^2}\bar{k}$. Then for any $\varepsilon > 0$, for $T \geq \left\lceil \frac{L_s}{\nu_s} \log\left(\frac{(L_s - \nu_s)\|\boldsymbol{w}_0 - \bar{\boldsymbol{w}}\|^2}{2\varepsilon(1-\rho)}\right)\right\rceil + 1 = \mathcal{O}(\kappa_s \log(\frac{1}{\varepsilon}))$, the iterates of IHT-TSP satisfy:

$$\min_{t \in [T]} R(\boldsymbol{w}_t) \leq (1 + 2\rho) R(\bar{\boldsymbol{w}}) + \varepsilon. \tag{3.8}$$

Further, if $\bar{\boldsymbol{w}}$ is a global minimizer of R over $\mathcal{B}_0(k) := \{\boldsymbol{w} : \|\boldsymbol{w}\|_0 \leq k\}$, then, with $\rho = 0.5$ in the expressions of k and T above: $\min_{t \in [T]} R(\boldsymbol{w}_t) \leq R(\bar{\boldsymbol{w}}) + \varepsilon$.

Proof Sketch. To obtain the proof for general Γ, we reiterate a similar proof as for Theorem 3, but this time, instead of Lemma 11, we use our more general Lemma 13, adapted to general Γ and to our two-step projection technique, to obtain (see the Proof Sketch of Thm. 3 for the definition of $\boldsymbol{v}_t$):

$$R(\boldsymbol{w}_t) \leq R(\bar{\boldsymbol{w}}) + \frac{L_s - \nu_s}{2}\|\boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\|^2 - \frac{L_s}{2}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 + \frac{L_s}{2}\sqrt{\beta}\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2. \tag{3.9}$$

Finally, taking a convex combination of equations 3.7 ($\times\rho$) and 3.9 ($\times(1-\rho)$) for $\rho \in (0, 0.5]$, using the bound $\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 \leq \|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2$ (non-expansiveness of convex projection onto Γ), and carefully tuning k depending on ρ (resulting in our final trade-off between sparsity and optimality), we can fall back to a telescopic sum and conclude the proof. □

Remark 13. *Theorem 4 therefore provides a global convergence guarantee in objective value. However, contrary to usual guarantees for IHT algorithms under RSS/RSC conditions (which are bounds of the form $R(\boldsymbol{w}_t) \leq R(\bar{\boldsymbol{w}}) + \varepsilon$ for some t) , our bound is of the form $R(\boldsymbol{w}_t) \leq (1 + 2\rho)R(\bar{\boldsymbol{w}}) + \varepsilon$. There is a trade-off about the choice of $\rho \in (0, 0.5]$. On one hand, $\rho \to 0$ is preferred in view of the RHS of above bound. On the other hand, the sparsity-level relaxation condition $k \geq \frac{4(1-\rho)^2 L_s^2}{\rho^2 \nu_s^2}\bar{k}$ prefers $\rho \to 0.5$. We illustrate such a trade-off on some synthetic experiments in Section 3.8.1.*

3.5 Proofs for Deterministic Optimization

3.5.1 Proof of Lemmas 11 and 13

3.5.1.1 Useful Lemmas

We first recall some useful definitions and lemmas from the literature.

Definition 13 (Relative concavity [90])**.** *The relative concavity coefficient $\gamma_{k,\beta}$ of a k-sparse projection operator $\mathcal{H}_k$, of relative sparsity $\beta := \frac{\bar{k}}{k}$ with $\bar{k} \leq k$ is defined as:*

$$\gamma_{k,\beta}(\mathcal{H}_k) = \sup\left\{\frac{\langle \boldsymbol{y} - \mathcal{H}_k(\boldsymbol{z}), \boldsymbol{z} - \mathcal{H}_k(\boldsymbol{z})\rangle}{\|\boldsymbol{y} - \mathcal{H}_k(\boldsymbol{z})\|_2^2} \quad \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^d, \|\boldsymbol{y}\|_0 \leq \beta k, \boldsymbol{y} \neq \mathcal{H}_k(\boldsymbol{z})\right\}.$$

Lemma 14 (Lemma 4.1 [90])**.** *When $\mathcal{H}_k$ is the hard-thresholding operator at sparsity level k, we have:*

$$\gamma_{k,\beta}\left(\mathcal{H}_k\right) = \frac{\sqrt{\beta}}{2} = \frac{1}{2}\sqrt{\frac{\bar{k}}{k}}. \tag{3.10}$$

Proof of Lemma 14. Proof in [90]. □

3.5.1.2 Proof of Lemma 11

Proof of Lemma 11. We have:

$$\begin{aligned}
\|\boldsymbol{w} - \bar{\boldsymbol{w}}\|^2 &= \|\boldsymbol{w} - \mathcal{H}_k(\boldsymbol{w})\|^2 + \|\mathcal{H}_k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2 + 2\langle \boldsymbol{w} - \mathcal{H}_k(\boldsymbol{w}), \mathcal{H}_k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\rangle \\
&\overset{(a)}{\geq} \|\boldsymbol{w} - \mathcal{H}_k(\boldsymbol{w})\|^2 + \|\mathcal{H}_k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2 - 2\gamma_{k,\rho}\|\mathcal{H}_k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2 \\
&= \|\boldsymbol{w} - \mathcal{H}_k(\boldsymbol{w})\|^2 + (1 - 2\gamma_{k,\rho})\|\mathcal{H}_k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2 \\
&\overset{(b)}{=} \|\boldsymbol{w} - \mathcal{H}_k(\boldsymbol{w})\|^2 + \left(1 - \sqrt{\frac{\bar{k}}{k}}\right)\|\mathcal{H}_k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2,
\end{aligned} \tag{3.11}$$

where (a) follows from Definition 13 and (b) follows from Lemma 14. Therefore, rearranging, we obtain:

$$\|\mathcal{H}_k(\boldsymbol{w}) - \boldsymbol{w}\|^2 \leq \|\boldsymbol{w} - \bar{\boldsymbol{w}}\|^2 - \left(1 - \sqrt{\frac{\bar{k}}{k}}\right)\|\mathcal{H}_k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2. \tag{3.12}$$

The proof is completed. □

3.5.1.3 Proof of Lemma 12

Of Lemma 12. Let $a = \sqrt{\frac{k}{\bar{k}}}$ and $b = \frac{\rho+1}{2} \in (\rho, 1)$. Consider

$$\boldsymbol{w} = [\underbrace{1, ..., 1}_{k}, \underbrace{b, ...b}_{\bar{k}}] \in \mathbb{R}^{k+\bar{k}}, \quad \bar{\boldsymbol{w}} = [\underbrace{0, ..., 0}_{k}, \underbrace{a, ...a}_{\bar{k}}] \in \mathbb{R}^{k+\bar{k}}.$$

Then we have $\mathcal{H}_k(\boldsymbol{w}) = [\underbrace{1, ..., 1}_{k}, \underbrace{0, ...0}_{\bar{k}}]$ and

$$\|\boldsymbol{w} - \mathcal{H}_k(\boldsymbol{w})\|^2 = b^2\bar{k}, \quad \|\boldsymbol{w} - \bar{\boldsymbol{w}}\|^2 = k + (a-b)^2\bar{k}, \quad \|\mathcal{H}_k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2 = k + a^2\bar{k}.$$

It can be verified that

$$\frac{\|\boldsymbol{w} - \mathcal{H}_k(\boldsymbol{w})\|^2 - \|\boldsymbol{w} - \bar{\boldsymbol{w}}\|^2 + \|\mathcal{H}_k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2}{\|\mathcal{H}_k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2} = \frac{2ab\bar{k}}{k + a^2\bar{k}} = b\sqrt{\frac{\bar{k}}{k}} > \rho\sqrt{\frac{\bar{k}}{k}}.$$

This proves the desired inequality. □

3.5.1.4 Proof of Lemma 13

Proof of Lemma 13. Let us abbreviate $\boldsymbol{v}_k := \mathcal{H}_k(\boldsymbol{w})$. It can be verified that

$$\begin{aligned}
\|\bar{\Pi}_\Gamma^k(\boldsymbol{w}) - \boldsymbol{w}\|^2 &= \left\|\bar{\Pi}_\Gamma^k(\boldsymbol{w}) - \boldsymbol{v}_k + \boldsymbol{v}_k - \boldsymbol{w}\right\|^2 \\
&\overset{(a)}{=} \left\|\bar{\Pi}_\Gamma^k(\boldsymbol{w}) - \boldsymbol{v}_k\right\|^2 + \|\boldsymbol{v}_k - \boldsymbol{w}\|^2 \\
&\overset{(b)}{\leq} \|\boldsymbol{v}_k - \bar{\boldsymbol{w}}\|^2 - \|\bar{\Pi}_\Gamma^k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2 + \|\boldsymbol{w} - \bar{\boldsymbol{w}}\|^2 - \left(1 - \sqrt{\beta}\right)\|\boldsymbol{v}_k - \bar{\boldsymbol{w}}\|^2 \\
&= \|\boldsymbol{w} - \bar{\boldsymbol{w}}\|^2 - \|\bar{\Pi}_\Gamma^k(\boldsymbol{w}) - \bar{\boldsymbol{w}}\|^2 + \sqrt{\beta}\|\boldsymbol{v}_k - \bar{\boldsymbol{w}}\|^2, \qquad (3.13)
\end{aligned}$$

where (a) is due to Assumption 9 and the definition of the two-step projection, which imply that $\bar{\Pi}_\Gamma^k(\boldsymbol{w}) - \boldsymbol{v}_k$ and $\boldsymbol{v}_k - \boldsymbol{w}$ have disjoint supporting sets, and (b) uses the three-point-lemma for projection onto a convex set Γ , as well as Lemma 11. The proof is completed. □

3.5.2 Proof of Theorems 3 and 4

3.5.2.1 Proof of Theorem 3

In this section, we present the proof of Theorem 3 for the convergence of Algorithm 7 without the additional constraint, which as mentioned above, is needed for the proof of Theorem 4, but also, as a byproduct, illustrates how the three-points lemma simplifies previous proofs of Iterative Hard-Thresholding.

Proof of Theorem 3. The L_s- restricted smoothness of R implies that

$$\begin{aligned}
&R(\boldsymbol{w}_t) \\
\leq& R(\boldsymbol{w}_{t-1}) + \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle + \frac{L_s}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 \\
=& R(\boldsymbol{w}_{t-1}) + \frac{L_s}{2}\left\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1} + \frac{1}{L_s}\nabla R(\boldsymbol{w}_{t-1})\right\|^2 - \frac{1}{2L_s}\|\nabla R(\boldsymbol{w}_{t-1})\|^2 \\
\overset{(a)}{\leq}& R(\boldsymbol{w}_{t-1}) + \frac{L_s}{2}\left\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1} + \frac{1}{L_s}\nabla R(\boldsymbol{w}_{t-1})\right\|^2 - \frac{L_s}{2}(1 - \sqrt{\beta})\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 \\
&- \frac{1}{2L_s}\|\nabla R(\boldsymbol{w}_{t-1})\|^2 \\
=& R(\boldsymbol{w}_{t-1}) + \langle \nabla R(\boldsymbol{w}_{t-1}), \bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\rangle + \frac{L_s}{2}\|\boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\|^2 - \frac{L_s}{2}(1 - \sqrt{\beta})\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 \\
\overset{(b)}{\leq}& R(\bar{\boldsymbol{w}}) + \frac{L_s - \nu_s}{2}\|\boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\|^2 - \frac{L_s}{2}(1 - \sqrt{\beta})\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 \\
\leq& R(\bar{\boldsymbol{w}}) + \frac{L_s - \nu_s}{2}\|\boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\|^2 - \frac{2L_s - \nu_s}{4}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2, \qquad (3.14)
\end{aligned}$$

where (a) uses Lemma 11, (b) is due to the ν_s-restricted strong-convexity of R, while the last step is implied by the condition on the sparsity level k from the theorem ($k \geq \frac{4L_s^2}{\nu_s^2}\bar{k}$), and the definition of β ($\beta = \sqrt{\frac{\bar{k}}{k}}$).

The update rule composed of the gradient step and the projection from Algorithm 7 can be rewritten into the following (given that the learning rate is $\eta = \frac{1}{L_s}$, and by definition of a projection):

$$\begin{aligned}
\boldsymbol{w}_t &= \arg \min_{\boldsymbol{w} \text{ s.t.} \|\boldsymbol{w}\|_0 \leq k} \left\| \boldsymbol{w} - \left(\boldsymbol{w}_{t-1} - \frac{1}{L_s} \nabla R(\boldsymbol{w}_{t-1}) \right) \right\|^2 \\
&= \arg \min_{\boldsymbol{w} \text{ s.t.} \|\boldsymbol{w}\|_0 \leq k} \frac{2}{L_s} \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w} - \boldsymbol{w}_{t-1} \rangle + \|\boldsymbol{w} - \boldsymbol{w}_{t-1}\|^2 + \frac{1}{L_s^2} \|\nabla R(\boldsymbol{w}_{t-1})\|^2 \\
&= \arg \min_{\boldsymbol{w} \text{ s.t.} \|\boldsymbol{w}\|_0 \leq k} R(\boldsymbol{w}_{t-1}) + \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w} - \boldsymbol{w}_{t-1} \rangle + \frac{L_s}{2} \|\boldsymbol{w} - \boldsymbol{w}_{t-1}\|^2. \qquad (3.15)
\end{aligned}$$

Therefore, by definition of an arg min, we have:

$$\begin{aligned}
&R(\boldsymbol{w}_{t-1}) + \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_t - \boldsymbol{w}_{t-1} \rangle + \frac{L_s}{2} \|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 \\
&\leq R(\boldsymbol{w}_{t-1}) + \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \boldsymbol{w}_{t-1} \rangle + \frac{L_s}{2} \|\boldsymbol{w}_{t-1} - \boldsymbol{w}_{t-1}\|^2 \\
&= R(\boldsymbol{w}_{t-1}). \qquad (3.16)
\end{aligned}$$

And from the L_s smoothness of R, we also have:

$$R(\boldsymbol{w}_t) \leq R(\boldsymbol{w}_{t-1}) + \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_t - \boldsymbol{w}_{t-1} \rangle + \frac{L_s}{2} \|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2. \qquad (3.17)$$

Therefore, combining equations 3.16 and 3.17, we obtain:

$$R(\boldsymbol{w}_t) \leq R(\boldsymbol{w}_{t-1}). \qquad (3.18)$$

That is, the sequence $\{R(\boldsymbol{w}_t)\}_{t \geq 0}$ of risk is non-increasing.

Let us now consider

$$T := \left\lceil \frac{2L_s}{\nu_s} \log \left(\frac{(L_s - \nu_s)\|\boldsymbol{w}_0 - \bar{\boldsymbol{w}}\|^2}{2\varepsilon} \right) \right\rceil.$$

We claim that $R(\boldsymbol{w}_t) \leq R(\bar{\boldsymbol{w}}) + \varepsilon$ for $t \geq T + 1$. To show this, suppose that $\exists t \in [T]$ such that $R(\boldsymbol{w}_t) \leq R(\bar{\boldsymbol{w}}) + \varepsilon$. Then the claim is naturally true by monotonicity. Otherwise assume that $R(\boldsymbol{w}_t) > R(\bar{\boldsymbol{w}}) + \varepsilon$ for all $t \in [T]$. Then in view of the inequality equation 3.14 we know that

$$\|\boldsymbol{w}_T - \bar{\boldsymbol{w}}\|^2 \leq \frac{2L_s - 2\nu_s}{2L_s - \nu_s} \|\boldsymbol{w}_{T-1} - \bar{\boldsymbol{w}}\|^2$$

$$
\begin{aligned}
&\leq \left(1 - \frac{\nu_s}{2L_s}\right) \|\boldsymbol{w}_{T-1} - \bar{\boldsymbol{w}}\|^2 \\
&\leq \left(1 - \frac{\nu_s}{2L_s}\right)^T \|\boldsymbol{w}_0 - \bar{\boldsymbol{w}}\|^2 \\
&= \exp\left(T \log\left(1 - \frac{\nu_s}{2L_s}\right)\right) \|\boldsymbol{w}_0 - \bar{\boldsymbol{w}}\|^2 \\
&\leq \exp\left(\frac{2L_s}{\nu_s} \log\left(\frac{(L_s - \nu_s)\|\boldsymbol{w}_0 - \bar{\boldsymbol{w}}\|^2}{2\varepsilon} + 1\right) \log\left(1 - \frac{\nu_s}{2L_s}\right)\right) \|\boldsymbol{w}_0 - \bar{\boldsymbol{w}}\|^2 \\
&= \left(1 - \frac{\nu_s}{2L_s}\right) \exp\left(\frac{2L_s}{\nu_s} \log\left(\frac{(L_s - \nu_s)\|\boldsymbol{w}_0 - \bar{\boldsymbol{w}}\|^2}{2\varepsilon}\right) \log\left(1 - \frac{\nu_s}{2L_s}\right)\right) \|\boldsymbol{w}_0 - \bar{\boldsymbol{w}}\|^2 \\
&\overset{(a)}{\leq} \left(1 - \frac{\nu_s}{2L_s}\right) \exp\left(\frac{2L_s}{\nu_s} \log\left(\frac{2\varepsilon}{(L_s - \nu_s)\|\boldsymbol{w}_0 - \bar{\boldsymbol{w}}\|^2}\right) \frac{\nu_s}{2L_s}\right) \|\boldsymbol{w}_0 - \bar{\boldsymbol{w}}\|^2 \\
&= \left(1 - \frac{\nu_s}{2L_s}\right) \frac{2\varepsilon}{L_s - \nu_s} \overset{(b)}{\leq} \frac{2\varepsilon}{L_s - \nu_s},
\end{aligned} \tag{3.19}
$$

where (a) follows from the fact that for all x in $(-\infty, 1)$: $\log(1-x) \leq -x$, and (b) uses the fact that $\left(1 - \frac{\nu_s}{2L_s}\right) \leq 1$.

Then according to equation 3.14 we must have

$$
R(\boldsymbol{w}_{T+1}) \leq R(\bar{\boldsymbol{w}}) + \frac{L_s - \nu_s}{2}\|\boldsymbol{w}_T - \bar{\boldsymbol{w}}\|^2 \leq R(\bar{\boldsymbol{w}}) + \varepsilon,
$$

which implies the desired claim. The proof is completed. □

Remark 14. *Theorem 3 recovers the result of Theorem 1 from [76]. Our proof is shorter yet more intuitive than in that paper.*

3.5.2.2 Proof of Theorem 4

Using the above results, we can now proceed to the full proof of convergence of Theorem 4 below.

Proof of Theorem 4. Denote $\boldsymbol{v}_t = \mathcal{H}_k(\boldsymbol{w}_{t-1} - \frac{1}{L_s}\nabla R(\boldsymbol{w}_{t-1}))$ for any $t \in \mathbb{N}$. Similar to the arguments for equation 3.14, based on the L_s-restricted smoothness of R we can show that:

$$
\begin{aligned}
&R(\boldsymbol{w}_t) \\
\leq& R(\boldsymbol{w}_{t-1}) + \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_t - \boldsymbol{w}_{t-1} \rangle + \frac{L_s}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 \\
=& R(\boldsymbol{w}_{t-1}) + \frac{L_s}{2}\left\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1} + \frac{1}{L_s}\nabla R(\boldsymbol{w}_{t-1})\right\|^2 - \frac{1}{2L_s}\|\nabla R(\boldsymbol{w}_{t-1})\|^2 \\
\overset{(a)}{\leq}& R(\boldsymbol{w}_{t-1}) + \frac{L_s}{2}\left\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1} + \frac{1}{L_s}\nabla R(\boldsymbol{w}_{t-1})\right\|^2 - \frac{L_s}{2}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{L_s}{2}\sqrt{\beta}\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2-\frac{1}{2L_s}\|\nabla R(\boldsymbol{w}_{t-1})\|^2\\
=&R(\boldsymbol{w}_{t-1})+\langle\nabla R(\boldsymbol{w}_{t-1}),\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\rangle+\frac{L_s}{2}\|\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\|^2-\frac{L_s}{2}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2\\
&+\frac{L_s}{2}\sqrt{\beta}\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2\\
\overset{(b)}{\le}&R(\bar{\boldsymbol{w}})+\frac{L_s-\nu_s}{2}\|\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\|^2-\frac{L_s}{2}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2+\frac{L_s}{2}\sqrt{\beta}\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2\\
\le&R(\bar{\boldsymbol{w}})+\frac{L_s-\nu_s}{2}\|\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\|^2-\frac{L_s}{2}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2+\frac{\rho\nu_s}{4(1-\rho)}\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2, \qquad (3.20)
\end{aligned}
$$

where (a) uses Lemma 11, (b) is due to the ν_s-restricted strong-convexity of R, and the last step is due to the condition on sparsity level k from the theorem ($k \ge \frac{4L_s^2(1-\rho)^2}{\nu_s^2\rho^2}\bar{k}$), and the definition of $\beta = \sqrt{\frac{\bar{k}}{k}}$.

In view of equation 3.14, which is valid under the given conditions, we know that

$$
R(\boldsymbol{v}_t) \le R(\bar{\boldsymbol{w}})+\frac{L_s-\nu_s}{2}\|\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\|^2-\frac{2L_s-\nu_s}{4}\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2. \qquad (3.21)
$$

After proper scaling and summing both sides of equation 3.20 and equation 3.21 yields that

$$
\begin{aligned}
&(1-\rho)R(\boldsymbol{w}_t)+\rho R(\boldsymbol{v}_t)\\
\le&R(\bar{\boldsymbol{w}})+\frac{L_s-\nu_s}{2}\|\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\|^2-\frac{(1-\rho)L_s}{2}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2-\frac{\rho(L_s-\nu_s)}{2}\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2\\
=&R(\bar{\boldsymbol{w}})+\frac{L_s-\nu_s}{2}\|\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\|^2-\frac{L_s-\rho\nu_s}{2}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2, \qquad (3.22)
\end{aligned}
$$

where in the second inequality we have used $\bar{\boldsymbol{w}} \in \Gamma$ and the non-expansiveness of projection over convex sets.

Let us now consider

$$
T := \left\lceil \frac{2L_s}{\nu_s}\log\left(\frac{(L_s-\nu_s)\|\boldsymbol{w}_0-\bar{\boldsymbol{w}}\|^2}{2\varepsilon}\right)\right\rceil. \qquad (3.23)
$$

We claim that:

$$
\min_{t\in[T+1]}\{(1-\rho)R(\boldsymbol{w}_t)+\rho R(\boldsymbol{v}_t)\} \le R(\bar{\boldsymbol{w}})+\varepsilon. \qquad (3.24)
$$

To show this, suppose that $\exists t \in [T]$ such that $(1-\rho)R(\boldsymbol{w}_t)+\rho R(\boldsymbol{v}_t) \le R(\bar{\boldsymbol{w}})+\varepsilon$. Then the claim is naturally true. Otherwise assume that $(1-\rho)R(\boldsymbol{w}_t)+\rho R(\boldsymbol{v}_t) > R(\bar{\boldsymbol{w}})+\varepsilon$ for all $t \in [T]$. Then in view of the inequality equation 3.22 we know that

$$
\|\boldsymbol{w}_T-\bar{\boldsymbol{w}}\|^2 \le \frac{L_s-\nu_s}{L_s-\rho\nu_s}\|\boldsymbol{w}_{T-1}-\bar{\boldsymbol{w}}\|^2 \le \left(1-\frac{(1-\rho)\nu_s}{L_s}\right)\|\boldsymbol{w}_{T-1}-\bar{\boldsymbol{w}}\|^2
$$

$$\leq \left(1 - \frac{(1-\rho)\nu_s}{L_s}\right)^T \|\boldsymbol{w}_0 - \bar{\boldsymbol{w}}\|^2 \leq \frac{2\varepsilon}{L_s - \nu_s}. \tag{3.25}$$

Then according to equation 3.22 we must have

$$(1-\rho)R(\boldsymbol{w}_{T+1}) + \rho R(\boldsymbol{v}_{T+1}) \leq R(\bar{\boldsymbol{w}}) + \frac{L_s - \nu_s}{2}\|\boldsymbol{w}_T - \bar{\boldsymbol{w}}\|^2 \leq R(\bar{\boldsymbol{w}}) + \varepsilon, \tag{3.26}$$

which proves the claim from equation 3.24. Now, recall that we have assumed in the Assumptions of Theorem 4, without loss of generality, that R is non-negative (if not, we can redefine R by adding a constant, without modifying the gradient of R, keeping the algorithm untouched), which implies that $R(\boldsymbol{v}_t) \geq 0$. Plugging this in equation 3.24, for $T \geq \left\lceil \frac{2L_s}{\nu_s} \log\left(\frac{(L_s-\nu_s)\|\boldsymbol{w}_0 - \bar{\boldsymbol{w}}\|^2}{2\varepsilon'(1-\rho)}\right)\right\rceil + 1$ implies that:

$$\min_{t\in[T]} R(\boldsymbol{w}_t) \leq \frac{1}{1-\rho}R(\bar{\boldsymbol{w}}) + \frac{\varepsilon}{1-\rho} \leq (1+2\rho)R(\bar{\boldsymbol{w}}) + \frac{\varepsilon}{1-\rho}. \tag{3.27}$$

Plugging the change of variable $\varepsilon' = \frac{\varepsilon}{1-\rho}$ into equation 3.27 above, and in 3.23, we obtain that when $T \geq \left\lceil \frac{2L_s}{\nu_s} \log\left(\frac{(L_s-\nu_s)\|\boldsymbol{w}_0 - \bar{\boldsymbol{w}}\|^2}{2\varepsilon'(1-\rho)}\right)\right\rceil + 1$:

$$\min_{t\in[T]} R(\boldsymbol{w}_t) \leq (1+2\rho)R(\bar{\boldsymbol{w}}) + \varepsilon'. \tag{3.28}$$

Further, consider an ideal case where $\bar{\boldsymbol{w}}$ is a global minimizer of R over $\mathcal{B}_0(k) := \{\boldsymbol{w} : \|\boldsymbol{w}\|_0 \leq k\}$. Then $R(\boldsymbol{v}_t) \geq R(\bar{\boldsymbol{w}})$ is always true for all $t \geq 1$. It follows that the bound in equation 3.24 yields, for $T \geq \left\lceil \frac{2L_s}{\nu_s} \log\left(\frac{(L_s-\nu_s)\|\boldsymbol{w}_0 - \bar{\boldsymbol{w}}\|^2}{2\varepsilon}\right)\right\rceil + 1$:

$$\min_{t\in[T]} \{(1-\rho)R(\boldsymbol{w}_t) + \rho R(\bar{\boldsymbol{w}})\} \leq \min_{t\in[T]} \{(1-\rho)R(\boldsymbol{w}_t) + \rho R(\boldsymbol{v}_t)\} \leq R(\bar{\boldsymbol{w}}) + \varepsilon,$$

which implies: $\min_{t\in[T]} R(\boldsymbol{w}_t) \leq R(\bar{\boldsymbol{w}}) + \frac{\varepsilon}{1-\rho}$. In this case, we can simply set $\rho = 0.5$, and define $\varepsilon' = \frac{\varepsilon}{1-\rho} = 2\varepsilon$ similarly as above. This implies the desired claims. The proof is completed.

□

3.5.3 Lower Bound on the Sparsity Relaxation

Consider $\kappa > 1$, $p = \bar{k} + \kappa^2\bar{k}$ and the following defined diagonal matrix A of size $p \times p$ and vector $\boldsymbol{b}$ of size p:

$$\boldsymbol{A} = \begin{bmatrix} \kappa & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & 0 \\ 0 & 0 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{p\times p}, \quad \boldsymbol{b} = [\underbrace{1, \kappa, \ldots, \kappa}_{\bar{k}}, \underbrace{1, \ldots, 1}_{\kappa^2\bar{k}}]^\top \in \mathbb{R}^p.$$

Clearly, $\boldsymbol{A}$ is κ-smooth and 1-strongly convex. Let us consider the following quadratic objective function:

$$f(\boldsymbol{w}) = \frac{1}{2}(\boldsymbol{w} - \boldsymbol{b})^\top \boldsymbol{A}(\boldsymbol{w} - \boldsymbol{b}).$$

Let $k \in [\bar{k}, \kappa^2 \bar{k}]$ be the relaxed sparsity level used for IHT, and being an even number (without loss of generality). Consider the following defined p-dimensional sparse vectors such that $\|\bar{x}\|_0 = \bar{k}$ and $\|x\|_0 = k$:

$$\bar{\boldsymbol{w}} = [\underbrace{1, \kappa, \ldots, \kappa}_{\bar{k}}, \underbrace{0, \ldots, 0}_{\kappa^2 \bar{k}}]^\top \in \mathbb{R}^p, \quad \boldsymbol{w} = [\underbrace{0, \ldots, 0}_{\bar{k}/2}, \underbrace{\kappa, \ldots, \kappa}_{\bar{k}/2}, \underbrace{1, \ldots, 1}_{k - \bar{k}/2}, \underbrace{0, \ldots, 0}_{\kappa^2 \bar{k} - k + \bar{k}/2}]^\top \in \mathbb{R}^p.$$

We next prove the following theorem which shows that $k \geq \mathcal{O}(\kappa^2 \bar{k})$ is indeed necessary for IHT to converge in some extreme cases for optimizing f.

Theorem 5. *If $\bar{k} \geq 4$ and $k \leq \frac{\kappa^2 \bar{k}}{8}$, then it holds that*

$$f(\boldsymbol{w}) \geq f(\bar{\boldsymbol{w}}) + \frac{\kappa^2 \bar{k}}{16},$$

while $\boldsymbol{w}$ is a fixed point of IHT with sparsity level k and step-size $\eta = \frac{1}{2\kappa}$, i.e.,

$$\boldsymbol{w} = \mathcal{H}_k\left(\boldsymbol{w} - \eta \nabla f(\boldsymbol{w})\right).$$

Proof. It can be seen that $f(\bar{\boldsymbol{w}}) = \frac{1}{2}\kappa^2 \bar{k}$ and

$$f(\boldsymbol{w}) = \frac{1}{2}\left(\kappa + \left(\frac{\bar{k}}{2} - 1\right)\kappa^2 + \kappa^2 \bar{k} + \frac{\bar{k}}{2} - k\right).$$

Therefore

$$\begin{aligned} f(\boldsymbol{w}) - f(\bar{\boldsymbol{w}}) =& \frac{1}{2}\left(\kappa + \left(\frac{\bar{k}}{2} - 1\right)\kappa^2 + \frac{\bar{k}}{2} - k\right) \\ \geq& \frac{1}{2}\left(\left(\frac{\bar{k}}{2} - 1\right)\kappa^2 - k\right) \\ \overset{\zeta_1}{\geq}& \frac{1}{2}\left(\frac{\bar{k}}{4}\kappa^2 - k\right) \geq \frac{\kappa^2 \bar{k}}{16}, \end{aligned}$$

where ζ_1 uses $\bar{k} \geq 4$, and the last inequality is due to $k \leq \frac{\kappa^2 \bar{k}}{8}$. Note that

$$\nabla f(\boldsymbol{w}) = \boldsymbol{A}(\boldsymbol{w} - \boldsymbol{b}) = [\underbrace{-\kappa, \ldots, -\kappa}_{\bar{k}/2}, \underbrace{0, \ldots, 0}_{k}, \underbrace{-1, \ldots, -1}_{\kappa^2 \bar{k} - k + \bar{k}/2}]^\top.$$

Given $\eta = \frac{1}{2\kappa}$, we can show that

$$\boldsymbol{w} - \eta \nabla f(\boldsymbol{w}) = [\underbrace{0.5, \ldots, 0.5}_{\bar{k}/2}, \underbrace{\kappa, \ldots, \kappa}_{\bar{k}/2}, \underbrace{1, \ldots, 1}_{k - \bar{k}/2}, \underbrace{0.5/\kappa, \ldots, 0.5/\kappa}_{\kappa^2 \bar{k} - k + \bar{k}/2}]^\top,$$

which directly yields (as $\kappa > 1$)

$$\boldsymbol{w} = \mathcal{H}_k(\boldsymbol{w} - \eta \nabla f(\boldsymbol{w})),$$

and thus $\boldsymbol{w}$ is a fixed point of IHT with sparsity level k and step-size $\eta = \frac{1}{2\kappa}$. □

Remark 15. *The example is inspired by the one from [6], though slightly simpler. A main difference is that in our example the supporting sets of $\boldsymbol{w}$ and $\bar{\boldsymbol{w}}$ are allowed to be significantly overlapped, while in theirs the supporting sets of the two vectors are constructed to be disjoint.*

3.6 Extensions: Stochastic and Zeroth-Order Cases

In this section, we provide extensions of Algorithm 7 to the stochastic and zeroth-order sparse optimization problems, and provide the corresponding convergence guarantees in objective value without system error.

3.6.1 Stochastic Optimization

In this section, we consider the previous risk minimization problem, in a finite-sum setting, i.e. with $R(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n} R_i(\boldsymbol{w})$, as in [116, 163]: indeed, stochastic algorithms can tackle more easily large-scale datasets where estimating the full $\nabla R(\boldsymbol{w})$ is expensive.

3.6.1.1 Algorithm

We describe the stochastic variant of our previous Algorithm 7 in Algorithm 8 below, which is an extension of the algorithm from [163], to the considered mixed constraints problem setting, using our two-step projection. More precisely, we approximate the gradient of R by a minibatch stochastic gradient with a batch-size increasing exponentially along training, and following the gradient step, we apply our two-step projection operator.

Algorithm 8: Hybrid Stochastic IHT with Extra Constraints (HSG-HT-TSP)

```
Input: $\boldsymbol{w}_0$: initial point, $\eta$: learning rate, $T$: number of iterations, $\{s_t\}$:
       mini-batch sizes.
for $t = 1$ to $T$ do
    Uniformly sample $s_t$ indices $\mathcal{S}_t$ from $[n]$ without replacement;
    Compute the approximate gradient $\boldsymbol{g}_{t-1} = \frac{1}{s_{t-1}} \sum_{i_t \in \mathcal{S}_t} \nabla R_{i_t}(\boldsymbol{w}_{t-1})$;
    $\boldsymbol{w}_t = \bar{\Pi}^k_\Gamma(\boldsymbol{w}_{t-1} - \eta \boldsymbol{g}_{t-1})$;
end
Output: $\hat{\boldsymbol{w}}_T = \arg\min_{\boldsymbol{w} \in \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_T\}} R(\boldsymbol{w})$.
```

3.6.1.2 Convergence Analysis

Before proceeding with the convergence analysis, we make an additional assumption on the population variance of the stochastic gradients, similar to the one in [104].

Assumption 10 (Bounded stochastic gradient variance)**.** *For any $\boldsymbol{w}$, the population variance of the gradient estimator is bounded by B:*

$$\frac{1}{n}\sum_{i=1}^{n}\|\nabla R_i(\boldsymbol{w}) - \nabla R(\boldsymbol{w})\|^2 \leq B. \tag{3.29}$$

We now present our convergence analysis, first with $\Gamma = \mathbb{R}^d$, retrieving Theorem 2 from [163].

Theorem 6 (Equivalent to Theorem 2 from [163], Proof in Section 3.7.2.2)**.** *Assume that $\Gamma = \mathbb{R}^d$. Suppose that Assumption 7, Assumption 8 and Assumption 10 hold. Let $s = 2k$. Let $\bar{\boldsymbol{w}}$ be an arbitrary $\bar{k}$-sparse vector. Let C be an arbitrary positive constant. Assume that we run HSG-HT-TSP (Algorithm 8) for T timesteps, with $\eta = \frac{1}{L_s+C}$, and denote $\alpha := \frac{C}{L_s} + 1$ and $\kappa_s := \frac{L_s}{\nu_s}$. Suppose that $k \geq 4\alpha^2\kappa_s^2\bar{k}$. Finally, assume that we take the following batch-size: $s_t := \left\lceil \frac{\tau}{\omega^t} \right\rceil$ with $\omega := 1 - \frac{1}{4\alpha\kappa_s}$ and $\tau := \frac{\eta B}{C}$. Then, we have the following convergence rate:*

$$\mathbb{E}R(\hat{\boldsymbol{w}}_T) - R(\bar{\boldsymbol{w}}) \leq 2\alpha^2 L_s\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right). \tag{3.30}$$

Such a Theorem is equivalent to Theorem 2 from [163], however, the proof from [163] is based on the same framework as [76], which makes it more complex. Our proof, on the other hand, is very similar to our proof of Theorem 3 above (i.e. closer to convex constrained optimization proofs as discussed above), and simply incorporates the variance of the stochastic gradient estimator (exponentially decreasing thanks to the exponentially increasing batch-size) in a properly weighted telescopic sum (with a technique inspired from [90]). We believe this makes the proof more readily usable for future extensions of IHT. And in particular, using a similar technique as for Theorem 4, we can extend our result to the case with an extra constraint Γ verifying Assumption 9: we present such extension in Theorem 7 below.

Theorem 7 (Proof in Section 3.7.2.3)**.** *Suppose that Assumptions 7 8, 9 and 10 hold, and that R is non-negative (without loss of generality). Let $s = 2k$. Let $\bar{\boldsymbol{w}}$ be an arbitrary $\bar{k}$-sparse vector. Let C be an arbitrary positive constant. Assume that we run HSG-HT-TSP (Algorithm 8) for T timesteps, with $\eta = \frac{1}{L_s+C}$, and denote $\alpha := \frac{C}{L_s} + 1$ and $\kappa_s := \frac{L_s}{\nu_s}$. Suppose that $k \geq 4\alpha^2\frac{1}{\rho^2}\kappa_s^2\bar{k}$ for some $\rho \in (0,1)$. Finally, assume that we take the following batch-size: $s_t := \left\lceil \frac{\tau}{\omega^t} \right\rceil$ with $\omega := 1 - \frac{1}{4\alpha\frac{1}{\rho}\kappa_s}$ and $\tau := \frac{\eta B}{C}$. Then, we have the following convergence rate:*

$$\mathbb{E}\min_{t\in[T]} R(\boldsymbol{w}_t) - (1+2\rho)R(\bar{\boldsymbol{w}}) \leq 2\frac{\alpha^2}{\rho(1-\rho)}L_s\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right). \tag{3.31}$$

Further, if $\bar{\boldsymbol{w}}$ is a global minimizer of R over $\mathcal{B}_0(k) := \{\boldsymbol{w} : \|\boldsymbol{w}\|_0 \leq k\}$, then, with $\rho = 0.5$:

$$\mathbb{E}\min_{t\in[T]} R(\boldsymbol{w}_t) - R(\bar{\boldsymbol{w}}) \leq 8\alpha^2 L_s\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right). \tag{3.32}$$

Corollary 3 (Proof in Section 3.7.3.)**.** *Therefore, the number of calls to a gradient ∇R_i (#IFO), and the number of hard thresholding operations (#HT) such that the left-hand sides in Theorem 7 above are smaller than some $\varepsilon > 0$, are respectively: $\#HT = \mathcal{O}(\kappa_s \log(\frac{1}{\varepsilon}))$ and $\#IFO = \mathcal{O}\left(\frac{\kappa_s}{\nu_s \varepsilon}\right)$.*

3.6.2 Zeroth-Order Optimization (ZOO)

We now consider the zeroth-order (ZO) case [114], in which one does not have access to the gradient $\nabla R(\boldsymbol{w})$, but only to function values $R(\boldsymbol{w})$, which arises for instance when the dataset is private as in distributed learning [66, 161] or the model is private as in black-box adversarial attacks [95], or when computing $\nabla R(\boldsymbol{w})$ is too expensive such as in certain graphical modeling tasks [146]. The idea is then to approximate $\nabla R(\boldsymbol{w})$ using finite differences. We refer the reader to [18] and [93] for an overview of ZO methods.

3.6.2.1 Algorithm

In this section, we describe the ZO version of our algorithm. At its core, it uses the ZO estimator from [46]. We present the full algorithm in Algorithm 9, where $\mathcal{D}_{s_2}$ is a uniform probability distribution on the following set $\mathcal{B}$, which is the set of unit spheres supported on supports of size $s_2 \leq d$: $\mathcal{B} = \{\boldsymbol{w} \in \mathbb{R}^d : \|\boldsymbol{w}\|_0 \leq s_2, \|\boldsymbol{w}\|_2 \leq 1\}$. We can sample from this set by first sampling a random support of size s_2, and then sampling from the unit sphere on that support. Note that if we choose $s_2 := d$, this estimator simply becomes the vanilla ZO estimator with unit-sphere smoothing [93]. Choosing $s_2 < d$ allows to avoid the full-smoothness assumption and can reduce memory consumption by allowing to sample random vectors of size s_2 instead of d. We refer to [46] for more details on such a ZO estimator. The difference with [46] (in addition to the mixed constraint setting and the use of the TSP) is that in our case we sample an exponentially increasing number of random directions, which allows us, for the first time up to our knowledge, to obtain convergence in risk for a ZO hard-thresholding algorithm without any system error (except the unavoidable system error due to the smoothing μ).

Algorithm 9: Hybrid ZO IHT with Extra Constraints (HZO-HT-TSP)

```
Input: $\boldsymbol{w}_0$: initial point, $\eta$: learning rate, $T$: number of iterations, $s_2$: size of the
       random supports, $\{q_t\}$: number of random directions.
for $t = 1$ to $T$ do
    Uniformly sample $q_{t-1}$ i.i.d. random directions $\{\boldsymbol{u}_i\}_{i=1}^{q_{t-1}} \sim \mathcal{D}_{s_2}$
    Compute the approximate gradient
     $\boldsymbol{g}_t = \frac{1}{q_{t-1}} \sum_{i=1}^{q_{t-1}} \frac{d}{\mu} \left( R(\boldsymbol{w}_{t-1} + \mu \boldsymbol{u}_i) - R(\boldsymbol{w}_{t-1}) \right) \boldsymbol{u}_i$
    $\boldsymbol{w}_t = \bar{\Pi}_{\Gamma}^{k}(\boldsymbol{w}_{t-1} - \eta \boldsymbol{g}_{t-1})$
end
Output: $\hat{\boldsymbol{w}}_T = \arg\min_{\boldsymbol{w} \in \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_T\}} R(\boldsymbol{w})$.
```

3.6.2.2 Convergence Analysis

Assumption 11 ((L_s, s)-RSS'). *([116, 133]) R is L_s-restricted strongly smooth with sparsity level s, i.e. it is differentiable, and there exist a generic constant L_s such that for all $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^d$ with $\|\boldsymbol{x}-\boldsymbol{y}\|_0 \leq s$: $\|\nabla R(\boldsymbol{x}) - \nabla R(\boldsymbol{y})\| \leq L_s \|\boldsymbol{x}-\boldsymbol{y}\|$.*

Remark 16. *Note that if a convex function R is (L_s, s)-RSS', then it is also (L_s, s)-RSS (this can be proven in the same way as for usual smoothness in convex optimization (see Lemma 1.2.3 from [110]). However, the converse is not true here, contrary to what holds for usual smooth and convex functions (cf. Theorem 2.1.5 from [110]), as we show through some counter-example in Section 3.7.1. Assumption 11 is indeed slightly more restrictive than Assumption 8, but it is necessary when working with ZO gradient estimators (see more details in [46]).*

We now present our main convergence theorem for the ZO setting, first when $\Gamma = \mathbb{R}^d$.

Theorem 8 (Proof in Section 3.7.4.2)**.** *Assume that $\Gamma = \mathbb{R}^d$. Let $\bar{\boldsymbol{w}}$ be an arbitrary $\bar{k}$-sparse vector. Let $s = 3k$, and $s_2 \in \{1, ..., d\}$. Assume that R is $(L_{s'}, s')$-RSS' with $s' = \max(s_2, s)$, and ν_s-restricted strongly convex. Denote $\kappa_s := \frac{L_{s'}}{\nu_s}$. Let C be an arbitrary positive constant, and denote $\varepsilon_F := \frac{2d}{(s_2+2)}\left(\frac{(s-1)(s_2-1)}{d-1}+3\right)$, $\varepsilon_{abs} := 2dL_{s'}^2 s s_2 \left(\frac{(s-1)(s_2-1)}{d-1}+1\right)$, and $\varepsilon_\mu := L_{s'}^2 sd$. Assume that we run HZO-HT-TSP (Algorithm 9) for T timesteps, with $\eta = \frac{1}{L_{s'}+C} = \frac{1}{\alpha L_{s'}}$, with $\alpha := \frac{C}{L_{s'}} + 1$. Suppose that $k \geq 16\alpha^2\kappa_s^2\bar{k}$. Finally, assume that we take the following number q_t of random directions at each iteration: $q_t := \left\lceil \frac{\tau}{\omega^t} \right\rceil$ with $\omega := 1 - \frac{1}{8\alpha\kappa_s}$ and $\tau := 16\kappa_s \frac{\varepsilon_F}{(\alpha-1)}$. Then, we have the following convergence rate:*

$$\mathbb{E}R(\hat{\boldsymbol{w}}_T) - R(\bar{\boldsymbol{w}}) \leq 4\alpha^2 L_{s'}\kappa_s\omega^T \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{1}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{\kappa_s L_{s'}}\right) + Z\mu^2, \text{ with} \tag{3.33}$$

with $Z = \varepsilon_\mu \left(\frac{2}{\nu_s} + \frac{1}{C}\right) + \frac{\varepsilon_{abs}}{C}$.

Such a novel result illustrates the power of proof techniques based on our three-point lemma. Up to our knowledge, it is the first global convergence guarantee without system error for a ZO hard-thresholding algorithm (see Table 3.1), and as such, is a significant improvement over the result from [46]. Our proof differs from the one in [46]: that latter uses a bound on the expansivity of the hard-thresholding operator, and only provides a result in terms of $\|\boldsymbol{w} - \bar{\boldsymbol{w}}\|$, with a non-vanishing system error which depends on $\nabla R(\boldsymbol{w})$ (cf. Table 3.1). We now present our Theorem in the case of a general support-preserving convex set Γ.

Theorem 9 (Proof in Section 3.7.4.3)**.** *Suppose that Assumptions 7, 9, and 11 hold, and that R is non-negative (without loss of generality). Let $s = 3k$, and let $\bar{\boldsymbol{w}}$ be an arbitrary $\bar{k}$-sparse vector. Let $s_2 \in \{1, ..., d\}$. Assume that R is $(L_{s'}, s')$-RSS' with $s' = \max(s_2, s)$, and ν_s-RSC. Denote $\kappa_s := \frac{L_{s'}}{\nu_s}$. Let C be an arbitrary positive constant, and denote $\varepsilon_F := \frac{2d}{(s_2+2)}\left(\frac{(s-1)(s_2-1)}{d-1}+3\right)$, $\varepsilon_{abs} := 2dL_{s'}^2 s s_2\left(\frac{(s-1)(s_2-1)}{d-1}+1\right)$, and $\varepsilon_\mu := L_{s'}^2 sd$. Assume* *that we run HZO-HT-TSP (Algorithm 9) for T timesteps, with $\eta = \frac{1}{L_{s'}+C} = \frac{1}{\alpha L_{s'}}$, with $\alpha := \frac{C}{L_{s'}} + 1$. Suppose that $k \geq 16\frac{\alpha^2}{\rho^2}\kappa_s^2\bar{k}$ for some $\rho \in (0,1)$. Finally, assume that we take q_t random directions at each iteration, with $q_t := \left\lceil \frac{\tau}{\omega^t} \right\rceil$ with $\omega := 1 - \frac{1}{8\frac{1}{\rho}\alpha\kappa_s}$ and $\tau := 16\kappa_s\frac{\varepsilon_F}{(\alpha-1)}$. Then, we have the following convergence rate:*

$$\mathbb{E}\min_{t\in[T]} R(\boldsymbol{w}_t) - (1+2\rho)R(\bar{\boldsymbol{w}}) \leq 4\frac{\alpha^2}{\rho(1-\rho)}L_{s'}\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{1}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{\kappa_s L_{s'}}\right) + Z\mu^2, \tag{3.34}$$

with $Z = \frac{1}{1-\rho}\left(\varepsilon_\mu\left(\frac{2}{\nu_s} + \frac{1}{C}\right) + \frac{\varepsilon_{abs}}{C}\right)$. Further, if $\bar{\boldsymbol{w}}$ is a global minimizer of R over $\mathcal{B}_0(k) := \{\boldsymbol{w} : \|\boldsymbol{w}\|_0 \leq k\}$, then, with $\rho = 0.5$:

$$\mathbb{E}\min_{t\in[T]} R(\boldsymbol{w}_t) - R(\bar{\boldsymbol{w}}) \leq 16\alpha^2 L_{s'}\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{1}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{\kappa_s L_{s'}}\right) + Z\mu^2. \tag{3.35}$$

Corollary 4 (Proof in Section 3.7.5.)**.** *Additionally, the number of calls to R (#IZO), and the number of hard thresholding operations (#HT) such that the left-hand sides in Theorem 9 above are smaller than $\varepsilon + Z\mu^2$, for some $\varepsilon > 0$ are respectively: #HT $= \mathcal{O}(\kappa_s \log(\frac{1}{\varepsilon}))$ and #IZO $= \mathcal{O}\left(\varepsilon_F\frac{\kappa_s^3 L_s}{\varepsilon}\right)$. Note that if $s_2 = d$ (in which case Assumption 11 becomes the usual (unrestricted) smoothness assumption), we have $\varepsilon_F = \mathcal{O}(s) = \mathcal{O}(k)$, and therefore we obtain a query complexity that is dimension independent.*

Such a query complexity result also holds when $\Gamma = \mathbb{R}^d$ (cf. Corollary 6 in Section). [46] also achieved a dimension independent rate, but their convergence result exhibited a potentially large non-vanishing system error (cf. Table 3.1), which we do not have in Theorems 8 and 9. In strongly convex and smooth ZOO, a dimension independent query complexity is impossible to achieve [77], unless with additional assumptions [7,30,31,31,61,77,91,117,135,149,160]. Our work confirms that, instead of making extra assumptions, a possible way to obtain a dimension independent query complexity is to instead consider optimization with ℓ_0 constraints.

3.7 Proofs for Stochastic and Zeroth-Order Optimization

3.7.1 Discussion on Restricted Smoothness Assumptions

In this section, we provide additional details on the difference between Assumptions 8 and 11. First, we recall the standard definition of smoothness:

Definition 14. *A differentiable function f is L-smooth if for all $\boldsymbol{x}, \boldsymbol{y} \in (\mathbb{R}^d)^2$:*

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\| \tag{3.36}$$

We now provide the counter-example below, illustrating that Assumptions 8 and 11 are not always equivalent, even if f is convex (and that those two assumptions are also different from the usual smoothness assumption).

Lemma 15. *Let us consider the following convex function $f : \mathbb{R}^2 \to \mathbb{R}$ defined as*

$$\forall (x_1, x_2) \in \mathbb{R}^2 : f(x_1, x_2) = x_1^2 + x_2^2 + x_1 x_2 \tag{3.37}$$

f has the following regularity properties, with the given constants being each time the smallest possible:

- *(i) 3-smooth*
- *(ii) 2-restricted smooth (Assumption 8) with sparsity level 1*
- *(iii) $\sqrt{5}$-restricted strongly smooth (Assumption 11) with sparsity level 1*

Proof.

3.7.1.1 Proof of (i)

The Hessian of f is:

$$\boldsymbol{H} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},$$

and its diagonalization is:

$$\boldsymbol{H} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1}, \tag{3.38}$$

with:

$$\boldsymbol{P} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \boldsymbol{P}^{-1} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \text{ and } \boldsymbol{D} = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}.$$

Therefore, the smallest L such that we have $\boldsymbol{H} \preccurlyeq L\boldsymbol{I}_{2\times 2}$ is 3, which implies from Lemma 1.2.2 in [112] that f is smooth with smoothness constant 3.

3.7.1.2 Proof of (ii):

Let us take two $\boldsymbol{x}, \boldsymbol{y}$ in $(\mathbb{R}^d)^2$ such that $\|\boldsymbol{x} - \boldsymbol{y}\|_0 \leq 1$, which therefore implies that: $x_1 = y_1$ or $x_2 = y_2$ (or both). Let us suppose that (E): $x_2 = y_2$. Note that this implies that $\|\boldsymbol{x} - \boldsymbol{y}\|_2 = (x_1 - y_1)^2$. We now need to find the smallest L such that:

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \tag{3.39}$$

$$\Leftrightarrow \tag{3.40}$$

$$y_1^2 + y_2^2 + y_1 y_2 \leq x_1^2 + x_2^2 + x_1 x_2 + (2x_1 + x_2)(y_1 - x_1) + (2x_2 + x_1)(y_2 - x_2) + \frac{L}{2}(x_1 - y_1)^2 \tag{3.41}$$

$$\overset{(E)}{\Leftrightarrow} \tag{3.42}$$

$$y_1^2 + x_2^2 + y_1 x_2 \leq x_1^2 + x_2^2 + x_1 x_2 + (2x_1 + x_2)(y_1 - x_1) + (2x_2 + x_1)(x_2 - x_2) + \frac{L}{2}(x_1 - y_1)^2 \tag{3.43}$$

$$\Leftrightarrow \tag{3.44}$$

$$y_1^2 + x_1^2 - 2y_1x_1 \leq \frac{L}{2}(x_1 - y_1)^2 \tag{3.45}$$

$$\Leftrightarrow \tag{3.46}$$

$$(x_1 - y_1)^2 \leq \frac{L}{2}(x_1 - y_1)^2 \tag{3.47}$$

Therefore, the smallest L possible which can verify the above is $L = 2$. By symmetry, we would have the same chain of equivalence in the alternative case where we would replace $x_2 = y_2$ by $x_1 = y_1$. Therefore, we need some L that will work for both cases, so again, such smallest L is 2.

3.7.1.3 Proof of (iii)

Let us take two $\boldsymbol{x}, \boldsymbol{y}$ such that $\|\boldsymbol{x}-\boldsymbol{y}\|_0 \leq 1$, which therefore implies that: $x_1 = y_1$ or $x_2 = y_2$ (or both). Let us suppose that (E): $x_2 = y_2$. Note that this means that $\|\boldsymbol{x}-\boldsymbol{y}\|_2 = (x_1-y_1)^2$. What we need to find is the smallest L such that:

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|^2 \leq L^2 \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \tag{3.48}$$

$$\Leftrightarrow \tag{3.49}$$

$$(2x_1 + x_2 - (2y_1 + y_2))^2 + (2x_2 + x_1 - (2y_2 + y_1))^2 \leq L^2(x_1 - y_1)^2 \tag{3.50}$$

$$\overset{(E)}{\Leftrightarrow} \tag{3.51}$$

$$(2x_1 + x_2 - (2y_1 + x_2))^2 + (2x_2 + x_1 - (2x_2 + y_1))^2 \leq L^2(x_1 - y_1)^2 \tag{3.52}$$

$$\Leftrightarrow \tag{3.53}$$

$$4(x_1 - y_1)^2 + (x_1 - y_1)^2 \leq L^2(x_1 - y_1)^2 \tag{3.54}$$

$$\Leftrightarrow \tag{3.55}$$

$$5(x_1 - y_1)^2 \leq L^2(x_1 - y_1)^2 \tag{3.56}$$

Therefore, the smallest L possible which can verify the above is $L = \sqrt{5}$. By symmetry, we would have the same chain of equivalence in the alternative case where we would replace $x_2 = y_2$ by $x_1 = y_1$. So therefore we need some L that will work for both cases, so again, that smallest L is $\sqrt{5}$.

□

3.7.2 Proof of Theorems 6 and 7

For the proof of Theorem 7, we use a similar technique as in Theorem 4 to deal with the extra constraint, i.e. we start first from the case $\Gamma = \mathbb{R}^d$ (Theorem 6). Based on our ℓ_0 three-point lemma (Lemma 11), such proof of Theorem 6 is simpler than the corresponding proof of [163] (Proof of Theorem 2, Section B.3). Also, compared to the deterministic setting, here, we need to carefully incorporate the exponentially decreasing error of the gradient estimator into a properly weighted telescopic sum containing terms in $\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2$. Below we provide several intermediary results needed for the proof of Theorem 7. Then, the proof of Theorem 7 will be provided in Section 3.7.2.3.

3.7.2.1 Useful Lemma

Before starting the proof, we present the following lemma from [104], which relates the batch-size s_t and the error of the gradient estimator:

Lemma 16 ([104], Lemma 1). *Let $\boldsymbol{w}_t \in \mathbb{R}^d$. Assume that $\boldsymbol{g}_t$ is the sampled gradient in Algorithm 8 and that the population variance of $R_i(\boldsymbol{w}_t)$ is bounded by B as in Assumption 10. Then the gradient estimate $\boldsymbol{g}_t$ is an unbiased estimate of $\nabla R(\boldsymbol{w}_t)$, and its variance is as follows:*

$$\mathbb{E}\left\|\boldsymbol{g}_t - \nabla R\left(\boldsymbol{w}_t\right)\right\|^2 \leq \frac{n-s_t}{n-1}\frac{1}{s_t}B, \tag{3.57}$$

Note that the original Lemma from [104] is written as an equality, in terms of the exact population variance of a random variable, denoted σ^2, but we rewrite it as an inequality here for simplicity, in order to have a general bound that applies at each iteration.

Proof of Lemma 16. Proof in [104]. □

3.7.2.2 Proof of Theorem 6

Below we now first present a proof for the convergence of Algorithm 8 without the additional constraint (Theorem 6), which is needed for the proof of Theorem 7, and also, as a byproduct, illustrates how the three-point lemma simplifies such proof.

Proof of Theorem 6. The L_s-smoothness of R implies that

$$\begin{aligned}
&R(\boldsymbol{w}_t)\\
\leq& R(\boldsymbol{w}_{t-1}) + \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle + \frac{L_s}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2\\
=& R(\boldsymbol{w}_{t-1}) + \langle \boldsymbol{g}_{t-1}, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle + \frac{L_s}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 + \langle \nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle\\
=& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\left[\|\boldsymbol{w}_t - (\boldsymbol{w}_{t-1} - \eta\boldsymbol{g}_{t-1})\|^2 - \eta^2\|\boldsymbol{g}_{t-1}\|^2 - \|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2\right] + \frac{L_s}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2\\
&+ \langle \nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle\\
=& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\|\boldsymbol{w}_t - (\boldsymbol{w}_{t-1} - \eta\boldsymbol{g}_{t-1})\|^2 - \frac{\eta}{2}\|\boldsymbol{g}_{t-1}\|^2 + \left[\frac{L_s - \frac{1}{\eta}}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2\\
&+ \langle \nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle\\
\overset{(a)}{\leq}& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\left[\|\bar{\boldsymbol{w}} - (\boldsymbol{w}_{t-1} - \eta\boldsymbol{g}_{t-1})\|^2 - (1-\sqrt{\beta})\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2\right] - \frac{\eta}{2}\|\boldsymbol{g}_{t-1}\|^2\\
&+ \left[\frac{L_s - \frac{1}{\eta}}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 + \langle \nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle\\
=& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\left[\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 + \eta^2\|\boldsymbol{g}_{t-1}\|^2 - 2\langle \eta\boldsymbol{g}_{t-1}, \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle\right] - \frac{1}{2\eta}(1-\sqrt{\beta})\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2
\end{aligned}$$

$$
\begin{aligned}
&-\frac{\eta}{2}\|\boldsymbol{g}_{t-1}\|^2+\left[\frac{L_s-\frac{1}{\eta}}{2}\right]\|\boldsymbol{w}_t-\boldsymbol{w}_{t-1}\|^2+\langle\nabla R(\boldsymbol{w}_{t-1})-\boldsymbol{g}_{t-1},\boldsymbol{w}_t-\boldsymbol{w}_{t-1}\rangle\\
=&R(\boldsymbol{w}_{t-1})+\frac{1}{2\eta}\left[\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2-2\langle\eta\boldsymbol{g}_{t-1},\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\rangle\right]-\frac{1}{2\eta}(1-\sqrt{\beta})\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2\\
&+\left[\frac{L_s-\frac{1}{\eta}}{2}\right]\|\boldsymbol{w}_t-\boldsymbol{w}_{t-1}\|^2+\langle\nabla R(\boldsymbol{w}_{t-1})-\boldsymbol{g}_{t-1},\boldsymbol{w}_t-\boldsymbol{w}_{t-1}\rangle\\
\overset{(b)}{=}&R(\boldsymbol{w}_{t-1})+\frac{1}{2\eta}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2-\langle\boldsymbol{g}_{t-1},\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\rangle-\frac{1}{2\eta}(1-\sqrt{\beta})\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2\\
&+\left[\frac{L_s-\frac{1}{\eta}+C}{2}\right]\|\boldsymbol{w}_t-\boldsymbol{w}_{t-1}\|^2+\frac{1}{2C}\|\nabla R(\boldsymbol{w}_{t-1})-\boldsymbol{g}_{t-1}\|^2,
\end{aligned}
$$

where (a) follows from Lemma 11 and (b) follows from the inequality $\langle a,b\rangle\leq\frac{C}{2}a^2+\frac{1}{2C}b^2$, for any $(a,b)\in(\mathbb{R}^d)^2$ with $C>0$ an arbitrary strictly positive constant.

Let us now assume that $\eta=\frac{1}{L_s+C}$: therefore the term $\left[\frac{L_s-\frac{1}{\eta}+C}{2}\right]\|\boldsymbol{w}_t-\boldsymbol{w}_{t-1}\|^2$ above is 0. We now take the conditional expectation (conditioned on w_{t-1}, which is the random variable which realizations are $\boldsymbol{w}_{t-1}$), on both sides, and from Lemma 16 we obtain the inequality below (we slightly abuse notations and denote $\mathbb{E}[\cdot|\mathrm{w}_{t-1}=\boldsymbol{w}_{t-1}]$ by $\mathbb{E}[\cdot|\boldsymbol{w}_{t-1}]$):

$$
\begin{aligned}
\mathbb{E}[R(\boldsymbol{w}_t)|\boldsymbol{w}_{t-1}]\leq&R(\boldsymbol{w}_{t-1})+\frac{1}{2\eta}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2-\langle\nabla R(\boldsymbol{w}_{t-1}),\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\rangle\\
&-\frac{1}{2\eta}(1-\sqrt{\beta})\mathbb{E}\left[\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]+\frac{B(n-s_{t-1})}{2Cs_{t-1}(n-1)}\\
\overset{(a)}{\leq}&R(\boldsymbol{w}_{t-1})+\frac{1}{2\eta}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2+\left[R(\bar{\boldsymbol{w}})-R(\boldsymbol{w}_{t-1})-\frac{\nu_s}{2}\|\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\|^2\right]\\
&-\frac{1}{2\eta}(1-\sqrt{\beta})\mathbb{E}\left[\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]+\frac{B}{2Cs_{t-1}}\\
=&R(\bar{\boldsymbol{w}})+\left[\frac{\frac{1}{\eta}-\nu_s}{2}\right]\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2-\frac{1}{2\eta}(1-\sqrt{\beta})\mathbb{E}\left[\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]\\
&+\frac{B}{2Cs_{t-1}},
\end{aligned}
\tag{3.58}
$$

where (a) follows from the RSC condition, and the fact that $s_{t-1}\in\mathbb{N}^*$.

We recall that $\eta=\frac{1}{L_s+C}$. Let us define $\alpha:=\frac{C}{L_s}+1$. Then $C=(\alpha-1)L_s$, and $\eta=\frac{1}{\alpha L_s}$. Also recall that $\kappa_s=\frac{L_s}{\nu_s}$.

We can simplify the inequality above into:

$$
\begin{aligned}
\mathbb{E}[R(\boldsymbol{w}_t)|\boldsymbol{w}_{t-1}]-R(\bar{\boldsymbol{w}})\leq&\frac{1}{2\eta}\left[\left(1-\frac{1}{\alpha\kappa_s}\right)\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2-(1-\sqrt{\beta})\mathbb{E}\left[\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]\right.\\
&\left.+\frac{\eta B}{Cs_{t-1}}\right].
\end{aligned}
\tag{3.59}
$$

We now take the expectation over w_{t-1} of the above inequality (i.e. we take $\mathbb{E}_{\mathrm{w}_{t-1}}[\cdot]$): using the law of total expectation ($\mathbb{E}[\cdot] = \mathbb{E}_{\mathrm{w}_{t-1}}[\mathbb{E}[\cdot|\boldsymbol{w}_{t-1}]]$) we obtain:

$$\mathbb{E}R(\boldsymbol{w}_t) - R(\bar{\boldsymbol{w}}) \leq \frac{1}{2\eta}\left[\left(1 - \frac{1}{\alpha\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - (1-\sqrt{\beta})\mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 + \frac{\eta B}{Cs_{t-1}}\right] \tag{3.60}$$

Similarly as in [90], we now take a weighted sum over $t = 1, ..., T$, to obtain:

$$\begin{aligned}
&\sum_{t=1}^{T} 2\eta \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \mathbb{E}[R(\boldsymbol{w}_t) - R(\bar{\boldsymbol{w}})] \\
\leq &\sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \left[\left(1 - \frac{1}{\alpha\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - (1-\sqrt{\beta})\mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 + \frac{\eta B}{Cs_{t-1}}\right] \\
= &\sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \left[\left(1 - \frac{1}{\alpha\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - (1-\sqrt{\beta})\mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2\right] \\
&+ \sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \frac{\eta B}{Cs_{t-1}} \\
= &(1-\sqrt{\beta}) \sum_{t=1}^{T} \left[\left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t+1} \mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2\right] \\
&+ \sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \frac{\eta B}{Cs_{t-1}} \\
\overset{(a)}{=} &(1-\sqrt{\beta}) \left[\left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T} \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 - \mathbb{E}\|\boldsymbol{w}_T - \bar{\boldsymbol{w}}\|^2\right] + \sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \frac{\eta B}{Cs_{t-1}} \\
\leq &(1-\sqrt{\beta}) \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T} \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \frac{\eta B}{Cs_{t-1}} \\
\leq &\left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T} \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \frac{\eta B}{Cs_{t-1}},
\end{aligned} \tag{3.61}$$

where (a) follows from simplifying the telescopic sum.

We now choose k and s_t as follows: we choose $k \geq 4\alpha^2\kappa_s^2\bar{k}$, which implies that:

$$\begin{aligned}
&\sqrt{\beta} \leq \frac{1}{2\alpha\kappa_s} \\
&\implies \sqrt{\beta} \leq \frac{1}{2\alpha\kappa_s - 1}
\end{aligned}$$

$$\begin{aligned}
&\implies 1-\sqrt{\beta} \geq 1-\frac{1}{2\alpha\kappa_s - 1} = \frac{2\alpha\kappa_s - 2}{2\alpha\kappa_s - 1} = \frac{1-\frac{1}{\alpha\kappa_s}}{1-\frac{1}{2\alpha\kappa_s}} \\
&\implies \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right) \leq 1 - \frac{1}{2\alpha\kappa_s}. \qquad (3.62)
\end{aligned}$$

And we choose $s_t := \left\lceil \frac{\tau}{\omega^t} \right\rceil$ with $\omega := 1 - \frac{1}{4\alpha\kappa_s}$ and $\tau := \frac{\eta B}{C}$.

Let us call $\nu := 1 - \frac{1}{2\alpha\kappa_s}$. Note that we have:

$$\nu \leq \omega. \qquad (3.63)$$

And that we have the inequality below:

$$\frac{\nu}{\omega} = \frac{1-\frac{1}{2\alpha\kappa_s}}{1-\frac{1}{4\alpha\kappa_s}} = \frac{4\alpha\kappa_s - 2}{4\alpha\kappa_s - 1} = 1 - \frac{1}{4\alpha\kappa_s - 1} \leq 1 - \frac{1}{4\alpha\kappa_s} = \omega. \qquad (3.64)$$

This allows us to simplify equation 3.61 into:

$$\begin{aligned}
\mathbb{E}\sum_{t=1}^{T} 2\eta \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} [R(\boldsymbol{w}_t) - R(\bar{\boldsymbol{w}})] &\leq \nu^T \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \sum_{t=1}^{T} \nu^{T-t}\omega^{t-1} \\
&= \nu^T \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\omega^T}{\omega}\sum_{t=1}^{T}\left(\frac{\nu}{\omega}\right)^{T-t} \\
&= \nu^T \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\omega^T}{\omega}\frac{1-\left(\frac{\nu}{\omega}\right)^T}{1-\left(\frac{\nu}{\omega}\right)} \\
&\leq \nu^T \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\omega^T}{\omega}\frac{1}{1-\left(\frac{\nu}{\omega}\right)} \\
&\overset{(a)}{\leq} \nu^T \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\omega^T}{\omega}\frac{1}{1-\omega} \\
&\overset{(b)}{\leq} \nu^T \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\omega^T\frac{1}{1-\omega} \\
&\overset{(c)}{\leq} \omega^T \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\omega^T\frac{1}{1-\omega} \\
&\overset{(d)}{\leq} \frac{\omega^T}{1-\omega} \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\omega^T\frac{1}{1-\omega} \\
&= \frac{\omega^T}{1-\omega}\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right) \\
&= 4\alpha\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right), \qquad (3.65)
\end{aligned}$$

where in the left hand side we have used the linearity of expectation, and where (a) uses equation 3.64, (b) uses the fact that $\frac{1}{\omega} = \frac{1}{1-\frac{1}{4\alpha\kappa_s}} \leq \frac{1}{1-\frac{1}{4}} = \frac{4}{3}$ (since $\kappa_s \geq 1$ and $\alpha \geq 1$ (indeed, from the theorem's assumption $\alpha = \frac{C}{L_s} + 1$ with $C > 0$)), (c) uses equation 3.63, and (d) uses the fact that $\omega < 1$ so $1 < \frac{1}{1-\omega}$.

Let us now normalize the above inequality:

$$\mathbb{E}\frac{\sum_{t=1}^{T} 2\eta \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} R(\boldsymbol{w}_t)}{\sum_{t=1}^{T} 2\eta \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}} \leq R(\bar{\boldsymbol{w}}) + \frac{4\alpha\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{\sum_{t=1}^{T} 2\eta \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}}. \tag{3.66}$$

The left hand side above is a weighted sum, which is an upper bound on the smallest term of the sum. Regarding the right hand side, we can simplify it using the fact that $0 < \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)$, and therefore:

$$\sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \geq 1. \tag{3.67}$$

Therefore, we obtain:

$$\mathbb{E}\min_{t\in\{1,..,T\}} R(\boldsymbol{w}_t) - R(\bar{\boldsymbol{w}}) \leq \frac{4\alpha\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{2\eta} = 2\alpha^2 L_s\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right) \tag{3.68}$$

Which can be simplified into the expression below, using the definition of $\hat{\boldsymbol{w}}_T$:

$$\mathbb{E}R(\hat{\boldsymbol{w}}_T) - R(\bar{\boldsymbol{w}}) \leq 2\alpha^2 L_s\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right). \tag{3.69}$$

The proof is completed. □

Corollary 5. *Under the assumptions of Theorem 6, let ε be a small enough positive number $\varepsilon > 0$. To achieve an error $\mathbb{E}R(\hat{\boldsymbol{w}}_T) - R(\bar{\boldsymbol{w}}) \leq \varepsilon$ using Algorithm 8 the number of calls to a gradient ∇R_i (#IFO), and the number of hard thresholding operations (#HT) are respectively:*

$$\#HT = \mathcal{O}(\kappa_s \log(\frac{1}{\varepsilon})), \quad \#IFO = \mathcal{O}\left(\frac{\kappa_s}{\nu_s\varepsilon}\right). \tag{3.70}$$

Proof of Corollary 5. Let $\varepsilon \in \mathbb{R}_+^*$. Let us find T to ensure that $\mathbb{E}R(\hat{\boldsymbol{w}}_T) - R(\bar{\boldsymbol{w}}) \leq \varepsilon$. This will be enforced if:

$$2\alpha^2 L_s\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right) \leq \varepsilon$$

$$\begin{aligned}
&\iff T\log(\omega) \leq \log\left(\frac{\varepsilon}{2\alpha^2 L_s\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\right)}\right) \\
&\iff T \geq \frac{1}{\log(\frac{1}{\omega})}\log\left(\frac{2\alpha^2 L_s\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\right)}{\varepsilon}\right). \qquad (3.71)
\end{aligned}$$

Therefore, let us take:

$$T := \left\lceil \frac{1}{\log(\frac{1}{\omega})}\log\left(\frac{2\alpha^2 L_s\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\right)}{\varepsilon}\right)\right\rceil. \qquad (3.72)$$

We can now derive the #IFO and #HT. First, we have one hard-thresholding operation at each iteration, therefore #HT$= T$. Using the fact that $\frac{1}{\log(\frac{1}{\omega})} = \frac{1}{-\log(\omega)} = \frac{1}{-\log(1-\frac{1}{4\alpha\kappa_s})} \leq \frac{1}{\frac{1}{4\alpha\kappa_s}} = 4\alpha\kappa_s$ (since by property of the logarithm, for all $x \in (-\infty,-1) : \log(1-x) \leq -x$), we obtain that #HT $= \mathcal{O}(\kappa_s \log\left(\frac{1}{\varepsilon}\right))$.

We now turn to computing the #IFO. At each iteration t we have s_t gradient evaluations, therefore:

$$\begin{aligned}
\#\text{IFO} &= \sum_{t=0}^{T-1} s_t \\
&\leq \sum_{t=0}^{T-1}\left(\frac{\tau}{\omega^t}+1\right) \\
&= T + \tau\frac{\left(\frac{1}{\omega}\right)^T-1}{\frac{1}{\omega}-1} \\
&\leq T + \frac{\tau}{\frac{1}{\omega}-1}\left(\frac{1}{\omega}\right)^T \\
&= T + \frac{\tau}{\frac{1}{\omega}-1}\exp\left(T\log\left(\frac{1}{\omega}\right)\right) \\
&\overset{(a)}{\leq} 1 + \frac{1}{\log(\frac{1}{\omega})}\log\left(\frac{2\alpha^2 L_s\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\right)}{\varepsilon}\right) \\
&\quad + \frac{\tau}{\frac{1}{\omega}-1}\exp\left(\log\left(\frac{1}{\omega}\right)\left[\frac{1}{\log(\frac{1}{\omega})}\log\left(\frac{2\alpha^2 L_s\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\right)}{\varepsilon}\right)+1\right]\right) \\
&= 1 + \frac{1}{\log(\frac{1}{\omega})}\log\left(\frac{2\alpha^2 L_s\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\right)}{\varepsilon}\right) + \frac{\frac{\tau}{\omega}}{\frac{1}{\omega}-1}\frac{2\alpha^2 L_s\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\right)}{\varepsilon} \\
&= 1 + \frac{1}{\log(\frac{1}{\omega})}\log\left(\frac{2\alpha^2 L_s\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\right)}{\varepsilon}\right) + \frac{\tau}{1-\omega}\frac{2\alpha^2 L_s\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\right)}{\varepsilon} \\
&= 1 + \frac{1}{\log(\frac{1}{\omega})}\log\left(\frac{2\alpha^2 L_s\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\right)}{\varepsilon}\right) + \tau\frac{8\alpha^3 L_s\kappa_s^2\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\right)}{\varepsilon}
\end{aligned}$$

$$
\begin{aligned}
&\overset{(b)}{=} 1 + \frac{1}{\log(\frac{1}{\omega})} \log\left(\frac{2\alpha^2 L_s \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{\varepsilon}\right) \\
&\quad + \frac{B}{\alpha L_s} \frac{1}{L_s(\alpha-1)} \frac{8\alpha^3 L_s}{\varepsilon} \frac{L_s}{\nu_s} \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right) \\
&= 1 + \frac{1}{\log(\frac{1}{\omega})} \log\left(\frac{2\alpha^2 L_s \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{\varepsilon}\right) + \frac{8B\alpha^2 \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{(\alpha-1)\nu_s} \frac{1}{\varepsilon},
\end{aligned}
\tag{3.73}
$$

where (a) follows from equation 3.72, and for (b) we recall that $\tau = \frac{\eta B}{C}$, $\eta = \frac{1}{\alpha L_s}$ and $C = L_s(\alpha - 1)$.

Therefore, overall, the IFO complexity is in $\mathcal{O}(\frac{\kappa_s}{\nu_s \varepsilon})$.

□

3.7.2.3 Proof of Theorem 7

We now proceed with the full proof of Theorem 7.

Proof of Theorem 7. Similary as in the proof of Theorem 6 in Section 3.7.2.2, let us take: $\eta := \frac{1}{L_s + C}$, and $\alpha := \frac{C}{L_s} + 1$. Then $C = (\alpha - 1)L_s$, and $\eta = \frac{1}{\alpha L_s}$. Recall that $\kappa_s := \frac{L_s}{\nu_s}$. Denote $\boldsymbol{v}_t = \mathcal{H}_k(\boldsymbol{w}_{t-1} - \eta \nabla R(\boldsymbol{w}_{t-1}))$ for any $t \in \mathbb{N}$.

Similarly as in Section 3.7.2.2, the L_s-smoothness of R implies that

$$
\begin{aligned}
&R(\boldsymbol{w}_t) \\
\le& R(\boldsymbol{w}_{t-1}) + \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_t - \boldsymbol{w}_{t-1} \rangle + \frac{L_s}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 \\
=& R(\boldsymbol{w}_{t-1}) + \langle \boldsymbol{g}_{t-1}, \boldsymbol{w}_t - \boldsymbol{w}_{t-1} \rangle + \frac{L_s}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 + \langle \nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}, \boldsymbol{w}_t - \boldsymbol{w}_{t-1} \rangle \\
=& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\left[\|\boldsymbol{w}_t - (\boldsymbol{w}_{t-1} - \eta \boldsymbol{g}_{t-1})\|^2 - \eta^2\|\boldsymbol{g}_{t-1}\|^2 - \|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2\right] + \frac{L_s}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 \\
&+ \langle \nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}, \boldsymbol{w}_t - \boldsymbol{w}_{t-1} \rangle \\
=& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\|\boldsymbol{w}_t - (\boldsymbol{w}_{t-1} - \eta \boldsymbol{g}_{t-1})\|^2 - \frac{\eta}{2}\|\boldsymbol{g}_{t-1}\|^2 + \left[\frac{L_s - \frac{1}{\eta}}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 \\
&+ \langle \nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}, \boldsymbol{w}_t - \boldsymbol{w}_{t-1} \rangle \\
\overset{(a)}{\le}& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\left[\|\bar{\boldsymbol{w}} - (\boldsymbol{w}_{t-1} - \eta \boldsymbol{g}_{t-1})\|^2 - \|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 + \sqrt{\beta}\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2\right] - \frac{\eta}{2}\|\boldsymbol{g}_{t-1}\|^2 \\
&+ \left[\frac{L_s - \frac{1}{\eta}}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 + \langle \nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}, \boldsymbol{w}_t - \boldsymbol{w}_{t-1} \rangle \\
=& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\left[\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 + \eta^2\|\boldsymbol{g}_{t-1}\|^2 - 2\langle \eta \boldsymbol{g}_{t-1}, \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}} \rangle\right] - \frac{1}{2\eta}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{\sqrt{\beta}}{2\eta}\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2-\frac{\eta}{2}\|\boldsymbol{g}_{t-1}\|^2+\left[\frac{L_s-\frac{1}{\eta}}{2}\right]\|\boldsymbol{w}_t-\boldsymbol{w}_{t-1}\|^2+\langle\nabla R(\boldsymbol{w}_{t-1})-\boldsymbol{g}_{t-1},\boldsymbol{w}_t-\boldsymbol{w}_{t-1}\rangle\\
=&R(\boldsymbol{w}_{t-1})+\frac{1}{2\eta}\left[\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2-2\langle\eta\boldsymbol{g}_{t-1},\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\rangle\right]-\frac{1}{2\eta}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2+\frac{\sqrt{\beta}}{2\eta}\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2 \qquad (3.74)\\
&+\left[\frac{L_s-\frac{1}{\eta}}{2}\right]\|\boldsymbol{w}_t-\boldsymbol{w}_{t-1}\|^2+\langle\nabla R(\boldsymbol{w}_{t-1})-\boldsymbol{g}_{t-1},\boldsymbol{w}_t-\boldsymbol{w}_{t-1}\rangle\\
\overset{(b)}{=}&R(\boldsymbol{w}_{t-1})+\frac{1}{2\eta}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2-\langle\boldsymbol{g}_{t-1},\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\rangle-\frac{1}{2\eta}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2+\frac{\sqrt{\beta}}{2\eta}\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2\\
&+\left[\frac{L_s-\frac{1}{\eta}+C}{2}\right]\|\boldsymbol{w}_t-\boldsymbol{w}_{t-1}\|^2+\frac{1}{2C}\|\nabla R(\boldsymbol{w}_{t-1})-\boldsymbol{g}_{t-1}\|^2, \qquad (3.75)
\end{aligned}
$$

where (a) follows from Lemma 13 and (b) follows from the inequality $\langle a,b\rangle\leq\frac{C}{2}a^2+\frac{1}{2C}b^2$, for any $(a,b)\in(\mathbb{R}^d)^2$ with $C>0$ an arbitrary strictly positive constant. Let us now take $\eta:=\frac{1}{L_s+C}$: therefore the term $\left[\frac{L_s-\frac{1}{\eta}+C}{2}\right]\|\boldsymbol{w}_t-\boldsymbol{w}_{t-1}\|^2$ above is 0. We now take the conditional expectation (conditioned on w_{t-1}, which is the random variable which realizations are $\boldsymbol{w}_{t-1}$), on both sides, and from Lemma 16 we obtain the inequality below (we slightly abuse notations and denote $\mathbb{E}[\cdot|\mathrm{w}_{t-1}=\boldsymbol{w}_{t-1}]$ by $\mathbb{E}[\cdot|\boldsymbol{w}_{t-1}]$):

$$
\begin{aligned}
\mathbb{E}[R(\boldsymbol{w}_t)|\boldsymbol{w}_{t-1}]\leq&R(\boldsymbol{w}_{t-1})+\frac{1}{2\eta}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2-\langle\nabla R(\boldsymbol{w}_{t-1}),\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\rangle\\
&-\frac{1}{2\eta}\mathbb{E}\left[\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]+\frac{\sqrt{\beta}}{2\eta}\mathbb{E}\left[\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]+\frac{B(n-s_{t-1})}{2Cs_{t-1}(n-1)} \qquad (3.76)\\
\overset{(a)}{\leq}&R(\boldsymbol{w}_{t-1})+\frac{1}{2\eta}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2+\left[R(\bar{\boldsymbol{w}})-R(\boldsymbol{w}_{t-1})-\frac{\nu_s}{2}\|\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\|^2\right]\\
&-\frac{1}{2\eta}\mathbb{E}\left[\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]+\frac{\sqrt{\beta}}{2\eta}\mathbb{E}\left[\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]+\frac{B}{2Cs_{t-1}}\\
=&R(\bar{\boldsymbol{w}})+\left[\frac{\frac{1}{\eta}-\nu_s}{2}\right]\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2-\frac{1}{2\eta}\mathbb{E}\left[\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]\\
&+\frac{\sqrt{\beta}}{2\eta}\mathbb{E}\left[\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]+\frac{B}{2Cs_{t-1}}, \qquad (3.77)
\end{aligned}
$$

where (a) follows from the RSC condition, and the fact that $s_{t-1}\in\mathbb{N}^*$.

Now recall that we have taken $\eta=\frac{1}{L_s+C}$, and let us define $\alpha:=\frac{C}{L_s}+1$. Then $C=(\alpha-1)L_s$, and $\eta=\frac{1}{\alpha L_s}$. Also recall that $\kappa_s=\frac{L_s}{\nu_s}$.

We can simplify the inequality above into:

$$\mathbb{E}[R(\boldsymbol{w}_t)|\boldsymbol{w}_{t-1}]-R(\bar{\boldsymbol{w}})$$

$$\leq \frac{1}{2\eta}\left[\left(1-\frac{1}{\alpha\kappa_s}\right)\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2 - \mathbb{E}\left[\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] + \sqrt{\beta}\mathbb{E}\left[\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] + \frac{\eta B}{Cs_{t-1}}\right]. \tag{3.78}$$

We now take the expectation over w_{t-1} of the above inequality (i.e. we take $\mathbb{E}_{\mathrm{w}_{t-1}}[\cdot]$): using the law of total expectation ($\mathbb{E}[\cdot] = \mathbb{E}_{\mathrm{w}_{t-1}}[\mathbb{E}[\cdot|\boldsymbol{w}_{t-1}]]$) we obtain:

$$\begin{aligned}\mathbb{E}R(\boldsymbol{w}_t) - R(\bar{\boldsymbol{w}}) \leq \frac{1}{2\eta}\Bigg[&\left(1-\frac{1}{\alpha\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2 - \mathbb{E}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2 \\ &+\sqrt{\beta}\mathbb{E}\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2 + \frac{\eta B}{Cs_{t-1}}\Bigg].\end{aligned} \tag{3.79}$$

Additionally, in view of equation 3.60 applied at $\boldsymbol{v}_t$ instead of $\boldsymbol{w}_t$, (since $\boldsymbol{v}_t$ here corresponds to the $\boldsymbol{w}_t$ from Section 3.7.2.2, i.e. $\boldsymbol{v}_t$ is the hard-thresholding of an iterate after a gradient step), we know that:

$$\mathbb{E}R(\boldsymbol{v}_t) - R(\bar{\boldsymbol{w}}) \leq \frac{1}{2\eta}\left[\left(1-\frac{1}{\alpha\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2 - (1-\sqrt{\beta})\mathbb{E}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2 + \frac{\eta B}{Cs_{t-1}}\right]. \tag{3.80}$$

We now take a convex combination similarly as in the case without additional constraint (section 3.5.2), for some $\rho \in (0,1)$.

$$\begin{aligned}&\mathbb{E}(1-\rho)R(\boldsymbol{w}_t) + \rho R(\boldsymbol{v}_t) \\ \leq& R(\bar{\boldsymbol{w}}) + \frac{1}{2\eta}\Bigg[\left(1-\frac{1}{\alpha\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2 - (1-\rho)\mathbb{E}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2 \\ &\quad + \left((1-\rho)\sqrt{\beta} - (1-\sqrt{\beta})\rho\right)\mathbb{E}\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2 + \frac{\eta B}{Cs_{t-1}}\Bigg] \\ =& R(\bar{\boldsymbol{w}}) + \frac{1}{2\eta}\Bigg[\left(1-\frac{1}{\alpha\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2 - (1-\rho)\mathbb{E}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2 \\ &\quad - \left(\rho-\sqrt{\beta}\right)\mathbb{E}\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2 + \frac{\eta B}{Cs_{t-1}}\Bigg] \\ \overset{(b)}{\leq}& R(\bar{\boldsymbol{w}}) + \frac{1}{2\eta}\Bigg[\left(1-\frac{1}{\alpha\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2 - (1-\rho)\mathbb{E}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2 \\ &\quad - \left(\rho-\sqrt{\beta}\right)\mathbb{E}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2 + \frac{\eta B}{Cs_{t-1}}\Bigg] \\ =& R(\bar{\boldsymbol{w}}) + \frac{1}{2\eta}\left[\left(1-\frac{1}{\alpha\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2 - (1-\sqrt{\beta})\mathbb{E}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2 + \frac{\eta B}{Cs_{t-1}}\right],\end{aligned} \tag{3.81}$$

where in (b), we have assumed that $\sqrt{\beta} \leq \rho$ (later we will verify that our choice of k ensures such a condition), and have used the fact that projection onto a convex set is non-expansive (which implies that $\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2 \geq \|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2$). Similarly as in 3.7.2.2, we now take a weighted sum over $t = 1, ..., T$, to obtain:

$$\begin{aligned}
&\sum_{t=1}^{T} 2\eta \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \mathbb{E}[(1-\rho)R(\boldsymbol{w}_t) + \rho R(\boldsymbol{v}_t) - R(\bar{\boldsymbol{w}})] \\
\leq &\sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \left[\left(1-\frac{1}{\alpha\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - (1-\sqrt{\beta})\mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 + \frac{\eta B}{Cs_{t-1}}\right] \\
= &\sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \left[\left(1-\frac{1}{\alpha\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - (1-\sqrt{\beta})\mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2\right] \\
&+ \sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \frac{\eta B}{Cs_{t-1}} \\
= &(1-\sqrt{\beta})\sum_{t=1}^{T} \left[\left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t+1} \mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2\right] \\
&+ \sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \frac{\eta B}{Cs_{t-1}} \\
\overset{(a)}{=} &(1-\sqrt{\beta})\left[\left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T} \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 - \mathbb{E}\|\boldsymbol{w}_T - \bar{\boldsymbol{w}}\|^2\right] + \sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \frac{\eta B}{Cs_{t-1}} \\
\leq &(1-\sqrt{\beta})\left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T} \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \frac{\eta B}{Cs_{t-1}} \\
\leq &\left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T} \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \frac{\eta B}{Cs_{t-1}},
\end{aligned} \tag{3.82}$$

where (a) follows from simplifying the telescopic sum.

We now choose k and s_t as follows: we choose $k \geq 4\frac{1}{\rho^2}\alpha^2\kappa_s^2\bar{k}$, which implies that: $\rho \geq \sqrt{\beta}$ (thereby verifying the assumption made earlier), and that:

$$\begin{aligned}
&\sqrt{\beta} \leq \frac{1}{2\alpha\frac{1}{\rho}\kappa_s} \\
&\implies \sqrt{\beta} \leq \frac{1}{2\alpha\frac{1}{\rho}\kappa_s - 1} \\
&\implies 1 - \sqrt{\beta} \geq 1 - \frac{1}{2\alpha\frac{1}{\rho}\kappa_s - 1} = \frac{2\alpha\frac{1}{\rho}\kappa_s - 2}{2\alpha\frac{1}{\rho}\kappa_s - 1} = \frac{1-\frac{1}{\alpha\frac{1}{\rho}\kappa_s}}{1-\frac{1}{2\alpha\frac{1}{\rho}\kappa_s}} \overset{(a)}{\geq} \frac{1-\frac{1}{\alpha\kappa_s}}{1-\frac{1}{2\alpha\frac{1}{\rho}\kappa_s}}
\end{aligned}$$

$$\implies \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right) \leq 1 - \frac{1}{2\alpha\frac{1}{\rho}\kappa_s}, \tag{3.83}$$

where (a) follows from the fact that $\rho \leq 1$.

And we now choose $s_t := \left\lceil \frac{\tau}{\omega^t} \right\rceil$, with $\omega := 1 - \frac{1}{4\alpha\frac{1}{\rho}\kappa_s}$ and $\tau := \frac{\eta B}{C}$.

Let us call $\nu := 1 - \frac{1}{2\alpha\frac{1}{\rho}\kappa_s}$. Note that we have:

$$\nu \leq \omega. \tag{3.84}$$

And that we have the inequality below:

$$\frac{\nu}{\omega} = \frac{1-\frac{1}{2\alpha\frac{1}{\rho}\kappa_s}}{1-\frac{1}{4\alpha\frac{1}{\rho}\kappa_s}} = \frac{4\alpha\frac{1}{\rho}\kappa_s - 2}{4\alpha\frac{1}{\rho}\kappa_s - 1} = 1 - \frac{1}{4\alpha\frac{1}{\rho}\kappa_s - 1} \leq 1 - \frac{1}{4\alpha\frac{1}{\rho}\kappa_s} = \omega. \tag{3.85}$$

This allows us to simplify equation 3.82 into:

$$\begin{aligned}
&\mathbb{E}\sum_{t=1}^{T} 2\eta \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} [(1-\rho)R(\boldsymbol{w}_t) + \rho R(\boldsymbol{v}_t) - R(\bar{\boldsymbol{w}})] \\
&\leq \nu^T \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \sum_{t=1}^{T} \nu^{T-t}\omega^{t-1} \\
&= \nu^T \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\omega^T}{\omega}\sum_{t=1}^{T}\left(\frac{\nu}{\omega}\right)^{T-t} \\
&= \nu^T \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\omega^T}{\omega}\frac{1-\left(\frac{\nu}{\omega}\right)^T}{1-\left(\frac{\nu}{\omega}\right)} \qquad (3.86) \\
&\leq \nu^T \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\omega^T}{\omega}\frac{1}{1-\left(\frac{\nu}{\omega}\right)} \\
&\overset{(a)}{\leq} \nu^T \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\omega^T}{\omega}\frac{1}{1-\omega} \\
&\overset{(b)}{\leq} \nu^T \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\omega^T\frac{1}{1-\omega} \\
&\overset{(c)}{\leq} \omega^T \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\omega^T\frac{1}{1-\omega} \\
&\overset{(d)}{\leq} \frac{\omega^T}{1-\omega} \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\omega^T\frac{1}{1-\omega} \\
&= \frac{\omega^T}{1-\omega}\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right) \\
&= 4\alpha\frac{1}{\rho}\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right), \qquad (3.87)
\end{aligned}$$

where in the left hand side we have used the linearity of expectation, and where (a) uses equation 3.85, (b) uses the fact that $\frac{1}{\omega} = \frac{1}{1-\frac{1}{4\alpha\frac{1}{\rho}\kappa_s}} \leq \frac{1}{1-\frac{1}{4}} = \frac{4}{3}$ (since $\kappa_s \geq 1$ and $\alpha \geq 1$ (indeed, from the theorem's assumption $\alpha = \frac{C}{L_s} + 1$ with $C > 0$), so consequently $\alpha\frac{1}{\rho} \geq 1$), (c) uses equation 3.84, and (d) uses the fact that $\omega < 1$ so $1 < \frac{1}{1-\omega}$.

Let us now normalize the above inequality:

$$\mathbb{E}\frac{\sum_{t=1}^{T} 2\eta\left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}(1-\rho)R(\boldsymbol{w}_t)+\rho R(\boldsymbol{v}_t)}{\sum_{t=1}^{T} 2\eta\left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}} \leq R(\bar{\boldsymbol{w}}) + \frac{4\alpha\frac{1}{\rho}\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\right)}{\sum_{t=1}^{T} 2\eta\left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}}. \tag{3.88}$$

The left hand side above is a weighted sum, which is an upper bound on the smallest term of the sum.

Regarding the right hand side, we can simplify it using the fact that $0 < \left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)$, and therefore:

$$\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \geq 1. \tag{3.89}$$

Therefore, we obtain:

$$\begin{aligned}\mathbb{E}\min_{t\in\{1,..,T\}}(1-\rho)R(\boldsymbol{w}_t)+\rho R(\boldsymbol{v}_t)-R(\bar{\boldsymbol{w}}) &\leq \frac{4\alpha\frac{1}{\rho}\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\right)}{2\eta}\\ &= 2\alpha^2\frac{1}{\rho}L_s\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\right).\end{aligned} \tag{3.90}$$

We denote by ε_T the right-hand side above:

$$\varepsilon_T = 2\alpha^2\frac{1}{\rho}L_s\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\right). \tag{3.91}$$

We now proceed similarly as in the proof of Theorem 4 above. Recall that we have assumed in the Assumptions of Theorem 7, without loss of generality, that R is non-negative, which implies that $R(\boldsymbol{v}_t) \geq 0$. Plugging this in equation 3.90 implies that:

$$\mathbb{E}\min_{t\in[T]} R(\boldsymbol{w}_t) \leq \frac{1}{1-\rho}R(\bar{\boldsymbol{w}}) + \frac{\varepsilon_T}{1-\rho} \leq (1+2\rho)R(\bar{\boldsymbol{w}}) + \frac{\varepsilon_T}{1-\rho}. \tag{3.92}$$

Plugging the change of variable $\varepsilon'_T = \frac{\varepsilon_T}{1-\rho}$ into equation 3.92 above, we obtain that:

$$\mathbb{E}\min_{t\in[T]} R(\boldsymbol{w}_t) \leq (1+2\rho)R(\bar{\boldsymbol{w}}) + \varepsilon'_T. \tag{3.93}$$

Further, consider an ideal case where $\bar{\boldsymbol{w}}$ is a global minimizer of R over $\mathcal{B}_0(k) := \{\boldsymbol{w} : \|\boldsymbol{w}\|_0 \leq k\}$. Then $R(\boldsymbol{v}_t) \geq R(\bar{\boldsymbol{w}})$ is always true for all $t \geq 1$. It follows that the bound in equation 3.92 yields:

$$\mathbb{E} \min_{t\in[T]} \{(1-\rho)R(\boldsymbol{w}_t) + \rho R(\bar{\boldsymbol{w}})\} \leq \mathbb{E} \min_{t\in[T]} \{(1-\rho)R(\boldsymbol{w}_t) + \rho R(\boldsymbol{v}_t)\} \leq R(\bar{\boldsymbol{w}}) + \varepsilon_T,$$

which implies: $\mathbb{E} \min_{t\in[T]} R(\boldsymbol{w}_t) \leq R(\bar{\boldsymbol{w}}) + \frac{\varepsilon_T}{1-\rho}$. In this case, we can simply set $\rho = 0.5$, and define $\varepsilon'_T = \frac{\varepsilon_T}{1-\rho} = 2\varepsilon_T$ similarly as above.. The proof is completed. □

3.7.3 Proof of Corollary 3

Proof of Corollary 3. We proceed similarly as in the proof of Corollary 5 in Section 3.7.2.2:

Let $\varepsilon \in \mathbb{R}_+^*$. Let us find T to ensure that $\mathbb{E} \min_{t\in\{1,..,T\}} (1-\rho)R(\boldsymbol{w}_t) + \rho R(\boldsymbol{v}_t) - R(\bar{\boldsymbol{w}}) \leq \varepsilon$ This will be enforced if:

$$\begin{aligned}
&2\alpha^2 \frac{1}{\rho} L_s \kappa_s \omega^T \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right) \leq \varepsilon \\
&\iff T \log(\omega) \leq \log\left(\frac{\varepsilon}{2\alpha^2 \frac{1}{\rho} L_s \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}\right) \\
&\iff T \geq \frac{1}{\log(\frac{1}{\omega})} \log\left(\frac{2\alpha^2 \frac{1}{\rho} L_s \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{\varepsilon}\right).
\end{aligned} \tag{3.94}$$

Therefore, let us take:

$$T := \left\lceil \frac{1}{\log(\frac{1}{\omega})} \log\left(\frac{2\alpha^2 \frac{1}{\rho} L_s \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{\varepsilon}\right) \right\rceil. \tag{3.95}$$

We can now derive the #IFO and #HT. First, we have one hard-thresholding operation at each iteration, therefore #HT$= T$. Using the fact that $\frac{1}{\log(\frac{1}{\omega})} = \frac{1}{-\log(\omega)} = \frac{1}{-\log(1-\frac{1}{4\alpha\frac{1}{\rho}\kappa_s})} \leq \frac{1}{\frac{1}{4\alpha\frac{1}{\rho}\kappa_s}} = 4\alpha\frac{1}{\rho}\kappa_s$ (since by property of the logarithm, for all $x \in (-\infty, -1) : \log(1-x) \leq -x$), we obtain that #HT $= \mathcal{O}(\kappa_s \log\left(\frac{1}{\varepsilon}\right))$.

We now turn to computing the #IFO. At each iteration t we have s_t gradient evaluations, therefore:

$$\begin{aligned}
\#\text{IFO} &= \sum_{t=0}^{T-1} s_t \\
&\leq \sum_{t=0}^{T-1} \left(\frac{\tau}{\omega^t} + 1\right)
\end{aligned}$$

$$
\begin{aligned}
&= T + \tau \frac{\left(\frac{1}{\omega}\right)^{T} - 1}{\frac{1}{\omega} - 1} \\
&\leq T + \frac{\tau}{\frac{1}{\omega} - 1} \left(\frac{1}{\omega}\right)^{T} \\
&= T + \frac{\tau}{\frac{1}{\omega} - 1} \exp\left(T \log\left(\frac{1}{\omega}\right)\right) \\
&\overset{(a)}{\leq} 1 + \frac{1}{\log(\frac{1}{\omega})} \log\left(\frac{2\alpha^2 \frac{1}{\rho} L_s \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{\varepsilon}\right) \\
&\quad + \frac{\tau}{\frac{1}{\omega} - 1} \exp\left(\log\left(\frac{1}{\omega}\right) \left[\frac{1}{\log(\frac{1}{\omega})} \log\left(\frac{2\alpha^2 \frac{1}{\rho} L_s \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{\varepsilon}\right) + 1\right]\right) \\
&= 1 + \frac{1}{\log(\frac{1}{\omega})} \log\left(\frac{2\alpha^2 \frac{1}{\rho} L_s \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{\varepsilon}\right) + \frac{\frac{\tau}{\omega}}{\frac{1}{\omega} - 1} \frac{2\alpha^2 \frac{1}{\rho} L_s \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{\varepsilon} \\
&= 1 + \frac{1}{\log(\frac{1}{\omega})} \log\left(\frac{2\alpha^2 \frac{1}{\rho} L_s \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{\varepsilon}\right) + \frac{\tau}{1 - \omega} \frac{2\alpha^2 \frac{1}{\rho} L_s \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{\varepsilon} \\
&= 1 + \frac{1}{\log(\frac{1}{\omega})} \log\left(\frac{2\alpha^2 \frac{1}{\rho} L_s \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{\varepsilon}\right) + \tau \frac{8\alpha^3 \frac{1}{\rho^2} L_s \kappa_s^2 \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{\varepsilon} \\
&\overset{(b)}{=} 1 + \frac{1}{\log(\frac{1}{\omega})} \log\left(\frac{2\alpha^2 \frac{1}{\rho} L_s \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{\varepsilon}\right) \\
&\quad + \frac{B}{\alpha L_s} \frac{1}{L_s(\alpha - 1)} \frac{8\alpha^3 \frac{1}{\rho^2} L_s}{\varepsilon} \frac{L_s}{\nu_s} \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right) \\
&= 1 + \frac{1}{\log(\frac{1}{\omega})} \log\left(\frac{2\alpha^2 \frac{1}{\rho} L_s \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{\varepsilon}\right) + \frac{8B\alpha^2 \frac{1}{\rho^2} \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\right)}{(\alpha - 1)\nu_s} \frac{1}{\varepsilon},
\end{aligned}
\tag{3.96}
$$

where (a) follows from equation 3.95, and for (b) we recall that $\tau = \frac{\eta B}{C}$, $\eta = \frac{1}{\alpha L_s}$ and $C = L_s(\alpha - 1)$. Therefore, overall, the IFO complexity is in $\mathcal{O}(\frac{\kappa_s}{\nu_s \varepsilon})$.

□

3.7.4 Proof of Theorems 8 and 9

Our proof for Theorem 9 is similar to the one for Theorem 7, though we needed to refine some results from [46] to properly express the variance of the ZO gradient estimator and incorporate it into the telescopic sum. Before proving the main Theorem 9, below we provide several intermediary results needed for the proof of Theorem 9. Then, the proof of Theorem 7 will be provided in Section 3.7.4.3.

3.7.4.1 Useful Lemmas

We first recall the following results from [46]:

Proposition 2 (Proposition 1 (i) [46])**.** *Let us consider any support $F \subseteq [d]$ of size s ($|F| = s$). For the Z0 gradient estimator $\boldsymbol{g}_t$ in Algorithm 9 at $\boldsymbol{w}_t$, with q_t random directions, and random supports of size s_2, and assuming that R is (L_{s_2}, s_2)-RSS' , we have, with $[\boldsymbol{u}]_F$ denoting the hard thresholding of a vector $\boldsymbol{u}$ on F (that is, we set all coordinates not in F to 0):*

$$\|[\mathbb{E}\boldsymbol{g}_t]_F - [\nabla R(\boldsymbol{w}_t)]_F\|^2 \leq \varepsilon_\mu \mu^2 \tag{3.97}$$

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

Proof of Proposition 2. Proof in [46]. □

Lemma 17 (Lemma C.2 [46])**.** *For any (L_{s_2}, s_2)-RSS' function R, using the gradient estimator $\boldsymbol{g}_t$ defined in Algorithm 9 with $q_t = 1$, we have, for any support $F \subseteq [d]$, with $|F| = s$, and $F^c := [d] \setminus F$:*

$$\mathbb{E}\|[\boldsymbol{g}_t]_F\|^2 = \varepsilon_F \|[\nabla R(\boldsymbol{w}_t)]_F\|^2 + \varepsilon_{F^c} \|[\nabla R(\boldsymbol{w}_t)]_{F^c}\|^2 + \varepsilon_{abs}\mu^2 \tag{3.98}$$

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 of Lemma 17. Proof in [46]. □

We now use the above lemma to bound the variance of the zeroth-order gradient estimator $\boldsymbol{g}_t$.

Lemma 18. *The gradient estimator $\boldsymbol{g}_t$ defined in Algorithm 9 verifies the following properties for any $q_t \in \mathbb{N}^*$:*

$$\mathbb{E}\|[\boldsymbol{g}_t]_F - \mathbb{E}[\boldsymbol{g}_t]_F\|^2 \leq \frac{\varepsilon_F}{q_t} \|\nabla R(\boldsymbol{w})\|^2 + \frac{\varepsilon_{abs}}{q_t}\mu^2 \tag{3.99}$$

with ε_F and ε_{abs} defined above in Lemma 17

Proof of Lemma 18. If $q_t = 1$, we have:

$$\begin{aligned}
\mathbb{E}\|[\boldsymbol{g}_t]_F - \mathbb{E}[\boldsymbol{g}_t]_F\|^2 &\overset{(a)}{=} \mathbb{E}\|[\boldsymbol{g}_t]_F\|^2 - \|[\mathbb{E}\boldsymbol{g}]_F\|^2 \\
&\leq \mathbb{E}\|[\boldsymbol{g}_t]_F\|^2 \\
&\overset{(3.98)}{\leq} \varepsilon_F\|[\nabla R(\boldsymbol{w})]_F\|^2 + \varepsilon_{F^c}\|[\nabla R(\boldsymbol{w})]_{F^c}\|^2 + \varepsilon_{\text{abs}}\mu^2 \\
&\overset{(b)}{\leq} \varepsilon_F\|\nabla R(\boldsymbol{w})\|^2 + \varepsilon_{abs}\mu^2,
\end{aligned} \tag{3.100}$$

where (a) follows from the bias-variance formula $\mathbb{E}\|X - E[X]\|_2^2 = \mathbb{E}\|X\|_2^2 - \|\mathbb{E}X\|_2^2$ for a multidimensional random variable X, and (b) follows from the fact that

$$\varepsilon_F = \frac{2d}{s_2+2}\left(\frac{s(s_2-1)}{d-1} + 3 - \frac{s_2-1}{d}\right) > \frac{2d}{s_2+2}\left(\frac{s(s_2-1)}{d-1}\right) = \varepsilon_{F^c} \tag{3.101}$$

(since $s_2 \leq d$), and since $\|[\nabla R(\boldsymbol{w})]_F\|^2 + \|[\nabla R(\boldsymbol{w})]_{F^c}\|^2 = \|\nabla R(\boldsymbol{w})\|^2$ (by definition of the Euclidean norm).

Now, if $q_t \geq 1$, we know that the variance of an average of q_t i.i.d. realizations of a random variable of total variance σ^2 is $\frac{\sigma^2}{q_t}$ (and its expected value remains the same by linearity of expectation): indeed, for any random multidimensional random variable X, for which we consider the q i.i.d. random variables X_i of same distribution, we have:

$$\begin{aligned}
\mathbb{E}\left\|\frac{1}{q_t}\sum_{i=1}^{q_t} X_i - \mathbb{E}\left[\frac{1}{q_t}\sum_{i=1}^{q_t} X_i\right]\right\|_2^2 &= \mathbb{E}\left\|\frac{1}{q_t}\sum_{i=1}^{q_t}(X_i - \mathbb{E}X_i)\right\|_2^2 \\
&= \frac{1}{q_t^2}\left(\sum_{i=1}^{q_t}(X_i - \mathbb{E}X_i)\right)^\top \left(\sum_{i=1}^{q_t}(X_i - \mathbb{E}X_i)\right) \\
&\overset{(a)}{=} \frac{1}{q_t^2}\sum_{i=1}^{q_t}\|X_i - \mathbb{E}X_i\|_2^2 \\
&= \frac{1}{q_t^2}\sum_{i=1}^{q_t}\|X - \mathbb{E}X\|_2^2 \\
&= \frac{1}{q_t^2} q_t \|X - \mathbb{E}X\|_2^2 \\
&= \frac{1}{q_t}\|X - \mathbb{E}X\|_2^2,
\end{aligned} \tag{3.102}$$

where (a) follows from the fact that X_i are i.i.d hence for $i \neq j$: $\mathrm{Cov}(X_i, X_j) = \mathbb{E}(X_i - \mathbb{E}X_i)^\top(X_j - \mathbb{E}X_j) = 0$. Applying this to the random variable which realizations are $[\boldsymbol{g}_t]_F$, this concludes the proof. □

3.7.4.2 Proof of Theorem 8

Below we now first present some results (and their proofs) for the convergence of Algorithm 9 without the additional constraint, which is needed for the proof of Theorem 9, and also, as a byproduct, provides, up to our knowledge, the first convergence guarantee in objective value without system error for a zeroth-order hard-thresholding algorithm.

Proof of Theorem 8. Let us denote for simplicity: $C_1 := \frac{\varepsilon_F}{q_t}$, $C_2 := \frac{\varepsilon_{abs}}{q_t}$, and $C_3 := \varepsilon_\mu \mu^2$. Moreover, let us denote $F := \mathrm{supp}(\boldsymbol{w}_t) \cup \mathrm{supp}(\boldsymbol{w}_{t-1}) \cup \mathrm{supp}(\bar{\boldsymbol{w}})$, where supp denotes the support of a vector, i.e. the set of coordinates of its non-zero components. Note that therefore we have $|F| \leq 2k + \bar{k} \leq 3k$. In addition $[\boldsymbol{u}]_F$ denotes the thresholding of $\boldsymbol{u}$ to the support F, that is, the vector $\boldsymbol{u}$ with its components that are not in F set to 0.

The fact that R is $(L_{s'}, s')$-RSS', therefore also $(L_{s'}, s)$-RSS', implies from the remark in 11 that it is also $(L_{s'}, s)$-RSS, therefore:

$$
\begin{aligned}
&R(\boldsymbol{w}_t)\\
\leq& R(\boldsymbol{w}_{t-1}) + \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle + \frac{L_{s'}}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2\\
=& R(\boldsymbol{w}_{t-1}) + \langle \boldsymbol{g}_{t-1}, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle + \frac{L_{s'}}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 + \langle \nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle\\
=& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\left[\|\boldsymbol{w}_t - (\boldsymbol{w}_{t-1} - \eta \boldsymbol{g}_{t-1})\|^2 - \eta^2\|\boldsymbol{g}_{t-1}\|^2 - \|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2\right] + \frac{L_{s'}}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2\\
&+ \langle \nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle \qquad (3.103)\\
=& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\|\boldsymbol{w}_t - (\boldsymbol{w}_{t-1} - \eta \boldsymbol{g}_{t-1})\|^2 - \frac{\eta}{2}\|\boldsymbol{g}_{t-1}\|^2 + \left[\frac{L_{s'} - \frac{1}{\eta}}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2\\
&+ \langle [\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle\\
\overset{(a)}{\leq}& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\left[\|\bar{\boldsymbol{w}} - (\boldsymbol{w}_{t-1} - \eta \boldsymbol{g}_{t-1})\|^2 - (1 - \sqrt{\beta})\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2\right] - \frac{\eta}{2}\|\boldsymbol{g}_{t-1}\|^2\\
&+ \left[\frac{L_{s'} - \frac{1}{\eta}}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 + \langle [\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle\\
=& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\left[\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 + \eta^2\|\boldsymbol{g}_{t-1}\|^2 - 2\langle \eta \boldsymbol{g}_{t-1}, \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle\right] - \frac{1}{2\eta}(1 - \sqrt{\beta})\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2\\
&- \frac{\eta}{2}\|\boldsymbol{g}_{t-1}\|^2 + \left[\frac{L_{s'} - \frac{1}{\eta}}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 + \langle [\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle\\
=& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\left[\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - 2\langle \eta \boldsymbol{g}_{t-1}, \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle\right] - \frac{1}{2\eta}(1 - \sqrt{\beta})\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2\\
&+ \left[\frac{L_{s'} - \frac{1}{\eta}}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 + \langle [\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle\\
\overset{(b)}{=}& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \boldsymbol{g}_{t-1}, \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle - \frac{1}{2\eta}(1 - \sqrt{\beta})\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2\\
&+ \left[\frac{L_{s'} - \frac{1}{\eta} + C}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 + \frac{1}{2C}\|[\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F\|^2\\
=& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle + \langle [\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F, \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle\\
&- \frac{1}{2\eta}(1 - \sqrt{\beta})\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 + \left[\frac{L_{s'} - \frac{1}{\eta} + C}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 + \frac{1}{2C}\|[\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F\|^2,\\
&\qquad (3.104)
\end{aligned}
$$

where (a) follows from Lemma 11 and (b) follows from the inequality $\langle a, b\rangle \leq \frac{C}{2}a^2 + \frac{1}{2C}b^2$, for any $(a, b) \in (\mathbb{R}^d)^2$ with $C > 0$ an arbitrary strictly positive constant.

Let us now choose $\eta := \frac{1}{L_{s'}+C}$: therefore the term $\left[\frac{L_{s'}-\frac{1}{\eta}+C}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2$ above is 0. We now take the conditional expectation (conditioned on w_{t-1}, which is the random variable which realizations are $\boldsymbol{w}_{t-1}$), on both sides, and from Lemma 16 we obtain the inequality below (we slightly abuse notations and denote $\mathbb{E}[\cdot|\mathrm{w}_{t-1} = \boldsymbol{w}_{t-1}]$ by $\mathbb{E}[\cdot|\boldsymbol{w}_{t-1}]$):

$$\begin{aligned}
&\mathbb{E}[R(\boldsymbol{w}_t)|\boldsymbol{w}_{t-1}] && (3.105)\\
\leq& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle\\
&- \frac{1}{2\eta}(1-\sqrt{\beta})\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] + \langle [\nabla R(\boldsymbol{w}_{t-1}) - \mathbb{E}\left[\boldsymbol{g}_{t-1}|\boldsymbol{w}_{t-1}\right]]_F, \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle\\
&+ \mathbb{E}\left[\frac{1}{2C}\|[\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F\|^2|\boldsymbol{w}_{t-1}\right]\\
\overset{(a)}{\leq}& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle\\
&- \frac{1}{2\eta}(1-\sqrt{\beta})\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] + \frac{G}{2}\|[\nabla R(\boldsymbol{w}_{t-1}) - \mathbb{E}[\boldsymbol{g}_{t-1}|\boldsymbol{w}_{t-1}]]_F\|^2\\
&+ \frac{1}{2G}\|\boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\|^2 + \frac{1}{2C}\mathbb{E}\left[\|\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}\|^2|\boldsymbol{w}_{t-1}\right]\\
=& R(\boldsymbol{w}_{t-1}) + \left[\frac{1}{2\eta} + \frac{1}{2G}\right]\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle\\
&- \frac{1}{2\eta}(1-\sqrt{\beta})\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] + \frac{G}{2}\|[\nabla R(\boldsymbol{w}_{t-1}) - \mathbb{E}[\boldsymbol{g}_{t-1}|\boldsymbol{w}_{t-1}]]_F\|^2\\
&+ \frac{1}{2C}\mathbb{E}\left[\|[\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F\|^2|\boldsymbol{w}_{t-1}\right]\\
\overset{(b)}{\leq}& R(\boldsymbol{w}_{t-1}) + \left[\frac{1}{2\eta} + \frac{1}{2G}\right]\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle\\
&- \frac{1}{2\eta}(1-\sqrt{\beta})\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] + \frac{G}{2}\|[\nabla R(\boldsymbol{w}_{t-1}) - \mathbb{E}[\boldsymbol{g}_{t-1}|\boldsymbol{w}_{t-1}]]_F\|^2\\
&+ \frac{1}{2C}\left(2\|[\nabla R(\boldsymbol{w}_{t-1}) - \mathbb{E}[\boldsymbol{g}_{t-1}|\boldsymbol{w}_{t-1}]]_F\|^2 + 2\|[\boldsymbol{g}_{t-1} - \mathbb{E}[\boldsymbol{g}_{t-1}|\boldsymbol{w}_{t-1}]]_F\|^2\right)\\
\overset{(3.97)+(3.99)}{\leq}& R(\boldsymbol{w}_{t-1}) + \left[\frac{1}{2\eta} + \frac{1}{2G}\right]\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle\\
&- \frac{1}{2\eta}(1-\sqrt{\beta})\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] + \frac{G}{2}C_3\\
&+ \frac{1}{2C}\left(2C_3 + 2C_1\|\nabla R(\boldsymbol{w}_{t-1})\|^2 + 2C_2\mu^2\right)\\
\overset{(c)}{\leq}& R(\boldsymbol{w}_{t-1}) + \left[\frac{1}{2\eta} + \frac{1}{2G}\right]\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle && (3.106)\\
&- \frac{1}{2\eta}(1-\sqrt{\beta})\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] + \frac{G}{2}C_3 && (3.107)
\end{aligned}$$

$$+\frac{1}{2C}\left(2C_1\left(2\|\nabla R(\boldsymbol{w}_{t-1})-\nabla R(\bar{\boldsymbol{w}})\|^2+2\|\nabla R(\bar{\boldsymbol{w}})\|^2\right)+2C_2\mu^2+2C_3\right) \tag{3.108}$$

$$\overset{(d)}{\leq} R(\boldsymbol{w}_{t-1})+\left[\frac{1}{2\eta}+\frac{1}{2G}\right]\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2-\langle\nabla R(\boldsymbol{w}_{t-1}),\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\rangle \tag{3.109}$$

$$-\frac{1}{2\eta}(1-\sqrt{\beta})\mathbb{E}\left[\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]+\frac{G}{2}C_3 \tag{3.110}$$

$$+\frac{1}{2C}\left(2C_1\left(2L_{s'}^2\|\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\|^2+2\|\nabla R(\bar{\boldsymbol{w}})\|^2\right)+2C_2\mu^2+2C_3\right) \tag{3.111}$$

$$=R(\boldsymbol{w}_{t-1})+\left[\frac{1}{2\eta}+\frac{1}{2G}+\frac{2C_1L_{s'}^2}{C}\right]\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2-\langle\nabla R(\boldsymbol{w}_{t-1}),\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\rangle \tag{3.112}$$

$$-\frac{1}{2\eta}(1-\sqrt{\beta})\mathbb{E}\left[\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]+\frac{G}{2}C_3+\frac{1}{C}\left(2C_1\|\nabla R(\bar{\boldsymbol{w}})\|^2+C_2\mu^2+C_3\right) \tag{3.113}$$

$$\overset{(e)}{\leq} R(\boldsymbol{w}_{t-1})+\left[\frac{1}{2\eta}+\frac{1}{2G}+\frac{2C_1L_{s'}^2}{C}\right]\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2 \tag{3.114}$$

$$+\left[R(\bar{\boldsymbol{w}})-R(\boldsymbol{w}_{t-1})-\frac{\nu_s}{2}\|\boldsymbol{w}_{t-1}-\bar{\boldsymbol{w}}\|^2\right] \tag{3.115}$$

$$-\frac{1}{2\eta}(1-\sqrt{\beta})\mathbb{E}\left[\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]+\frac{G}{2}C_3+\frac{1}{C}\left(2C_1\|\nabla R(\bar{\boldsymbol{w}})\|^2+C_2\mu^2+C_3\right) \tag{3.116}$$

$$=R(\bar{\boldsymbol{w}})+\left[\frac{\frac{1}{\eta}-\nu_s}{2}+\frac{1}{2G}+\frac{2C_1L_{s'}^2}{C}\right]\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2 \tag{3.117}$$

$$-\frac{1}{2\eta}(1-\sqrt{\beta})\mathbb{E}\left[\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]+\frac{G}{2}C_3+\frac{1}{C}\left(2C_1\|\nabla R(\bar{\boldsymbol{w}})\|^2+C_2\mu^2+C_3\right) \tag{3.118}$$

$$\overset{(f)}{\leq} R(\bar{\boldsymbol{w}})+\left[\frac{\frac{1}{\eta}-\nu_s}{2}+\frac{1}{2G}+\frac{2\varepsilon_FL_{s'}^2}{\tau C}\right]\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2 \tag{3.119}$$

$$-\frac{1}{2\eta}(1-\sqrt{\beta})\mathbb{E}\left[\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]+\frac{G}{2}C_3+\frac{1}{C}\left(2C_1\|\nabla R(\bar{\boldsymbol{w}})\|^2+C_2\mu^2+C_3\right), \tag{3.120}$$

where (a) follows from the inequality $\langle a,b\rangle \leq \frac{G}{2}a^2+\frac{1}{2G}b^2$, for any $(a,b)\in(\mathbb{R}^d)^2$ with $G>0$ an arbitrary strictly positive constant, (b) and (c) follow from the inequality $\|a+b\|^2\leq 2\|a\|^2+2\|b\|^2$ for any $(a,b)\in(\mathbb{R}^d)^2$, (d) follows from the fact that R is $(L_{s'}, s')$-RSS' (Assumption 11 with sparsity level s'), therefore it is also $(L_{s'}, s_2)$-RSS', (e) follows from the RSC condition, and for (f), we recall that $C_1=\frac{\varepsilon_F}{q_t}$, and we define $q_t=\left\lceil\frac{\tau}{\omega^t}\right\rceil$, for some $\omega>1$ and $\tau>0$ that will be chosen later in the proof. Recall that we have chosen $\eta=\frac{1}{L_{s'}+C}$. Let us define $\alpha:=\frac{C}{L_{s'}}+1$. Then $C=(\alpha-1)L_{s'}$, and $\eta=\frac{1}{\alpha L_{s'}}$. Also recall that $\kappa_s=\frac{L_{s'}}{\nu_s}$.

We will now choose the constant G and C, in order to simplify the inequality above, such that it matches as much as possible the structure of the previous proofs:

We will seek to rewrite:
$\left[\frac{\frac{1}{\eta}-\nu_s}{2}+\frac{1}{2G}+\frac{2\frac{\varepsilon_F}{\tau}L_{s'}^2}{C}\right]\left(=\frac{1}{2\eta}\left[1+\frac{1}{G\alpha L_{s'}}+\frac{4L_{s'}^2\frac{\varepsilon_F}{\tau}}{(\alpha-1)\alpha L_{s'}^2}-\frac{1}{\alpha\kappa_s}\right]\right)$, into :
$\frac{1}{2\eta}\left[1-\frac{1}{\alpha'\kappa_s}\right]$ for some $\alpha'>0$ (we will seek $\alpha'\propto\alpha$, with a dimensionless proportionality constant for simplicity).

Therefore, let us choose $G:=\frac{4}{\nu_s}$, which implies:

$$\frac{1}{G\alpha L_{s'}}=\frac{1}{4\alpha\kappa_s}. \tag{3.121}$$

And let us choose $\tau:=\frac{16\kappa_s\varepsilon_F}{(\alpha-1)}$, which implies:

$$\frac{4L_{s'}^2\frac{\varepsilon_F}{\tau}}{(\alpha-1)\alpha L_{s'}^2}=\frac{1}{4\alpha\kappa_s}. \tag{3.122}$$

Therefore, using equations 3.121 and 3.122, we obtain:

$$\begin{aligned}\left[\frac{\frac{1}{\eta}-\nu_s}{2}+\frac{1}{2G}+\frac{2\frac{\varepsilon_F}{\tau}L_{s'}^2}{C}\right]&=\frac{1}{2\eta}\left[1+\frac{1}{G\alpha L_{s'}}+\frac{4L_{s'}^2\frac{\varepsilon_F}{\tau}}{(\alpha-1)\alpha L_{s'}^2}-\frac{1}{\alpha\kappa_s}\right]\\&=\frac{1}{2\eta}\left[1+\frac{1}{4\alpha\kappa_s}+\frac{1}{4\alpha\kappa_s}-\frac{1}{\alpha\kappa_s}\right]\\&=\frac{1}{2\eta}\left[1-\frac{1}{2\alpha\kappa_s}\right]=\frac{1}{2\eta}\left[1-\frac{1}{\alpha'\kappa_s}\right],\end{aligned} \tag{3.123}$$

where for simplicity we have denoted $\alpha'=2\alpha$. We can therefore simplify (3.120) into:

$$\begin{aligned}\mathbb{E}[R(\boldsymbol{w}_t)|\boldsymbol{w}_{t-1}]-R(\bar{\boldsymbol{w}})\leq&\frac{1}{2\eta}\left[\left(1-\frac{1}{\alpha'\kappa_s}\right)\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2-(1-\sqrt{\beta})\mathbb{E}\left[\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]\right.\\&\left.+2\eta\left(\frac{G}{2}C_3+\frac{1}{C}\left(2C_1\|\nabla R(\bar{\boldsymbol{w}})\|^2+C_2\mu^2+C_3\right)\right)\right].\end{aligned} \tag{3.124}$$

We now take the expectation over w_{t-1} of the above inequality (i.e. we take $\mathbb{E}_{\mathrm{w}_{t-1}}[\cdot]$): using the law of total expectation ($\mathbb{E}[\cdot]=\mathbb{E}_{\mathrm{w}_{t-1}}[\mathbb{E}[\cdot|\boldsymbol{w}_{t-1}]]$) we obtain:

$$\begin{aligned}\mathbb{E}R(\boldsymbol{w}_t)-R(\bar{\boldsymbol{w}})\leq&\frac{1}{2\eta}\left[\left(1-\frac{1}{\alpha'\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2-(1-\sqrt{\beta})\mathbb{E}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2\right.\\&\left.+2\eta\left(\frac{G}{2}C_3+\frac{1}{C}\left(2C_1\|\nabla R(\bar{\boldsymbol{w}})\|^2+C_2\mu^2+C_3\right)\right)\right]\end{aligned} \tag{3.125}$$

Let us call $A:=2\eta\left(\frac{G}{2}C_3+\frac{1}{C}\left(2C_1\|\nabla R(\bar{\boldsymbol{w}})\|^2+C_2\mu^2+C_3\right)\right)$ for simplicity. Similarly as in [90], we now take a weighted sum over $t=1,...,T$, to obtain:

$$\sum_{t=1}^{T}2\eta\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}\mathbb{E}[R(\boldsymbol{w}_t)-R(\bar{\boldsymbol{w}})]$$

$$
\begin{aligned}
&\leq \sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \left[\left(1-\frac{1}{\alpha'\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2-(1-\sqrt{\beta})\mathbb{E}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2+A\right]\\
&= \sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \left[\left(1-\frac{1}{\alpha'\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2-(1-\sqrt{\beta})\mathbb{E}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2\right]\\
&\quad+\sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} A\\
&=(1-\sqrt{\beta})\sum_{t=1}^{T}\left[\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t+1}\mathbb{E}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2-\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}\mathbb{E}\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2\right]\\
&\quad+\sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} A\\
&\overset{(a)}{=}(1-\sqrt{\beta})\left[\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2-\mathbb{E}\|\boldsymbol{w}_T-\bar{\boldsymbol{w}}\|^2\right]+\sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} A\\
&\leq(1-\sqrt{\beta})\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} A\\
&\leq\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} A\\
&=\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2\\
&\quad+\sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} 2\eta\left(\frac{G}{2}C_3+\frac{1}{C}\left(2C_1\|\nabla R(\bar{\boldsymbol{w}})\|^2+C_2\mu^2+C_3\right)\right)\\
&=\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2\\
&\quad+\sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} 2\eta\left(\frac{G}{2}C_3+\frac{1}{C}\left(2\frac{\varepsilon_F}{q_t}\|\nabla R(\bar{\boldsymbol{w}})\|^2+\frac{\varepsilon_{abs}\mu^2}{q_t}+C_3\right)\right)\\
&=\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}\frac{2\eta}{q_t}\left(\frac{2\varepsilon_F\|\nabla R(\bar{\boldsymbol{w}})\|^2+\varepsilon_{abs}\mu^2}{C}\right)\\
&\quad+\sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} 2\eta C_3\left(\frac{G}{2}+\frac{1}{C}\right)\\
&=\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\sum_{t=1}^{T} \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}\frac{2\eta}{q_t}\left(\frac{2\varepsilon_F\|\nabla R(\bar{\boldsymbol{w}})\|^2}{C}\right)
\end{aligned}
$$

$$
\begin{aligned}
&+\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} 2\eta\mu^2\left(\varepsilon_\mu\left(\frac{G}{2}+\frac{1}{C}\right)+\frac{\varepsilon_{abs}}{Cq_t}\right)\\
\leq&\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}\frac{2\eta}{q_t}\left(\frac{2\varepsilon_F\|\nabla R(\bar{\boldsymbol{w}})\|^2}{C}\right)\\
&+\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} 2\eta\mu^2\left(\varepsilon_\mu\left(\frac{G}{2}+\frac{1}{C}\right)+\frac{\varepsilon_{abs}}{C}\right),
\end{aligned}
\tag{3.126}
$$

where (a) follows from simplifying the telescopic sum. Let us denote for simplicity $\zeta := \frac{2\eta(2\varepsilon_F\|\nabla R(\bar{\boldsymbol{w}})\|^2)}{C} = \frac{4\eta\varepsilon_F\|\nabla R(\bar{\boldsymbol{w}})\|^2}{C}$ and $Z := \varepsilon_\mu\left(\frac{G}{2}+\frac{1}{C}\right)+\frac{\varepsilon_{abs}}{C}$.

We now choose k and q_t as follows: we choose $k \geq 4\alpha'^2\kappa_s^2\bar{k}$, which implies that:

$$
\begin{aligned}
&\sqrt{\beta}\leq\frac{1}{2\alpha'\kappa_s}\\
&\implies \sqrt{\beta}\leq\frac{1}{2\alpha'\kappa_s-1}\\
&\implies 1-\sqrt{\beta}\geq 1-\frac{1}{2\alpha'\kappa_s-1}=\frac{2\alpha'\kappa_s-2}{2\alpha'\kappa_s-1}=\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\frac{1}{2\alpha'\kappa_s}}\\
&\implies \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)\leq 1-\frac{1}{2\alpha'\kappa_s}.
\end{aligned}
\tag{3.127}
$$

We recall that we previously defined $q_t = \left\lceil\frac{\tau}{\omega^t}\right\rceil$, with $\tau := \frac{16\kappa_s\varepsilon_F}{(\alpha-1)}$. We now set the value of ω, to $\omega := 1-\frac{1}{4\alpha'\kappa_s}$.

Let us call $\nu := 1-\frac{1}{2\alpha'\kappa_s}$. Note that we have:

$$\nu\leq\omega. \tag{3.128}$$

And that we have the inequality below:

$$\frac{\nu}{\omega}=\frac{1-\frac{1}{2\alpha'\kappa_s}}{1-\frac{1}{4\alpha'\kappa_s}}=\frac{4\alpha'\kappa_s-2}{4\alpha'\kappa_s-1}=1-\frac{1}{4\alpha'\kappa_s-1}\leq 1-\frac{1}{4\alpha'\kappa_s}=\omega. \tag{3.129}$$

This allows us to simplify equation 3.126 into:

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}2\eta\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}[R(\boldsymbol{w}_t)-R(\bar{\boldsymbol{w}})]\right]\\
&\leq\nu^T\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{\zeta}{\tau}\sum_{t=1}^{T}\nu^{T-t}\omega^{t-1}+\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}2\eta Z\mu^2
\end{aligned}
$$

$$\begin{aligned}
&= \nu^T\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\zeta}{\tau}\frac{\omega^T}{\omega}\sum_{t=1}^{T}\left(\frac{\nu}{\omega}\right)^{T-t} + \sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} 2\eta Z\mu^2 \\
&= \nu^T\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\zeta}{\tau}\frac{\omega^T}{\omega}\frac{1-\left(\frac{\nu}{\omega}\right)^T}{1-\left(\frac{\nu}{\omega}\right)} + \sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} 2\eta Z\mu^2 \\
&\le \nu^T\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\zeta}{\tau}\frac{\omega^T}{\omega}\frac{1}{1-\left(\frac{\nu}{\omega}\right)} + \sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} 2\eta Z\mu^2 \\
&\overset{(a)}{\le} \nu^T\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\zeta}{\tau}\frac{\omega^T}{\omega}\frac{1}{1-\omega} + \sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} 2\eta Z\mu^2 \\
&\overset{(b)}{\le} \nu^T\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\zeta}{\tau}\frac{4}{3}\omega^T\frac{1}{1-\omega} + \sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} 2\eta Z\mu^2 \\
&\overset{(c)}{\le} \omega^T\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\zeta}{\tau}\frac{4}{3}\omega^T\frac{1}{1-\omega} + \sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} 2\eta Z\mu^2 \\
&\overset{(d)}{\le} \frac{\omega^T}{1-\omega}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\zeta}{\tau}\frac{4}{3}\omega^T\frac{1}{1-\omega} + \sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} 2\eta Z\mu^2 \\
&= \frac{\omega^T}{1-\omega}\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\zeta}{\tau}\frac{4}{3}\right) + \sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} 2\eta Z\mu^2 \\
&= 4\alpha'\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{\zeta}{\tau}\frac{4}{3}\right) + \sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} 2\eta Z\mu^2, \qquad (3.130)
\end{aligned}$$

where in the left hand side we have used the linearity of expectation, and where (a) uses equation 3.129, (b) uses the fact that $\frac{1}{\omega} = \frac{1}{1-\frac{1}{4\alpha'\kappa_s}} \le \frac{1}{1-\frac{1}{4}} = \frac{4}{3}$ (since $\kappa_s \ge 1$ and $\alpha' \ge 1$ (indeed, we have $\alpha' = 2\alpha = 2(\frac{C}{L_{s'}} + 1)$ with $C > 0$)), (c) uses equation 3.128, and (d) uses the fact that $\omega < 1$ so $1 < \frac{1}{1-\omega}$.

Let us now normalize the above inequality:

$$\mathbb{E}\frac{\sum_{t=1}^{T} 2\eta\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} R(\boldsymbol{w}_t)}{\sum_{t=1}^{T} 2\eta\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}} \le R(\bar{\boldsymbol{w}}) + \frac{4\alpha'\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\frac{\zeta}{\tau}\right)}{\sum_{t=1}^{T} 2\eta\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}} + Z\mu^2. \qquad (3.131)$$

The left hand side above is a weighted sum, which is an upper bound on the smallest term of the sum.

Regarding the right hand side, we can simplify it using the fact that $0 < \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)$, and therefore:

$$\sum_{t=1}^{T} \left(\frac{1 - \frac{1}{\alpha' \kappa_s}}{1 - \sqrt{\beta}} \right)^{T-t} \geq 1. \tag{3.132}$$

Therefore, we obtain:

$$\begin{aligned} \mathbb{E} \min_{t \in \{1,..,T\}} R(\boldsymbol{w}_t) - R(\bar{\boldsymbol{w}}) &\leq \frac{4\alpha' \kappa_s \omega^T \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3} \frac{\zeta}{\tau} \right)}{2\eta} + Z\mu^2 \\ &= 4\alpha^2 L_{s'} \kappa_s \omega^T \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3} \frac{\zeta}{\tau} \right) + Z\mu^2. \end{aligned} \tag{3.133}$$

Which can be simplified into the expression below, using the definition of $\hat{\boldsymbol{w}}_T$:

$$\mathbb{E} R(\hat{\boldsymbol{w}}_T) - R(\bar{\boldsymbol{w}}) \leq 4\alpha^2 L_{s'} \kappa_s \omega^T \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3} \frac{\zeta}{\tau} \right) + Z\mu^2. \tag{3.134}$$

To simplify the above result, we recall the assumptions made earlier on: we have chosen $\tau = \frac{16\kappa_s \varepsilon_F}{(\alpha - 1)}$, and $G = \frac{4}{\nu_s}$.

Therefore, to sum up, we have:

$$Z = \varepsilon_\mu \left(\frac{G}{2} + \frac{1}{C} \right) + \frac{\varepsilon_{abs}}{C} = \varepsilon_\mu \left(\frac{2}{\nu_s} + \frac{1}{C} \right) + \frac{\varepsilon_{abs}}{C}. \tag{3.135}$$

$$\omega = 1 - \frac{1}{4\alpha' \kappa_s} = 1 - \frac{1}{8\alpha \kappa_s} \tag{3.136}$$

$$\zeta = \frac{4\eta \varepsilon_F \|\nabla R(\bar{\boldsymbol{w}})\|^2}{C} \tag{3.137}$$

The last inequality implies: $\frac{\zeta}{\tau} = \frac{\frac{4\eta \varepsilon_F \|\nabla R(\bar{\boldsymbol{w}})\|^2}{C}}{16\kappa_s L_{s'} \frac{\varepsilon_F}{C}} = \frac{\eta \|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}$. □

Corollary 6. *Additionally, the number of calls to the function R (#IZO), and the number of hard thresholding operations (#HT) such that the upper bound in Theorem 7 above is smaller than $\varepsilon + Z\mu$, with $\varepsilon > 0$ are respectively: $\#HT = \mathcal{O}(\kappa_s \log(\frac{1}{\varepsilon}))$ and $\#IZO = \mathcal{O}\left(\frac{\varepsilon_F \kappa_s^3 L_s}{\varepsilon}\right)$. Note that if $s_2 = d$, we have $\varepsilon_F = \mathcal{O}(s) = \mathcal{O}(k)$, and therefore we obtain a query complexity that is dimension independent.*

Proof of Corollary 6. Let $\varepsilon \in \mathbb{R}_+^*$. Let us find T to ensure that $\mathbb{E} R(\hat{\boldsymbol{w}}_T) - R(\bar{\boldsymbol{w}}) \leq \varepsilon + Z\mu^2$ This will be enforced if:

$$4\alpha^2 L_{s'} \kappa_s \omega^T \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3} \frac{\eta \|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}} \right) \leq \varepsilon$$

$$\begin{aligned}
&\iff T\log(\omega) \le \log\left(\frac{\varepsilon}{4\alpha^2 L_{s'}\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right)}\right)\\
&\iff T \ge \frac{1}{\log(\frac{1}{\omega})}\log\left(\frac{4\alpha^2 L_{s'}\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right)}{\varepsilon}\right).
\end{aligned}\tag{3.138}$$

Therefore, let us take:

$$T := \left\lceil \frac{1}{\log(\frac{1}{\omega})}\log\left(\frac{4\alpha^2 L_{s'}\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right)}{\varepsilon}\right)\right\rceil. \tag{3.139}$$

We can now derive the #IZO and #HT. First, we have one hard-thresholding operation at each iteration, therefore #HT$= T$. Using the fact that $\frac{1}{\log(\frac{1}{\omega})} = \frac{1}{-\log(\omega)} = \frac{1}{-\log(1-\frac{1}{8\alpha\kappa_s})} \le \frac{1}{\frac{1}{8\alpha\kappa_s}} = 8\alpha\kappa_s$ (since by property of the logarithm, for all $x \in (-\infty, -1) : \log(1-x) \le -x$), and the fact that $\alpha = \frac{C}{L_{s'}}$ is independent of κ_s, we obtain that #HT $= \mathcal{O}(\kappa_s \log\left(\frac{1}{\varepsilon}\right))$.

We now turn to computing the #IZO. At each iteration t we have q_t function evaluations, therefore:

$$\begin{aligned}
&\#\text{IFO} = \sum_{t=0}^{T-1} q_t\\
&\le \sum_{t=0}^{T-1}\left(\frac{\tau}{\omega^t}+1\right)\\
&= T + \tau\frac{\left(\frac{1}{\omega}\right)^T - 1}{\frac{1}{\omega}-1}\\
&\le T + \frac{\tau}{\frac{1}{\omega}-1}\left(\frac{1}{\omega}\right)^T\\
&= T + \frac{\tau}{\frac{1}{\omega}-1}\exp\left(T\log\left(\frac{1}{\omega}\right)\right)\\
&\overset{(a)}{\le} 1 + \frac{1}{\log(\frac{1}{\omega})}\log\left(\frac{4\alpha^2 L_{s'}\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right)}{\varepsilon}\right)\\
&\qquad + \frac{\tau}{\frac{1}{\omega}-1}\exp\left(\log\left(\frac{1}{\omega}\right)\left[\frac{1}{\log(\frac{1}{\omega})}\log\left(\frac{4\alpha^2 L_{s'}\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right)}{\varepsilon}\right)+1\right]\right)\\
&= 1 + \frac{1}{\log(\frac{1}{\omega})}\log\left(\frac{4\alpha^2 L_{s'}\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right)}{\varepsilon}\right)
\end{aligned}$$

$$
\begin{aligned}
&+\frac{\frac{\tau}{\omega}}{\frac{1}{\omega}-1}\frac{4\alpha^2 L_{s'}\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right)}{\varepsilon}\\
&=1+\frac{1}{\log(\frac{1}{\omega})}\log\left(\frac{4\alpha^2 L_{s'}\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right)}{\varepsilon}\right)\\
&+\frac{\tau}{1-\omega}\frac{4\alpha^2 L_{s'}\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right)}{\varepsilon} \qquad (3.140)\\
&=1+\frac{1}{\log(\frac{1}{\omega})}\log\left(\frac{4\alpha^2 L_{s'}\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right)}{\varepsilon}\right)\\
&+\tau\frac{32\alpha^3 L_{s'}\kappa_s^2\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right)}{\varepsilon}, \qquad (3.141)
\end{aligned}
$$

where (a) follows from equation 3.139.

And we recall that $\tau := \frac{16\kappa_s\varepsilon_F}{(\alpha-1)}$, which implies that:

$$
\tau\frac{32\alpha^3 L_{s'}\kappa_s^2\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{2\gamma\kappa_s L_{s'}}\right)}{\varepsilon}=\mathcal{O}\left(\frac{\varepsilon_F}{\varepsilon}\left(\kappa_s^3 L_{s'}+\frac{\kappa_s}{\nu_s}\right)\right).
$$

Therefore, overall, the # IZO complexity is in $\mathcal{O}\left(\frac{\varepsilon_F}{\varepsilon}\kappa_s^3 L_{s'}\right)$.

□

3.7.4.3 Proof of Theorem 9

Using the results above, we can now proceed to the proof of Theorem 9.

Proof of Theorem 9. Let us denote for simplicity: $C_1 := \frac{\varepsilon_F}{qt}$, $C_2 := \frac{\varepsilon_{abs}}{qt}$, and $C_3 := \varepsilon_\mu \mu^2$. Moreover, let us denote $F := \operatorname{supp}(\boldsymbol{w}_t) \cup \operatorname{supp}(\boldsymbol{w}_{t-1}) \cup \operatorname{supp}(\bar{\boldsymbol{w}})$, where supp denotes the support of a vector, i.e. the set of coordinates of its non-zero components. Note that therefore we have $|F| \leq 2k + \bar{k} \leq 3k$. In addition $[\boldsymbol{u}]_F$ denotes the thresholding of $\boldsymbol{u}$ to the support F, that is, the vector $\boldsymbol{u}$ with its components that are not in F set to 0. Since R is $L_{s'}$-RSS', with $s' = \max(s_2, s)$, R is also s-RSS' and s_2-RSS', with Lipschitz constant $L_{s'}$.

Denote $\boldsymbol{v}_t = \mathcal{H}_k(\boldsymbol{w}_{t-1} - \eta\nabla R(\boldsymbol{w}_{t-1}))$ for any $t \in \mathbb{N}$. The fact that R is $(L_{s'}, s')$-RSS', therefore also $(L_{s'}, s)$-RSS', implies from the remark in Assumption 11 that it is also $(L_{s'}, s)$-RSS, therefore:

$$
\begin{aligned}
&R(\boldsymbol{w}_t)\\
\leq& R(\boldsymbol{w}_{t-1}) + \langle\nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle + \frac{L_s}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2\\
=& R(\boldsymbol{w}_{t-1}) + \langle\boldsymbol{g}_{t-1}, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle + \frac{L_s}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 + \langle\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle
\end{aligned}
$$

$$
\begin{aligned}
&= R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\left[\|\boldsymbol{w}_t - (\boldsymbol{w}_{t-1} - \eta \boldsymbol{g}_{t-1})\|^2 - \eta^2\|\boldsymbol{g}_{t-1}\|^2 - \|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2\right] + \frac{L_s}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 \\
&\quad + \langle \nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle \\
&= R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\|\boldsymbol{w}_t - (\boldsymbol{w}_{t-1} - \eta \boldsymbol{g}_{t-1})\|^2 - \frac{\eta}{2}\|\boldsymbol{g}_{t-1}\|^2 + \left[\frac{L_s - \frac{1}{\eta}}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 \\
&\quad + \langle [\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle \\
&\overset{(a)}{\leq} R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\left[\|\bar{\boldsymbol{w}} - (\boldsymbol{w}_{t-1} - \eta \boldsymbol{g}_{t-1})\|^2 - \|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 + \sqrt{\beta}\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2\right] - \frac{\eta}{2}\|\boldsymbol{g}_{t-1}\|^2 \\
&\quad + \left[\frac{L_s - \frac{1}{\eta}}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 + \langle [\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle \\
&= R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\left[\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 + \eta^2\|\boldsymbol{g}_{t-1}\|^2 - 2\langle \eta \boldsymbol{g}_{t-1}, \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle\right] \\
&\quad - \frac{1}{2\eta}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 + \frac{\sqrt{\beta}}{2\eta}\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2 \\
&\quad - \frac{\eta}{2}\|\boldsymbol{g}_{t-1}\|^2 + \left[\frac{L_s - \frac{1}{\eta}}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 + \langle [\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle \\
&= R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\left[\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - 2\langle \eta \boldsymbol{g}_{t-1}, \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle\right] - \frac{1}{2\eta}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 + \frac{\sqrt{\beta}}{2\eta}\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2 \\
&\quad + \left[\frac{L_s - \frac{1}{\eta}}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 + \langle [\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F, \boldsymbol{w}_t - \boldsymbol{w}_{t-1}\rangle \\
&\overset{(b)}{=} R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \boldsymbol{g}_{t-1}, \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle - \frac{1}{2\eta}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 + \frac{\sqrt{\beta}}{2\eta}\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2 \\
&\quad + \left[\frac{L_s - \frac{1}{\eta} + C}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 + \frac{1}{2C}\|[\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F\|^2 \\
&= R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle + \langle \nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}, \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle \\
&\quad - \frac{1}{2\eta}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 + \frac{\sqrt{\beta}}{2\eta}\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2 + \left[\frac{L_{s'} - \frac{1}{\eta} + C}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2 \\
&\quad + \frac{1}{2C}\|[\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F\|^2, \qquad (3.142)
\end{aligned}
$$

where (a) follows from Lemma 11 and (b) follows from the inequality $\langle a, b\rangle \leq \frac{C}{2}a^2 + \frac{1}{2C}b^2$, for any $(a, b) \in (\mathbb{R}^d)^2$ with $C > 0$ an arbitrary strictly positive constant.

Let us now assume that $\eta := \frac{1}{L_{s'} + C}$: therefore the term $\left[\frac{L_{s'} - \frac{1}{\eta} + C}{2}\right]\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2$ above is 0. We now take the conditional expectation (conditioned on w_{t-1}, which is the random variable which realizations are $\boldsymbol{w}_{t-1}$), on both sides, and from Lemma 16 we obtain the inequality below (we slightly abuse notations and denote $\mathbb{E}[\cdot|\mathrm{w}_{t-1} = \boldsymbol{w}_{t-1}]$ by $\mathbb{E}[\cdot|\boldsymbol{w}_{t-1}]$):

$$
\begin{aligned}
&\mathbb{E}[R(\boldsymbol{w}_t)|\boldsymbol{w}_{t-1}] \\
\le& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle \\
& - \frac{1}{2\eta}\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] + \frac{\sqrt{\beta}}{2\eta}\mathbb{E}\left[\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] \\
& + \langle [\nabla R(\boldsymbol{w}_{t-1}) - \mathbb{E}\left[\boldsymbol{g}_{t-1}|\boldsymbol{w}_{t-1}\right]]_F, \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle \\
& + \mathbb{E}\left[\frac{1}{2C}\|[\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F\|^2|\boldsymbol{w}_{t-1}\right] \\
\overset{(a)}{\le}& R(\boldsymbol{w}_{t-1}) + \frac{1}{2\eta}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle \\
& - \frac{1}{2\eta}\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] + \frac{\sqrt{\beta}}{2\eta}\mathbb{E}\left[\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] \\
& + \frac{G}{2}\|[\nabla R(\boldsymbol{w}_{t-1}) - \mathbb{E}[\boldsymbol{g}_{t-1}|\boldsymbol{w}_{t-1}]]_F\|^2 + \frac{1}{2G}\|\boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\|^2 \\
& + \frac{1}{2C}\mathbb{E}\left[\|\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}\|^2|\boldsymbol{w}_{t-1}\right] \\
=& R(\boldsymbol{w}_{t-1}) + \left[\frac{1}{2\eta} + \frac{1}{2G}\right]\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle \\
& - \frac{1}{2\eta}\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] + \frac{\sqrt{\beta}}{2\eta}\mathbb{E}\left[\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] \qquad (3.143) \\
& + \frac{G}{2}\|[\nabla R(\boldsymbol{w}_{t-1}) - \mathbb{E}[\boldsymbol{g}_{t-1}|\boldsymbol{w}_{t-1}]]_F\|^2 \\
& + \frac{1}{2C}\mathbb{E}\left[\|[\nabla R(\boldsymbol{w}_{t-1}) - \boldsymbol{g}_{t-1}]_F\|^2|\boldsymbol{w}_{t-1}\right] \\
\overset{(b)}{\le}& R(\boldsymbol{w}_{t-1}) + \left[\frac{1}{2\eta} + \frac{1}{2G}\right]\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle \\
& - \frac{1}{2\eta}\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] + \frac{\sqrt{\beta}}{2\eta}\mathbb{E}\left[\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] \\
& + \frac{G}{2}\|[\nabla R(\boldsymbol{w}_{t-1}) - \mathbb{E}[\boldsymbol{g}_{t-1}|\boldsymbol{w}_{t-1}]]_F\|^2 \\
& + \frac{1}{2C}\left(2\|[\nabla R(\boldsymbol{w}_{t-1}) - \mathbb{E}[\boldsymbol{g}_{t-1}|\boldsymbol{w}_{t-1}]]_F\|^2 + 2\|[\boldsymbol{g}_{t-1} - \mathbb{E}[\boldsymbol{g}_{t-1}|\boldsymbol{w}_{t-1}]]_F\|^2\right) \\
\overset{(3.97)+(3.99)}{\le}& R(\boldsymbol{w}_{t-1}) + \left[\frac{1}{2\eta} + \frac{1}{2G}\right]\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle \\
& - \frac{1}{2\eta}\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] + \frac{\sqrt{\beta}}{2\eta}\mathbb{E}\left[\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] + \frac{G}{2}C_3 \\
& + \frac{1}{2C}\left(2C_3 + 2C_1\|\nabla R(\boldsymbol{w}_{t-1})\|^2 + 2C_2\mu^2\right) \\
\overset{(c)}{\le}& R(\boldsymbol{w}_{t-1}) + \left[\frac{1}{2\eta} + \frac{1}{2G}\right]\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle
\end{aligned}
$$

$$
\begin{aligned}
&- \frac{1}{2\eta}\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 | \boldsymbol{w}_{t-1}\right] + \frac{\sqrt{\beta}}{2\eta}\mathbb{E}\left[\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2 | \boldsymbol{w}_{t-1}\right] + \frac{G}{2}C_3 \\
&+ \frac{1}{2C}\left(2C_1\left(2\|\nabla R(\boldsymbol{w}_{t-1}) - \nabla R(\bar{\boldsymbol{w}})\|^2 + 2\|\nabla R(\bar{\boldsymbol{w}})\|^2\right) + 2C_2\mu^2 + 2C_3\right) \\
\overset{(d)}{\leq}& R(\boldsymbol{w}_{t-1}) + \left[\frac{1}{2\eta} + \frac{1}{2G}\right]\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle \\
&- \frac{1}{2\eta}\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 | \boldsymbol{w}_{t-1}\right] + \frac{\sqrt{\beta}}{2\eta}\mathbb{E}\left[\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2 | \boldsymbol{w}_{t-1}\right] + \frac{G}{2}C_3 \\
&+ \frac{1}{2C}\left(2C_1\left(2L_{s'}^2\|\boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\|^2 + 2\|\nabla R(\bar{\boldsymbol{w}})\|^2\right) + 2C_2\mu^2 + 2C_3\right) \\
=& R(\boldsymbol{w}_{t-1}) + \left[\frac{1}{2\eta} + \frac{1}{2G} + \frac{2C_1L_{s'}^2}{C}\right]\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \langle \nabla R(\boldsymbol{w}_{t-1}), \boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\rangle \\
&- \frac{1}{2\eta}\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 | \boldsymbol{w}_{t-1}\right] + \frac{\sqrt{\beta}}{2\eta}\mathbb{E}\left[\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2 | \boldsymbol{w}_{t-1}\right] + \frac{G}{2}C_3 \\
&+ \frac{1}{C}\left(2C_1\|\nabla R(\bar{\boldsymbol{w}})\|^2 + C_2\mu^2 + C_3\right) \qquad (3.144) \\
\overset{(e)}{\leq}& R(\boldsymbol{w}_{t-1}) + \left[\frac{1}{2\eta} + \frac{1}{2G} + \frac{2C_1L_{s'}^2}{C}\right]\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 \\
&+ \left[R(\bar{\boldsymbol{w}}) - R(\boldsymbol{w}_{t-1}) - \frac{\nu_s}{2}\|\boldsymbol{w}_{t-1} - \bar{\boldsymbol{w}}\|^2\right] \\
&- \frac{1}{2\eta}\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 | \boldsymbol{w}_{t-1}\right] + \frac{\sqrt{\beta}}{2\eta}\mathbb{E}\left[\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2 | \boldsymbol{w}_{t-1}\right] + \frac{G}{2}C_3 \\
&+ \frac{1}{C}\left(2C_1\|\nabla R(\bar{\boldsymbol{w}})\|^2 + C_2\mu^2 + C_3\right) \qquad (3.145) \\
=& R(\bar{\boldsymbol{w}}) + \left[\frac{\frac{1}{\eta} - \nu_s}{2} + \frac{1}{2G} + \frac{2C_1L_{s'}^2}{C}\right]\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 \\
&- \frac{1}{2\eta}\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 | \boldsymbol{w}_{t-1}\right] + \frac{\sqrt{\beta}}{2\eta}\mathbb{E}\left[\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2 | \boldsymbol{w}_{t-1}\right] + \frac{G}{2}C_3 \\
&+ \frac{1}{C}\left(2C_1\|\nabla R(\bar{\boldsymbol{w}})\|^2 + C_2\mu^2 + C_3\right) \\
\overset{(f)}{\leq}& R(\bar{\boldsymbol{w}}) + \left[\frac{\frac{1}{\eta} - \nu_s}{2} + \frac{1}{2G} + \frac{2\varepsilon_F L_{s'}^2}{\tau C}\right]\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 \\
&- \frac{1}{2\eta}\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 | \boldsymbol{w}_{t-1}\right] + \frac{\sqrt{\beta}}{2\eta}\mathbb{E}\left[\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2 | \boldsymbol{w}_{t-1}\right] + \frac{G}{2}C_3 \\
&+ \frac{1}{C}\left(2C_1\|\nabla R(\bar{\boldsymbol{w}})\|^2 + C_2\mu^2 + C_3\right) \qquad (3.146)
\end{aligned}
$$

Where (a) follows from the inequality $\langle a, b\rangle \leq \frac{G}{2}a^2 + \frac{1}{2G}b^2$, for any $(a, b) \in (\mathbb{R}^d)^2$ with $G > 0$ an arbitrary strictly positive constant, (b) and (c) follow from the inequality $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ for any $(a, b) \in (\mathbb{R}^d)^2$, (d) follows from the fact that R is $(L_{s'}, s')$-RSS' (Assumption 11 with sparsity level s'), therefore it is also $(L_{s'}, s)$-RSS', (e) follows from the RSC condition, and for (f), we recall that $C_1 = \frac{\varepsilon_F}{q_t}$, and we define $q_t = \left\lceil \frac{\tau}{\omega^t} \right\rceil$, for some $\omega > 1$ and $\tau > 0$ that will be chosen later in the proof.

Recall that we have chosen $\eta := \frac{1}{L_{s'}+C}$. Let us define $\alpha := \frac{C}{L_{s'}}+1$. Then $C = (\alpha-1)L_{s'}$, and $\eta = \frac{1}{\alpha L_{s'}}$. Also recall that $\kappa_s = \frac{L_{s'}}{\nu_s}$.

We will now choose the constant G and C, in order to simplify the inequality above, such that it matches as much as possible the structure of the previous proofs:

We will seek to rewrite:
$\left[\frac{\frac{1}{\eta}-\nu_s}{2} + \frac{1}{2G} + \frac{2\frac{\varepsilon_F}{\tau}L_{s'}^2}{C}\right] \left(= \frac{1}{2\eta}\left[1+\frac{1}{G\alpha L_{s'}} + \frac{4L_{s'}^2\frac{\varepsilon_F}{\tau}}{(\alpha-1)\alpha L_{s'}^2} - \frac{1}{\alpha\kappa_s}\right]\right)$, into :

$\frac{1}{2\eta}\left[1-\frac{1}{\alpha'\kappa_s}\right]$ for some $\alpha' > 0$ (we will seek $\alpha' \propto \alpha$, with a dimensionless proportionality constant for simplicity).

Therefore, let us choose $G := \frac{4}{\nu_s}$, which implies:

$$\frac{1}{G\alpha L_{s'}} = \frac{1}{4\alpha\kappa_s}. \tag{3.147}$$

And let us choose $\tau := \frac{16\kappa_s\varepsilon_F}{(\alpha-1)}$, which implies:

$$\frac{4L_{s'}^2\frac{\varepsilon_F}{\tau}}{(\alpha-1)\alpha L_{s'}^2} = \frac{1}{4\alpha\kappa_s}. \tag{3.148}$$

Therefore, using equations 3.147 and 3.148, we obtain:

$$\begin{aligned}\left[\frac{\frac{1}{\eta}-\nu_s}{2} + \frac{1}{2G} + \frac{2\frac{\varepsilon_F}{\tau}L_{s'}^2}{C}\right] &= \frac{1}{2\eta}\left[1+\frac{1}{G\alpha L_{s'}} + \frac{4L_{s'}^2\frac{\varepsilon_F}{\tau}}{(\alpha-1)\alpha L_{s'}^2} - \frac{1}{\alpha\kappa_s}\right] \\ &= \frac{1}{2\eta}\left[1+\frac{1}{4\alpha\kappa_s} + \frac{1}{4\alpha\kappa_s} - \frac{1}{\alpha\kappa_s}\right] \\ &= \frac{1}{2\eta}\left[1-\frac{1}{2\alpha\kappa_s}\right] = \frac{1}{2\eta}\left[1-\frac{1}{\alpha'\kappa_s}\right], \end{aligned} \tag{3.149}$$

where for simplicity we denote $\alpha' = 2\alpha$.

We can therefore simplify (3.146) into:

$$\begin{aligned}\mathbb{E}[R(\boldsymbol{w}_t)|\boldsymbol{w}_{t-1}] - R(\bar{\boldsymbol{w}}) \leq & \frac{1}{2\eta}\left[\left(1-\frac{1}{\alpha'\kappa_s}\right)\|\bar{\boldsymbol{w}}-\boldsymbol{w}_{t-1}\|^2 - \frac{1}{2\eta}\mathbb{E}\left[\|\boldsymbol{w}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right]\right. \\ & + \frac{\sqrt{\beta}}{2\eta}\mathbb{E}\left[\|\boldsymbol{v}_t-\bar{\boldsymbol{w}}\|^2|\boldsymbol{w}_{t-1}\right] \\ & \left. +2\eta\left(\frac{G}{2}C_3 + \frac{1}{C}\left(2C_1\|\nabla R(\bar{\boldsymbol{w}})\|^2 + C_2\mu^2 + C_3\right)\right)\right]. \end{aligned} \tag{3.150}$$

We now take the expectation over w_{t-1} of the above inequality (i.e. we take $\mathbb{E}_{\mathrm{w}_{t-1}}[\cdot]$): using the law of total expectation ($\mathbb{E}[\cdot] = \mathbb{E}_{\mathrm{w}_{t-1}}[\mathbb{E}[\cdot|\boldsymbol{w}_{t-1}]]$) we obtain:

$$\begin{aligned}\mathbb{E}R(\boldsymbol{w}_t) - R(\bar{\boldsymbol{w}}) \leq& \frac{1}{2\eta}\left[\left(1 - \frac{1}{\alpha'\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \frac{1}{2\eta}\mathbb{E}\left[\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2\right]\right. \qquad (3.151)\\ &+ \frac{\sqrt{\beta}}{2\eta}\mathbb{E}\left[\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2\right]\\ &\left.+2\eta\left(\frac{G}{2}C_3 + \frac{1}{C}\left(2C_1\|\nabla R(\bar{\boldsymbol{w}})\|^2 + C_2\mu^2 + C_3\right)\right)\right]. \qquad (3.152)\end{aligned}$$

Let us call $A := 2\eta\left(\frac{G}{2}C_3 + \frac{1}{C}\left(2C_1\|\nabla R(\bar{\boldsymbol{w}})\|^2 + C_2\mu^2 + C_3\right)\right)$ for simplicity.

This gives:

$$\begin{aligned}\mathbb{E}R(\boldsymbol{w}_t) - R(\bar{\boldsymbol{w}}) \leq \frac{1}{2\eta}&\left[\left(1 - \frac{1}{\alpha'\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \frac{1}{2\eta}\mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2\right.\\ &\left.+\frac{\sqrt{\beta}}{2\eta}\mathbb{E}\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2 + A\right]. \qquad (3.153)\end{aligned}$$

Additionally, in view of equation 3.125 applied at $\boldsymbol{v}_t$ instead of $\boldsymbol{w}_t$, (since $\boldsymbol{v}_t$ here corresponds to the $\boldsymbol{w}_t$ from Section 3.7.2.2, i.e. $\boldsymbol{v}_t$ is the hard-thresholding of an iterate after a gradient step), we know that:

$$\mathbb{E}R(\boldsymbol{v}_t) - R(\bar{\boldsymbol{w}}) \leq \frac{1}{2\eta}\left[\left(1 - \frac{1}{\alpha'\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - (1 - \sqrt{\beta})\mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 + A\right]. \qquad (3.154)$$

We now take a convex combination similarly as in the case without additional constraint (section 3.5.2), for some $\rho \in (0, 1)$.

$$\begin{aligned}&\mathbb{E}(1-\rho)R(\boldsymbol{w}_t) + \rho R(\boldsymbol{v}_t)\\ \leq& R(\bar{\boldsymbol{w}}) + \frac{1}{2\eta}\left[\left(1 - \frac{1}{\alpha'\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - (1-\rho)\mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2\right.\\ &\left.+\left((1-\rho)\sqrt{\beta} - (1-\sqrt{\beta})\rho\right)\mathbb{E}\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2 + A\right]\\ =& R(\bar{\boldsymbol{w}}) + \frac{1}{2\eta}\left[\left(1 - \frac{1}{\alpha'\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - (1-\rho)\mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2\right.\\ &\left.-\left(\rho - \sqrt{\beta}\right)\mathbb{E}\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2 + A\right]\\ \overset{(b)}{\leq}& R(\bar{\boldsymbol{w}}) + \frac{1}{2\eta}\left[\left(1 - \frac{1}{\alpha'\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - (1-\rho)\mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2\right.\\ &\left.-\left(\rho - \sqrt{\beta}\right)\mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 + A\right]\\ =& R(\bar{\boldsymbol{w}}) + \frac{1}{2\eta}\left[\left(1 - \frac{1}{\alpha'\kappa_s}\right)\mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - (1-\sqrt{\beta})\mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 + A\right]. \qquad (3.155)\end{aligned}$$

where in (b), we have assumed that $\sqrt{\beta} \leq \rho$ (later we will verify that our choice of k ensures such a condition), and have used the fact that projection onto a convex set is non-expansive (which implies that $\|\boldsymbol{v}_t - \bar{\boldsymbol{w}}\|^2 \geq \|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2$).

Similarly as in [90], we now take a weighted sum over $t = 1, ..., T$, to obtain:

$$
\begin{aligned}
&\sum_{t=1}^{T} 2\eta \left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T-t} \mathbb{E}[(1-\rho)R(\boldsymbol{w}_t) + \rho R(\boldsymbol{v}_t) - R(\bar{\boldsymbol{w}})] \\
\leq &\sum_{t=1}^{T} \left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T-t} \left[\left(1 - \frac{1}{\alpha'\kappa_s}\right) \mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - (1-\sqrt{\beta})\mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 + A \right] \\
= &\sum_{t=1}^{T} \left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T-t} \left[\left(1 - \frac{1}{\alpha'\kappa_s}\right) \mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - (1-\sqrt{\beta})\mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 \right] \\
&+ \sum_{t=1}^{T} \left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T-t} A \\
= &(1-\sqrt{\beta}) \sum_{t=1}^{T} \left[\left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T-t+1} \mathbb{E}\|\bar{\boldsymbol{w}} - \boldsymbol{w}_{t-1}\|^2 - \left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T-t} \mathbb{E}\|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\|^2 \right] \\
&+ \sum_{t=1}^{T} \left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T-t} A \\
\overset{(a)}{=} &(1-\sqrt{\beta}) \left[\left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T} \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 - \mathbb{E}\|\boldsymbol{w}_T - \bar{\boldsymbol{w}}\|^2 \right] + \sum_{t=1}^{T} \left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T-t} A \\
\leq &(1-\sqrt{\beta}) \left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T} \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \sum_{t=1}^{T} \left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T-t} A \\
\leq &\left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T} \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \sum_{t=1}^{T} \left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T-t} A \\
= &\left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T} \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \sum_{t=1}^{T} \left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T-t} 2\eta \left(\frac{G}{2} C_3 + \frac{1}{C} \left(2C_1 \|\nabla R(\bar{\boldsymbol{w}})\|^2 \right.\right. \\
&\qquad \left.\left. + C_2\mu^2 + C_3 \right)\right) \\
= &\left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T} \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \sum_{t=1}^{T} \left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T-t} 2\eta \left(\frac{G}{2} C_3 + \frac{1}{C} \left(2\frac{\varepsilon_F}{q_t} \|\nabla R(\bar{\boldsymbol{w}})\|^2 \right.\right. \\
&\qquad \left.\left. + \frac{\varepsilon_{abs}\mu^2}{q_t} + C_3 \right)\right) \\
= &\left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T} \|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \sum_{t=1}^{T} \left(\frac{1 - \frac{1}{\alpha'\kappa_s}}{1 - \sqrt{\beta}} \right)^{T-t} \frac{2\eta}{q_t} \left(\frac{2\varepsilon_F \|\nabla R(\bar{\boldsymbol{w}})\|^2 + \varepsilon_{abs}\mu^2}{C} \right)
\end{aligned}
$$

$$
\begin{aligned}
&+\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}2\eta C_3\left(\frac{G}{2}+\frac{1}{C}\right)\\
&=\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}\frac{2\eta}{q_t}\left(\frac{2\varepsilon_F\|\nabla R(\bar{\boldsymbol{w}})\|^2}{C}\right)\\
&+\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}2\eta\mu^2\left(\varepsilon_\mu\left(\frac{G}{2}+\frac{1}{C}\right)+\frac{\varepsilon_{abs}}{Cq_t}\right)\\
&\leq\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}\frac{2\eta}{q_t}\left(\frac{2\varepsilon_F\|\nabla R(\bar{\boldsymbol{w}})\|^2}{C}\right)\\
&+\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}2\eta\mu^2\left(\varepsilon_\mu\left(\frac{G}{2}+\frac{1}{C}\right)+\frac{\varepsilon_{abs}}{C}\right),
\end{aligned}
\tag{3.156}
$$

where (a) follows from simplifying the telescopic sum. Let us denote for simplicity $\zeta := \frac{2\eta(2\varepsilon_F\|\nabla R(\bar{\boldsymbol{w}})\|^2)}{C} = \frac{4\eta\varepsilon_F\|\nabla R(\bar{\boldsymbol{w}})\|^2}{C}$ and $Z := \varepsilon_\mu\left(\frac{G}{2}+\frac{1}{C}\right)+\frac{\varepsilon_{abs}}{C}$.

We now choose k and s_t as follows: we choose $k \geq 4\frac{\alpha'^2}{\rho}\kappa_s^2\bar{k}$, which implies that:

$$
\begin{aligned}
&\sqrt{\beta}\leq\frac{1}{2\frac{\alpha'}{\rho}\kappa_s}\\
&\implies\sqrt{\beta}\leq\frac{1}{2\frac{\alpha'}{\rho}\kappa_s-1}\\
&\implies 1-\sqrt{\beta}\geq 1-\frac{1}{2\frac{\alpha'}{\rho}\kappa_s-1}=\frac{2\frac{\alpha'}{\rho}\kappa_s-2}{2\frac{\alpha'}{\rho}\kappa_s-1}=\frac{1-\frac{1}{\frac{\alpha'}{\rho}\kappa_s}}{1-\frac{1}{2\frac{\alpha'}{\rho}\kappa_s}}\\
&\implies\left(\frac{1-\frac{1}{\frac{\alpha'}{\rho}\kappa_s}}{1-\sqrt{\beta}}\right)\leq 1-\frac{1}{2\frac{\alpha'}{\rho}\kappa_s}.
\end{aligned}
\tag{3.157}
$$

We recall that we previously defined $q_t = \left\lceil\frac{\tau}{\omega^t}\right\rceil$, with $\tau = 16\kappa_s\frac{\varepsilon_F}{(\alpha-1)}$. We now set the value of ω, to $\omega := 1-\frac{1}{\frac{\alpha'}{\rho}\kappa_s}$.

Let us call $\nu := 1-\frac{1}{2\frac{\alpha'}{\rho}\kappa_s}$. Note that we have:

$$
\nu\leq\omega. \tag{3.158}
$$

And that we have the inequality below:

$$
\frac{\nu}{\omega}=\frac{1-\frac{1}{2\frac{\alpha'}{\rho}\kappa_s}}{1-\frac{1}{4\frac{\alpha'}{\rho}\kappa_s}}=\frac{4\frac{\alpha'}{\rho}\kappa_s-2}{4\frac{\alpha'}{\rho}\kappa_s-1}=1-\frac{1}{4\frac{\alpha'}{\rho}\kappa_s-1}\leq 1-\frac{1}{4\frac{\alpha'}{\rho}\kappa_s}=\omega. \tag{3.159}
$$

This allows us to simplify equation 3.156 into:

$$
\begin{aligned}
&\mathbb{E}\sum_{t=1}^{T} 2\eta \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \quad [(1-\rho)R(\boldsymbol{w}_t)+\rho R(\boldsymbol{v}_t)-R(\bar{\boldsymbol{w}})] \\
&\leq \nu^T\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2 + \sum_{t=1}^{T}\nu^{T-t}\omega^{t-1} + \frac{\zeta}{\tau}\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \quad 2\eta Z\mu^2 \\
&= \nu^T\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2 + \frac{\omega^T}{\omega}\sum_{t=1}^{T}\left(\frac{\nu}{\omega}\right)^{T-t} + \frac{\zeta}{\tau}\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \quad 2\eta Z\mu^2 \\
&= \nu^T\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2 + \frac{\omega^T}{\omega}\frac{1-\left(\frac{\nu}{\omega}\right)^T}{1-\left(\frac{\nu}{\omega}\right)} + \frac{\zeta}{\tau}\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \quad 2\eta Z\mu^2 \\
&\leq \nu^T\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2 + \frac{\omega^T}{\omega}\frac{1}{1-\left(\frac{\nu}{\omega}\right)} + \frac{\zeta}{\tau}\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \quad 2\eta Z\mu^2 \\
&\overset{(a)}{\leq} \nu^T\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2 + \frac{\omega^T}{\omega}\frac{1}{1-\omega} + \frac{\zeta}{\tau}\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \quad 2\eta Z\mu^2 \\
&\overset{(b)}{\leq} \nu^T\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2 + \frac{4}{3}\omega^T\frac{1}{1-\omega} + \frac{\zeta}{\tau}\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \quad 2\eta Z\mu^2 \\
&\overset{(c)}{\leq} \omega^T\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2 + \frac{4}{3}\omega^T\frac{1}{1-\omega} + \frac{\zeta}{\tau}\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \quad 2\eta Z\mu^2 \\
&\overset{(d)}{\leq} \frac{\omega^T}{1-\omega}\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2 + \frac{4}{3}\omega^T\frac{1}{1-\omega} + \frac{\zeta}{\tau}\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \quad 2\eta Z\mu^2 \\
&= \frac{\omega^T}{1-\omega}\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2 + \frac{4}{3}\right) + \frac{\zeta}{\tau}\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \quad 2\eta Z\mu^2 \\
&= 4\frac{\alpha'}{\rho}\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2 + \frac{4}{3}\right) + \frac{\zeta}{\tau}\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \quad 2\eta Z\mu^2,
\end{aligned}
\tag{3.160}
$$

where in the left hand side we have used the linearity of expectation, and where (a) uses equation 3.159, (b) uses the fact that $\frac{1}{\omega} = \frac{1}{1-\frac{1}{4\frac{\alpha'}{\rho}\kappa_s}} \leq \frac{1}{1-\frac{1}{4}} = \frac{4}{3}$ (since $\kappa_s \geq 1$ and $\alpha' \geq 1$ (indeed, we have $\alpha' = 2\alpha = 2(\frac{C}{L_{s'}}+1)$ with $C > 0$), so consequently $\frac{\alpha'}{\rho} \geq 1$), (c) uses equation 3.158, and (d) uses the fact that $\omega < 1$ so $1 < \frac{1}{1-\omega}$.

Let us now normalize the above inequality:

$$\mathbb{E}\frac{\sum_{t=1}^{T} 2\eta \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} [(1-\rho)R(\boldsymbol{w}_t)+\rho R(\boldsymbol{v}_t)]}{\sum_{t=1}^{T} 2\eta \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}} \leq R(\bar{\boldsymbol{w}}) + \frac{4\frac{\alpha'}{\rho}\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\zeta}{\tau}\right)}{\sum_{t=1}^{T} 2\eta \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t}} + Z\mu^2. \tag{3.161}$$

The left hand side above is a weighted sum, which is an upper bound on the smallest term of the sum.

Regarding the right hand side, we can simplify it using the fact that $0 < \left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)$, and therefore:

$$\sum_{t=1}^{T}\left(\frac{1-\frac{1}{\alpha'\kappa_s}}{1-\sqrt{\beta}}\right)^{T-t} \geq 1. \tag{3.162}$$

Therefore, we obtain:

$$\begin{aligned}\mathbb{E}\min_{t\in\{1,..,T\}}[(1-\rho)R(\boldsymbol{w}_t)+\rho R(\boldsymbol{v}_t)-R(\bar{\boldsymbol{w}})] &\leq \frac{4\frac{\alpha'}{\rho}\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\zeta}{\tau}\right)}{2\eta} + Z\mu^2 \\ &= 4\frac{\alpha^2}{\rho}L_{s'}\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\zeta}{\tau}\right)+Z\mu^2,\end{aligned} \tag{3.163}$$

which can be simplified into the expression below, using the definition of $\hat{\boldsymbol{w}}_T$:

$$\mathbb{E}[\min_{t\in[T]}(1-\rho)R(\boldsymbol{w}_t)+\rho R(\boldsymbol{v}_t)-R(\bar{\boldsymbol{w}})] \leq 4\frac{\alpha^2}{\rho}L_{s'}\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\zeta}{\tau}\right)+Z\mu^2. \tag{3.164}$$

To simplify the above result, we recall the assumptions made earlier on: we have chosen $\tau = \frac{16\kappa_s\varepsilon_F}{(\alpha-1)}$, and $G = \frac{4}{\nu_s}$.

Therefore, to sum up, we have:

$$Z = \varepsilon_\mu\left(\frac{G}{2}+\frac{1}{C}\right)+\frac{\varepsilon_{abs}}{C} = \varepsilon_\mu\left(\frac{2}{\nu_s}+\frac{1}{C}\right)+\frac{\varepsilon_{abs}}{C} \tag{3.165}$$

$$\omega = 1-\frac{1}{4\frac{\alpha'}{\rho}\kappa_s} = 1-\frac{1}{8\frac{\alpha}{\rho}\kappa_s} \tag{3.166}$$

$$\zeta = \frac{4\eta\varepsilon_F\|\nabla R(\bar{\boldsymbol{w}})\|^2}{C} \tag{3.167}$$

The last inequality implies: $\frac{\zeta}{\tau} = \frac{\frac{4\eta\varepsilon_F\|\nabla R(\bar{\boldsymbol{w}})\|^2}{C}}{16\kappa_s L_{s'}\frac{\varepsilon_F}{C}} = \frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}$.

Let us denote by ε_T the right-hand side term from equation 3.164:

$$\varepsilon_T = 4\frac{\alpha^2}{\rho}L_{s'}\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right) + Z\mu^2. \tag{3.168}$$

We now proceed similarly as in the proof of Theorem 7 above. Recall that we have assumed in the Assumptions of Theorem 9, without loss of generality, that R is non-negative, which implies that $R(\boldsymbol{v}_t) \geq 0$. Plugging this in equation 3.164 implies that:

$$\mathbb{E}\min_{t\in[T]} R(\boldsymbol{w}_t) \leq \frac{1}{1-\rho}R(\bar{\boldsymbol{w}}) + \frac{\varepsilon_T}{1-\rho} + \frac{Z}{(1-\rho)}\mu^2 \leq (1+2\rho)R(\bar{\boldsymbol{w}}) + \frac{\varepsilon_T}{1-\rho} + \frac{Z}{1-\rho}\mu^2. \tag{3.169}$$

Plugging the change of variable $\varepsilon'_T = \frac{\varepsilon_T}{1-\rho}$ into equation 3.169 above, and redefining Z into $Z := \frac{1}{1-\rho}\left(\varepsilon_\mu\left(\frac{2}{\nu_s} + \frac{1}{C}\right) + \frac{\varepsilon_{abs}}{C}\right)$, we obtain that:

$$\mathbb{E}\min_{t\in[T]} R(\boldsymbol{w}_t) \leq (1+2\rho)R(\bar{\boldsymbol{w}}) + \varepsilon'_T + Z\mu^2. \tag{3.170}$$

Further, consider an ideal case where $\bar{\boldsymbol{w}}$ is a global minimizer of R over $\mathcal{B}_0(k) := \{\boldsymbol{w} : \|\boldsymbol{w}\|_0 \leq k\}$. Then $R(\boldsymbol{v}_t) \geq R(\bar{\boldsymbol{w}})$ is always true for all $t \geq 1$. It follows that the bound in equation 3.164 yields:

$$\mathbb{E}\min_{t\in[T]}\{(1-\rho)R(\boldsymbol{w}_t) + \rho R(\bar{\boldsymbol{w}})\} \leq \mathbb{E}\min_{t\in[T]}\{(1-\rho)R(\boldsymbol{w}_t) + \rho R(\boldsymbol{v}_t)\} \leq R(\bar{\boldsymbol{w}}) + \varepsilon_T,$$

which implies: $\mathbb{E}\min_{t\in[T]} R(\boldsymbol{w}_t) \leq R(\bar{\boldsymbol{w}}) + \frac{\varepsilon_T}{1-\rho}$. In this case, we can simply set $\rho = 0.5$, and define $\varepsilon'_T = \frac{\varepsilon_T}{1-\rho} = 2\varepsilon_T$ similarly as above. The proof is completed.

□

3.7.5 Proof of Corollary 4

Proof of Corollary 4. Let $\varepsilon \in \mathbb{R}_+^*$. Let us find T to ensure that $\mathbb{E}\min_{t\in\{1,..,T\}}(1-\rho)R(\boldsymbol{w}_t) + \rho R(\boldsymbol{v}_t) - R(\bar{\boldsymbol{w}}) \leq \varepsilon + Z\mu^2$

This will be enforced if:

$$\begin{aligned}
&4\alpha^2\frac{1}{\rho}L_{s'}\kappa_s\omega^T\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right) \leq \varepsilon \\
&\iff T\log(\omega) \leq \log\left(\frac{\varepsilon}{4\alpha^2\frac{1}{\rho}L_{s'}\kappa_s\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right)}\right) \\
&\iff T \geq \frac{1}{\log(\frac{1}{\omega})}\log\left(\frac{4\alpha^2\frac{1}{\rho}L_{s'}\kappa_s\left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right)}{\varepsilon}\right).
\end{aligned} \tag{3.171}$$

Therefore, let us take:

$$T := \left\lceil \frac{1}{\log(\frac{1}{\omega})} \log \left(\frac{4\alpha^2 \frac{1}{\rho} L_{s'} \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3} \frac{\eta \|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}} \right)}{\varepsilon} \right) \right\rceil . \tag{3.172}$$

We can now derive the #IZO and #HT. First, we have one hard-thresholding operation at each iteration, therefore #HT= T. Using the fact that $\frac{1}{\log(\frac{1}{\omega})} = \frac{1}{-\log(\omega)} = \frac{1}{-\log(1 - \frac{1}{8\alpha \frac{1}{\rho}\kappa_s})} \leq \frac{1}{\frac{1}{8\alpha\frac{1}{\rho}\kappa_s}} = 8\alpha \frac{1}{\rho}\kappa_s$ (since by property of the logarithm, for all $x \in (-\infty, -1) : \log(1-x) \leq -x$), and the fact that $\alpha = \frac{C}{L_{s'}}$ is independent of κ_s, we obtain that #HT $= \mathcal{O}(\kappa_s \log\left(\frac{1}{\varepsilon}\right))$.

We now turn to computing the #IZO. At each iteration t we have q_t function evaluations, therefore:

$$\begin{aligned}
\#\text{IZO} &= \sum_{t=0}^{T-1} q_t \\
&\leq \sum_{t=0}^{T-1} \left(\frac{\tau}{\omega^t} + 1 \right) \\
&= T + \tau \frac{\left(\frac{1}{\omega}\right)^T - 1}{\frac{1}{\omega} - 1} \\
&\leq T + \frac{\tau}{\frac{1}{\omega} - 1} \left(\frac{1}{\omega} \right)^T \\
&= T + \frac{\tau}{\frac{1}{\omega} - 1} \exp\left(T \log\left(\frac{1}{\omega} \right) \right) \\
&\overset{(a)}{\leq} 1 + \frac{1}{\log(\frac{1}{\omega})} \log \left(\frac{4\alpha^2 \frac{1}{\rho} L_{s'} \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3} \frac{\eta \|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}} \right)}{\varepsilon} \right) \\
&\quad + \frac{\tau}{\frac{1}{\omega} - 1} \exp \left(\log\left(\frac{1}{\omega} \right) \left[\frac{1}{\log(\frac{1}{\omega})} \log \left(\frac{4\alpha^2 \frac{1}{\rho} L_{s'} \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3} \frac{\eta \|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}} \right)}{\varepsilon} \right) \right. \right. \\
&\quad \left. \left. +1 \right] \right) \\
&= 1 + \frac{1}{\log(\frac{1}{\omega})} \log \left(\frac{4\alpha^2 \frac{1}{\rho} L_{s'} \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3} \frac{\eta \|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}} \right)}{\varepsilon} \right) \\
&\quad + \frac{\frac{\tau}{\omega}}{\frac{1}{\omega} - 1} \frac{4\alpha^2 \frac{1}{\rho} L_{s'} \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3} \frac{\eta \|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}} \right)}{\varepsilon} \\
&= 1 + \frac{1}{\log(\frac{1}{\omega})} \log \left(\frac{4\alpha^2 \frac{1}{\rho} L_{s'} \kappa_s \left(\|\bar{\boldsymbol{w}} - \boldsymbol{w}_0\|^2 + \frac{4}{3} \frac{\eta \|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}} \right)}{\varepsilon} \right)
\end{aligned}$$

$$
\begin{aligned}
&+ \frac{\tau}{1-\omega}\frac{4\alpha^2\frac{1}{\rho}L_{s'}\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right)}{\varepsilon}\\
&= 1+\frac{1}{\log(\frac{1}{\omega})}\log\left(\frac{4\alpha^2\frac{1}{\rho}L_{s'}\kappa_s\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right)}{\varepsilon}\right)\\
&+\tau\frac{32\alpha^3\frac{1}{\rho^2}L_{s'}\kappa_s^2\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{4\kappa_s L_{s'}}\right)}{\varepsilon},
\end{aligned}
\tag{3.173}
$$

where (a) follows from equation 3.172.

And we recall that $\tau = 16\kappa_s\frac{\varepsilon_F}{(\alpha-1)}$, which implies that:

$$
\tau\frac{32\alpha^3\frac{1}{\rho^2}L_{s'}\kappa_s^2\left(\|\bar{\boldsymbol{w}}-\boldsymbol{w}_0\|^2+\frac{4}{3}\frac{\eta\|\nabla R(\bar{\boldsymbol{w}})\|^2}{2\gamma\kappa_s L_{s'}}\right)}{\varepsilon} = \mathcal{O}\left(\frac{\varepsilon_F}{\varepsilon}\left(\kappa_s^3 L_{s'}+\frac{\kappa_s}{\nu_s}\right)\right).
$$

Therefore, overall, the IZO (query complexity) is in $\mathcal{O}\left(\frac{\varepsilon_F}{\varepsilon}\kappa_s^3 L_{s'}\right)$. The proof is completed. □

3.8 Experiments

In this section, we provide some experiments to validate experimentally our theoretical results. Before describing our experiments, we provide a short discussion about the settings and algorithms that we will illustrate. For constraints Γ for which the Euclidean projection onto $\mathcal{B}_0(k)\cap\Gamma$ has a closed form equal to the TSP, our algorithm is identical to a vanilla non-convex projected gradient descent baseline (see Remark 11). In such case, our contribution in this paper is on the theoretical side, by providing some global guarantees on the optimization, instead of the local guarantees from existing work (cf. Table 1). Additionally, there are case in which there exists a closed form for projection onto $\Gamma\cap\mathcal{B}_0(k)$, different from the TSP (e.g. when $\Gamma=\mathbb{R}_+^d$, cf. [97]). Although our framework allows us to get approximate global convergence results when using the TSP, still, at the iteration level, a gradient step followed by Euclidean projection (not TSP) is optimal, since it minimizes a constrained quadratic upper bound on R. Therefore, we may not expect much improvement of the TSP over the Euclidean projection in such case, except on the computational side. With this in mind, we provide below the outline of our experiments:

- In Section 3.8.1, we illustrate on a synthetic example the trade-off between sparsity and optimality that is introduced by the extra constraint Γ, and that is balanced by the parameter ρ.
- In Section 3.8.2, we consider a synthetic example, in a case where Euclidean Projection and Two Steps Projection do not coincide (as mentioned above), in order to compare those two methods.

- In Section 3.8.3, we consider a portfolio index tracking problem where the goal is to illustrate a real-life application of our methods.
- In section 3.8.4, we consider a multi-class logistic regression on a real life dataset, to illustrate in more details in particular the stochastic and the zeroth-order versions of our method.

3.8.1 Synthetic Experiments: Illustrating the Sparsity/Optimality Trade-Off

In the section below, we provide a synthetic experiment to illustrate our Theorem 4, i.e. the trade-off between sparsity and optimality that is introduced by the extra constraint Γ, and that is balanced by $\rho \in (0, 0.5]$. We consider the synthetic linear regression example from [6] (Section E), with the risk below:

$$R(\boldsymbol{w}) := \frac{1}{n} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 , \tag{3.174}$$

and where $\boldsymbol{X}$ is diagonal with:

$$\boldsymbol{X}_{ii} = \begin{cases} 1 & \text{if } i \in I_1 \\ \sqrt{\kappa} & \text{if } i \in I_2 \\ 1 & \text{if } i \in I_3 , \end{cases} \tag{3.175}$$

where $I_1 = [s], I_2 = [s+1, s(\kappa+1)], I_3 = [s(\kappa+1)+1, s(\kappa^2+\kappa+1)]$ for some $s \geq 1$ and $\kappa \geq 1$ (we choose $s = 50$ and $\kappa = 2$, which results in having $d = 350$), n denotes the number of rows of $\boldsymbol{X}$, and $\boldsymbol{y}$ is defined as

$$y_i = \begin{cases} \kappa\sqrt{1-4\delta} & \text{if } i \in I_1 \\ \sqrt{\kappa}\sqrt{1-2\delta} & \text{if } i \in I_2 \\ 1 & \text{if } i \in I_3 \end{cases} \tag{3.176}$$

for some small $\delta > 0$ used for tie-breaking (we set it to $1e-4$). We chose such an example as it is used by [6] to prove a lower bound on the fundamental trade-off between sparsity and optimality proper to IHT: they use it to show that the relaxation of the sparsity k, of the order $k = \Omega(\kappa^2 \bar{k})$ (see also Table 3.1) is in fact unavoidable for IHT-type algorithms.

Case without Extra Constraints. First, we illustrate our Theorem 3 which considers vanilla IHT, without extra constraints. In Figure 3.2, on the one hand, we plot in blue, for every $k \in [d]$, the value of $R(\hat{\boldsymbol{w}}_k)$ where $\hat{\boldsymbol{w}}_k$ is the result of running vanilla IHT with sparsity k up to convergence. Then, on the other hand, we go through every value of $\bar{k} \in [d]$, and for each of them, we plot a point $(K(\bar{k}), R(\bar{\boldsymbol{w}}_{\bar{k}}))$, where $K(\bar{k})$ denotes the value of k required in our Theorem 3, i.e.: $K(\bar{k}) := 4\kappa^2\bar{k}$, and $\bar{\boldsymbol{w}}_{\bar{k}} := \min_{\boldsymbol{w} \in \mathbb{R}^d : \|\boldsymbol{w}\|_0 \leq \bar{k}} R(\boldsymbol{w})$. Therefore, each of such point $R(\bar{\boldsymbol{w}}_{\bar{k}})$ constitutes an upper bound on the value of $R(\hat{\boldsymbol{w}}_{K(\bar{k})})$, as we can indeed observe on Figure 3.2.

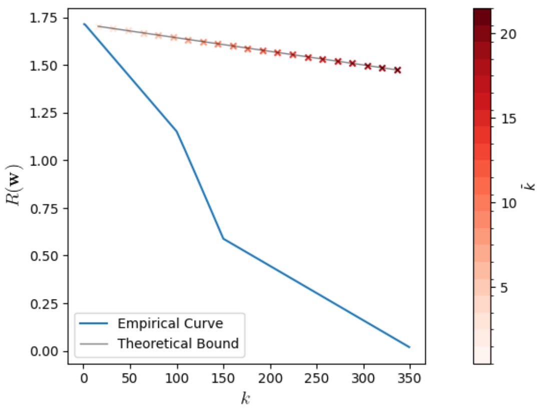


Figure 3.2: Illustration of Theorem 3 (i.e. $\Gamma = \mathbb{R}^d$).

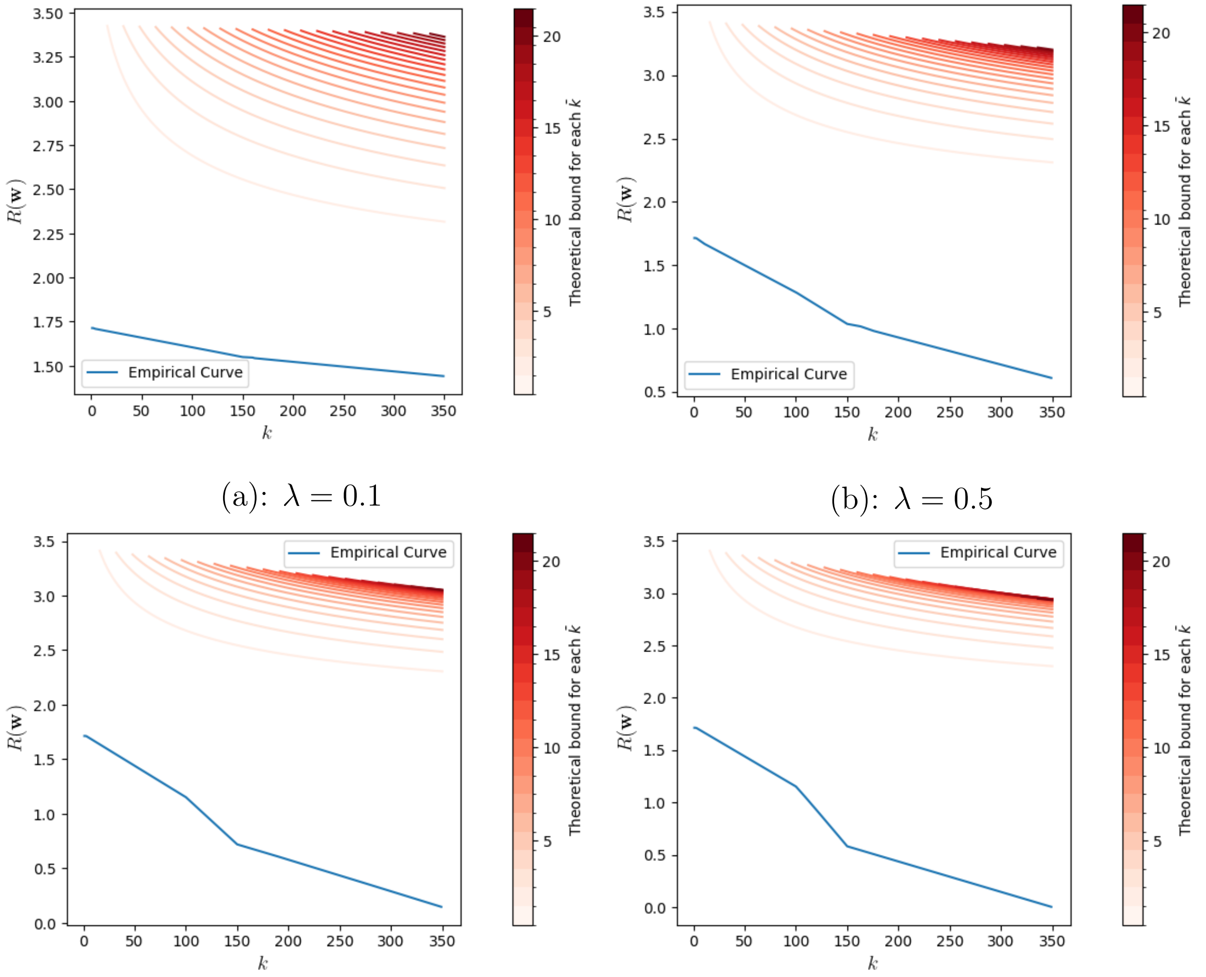
Case with Extra Constraints. We now illustrate the influence of the extra constraint Γ on the problem. We consider for Γ an ℓ_∞ norm constraint of radius $\lambda > 0$, that is: $\Gamma = \{\boldsymbol{w} \in \mathbb{R}^d : \forall i \in [d], |w_i| \leq \lambda\}$. In this new setting, we also go through every value of $\bar{k} \in [d]$, but this time, each of those values actually defines a curve parameterized by ρ, according to our Theorem 4: for each $\bar{k}$ we plot the parametric curve $(K(\bar{k}, \rho), (1+2\rho)R(\bar{\boldsymbol{w}}_{\bar{k}}))$, where, similarly as above, $K(\bar{k}, \rho)$ denotes the required value of k according to Theorem 4 (i.e., $K(\bar{k}, \rho) = \frac{4(1-\rho)^2\bar{k}\kappa^2}{\rho^2}$), and $\bar{\boldsymbol{w}}_{\bar{k}} := \min_{\boldsymbol{w}\in\mathbb{R}^d:\|\boldsymbol{w}\|_0\leq\bar{k}} R(\boldsymbol{w})$, and where ρ ranges in $(0, 0.5]$. We present the results for several values of λ in Figure 3.3 below. Note *that a priori*, the curves are allowed to cross, i.e. for a given k on the x-axis, one could have a point from a curve of small $\bar{k}$ (i.e. lighter shade of red) which could potentially also belong to a curve of larger $\bar{k}$ (let us denote it $\bar{k}'$) (darker shade of red), which would necessarily have a larger ρ (let us denote it ρ'), but for which the overall $(1 + 2\rho')R(\bar{\boldsymbol{w}}_{\bar{k}'})$ could be equal to $(1 + 2\rho)R(\bar{\boldsymbol{w}}_{\bar{k}})$ (since the problem will be less constrained with $\bar{k}'$ than with k). However, interestingly, this is not the case here due to the simplicity of the structure of the example. We can also observe that similarly as in the case where $\Gamma = \mathbb{R}^d$, the bound is a bit tighter in the small k regime (i.e. when $k \in [50, 100]$).

3.8.2 Synthetic Experiment: Comparing Two-Step Projection and Euclidean Projection

3.8.2.1 Differences Between Two-Step Projection and Euclidean Projection

In this section, we describe the differences between the two-step projection and the Euclidean projection onto the mixed constraints $\Gamma \cap \mathcal{B}_0(k)$. One can encounter several possible cases:

- **Case (i):** the two-step projection (TSP) and the Euclidean projection onto $\Gamma \cap \mathcal{B}_0(k)$ are identical (see e.g. Remark 11): in that case, the contribution of our paper are on the theoretical side: Theorems 4, 7, and 9 give global convergence guarantee which therefore in this case apply to the usual (non-convex) projected gradient descent algorithm with Euclidean projection.



(a): $\lambda = 0.1$

(b): $\lambda = 0.5$

(c): $\lambda = 1$

(d): $\lambda = 2$

Figure 3.3: Illustration of Theorem 3 (with Γ an ℓ_∞ ball of radius λ).

- **Case (ii):** the TSP and the Euclidean projection onto the mixed constraints are different: this case can be declined into several sub-cases as described below:
  - Case (a): the Euclidean projection onto the mixed constraint $\Gamma \cap \mathcal{B}_0(k)$ is unknown (such as for the constraints Γ used in the experiments from Section 3.8): in that case, the TSP can allow to fill such gap, since the TSP only requires the knowledge of the projection onto Γ, which is often known and easy to do.
  - Case (b): the Euclidean projection onto the mixed constraint $\Gamma \cap \mathcal{B}_0(k)$ is known, but computationally expensive: in that case, the TSP can provide a simpler and faster alternative to the Euclidean projection, while still enjoying some convergence guarantees as shown in this paper.
  - Case (c): the Euclidean projection onto the mixed constraint $\Gamma \cap \mathcal{B}_0(k)$ is known and is efficient enough (e.g. when Γ belongs to the set of positive symmetric sets

such as in [97]). In such cases, it is unclear whether the TSP can improve upon Euclidean projection since, at the iteration level, using the Euclidean projection is optimal (indeed, a (Euclidean) projected gradient descent step minimizes a quadratic upper bound on the objective value under constraints (derived from the smoothness of R)), and the TSP is therefore suboptimal in that sense (at the iteration level). *This is the case that we will analyze in this section*, in order to evaluate in practice the extend of such differences between TSP and Euclidean projection in such case.

3.8.2.2 Setting

As mentioned above, we analyze in more details the case (ii,c) above. We consider a simple synthetic linear regression setting with a correlated design matrix, i.e. where the design matrix $\boldsymbol{X}$ is formed by n i.i.d. samples from d (we take $d = 1000$, and $n = 5000$) correlated Gaussian random variables $\{X_1, .., X_d\}$ of zero mean and unit variance, such that:

$$\forall i \in \{1, \dots, d\} : \mathbb{E}[X_i] = 0, \mathbb{E}[X_i^2] = 1; \tag{3.177}$$

$$\forall (i,j) \in \{1, \dots, d\}^2, i \neq j : \mathbb{E}[X_i X_j] = \rho^{|i-j|}. \tag{3.178}$$

More precisely, we generate each feature X_i in an auto-regressive manner, from previous features, using a correlation $\rho \in [0, 1)$, in the following way: we have $X_1 \sim \mathcal{N}(0, 1)$ and $\sigma^2 = 1 - \rho^2$, and for all $j \in \{2, ..., d\}$: $X_{j+1} = \rho X_j + \epsilon_j$ where $\epsilon_j = \sigma \Delta$, with $\Delta \sim \mathcal{N}(0, 1)$. Additionally, the data is generated from a vector $\boldsymbol{w}^*$ of k^*-sparse support sampled uniformly at random, with $k^* = 20$, and with each non-zero entry sampled from a normal distribution, and $\boldsymbol{y}$ is obtained with a noise vector $\boldsymbol{\epsilon}$ created from i.i.d. samples from a normal distribution, rescaled to enforce a given signal to noise ratio (SNR), as follows:

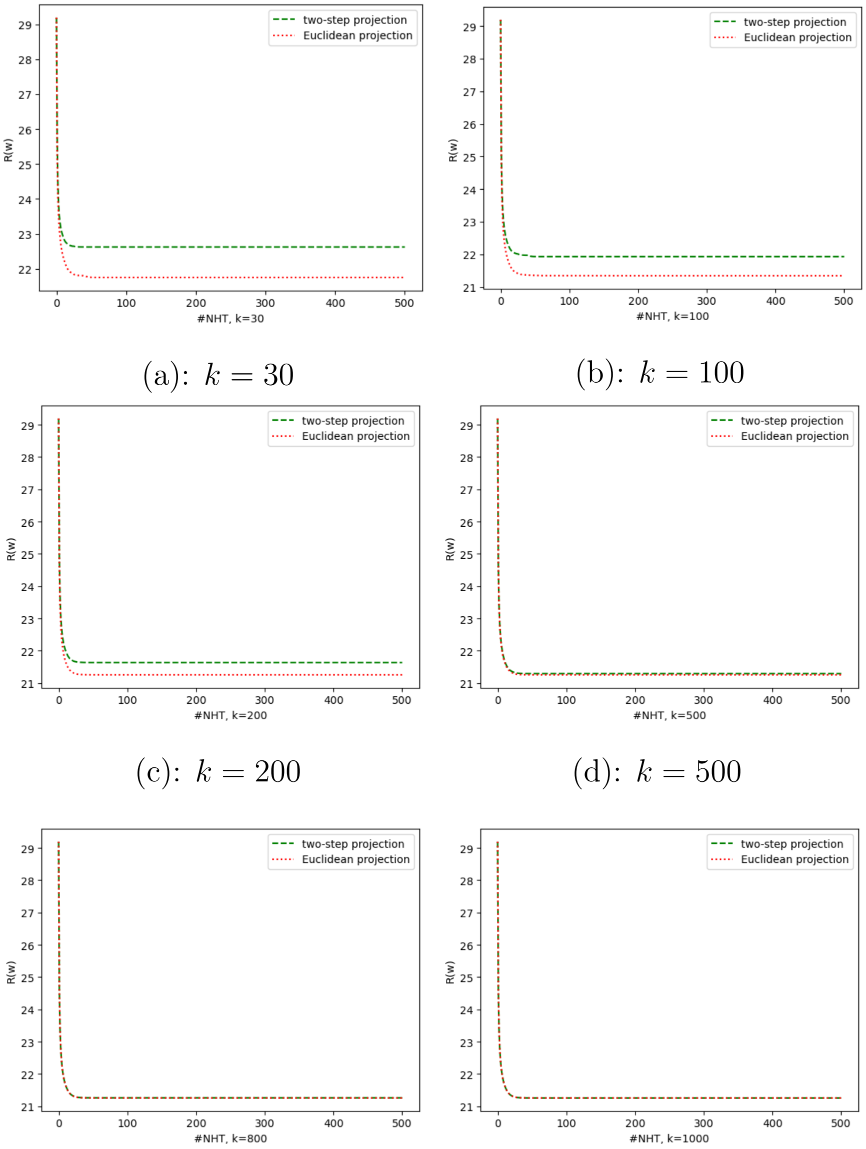
$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\epsilon} \tag{3.179}$$

with the signal to noise ratio defined as $\text{snr} = \frac{\|\boldsymbol{X}\boldsymbol{w}^*\|}{\|\boldsymbol{\epsilon}\|}$ (we choose $\text{snr} = 3$). We generate this dataset using the `make_correlated_data` function from the `benchopt` package [106]. The problem that we solve is:

$$\min_{\boldsymbol{w} \in \Gamma \cap \mathbb{R}^d} \frac{1}{n} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 \tag{3.180}$$

In such case, the Euclidean projection of $\boldsymbol{w} \in \mathbb{R}^d$ onto $\Gamma \cap \mathcal{B}_0(k)$ is given in [97], [14], and consists in simply sorting the entries in $\boldsymbol{w}$, $(w_1, ..., w_d)$ (not in absolute value), keeping the k largest ones (and setting the others to 0) to obtain $\boldsymbol{w'}$ and then replacing each coordinate w'_i by $\max(0, w'_i)$. The two-step projection (TSP) in such case is simply hard-thresholding of $\boldsymbol{w}$ to obtain a vector $\boldsymbol{w'}$ followed by replacing each coordinate w'_i by $\max(0, w'_i)$).

We plot the optimization curves for several values of k ($k \in \{30, 100, 200, 500, 800, 1000\}$ in Figure 3.4). In all curves, the learning rate is set to $1/L$ where L is the smoothness constant, equal to $\frac{2}{n}\|\boldsymbol{X}\|_s^2$ where $\|\boldsymbol{X}\|_s$ is the spectral norm of $\boldsymbol{X}$.



(a): $k = 30$ (b): $k = 100$

(c): $k = 200$ (d): $k = 500$

(e): $k = 800$ (f): $k = 1000$

Figure 3.4: Comparison of TSP vs. Euclidean projection for several k.

3.8.2.3 Discussion

As we can observe in Figure 3.4, the Euclidean projection onto $\Gamma \cap \mathcal{B}_0(k)$ performs better in terms of objective value than the TSP in some cases. However, the gap between the two methods closes as the enforced sparsity of the iterates k increases. We interpret it in the following way. First, (non-convex) projected gradient descent (i.e. using Euclidean projection) is guaranteed to converge to a (non-convex constraints version of a) stationary point of the objective function (see e.g. Theorem 1 from [152]), whereas our method does not possess such guarantee (indeed, our guarantees are of the global kind: we give upper bounds on the objective value for the output of the algorithm), and therefore, the TSP may in some cases not converge to a stationary point, which may explain why Euclidean projection sometimes performs better than TSP. However, for larger k, in both cases the projections operators (TSP or Euclidean projection) become closer to a simple projection onto Γ (i.e. without sparsity constraints), which explains why as k grows, the gap between the two methods reduces. Finally, the improved performance of the TSP when k is larger is consistent with our Theorem 4, since for larger k, the upper bound on R from Theorem 4 can be made smaller, since considering larger k implies that ρ can be taken smaller as per Remark 13, reducing our upper bound on the objective value.

In conclusion, these results show that in case (ii) from Section 3.8.2.1 above, the TSP introduced in this paper can be the most useful if the Euclidean projection onto $\Gamma \cap \mathcal{B}_0(k)$ is unknown, or too expensive computationally. Additionally, the gap between the two methods reduces if the enforced sparsity k of the iterates is large enough, or if the constraint forces iterates to stay close to 0.

3.8.3 Real Data Experiment: Portfolio Index Tracking

We now consider the following index tracking problem, originally presented in [138], and used as well in [14, 97]. It is also similar to the portfolio optimization problem presented in [84]. We seek to reproduce the performance of an index fund (such as S&P500), by investing only in a few key k assets, in order to limit transaction costs. The general problem can be formulated as a linear regression problem:

$$\min_{\boldsymbol{w} \in \mathcal{B}_0(k) \cap \Gamma} \|\boldsymbol{A}\boldsymbol{w} - \boldsymbol{y}\|^2 \tag{3.181}$$

where $\boldsymbol{w}$ represents the amount invested in each asset. For each $i \in [n]$ denoting a timestep , the i-th row of $\boldsymbol{A}$ denotes the returns of the d stocks at timestep i, and y_i the return of the index fund. In our scenario, we seek to limit to a value $D > 0$ the amount of transactions in each of c activity sector (group) of the portfolio (e.g. Industrials, Healthcare, etc.), denoted as G_i for $i \in [c]$. We ensure such constraint through an ℓ_1 norm constraint on each group: $\Gamma = \{\boldsymbol{w} \in \mathbb{R}^d : \forall i \in [c], \|\boldsymbol{w}_{G_i}\|_1 \leq D\}$, where $\boldsymbol{w}_{G_i}$ is the restriction of $\boldsymbol{w}$ to group G_i (i.e. for $j \in [d]$, $\boldsymbol{w}_{G_{i}j} = \boldsymbol{w}_j$ if $j \in G_i$ and 0 otherwise). In our case, $\boldsymbol{y}$ denotes the daily returns of a given portfolio index (e.g. S&P500) for a given time period (e.g. a given year), and $\boldsymbol{A}$ the returns of the corresponding d assets (over c sectors) of the index during such period.

Baselines. Up to our knowledge, there are no closed form for the Euclidean projection onto $\mathcal{B}_0(k) \cap \Gamma$, but the two-step projection can easily be done by projecting onto the ℓ_1 ball for each sector independently. We compare our algorithm (FG-HT-TSP) to two naive baselines: (a) the first one. called "$\text{PGD}(\Gamma) + \text{final}\Pi_{\mathcal{B}_0}$", consists in only ensuring the constraints in Γ, followed at the end of training by a simple hard-thresholding step to keep the k largest components of $\boldsymbol{w}$ in absolute value, and (b) the second one, called "$\text{PGD}(\mathcal{B}_0) + \text{final}\Pi_{\Gamma}$", consists in running vanilla IHT, followed at the end of training by a simple projection onto Γ to keep $\boldsymbol{w}$ in $\Gamma \cap \mathcal{B}_0$. We learn the weights of the portfolio on 80% of the considered period, and evaluate the out of sample (test set) performance on the remaining 20% (shaded area in the figure).

Datasets. We compare our algorithms on three portfolio indices datasets:

- **S&P500**: We take $k = 15$ and $D = 50$. $\boldsymbol{y}$ denotes the daily returns from January 1, 2021, to December 31, 2022, and $\boldsymbol{A}$ denotes the returns of the corresponding $d = 497$ assets (over $c = 11$ sectors). We plot our results in Figure 3.4(a).
- **HSI**: We take $k = 15$ and $D = 1000$. $\boldsymbol{y}$ denotes the daily returns from January 1, 2021, to December 31, 2022, and $\boldsymbol{A}$ denotes the returns of the corresponding $d = 72$ assets (over $c = 4$ sectors). We plot our results in Figure 3.4(b).
- **CSI300**: We take $k = 15$ and $D = 100$. $\boldsymbol{y}$ denotes the daily returns from March 1, 2021 (due to missing values in early 2021), to December 31, 2022, and $\boldsymbol{A}$ denotes the returns of the corresponding $d = 291$ assets (over $c = 10$ sectors). We plot our results in Figure 3.4(c).

The data for those three indices is scrapped from the web using the `beautifulsoup`[1] library to gather information about the index, and the `yfinance`[2] library to scrap the returns of such stocks during the considered time period. We provide in Table 3.2 below the respective dimensions of the train-sets used for the experiments (which constitutes, as we recall, 80% of the total dataset).

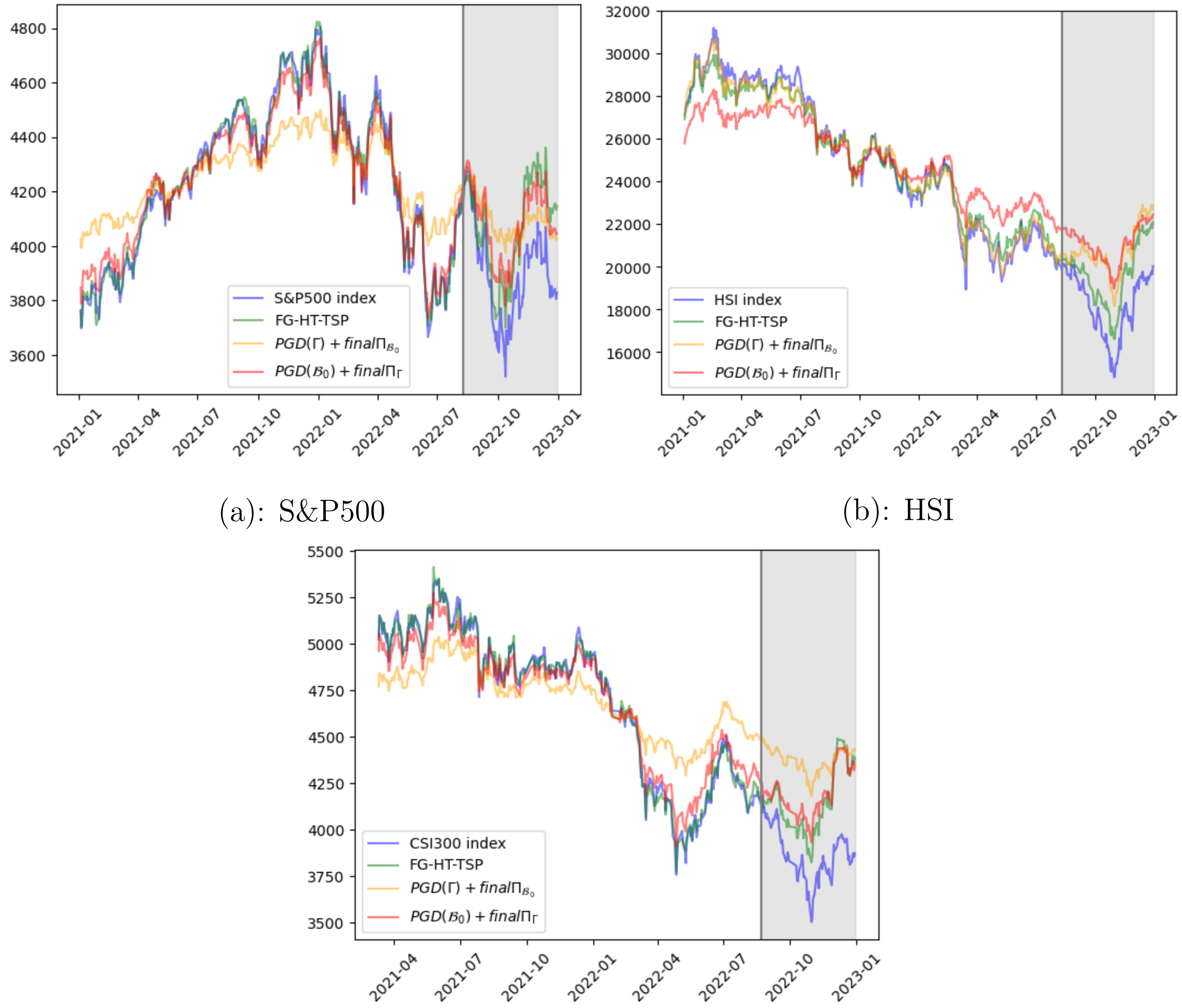
INDEX	$\boldsymbol{n}$	$\boldsymbol{d}$
S&P500	402	497
CSI300	353	291
HSI	394	72

Table 3.2: Number of samples (n) and dimension (d) of the training sets for the index tracking experiment.

[1] https://pypi.org/project/beautifulsoup4/

[2] https://github.com/ranaroussi/yfinance

Results. As we can observe on Figure 3.5, overall, the true index (blue curve) is more successfully tracked by our method (FG-HT-TSP, green curve), on the train-set of S&P500 and CSI300 and on the test-set of HSI and CSI300. Additionally, we have observed that for S&P500, our algorithm solution nonzero weights spans 9 of the 11 sectors for the S&P500 index, 7 sectors out of 10 for the CSI300 index, and 3 of the 4 sectors the one for the HSI index. Therefore, such portfolios are well diversified, as successfully enforced by our constraint.



(a): S&P500

(b): HSI

(c): CSI300

Figure 3.5: Index tracking with sector constraints for various indices

On the Verification of Assumptions 7 to 9: Note that such index tracking experiments verify Assumptions 7, 8 and 9:

- **Assumption 7** is verified since the cost function is quadratic, with a design matrix of size $n > d$ (except in the case of S&P500). As can be expected with such matrices in general, the Hessian $\boldsymbol{H} = 2\boldsymbol{A}^\top\boldsymbol{A}$ is positive-definite (we have indeed verified in our code that it is). Therefore the RSC constant is bounded below by $\lambda_{\min}$ where $\lambda_{\min}$ is the smallest eigenvalue of $2\boldsymbol{A}^\top\boldsymbol{A}$. Note that for S&P500, strong convexity is not verified since $d > n$: however, since we take $k = 15$, with high probability (i.e. unless we can find $s = 2k = 30$ columns of $\boldsymbol{A}$ that are exactly linearly dependent), RSC should be verified.

- **Assumption 8 and Assumption 11** are both verified since the cost function is quadratic, therefore the (strong) RSS constant is bounded above by $2\|\boldsymbol{A}\|_s^2$, where $\|\cdot\|_s$ denotes the spectral norm.

- **Assumption 9** is verified since projection onto Γ can be done group-wise, and for each group the projection is onto an ℓ_1 ball, which is a convex symmetric set (which is support-preserving from Remark 10), therefore, overall, Γ is support-preserving).

3.8.4 Real Data Experiment: Multiclass Logistic Regression

We now consider the multiclass logistic regression problem with class group-wise ℓ_2 norm constraint as follows. We have $R_i(\boldsymbol{w}) = \sum_{j=1}^{c}\left[\frac{\lambda}{c}\|\boldsymbol{w}_j\|_2^2 - \mathbf{1}\{y_i = j\}\log\frac{\exp\left(\boldsymbol{x}_i^\top\boldsymbol{w}_j\right)}{\sum_{l=1}^{c}\exp\left(\boldsymbol{x}_i^\top\boldsymbol{w}_l\right)}\right]$, where $\boldsymbol{y}_i$ is the target output of $\boldsymbol{x}_i$, c is the number of classes, and $\boldsymbol{w}_j$ is the weight vector specific to class j. In addition to the sparsity constraint $\mathcal{B}_0(k)$, we enforce the following additional constraint $\Gamma = \{\boldsymbol{w} \in \mathbb{R}^d : \forall j \in [c] : \|\boldsymbol{w}_j\|_2 \leq D\}$, for some constant $D \in \mathbb{R}_+$, where $d = p \times c$, with p the number of features of the samples $\boldsymbol{x}_i$. More precisely, in such multiclass logistic regression, we seek to ensure an extra regularization not only on the whole global weight vector $\boldsymbol{w}$ (with the used squared ℓ_2 penalty), but also on each weight vector related to each class (through Γ), in order to prevent a potential class-wise overfitting.

Up to our knowledge, there is no known closed form for the Euclidean projection onto such $\Gamma \cap \mathcal{B}_0(k)$. However, the two-step projection (TSP) can be done easily: once the first projection is done (projection onto $\mathcal{B}_0(k)$, i.e. hard-thresholding) and the sparse support S is identified as per Section 3.4.1, the projection onto Γ restricted to S can be easily done since Γ is class-wise decomposable, and therefore it suffices to project, for each $j \in [c]$, each $\boldsymbol{w}_j$ onto the ℓ_2 ball of radius D.

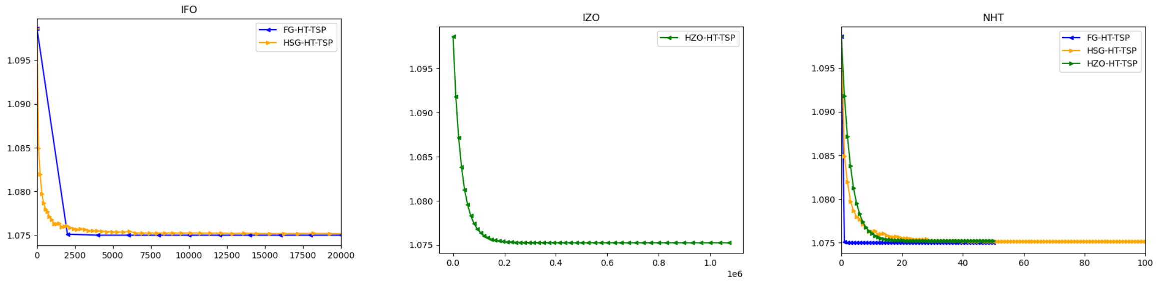
We have the smoothness constant L as below (see [21] for a derivation):

$$L = \sigma_{\max}\left(\frac{1}{2n}\left(\boldsymbol{I}_{c\times c} - \frac{1}{c}\mathbf{1}_c\mathbf{1}_c^\top\right) \otimes \boldsymbol{X}^\top\boldsymbol{X} + 2\lambda\boldsymbol{I}_{d\times d}\right) \tag{3.182}$$

Where $\otimes$ denotes the Kronecker product, $\sigma_{\max}$ the largest singular value of a matrix, $\boldsymbol{I}_{m\times m}$ the identity matrix of size $m \times m$ for some m, and $\mathbf{1}_c$ the vector $[1, 1, .., 1]^\top \in \mathbb{R}^c$.

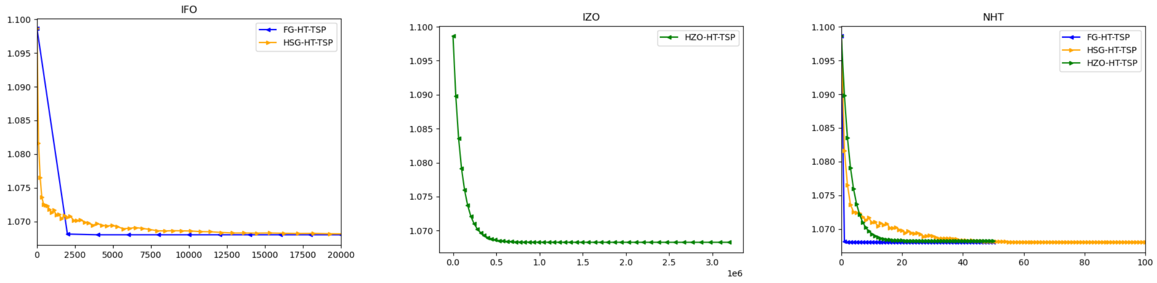
We consider the `dna` dataset from the LibSVM dataset repository [35], and we choose $D = 0.5$, $\lambda = 10$. For the stochastic case we take $B = 1e^5$, and for the stochastic and ZO case we take $\alpha = 2$. Note that in the stochastic case, if the growing batch-size required by Theorem 7 becomes larger than n, we keep it fixed to n (i.e. in such case we take the whole dataset at each step). In the zeroth-order case, we take $\mu = 1e - 6$. We set set all other hyperparameters as per Theorems 4, 7 and 9. In Figures 3.6, 3.7, 3.8 and 3.9, we plot the number of calls to a gradient ∇R_i (IFO: iterative first order oracle), and number of hard-thresholding operations (NHT), for various values of k and D (for the zeroth-order case, we plot the IZO (number of calls to the function R) instead of the IFO). We can observe that HSG-HT-TSP allows a smaller IFO than FG-HT-TSP in early iterations, since it does not need to compute a full gradient at each iteration.

In addition, to illustrate the theoretical improvement of our results on zeroth-order, even in the case where there is no additional constraint, we compare in Figures 3.10, 3.11 and 3.12 our algorithm HZO-HT with ZOHT [46], choosing for both algorithm an initial number of random direction as prescribed by our Theorem 9, and choosing, for the learning rate, in our case the one prescribed by Theorem 9, and for ZOHT, the one prescribed by Theorem 1 from [46] (and in both cases we fix $s = 3k$ as per Theorem 9): we can see that, in addition to being able to obtain a convergence in risk without system error, contrary to ZOHT (cf. Table 3.1), our Theorem 9 also prescribes a better (larger) learning rate (i.e. less conservative), leading to faster convergence.



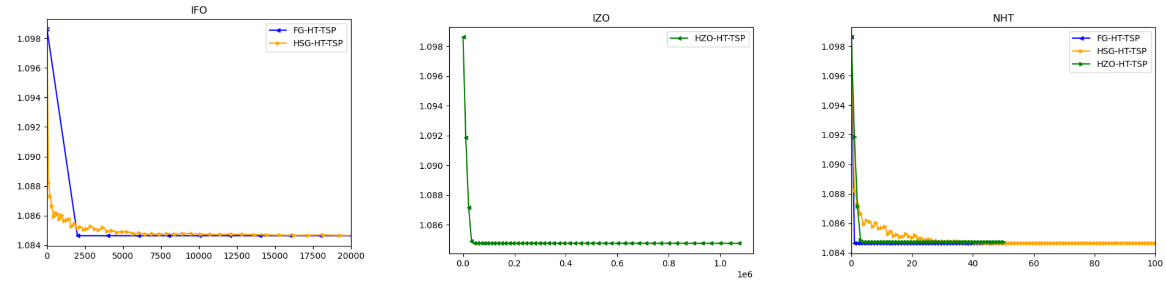
(a): #IFO (b): #IZO (c): #NHT

Figure 3.6: Multiclass Logistic Regression with TSP, $k = 50$, $D = 0.5$



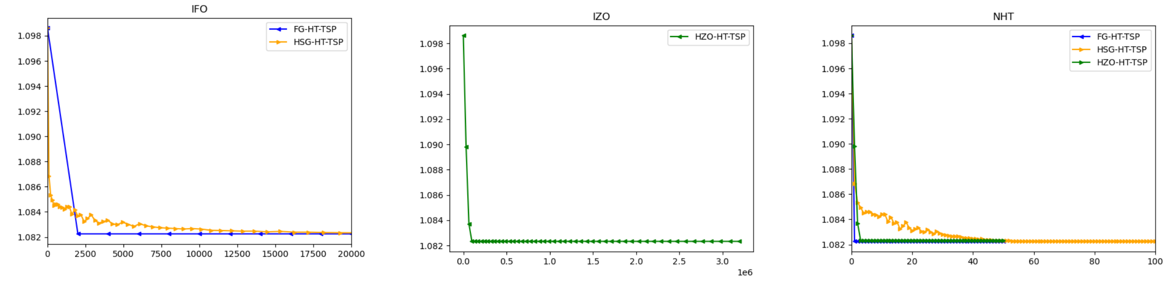
(a): #IFO (b): #IZO (c): #NHT

Figure 3.7: Multiclass Logistic Regression with TSP, $k = 150$, $D = 0.5$



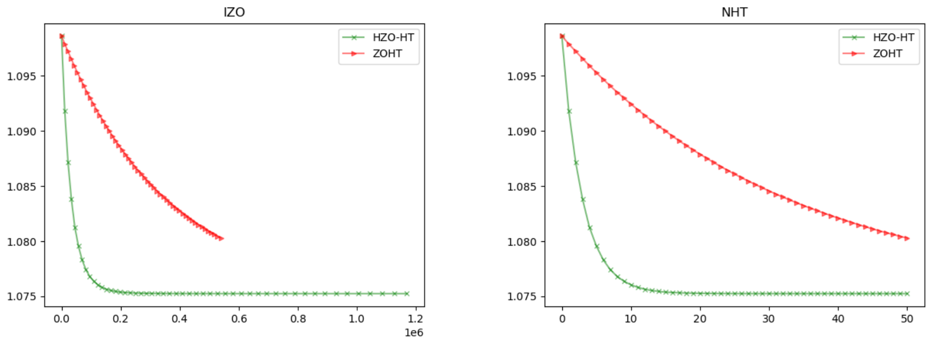
(a): #IFO (b): #IZO (c): #NHT

Figure 3.8: Multiclass Logistic Regression with TSP, $k = 50$, $D = 0.01$



(a): #IFO (b): #IZO (c): #NHT

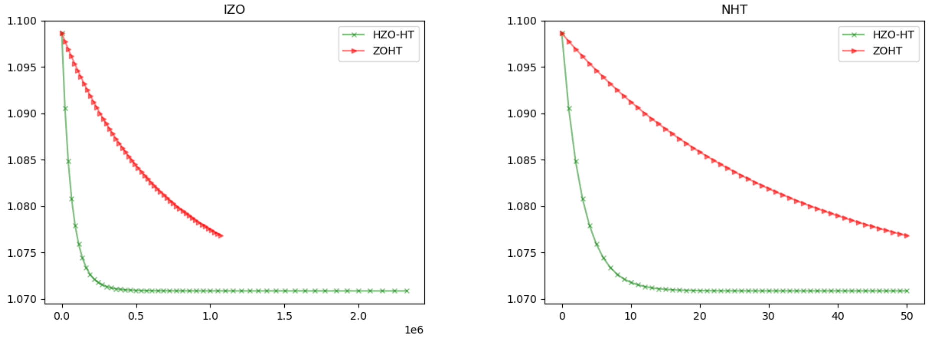
Figure 3.9: Multiclass Logistic Regression with TSP, $k = 150$, $D = 0.01$



(a): #IZO (b): #NHT

Figure 3.10: Multiclass Logistic Regression: HZO-HT vs. ZOHT, $k = 50$

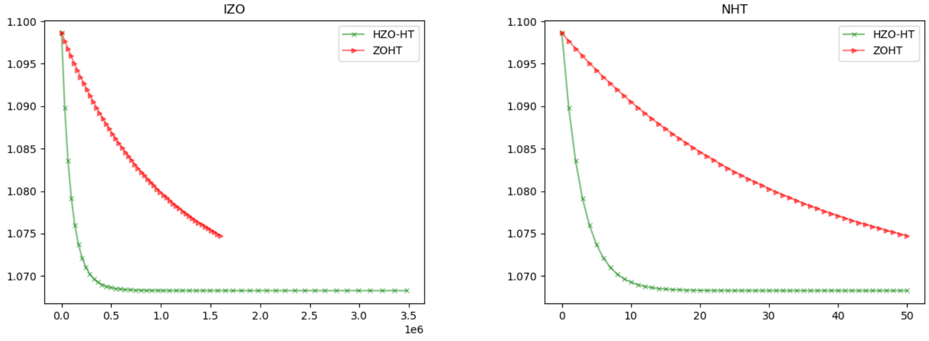
On the Verification of Assumptions 7 to 9: Note that such logistic regression experiments verify Assumptions 7, 8, 11 and 9:



(a): #IZO

(b): #NHT

Figure 3.11: Multiclass Logistic Regression: HZO-HT vs. ZOHT, $k = 100$



(a): #IZO

(b): #NHT

Figure 3.12: Multiclass Logistic Regression: HZO-HT vs. ZOHT, $k = 150$

- **Assumption 7** is verified thanks to the added squared ℓ_2 regularization, which makes the problem strongly convex and hence also restricted strongly convex.

- **Assumption 8 and Assumption 11** are both verified since the problem is smooth with a constant L as described above in equation 3.182, and therefore such constant is also a valid (strong) restricted-smoothness constant.

- **Assumption 9** is verified since, since, similarly as in the index tracking experiments from Section 3.8.3, projection onto Γ can be done group-wise, and for each group the projection is onto an ℓ_1 ball, which is a convex sign-free set (which is support-preserving from Remark 10), therefore, overall, Γ is support-preserving.

3.9 Conclusion

In this paper, we provided global convergence guarantees for variants of Iterative Hard Thresholding which can handle extra convex constraints which are support-preserving, via a two-step projection algorithm. We provided our analysis in the deterministic, stochastic, and zeroth-order settings. To that end, we used a variant of the three-point lemma, adapted to such mixed constraints, which allowed to simplified existing proofs for vanilla constraints (and to provide a new kind of result in the ZO setting), as well as obtaining new proofs in such combined constraints setting. Finally, it would also be interesting to extend this work to a broader family of sparsity structures and constraints, for instance to matrices or graphs. We leave this for future work.

Chapter 4

Iterative Regularization with k-Support Norm.

4.1 Interlude: a Dual Perspective on Iterative Hard-Thresholding

In this section, we transition to the last algorithm explored in this thesis, namely, Iterative Regularization with k-support Norm (IRKSN). So far, we considered the IHT algorithm, which is a (non-convex) projected gradient algorithm. For a set $\mathcal{S}$ for which a projection operator is well defined, projected gradient descent is as follows, where η_t is the step-size at iteration t, f is the function to optimize, and $\Pi_{\mathcal{S}}$ denotes the Euclidean projection onto $\mathcal{S}$:

$$\boldsymbol{x}_{t+1} = \Pi_{\mathcal{S}}(\boldsymbol{x}_t - \eta_t \nabla f(\boldsymbol{x}_t)) \tag{4.1}$$

In other words, at each iteration, we perform a gradient step, followed by a projection step. However, in [111], an alternative method is proposed to solve constrained optimization problems (although in the convex setting): the dual averaging method. It consists in accumulating the gradients in an unprojected so-called dual variable, from which the iterates $\boldsymbol{x}_t$ are obtained by a projection onto the feasible set. The update rule of dual averaging can be written as follows:

$$\boldsymbol{y}_{t+1} = \boldsymbol{y}_t - \eta_t \nabla f(\boldsymbol{x}_t) \tag{4.2}$$

$$\boldsymbol{x}_{t+1} = \Pi_{\mathcal{S}}(\boldsymbol{y}_{t+1}) \tag{4.3}$$

Such an algorithm is also called (lazy) mirror descent [27] or lazy online Convex Optimization [164]. To prove the convergence of such an algorithm, one usually needs to consider the following potential function ϕ, such that we have $\Pi_{\mathcal{S}} = \nabla \phi$. For instance, if $\mathcal{S}$ is the ℓ_2 unit ball $\mathcal{B} := \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| \leq 1\}$, we have $\Pi_{\mathcal{S}}(\cdot) = \nabla \phi(\cdot)$, with, for any $\boldsymbol{x} \in \mathbb{R}^d : \phi(\boldsymbol{x}) = \begin{cases} \frac{1}{2}\|\boldsymbol{x}\|^2 \text{ if } \|\boldsymbol{x}\|_2 \leq 1 \\ \|\boldsymbol{x}\|_2 - \frac{1}{2} \text{ otherwise} \end{cases}$ (one can show that ϕ is indeed smooth, as it is

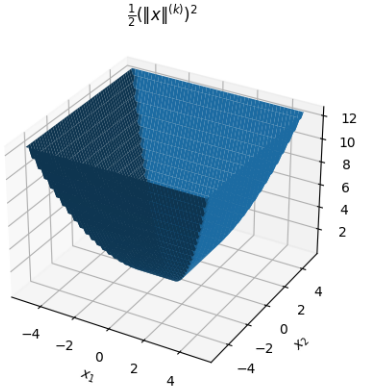


Figure 4.1: The half-squared top-k norm is not smooth.

the multidimensional Huber loss function, which is the Moreau-Yosida smoothing of the (unsquared) ℓ_2 norm function (and hence it is smooth by property of the Moreau-Yosida smoothing) (see Example 4.1 in [15])). Therefore, by analogy, we could consider as a potential ϕ, the following half-squared top-k norm function $\phi(\cdot) = \frac{1}{2}(\|\cdot\|^{(k)})^2$, where $\|\cdot\|^{(k)}$ is defined as the ℓ_2 norm of the largest k components of a vector. One can indeed check that for the points where $\frac{1}{2}(\|\cdot\|^{(k)})$ is differentiable, $\nabla\left[\frac{1}{2}(\|\cdot\|^{(k)})\right] = \mathcal{H}_k(\cdot)$: in other words, such a function ϕ can indeed be a potential associated with projection onto the ℓ_0 ball, i.e. a potential associated with the hard-thresholding operator. Unfortunately however, such a potential function ϕ is not smooth (as can be observed in Figure 4.1), and therefore one cannot use the same proof than in the convex case to prove convergence of a dual averaging version of the k-support norm. However, a potentially interesting idea is to replace the potential by its Moreau smoothing, which amounts to adding an squared ℓ_2 regularization to the dual of the (half-squared) top-k norm. The dual of the half-squared top-k norm is the half-squared k-support norm (see e.g. [24] Example 3.27, p. 94), where the k-support norm is the dual norm of the top-k norm. We denote the k-support norm by $\|\cdot\|_k^{sp}$ (see e.g. [3]) and we will describe it in more details later in this chapter. The Moreau smoothing of the top-k norm, can indeed be denoted by ϕ^δ, and expressed as follows, with ϕ^* denoting the Fenchel dual of a function ϕ [127], as using the fact that for some $\lambda > 0$, $(\lambda\phi(\cdot))^* = \lambda\phi^*(\frac{\cdot}{\lambda})$, and Section 3.1 from [121]:

$$\phi_\delta(\cdot) = \left(\quad \delta\frac{1}{2}(\|\frac{\cdot}{\delta}\|_k^{sp})^2 + \frac{1}{2}(\|\cdot\|_2^2) \quad \right)^* \tag{4.4}$$

Therefore, to sum up, without smoothing of ϕ, a dual averaging algorithm using the potential ϕ directly would be as such (where we use the subgradient since the potential is not smooth):

$$\boldsymbol{y}_{t+1} = \boldsymbol{y}_t - \eta_t \nabla f(\boldsymbol{x}_t) \tag{4.5}$$

$$\boldsymbol{x}_{t+1} \in \partial\phi(\boldsymbol{y}_{t+1}) \tag{4.6}$$

But using the smoothed ϕ_δ instead of ϕ, we obtain the following algorithm:

$$\boldsymbol{y}_{t+1} = \boldsymbol{y}_t - \eta_t \nabla f(\boldsymbol{x}_t) \tag{4.7}$$
$$\boldsymbol{x}_{t+1} = \nabla \phi_\delta(\boldsymbol{y}_{t+1}) \tag{4.8}$$

Since the gradient of the Moreau envelope of a function ϕ is equal to the proximal operator of its Fenchel dual ϕ^* (see the last equation in Section 3.1 [121]), we can rewrite the above update steps into:

$$\boldsymbol{y}_{t+1} = \boldsymbol{y}_t - \eta_t \nabla f(\boldsymbol{x}_t) \tag{4.9}$$
$$\boldsymbol{x}_{t+1} = \text{prox}_{\frac{1}{\delta}\phi^*(\cdot)}(\frac{\boldsymbol{y}_{t+1}}{\delta}) \tag{4.10}$$

This algorithm is sometimes known as Bregman iterations (see e.g. [29]). Taking ϕ^* to be the half-squared k-support norm, as described above, we get the following algorithm:

$$\boldsymbol{y}_{t+1} = \boldsymbol{y}_t - \eta_t \nabla f(\boldsymbol{x}_t) \tag{4.11}$$
$$\boldsymbol{x}_{t+1} = \text{prox}_{\frac{1}{2\delta}(\|\cdot\|_k^{sp})^2}(\frac{\boldsymbol{y}_{t+1}}{\delta}) \tag{4.12}$$

We can now describe a few properties of such an algorithm above:

- The iterates $\boldsymbol{x}_t$ may not be sparse anymore: the operator $\text{prox}_{\frac{1}{2\delta}(\|\cdot\|_k^{sp})^2}(\frac{\boldsymbol{y}_{t+1}}{\delta})$ can be seen as a *relaxed* version of hard-thresholding.
- 4.7 and 4.8 are characteristic equations of dual averaging algorithm, since $\frac{1}{2}(\|\cdot\|_k^{sp})^2 + \frac{1}{2}(\|\cdot\|_2^2)$ is strongly convex, cf. Definition 4 in [80]. Therefore such algorithm is a vanilla dual averaging algorithm, and as such, will minimize f if f is convex, which will not result in a sparse output of the algorithm in general (unless for instance when f admits a unique sparse minimizer).
- However, for overparameterized linear models, mirror descent (which is similar to dual averaging, see also [80]) is known to have an *implicit bias* [69], which can be sparsity enforcing for instance if the regularizer used is sparsity enforcing, see for instance [29]. Therefore such an algorithm above based on the k-support norm could still be useful to enforce some sparsity of the solution, in some cases.

In the next section, we will follow the recommendation from the last item above, and analyze a simpler case of the algorithm above, where f is taken as a quadratic function, and where the problem we seek to solve is the problem of sparse recovery. We will use an iterative regularization methods (see e.g. [99, 105]), which is a method similar to Bregman iterations as above, but with an early stopping stage, and which is known to have specific sparsity guarantees for the returned solution (hence their applicability for sparse recovery), as we seek. We will solve the following problem: we assume that we observe data of the following form, where $\boldsymbol{X}$ denotes a sensing matrix, $\boldsymbol{y}^\delta$ is a vector of noisy observations, $\boldsymbol{w}^*$

denotes the true sparse model which we wish to recover, and $\boldsymbol{\epsilon}$ denotes a vector of noise, of magnitude bounded by $\delta > 0$:

$$\boldsymbol{y}^{\delta} = \boldsymbol{X}\boldsymbol{w}^{*} + \boldsymbol{\epsilon}, \tag{4.13}$$

$$\text{with } \|\boldsymbol{\epsilon}\| \leq \delta. \tag{4.14}$$

In the above problem, we wish to recover $\boldsymbol{w}^{*}$. To that end, we will seek to solve the following problem:

$$\min_{\boldsymbol{x}} R(\boldsymbol{x}) \text{ s.t. } \boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}^{\delta}, \tag{4.15}$$

with $R(\boldsymbol{w}) = F(\boldsymbol{w}) + \frac{\alpha}{2}\|\boldsymbol{w}\|_2{}^2$ with $F(\boldsymbol{w}) = \frac{1-\alpha}{2}(\|\boldsymbol{w}\|_k^{sp})^2$. As mentioned above, we will solve this problem with a specific instantiation of the algorithm from [99], and which will be similar to an accelerated version of the Bregman Iterations algorithm with k-support norm described above. We now describe our full algorithm and setting in the sections below, which are based on our paper [44].

4.2 Introduction

Sparse recovery 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 [63], text data [85], or image data such as fMRI [16], 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

$$\boldsymbol{y}^{\delta} = \boldsymbol{X}\boldsymbol{w}^{*} + \boldsymbol{\epsilon} \tag{4.16}$$

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 [107].

However, most of those iterative methods are based on the ℓ_1 norm which requires restrictive applicability conditions and could fail in many cases. We discuss such related works in more details in the next section. 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. That norm was first

[1] An introduction to this topic, as well as an extensive review of its applications can be found in [55] and [150].

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, 102], into the primal-dual framework for iterative regularization described in [99].

We then provide some 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 elaborate on why such specific conditions include cases that are not included in some usual sufficient conditions for recovery with traditional methods based on the ℓ_1 norm (see Figure 4.2) (we describe such conditions for recovery with ℓ_1 norm in more details in Assumption 14). Since those types of conditions are still slightly opaque to interpret, we do as is common in the literature (such as in [78, 165]), 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 applicability of IRKSN on several experiments, including a support recovery experiment with a correlated design matrix, and show that it allows to identify the support more accurately than its competitors.

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 some sufficient 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 illustrate the applicability of our algorithm on several experiments, including a support recovery experiment with a correlated design matrix, and show that it allows support recovery with a higher F1 score than its competitors.

4.3 Preliminaries

Notations. We first recall a few definitions and notations used in the rest of the paper. We denote all vectors and matrices variables in bold font. For $S \subseteq [d]$, $\bar{S}$ denotes $[d] \setminus S$. 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 [62], $\|\boldsymbol{M}\|$ its nuclear norm, and $\boldsymbol{M}_S$ its column-restriction to a support $S \subseteq [d]$, i.e. the $n \times |S|$ matrix composed of the $|S|$ columns of $\boldsymbol{M}$ of indices in S. For a vector $\boldsymbol{w} \in \mathbb{R}^d$, $\text{supp}(\boldsymbol{w})$ 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.

Method	Condition on $\boldsymbol{X}$	Bound on $\lVert\hat{\boldsymbol{w}} - \boldsymbol{w}^*\rVert$	Complexity
IHT [20]	RIP	$O(\delta)$	$O(T)$
Lasso [139]	$\max_{\ell\in\bar{S}} \lvert\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \operatorname{sgn}(\boldsymbol{w}_S^*)\rangle\rvert < 1^{(2)}$	$O(\delta)$	$O(\Lambda T)$
ElasticNet [165]	-	-	$O(\Lambda T)$
KSN pen. [3]	-	-	$O(\Lambda T)$
OMP [141]	RIP	$O(\delta)$	$O(k)$
SRDI [118]	$\begin{cases} \exists \gamma \in (0,1]: \ \boldsymbol{X}_S^\top \boldsymbol{X}_S \geq n\gamma I_{d,d} \\ \exists \eta \in (0,1): \ \lVert \boldsymbol{X}_{\bar{S}} \boldsymbol{X}_S^\dagger \rVert_\infty \leq 1-\eta \end{cases}$	$O(\sigma\sqrt{\frac{k\log d}{n}})$ $^{(1)}$	$O(T)$
IROSR [145]	RIP	$O(\sigma\sqrt{\frac{k\log d}{n}})$ $^{(1)}$	$O(T)$
IRCR [105]	$\max_{\ell\in\bar{S}} \lvert\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \operatorname{sgn}(\boldsymbol{w}_S^*)\rangle\rvert < 1^{(2)}$	$O(\delta)$	$O(T)$
IRKSN (ours)	$\max_{\ell\in\bar{S}} \lvert\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^*\rangle\rvert < \min_{j\in S} \lvert\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_j, \boldsymbol{w}_S^*\rangle\rvert$	$O(\delta)$	$O(T)$

Table 4.1: Comparison of the existing algorithms for sparse recovery in the literature, including conditions on $\boldsymbol{X}$ and $\boldsymbol{w}^*$ sufficient 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.

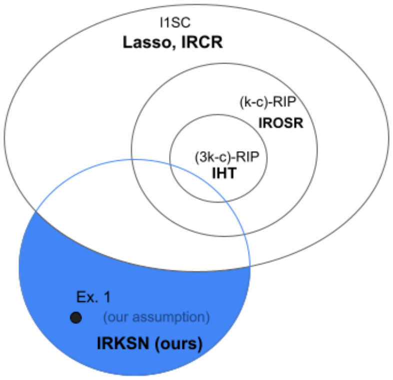


Figure 4.2: Conditions for recovery in various settings: l1SC corresponds to the condition $\max_{\ell\in\bar{S}} \lvert\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \operatorname{sgn}(\boldsymbol{w}_S^*)\rangle\rvert < 1$. "ours" denotes the condition $\max_{i\in\bar{S}} \lvert\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_i, \boldsymbol{w}_S^*\rangle\rvert < \min_{j\in S} \lvert\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_j, \boldsymbol{w}_S^*\rangle\rvert$. c denotes some constant in $[0,1]$. Here $3k$-RIP is shown for indicative purposes, corresponding to the condition for IHT as described in [20]. As we can see, for some cases (in blue), only IRKSN (our algorithm) can provably ensure sparse recovery.

More generally $\lVert\boldsymbol{w}\rVert_p$ denotes its ℓ_p norm for $p \in [1,+\infty)$, and $\lVert\boldsymbol{w}\rVert_0$ 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 $i \in S$. $\operatorname{sgn}(\boldsymbol{w})$ denotes the vector of its signs (with the additional convention that if $\boldsymbol{w}_i = 0$, $\operatorname{sgn}(\boldsymbol{w})_i = 0$).

Related works. 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 [20] 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) [20]. However, as mentioned in [76], this condition is very restrictive, and does not hold in most high-dimensional problems. Greedy methods, such as Orthogonal Matching Pursuit (OMP) [141], 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 [148], under some RIP condition.

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

$$(P) : \min_{\boldsymbol{w}} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}^\delta\|^2 + \lambda R(\boldsymbol{w}) \tag{4.17}$$

Where R is a regularizer, such as the ℓ_1 norm as is done in the Lasso method [139], 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 [139], has been proven in [65] to give a bound $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\| \leq O(\delta)$ under the so-called *source conditions* (described in Condition 4.3 from [65]) which are implied by the following more intuitive conditions: $\boldsymbol{X}_S$ is injective, and $\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \operatorname{sgn}(\boldsymbol{w}_S^*) \rangle| < 1$ (we detail this implication in Assumption 14). 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 [78], 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 penalty [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 [19, 53]. 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

$$\begin{aligned} (I): \quad & \min_{\boldsymbol{w}} R(\boldsymbol{w}) \\ \text{s.t.} \quad & \boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}^\delta \end{aligned} \tag{4.18}$$

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 [118], 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 [145] 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 [162]: under similar conditions, they also obtain a similar rate. Finally, [105] provide bounds of the form $\|\hat{\boldsymbol{w}} - \boldsymbol{w}\| \leq O(\delta)$, under the same *source conditions* as in [65].

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 sufficient conditions for recovery with Lasso that we discussed above) do not hold anymore. For instance, in gene array data [165], 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, which we tackle in this paper.

k-support Norm Regularization. We now introduce the k-support norm, which is the main component of our algorithm, as well as its proximal operator. 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 [100, 102], as well as successfully applied to several problems, including for instance fMRI [16, 60]. We give below its formal definition, with the following variational formula from [3]:

Definition 15 ([3, 101])**.** *Let $k \in \{1, ..., d\}$. The k-support norm $\|\cdot\|_k^{sp}$ is defined, for every $\boldsymbol{w} \in \mathbb{R}^d$, as:*

$$\|\boldsymbol{w}\|_k^{sp} = \min\left\{\sum_{I\in\mathcal{G}_k}\|\boldsymbol{v}_I\|_2 : \boldsymbol{v}_I \in \mathbb{R}^d, \operatorname{supp}(\boldsymbol{v}_I) \subseteq I,\right.$$
$$\left.\sum_{I\in\mathcal{G}_k}\boldsymbol{v}_I = \boldsymbol{w}\right\} \tag{4.19}$$

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 [37], 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 [75], when the set of overlapping groups is all possible subsets of $\{1, ..., d\}$ of cardinality at most k. Finally, we introduce the proximal operator [121] below, that will be used in our algorithm:

Definition 16 (Proximal operator, [121])**.** *The proximal operator for a function $h : \mathbb{R}^d \to \mathbb{R}$ is defined as:*

$$prox_h(\boldsymbol{z}) = \arg\min_{\boldsymbol{w}} h(\boldsymbol{w}) + \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{z}\|_2^2 \tag{4.20}$$

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 [102], which we will use in IRKSN, as described in Section 4.6.5.

4.4 The Algorithm

In this section, we describe the IRKSN (Iterative Regularization with k-Support Norm) algorithm. It is based on the general accelerated algorithm from [99], in which we plug a regularization function based on the k-support norm. More precisely, [99] 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):

$$\begin{aligned} (I_{ks}): \qquad & \min_{\boldsymbol{w}} R(\boldsymbol{w}) \\ \text{s.t.} \quad & \boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}^{\delta} \end{aligned} \tag{4.21}$$

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 [99], 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
$\hat{\boldsymbol{w}}_t \leftarrow \mathrm{prox}_{\alpha^{-1}F}(-\alpha^{-1}\boldsymbol{X}^\top \hat{\boldsymbol{v}}_t)$
# 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 [32,118]. In the optimization literature, it is mostly known as (Lazy) Mirror Descent [27], also called Dual Averaging [111,151]. The main idea in [99] is to early stop the algorithm at some iteration T, before convergence. We present the full accelerated version, IRKSN, in Algorithm 10.

```
Algorithm 10: IRKSN
Input   : $\hat{\boldsymbol{v}}_0 = \hat{\boldsymbol{z}}{-}1 = \hat{\boldsymbol{z}}_0 \in \mathbb{R}^d$, $\gamma = \alpha|\boldsymbol{X}|^{-2}$, $\theta_0 = 1$
for $t = 0$ to $T$ do
    $\hat{\boldsymbol{w}}_t \leftarrow \mathrm{prox}\,\alpha^{-1}F\left(-\alpha^{-1}\boldsymbol{X}^T\hat{\boldsymbol{z}}_{\boldsymbol{t}}\right)$ $\hat{\boldsymbol{r}}_t \leftarrow \mathrm{prox}\,\alpha^{-1}F\left(-\alpha^{-1}\boldsymbol{X}^T\hat{\boldsymbol{v}}_t\right)$
    $\hat{\boldsymbol{z}}_t \leftarrow \hat{\boldsymbol{v}}t + \gamma\left(\boldsymbol{X}\hat{\boldsymbol{r}}_{\boldsymbol{t}} - \boldsymbol{y}^\delta\right)$ $\theta_{t+1} \leftarrow \left(1 + \sqrt{1 + 4\theta_t^2}\right)/2$
    $\hat{\boldsymbol{v}}t + 1 = \hat{\boldsymbol{z}}t + \frac{\theta_t - 1}{\theta t+1}\left(\hat{\boldsymbol{z}}t - \hat{\boldsymbol{z}}t - 1\right)$
end
```

4.5 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.

4.5.1 Assumptions

We will present several sufficient conditions for recovery with the k-support norm, which are similar to the sufficient conditions needed for ℓ_1-based recovery that we describe in Assumption 14 (we will then elaborate on the differences between such conditions). The first assumption below is a variant of the usual feasibility assumption of the noiseless problem [55]: it simply states that $\boldsymbol{w}^*$, the true model that we wish to recover, is a feasible solution of the noiseless problem, and that it is k-sparse. Additionally, if several feasible solutions of same support than $\boldsymbol{w}^*$ exist, $\boldsymbol{w}^*$ should be the smallest norm one (we will elaborate on such condition in this section). Recall from the Introduction that $\boldsymbol{y}$ is the true target vector, i.e. uncorrupted by noise.

Assumption 12. *$\boldsymbol{w}^*$ 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:*

$$\boldsymbol{w}_S^* = \arg \min_{\boldsymbol{z} \in \mathbb{R}^k : \boldsymbol{X}_S \boldsymbol{z} = \boldsymbol{y}} \|\boldsymbol{z}\|_2 \tag{4.22}$$

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 13. *$\boldsymbol{w}^*$ verifies:*

$$\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^* \rangle| < \min_{j \in S} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_j, \boldsymbol{w}_S^* \rangle| \tag{4.23}$$

Up to our knowledge, we are the first to provide such assumptions for recovery with a k-support norm based algorithm: although [37] 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 values which are i.i.d. samples from a Gaussian distribution.

4.5.2 Discussion on the Assumptions

In this section, we attempt to interpret the assumptions above in simple terms, and to compare them to some similar sufficient conditions for recovery with ℓ_1 norm. More precisely, the condition below implies Condition 4.3 from [65], which latter is shown in [65] to be a necessary and sufficient condition for achieving a linear rate of recovery with ℓ_1 norm Tikhonov regularization. We prove such implication in Section 4.6.2.

Assumption 14 (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 12, the minimum ℓ_2 norm condition is actually not restrictive, compared to Assumption 14: indeed, in Assumption 14 $\boldsymbol{X}_S$ 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 is, we can obtain recovery in a regime where the number of samples n is even lower than the sparsity of the signal k). Note that for the Lasso, the condition $n \geq k$ is even *necessary*: indeed, when $n < k$, the Lasso is known to saturate [165] and recovery is impossible: interestingly, there is no such constraint when using a k-support norm regularizer (similarly to recovery with ElasticNet).

Dependence on the sign. As we can observe, Assumption 14 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 13 depends on $\boldsymbol{w}^*$ itself.

Case where $\boldsymbol{X}_S$ is injective. In the case where $\boldsymbol{X}_S$ is injective (as will happen in most cases in practice when $n > k$, i.e. unless there is some spurious exact linear dependence between columns), it is even easier to compare Assumptions 13 and 14. 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 13 can be rewritten into: $\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^{\dagger}\boldsymbol{x}_\ell, \boldsymbol{w}_S^*\rangle| < \min_{j \in S} |w_i^*|$, which is equivalent to:

$$\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^{\dagger}\boldsymbol{x}_\ell, \frac{\boldsymbol{w}_S^*}{\min_{j \in S} |w_i^*|}\rangle| < 1 \tag{4.24}$$

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 13 and 14 become equivalent (because then: $\frac{\boldsymbol{w}_S^*}{\min_{j \in S} |w_i^*|} = \operatorname{sgn}(\boldsymbol{w}_S^*)$). However, the two conditions 13 and 14 may differ depending on the *relative magnitudes* of the entries in $\boldsymbol{w}_S^*$. In particular, it may happen that our Assumption 13 is verified even if the Assumption 14 is not verified. We analyze such an example in Example 1.

4.5.3 Early Stopping Bound

We are now ready to state our main result:

Theorem 10 (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 12 and 13, 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 $t \geq 2$:*

$$\|\hat{\boldsymbol{w}}_t - \boldsymbol{w}^*\|_2 \leq at\delta + bt^{-1} \tag{4.25}$$

$$\textit{with} \quad a = 4\|\boldsymbol{X}\|^{-1} \quad \textit{and} \quad b = \frac{2\|\boldsymbol{X}\|\|(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*\|}{\alpha} \tag{4.26}$$

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

$$\|\hat{\boldsymbol{w}}_t - \boldsymbol{w}^*\|_2 \leq (a(c+1) + bc^{-1})\delta^{1/2} \tag{4.27}$$

Proof. Proof in Section 4.6.3. □

Discussion. We can notice in Theorem 10 above that b is large when α is small: therefore, if the inequality in 13 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 13 is fulfilled is, the better the retrieval of the true vector $\boldsymbol{w}^*$ is (because the larger we can choose α).

4.6 Proofs of the Main Results

4.6.1 Notations and Definitions

First, we describe some of the notations that will be used in this Section. $[\boldsymbol{v}]_S$ 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 $\boldsymbol{M}$ [62]. $\boldsymbol{I}_{r\times r}$ denotes the identity matrix in $\mathbb{R}^{r,r}$. $[d]$ denotes the set $\{1, ..., d\}$, and $\left[\begin{smallmatrix} d \\ k \end{smallmatrix}\right]$ 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 [127]. $\text{conv}(\mathcal{A})$ 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 17 (Legendre-Fenchel dual [127])**.** *For any function $f : \mathbb{R}^d \to \mathbb{R} \cup \{-\infty, +\infty\}$, the function $f^* : \mathbb{R}^d \to \mathbb{R}$ defined by*

$$f^*(\boldsymbol{y}) := \sup_{\boldsymbol{w}}\{\langle \boldsymbol{y}, \boldsymbol{w}\rangle - f(\boldsymbol{w})\} \tag{4.28}$$

is the Fenchel conjugate or dual to f.

Definition 18 (hard-thresholding operator [20])**.** *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:*

$$\pi_{HT}(\boldsymbol{z}) := \arg \min_{\boldsymbol{w}\in\mathbb{R}^d \ \ s.t. \|\boldsymbol{w}\|_0 \leq k} \|\boldsymbol{w} - \boldsymbol{z}\|_2^2 \tag{4.29}$$

Remark 17. *$\pi_{HT}(\boldsymbol{z})$ 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 2. *With $k = 1$: $\pi_{HT}((2,1)) = \{(2,0)\}$ and $\pi_{HT}((2,2)) = \{(2,0),(0,2)\}$*

Definition 19 (top-k norm)**.** *We define the following top-k norm $\|\cdot\|_{(k)}$, for all $\boldsymbol{w} \in \mathbb{R}^d$:*

$$\|\boldsymbol{w}\|_{(k)} = \|\pi^*_{HT}(\boldsymbol{w})\|_2 \tag{4.30}$$

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 $\boldsymbol{w}$.

4.6.2 Recall on the Conditions for Recovery with ℓ_1 Regularization

In this section, we briefly recall a conditions for sparse recovery with ℓ_1 norm regularization from [65], and why it is implied by Assumption 14. The authors of [65] proved in their Theorem 4.7 that such Assumption 15 below is a necessary and sufficient condition for achieving a linear rate of convergence for Tikhonov regularization with a priori parameter choice. We present below such condition 15.

Assumption 15 (Cond. 4.3 [65])**.**

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 :*
$$\text{(i): } \boldsymbol{X}^\top \boldsymbol{\lambda} \in \partial\|\cdot\|_1(\boldsymbol{w}^*) \quad \text{and (ii): } \quad |\langle \boldsymbol{x}_i, \boldsymbol{\lambda}\rangle| < 1 \quad \text{for} \quad i \notin \mathit{supp}(\boldsymbol{w}^*) \tag{4.31}$$
where 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 $\boldsymbol{X}_{\mathit{supp}(\boldsymbol{w}^*)}$ is injective.*

We now show that this Assumption 15 is implied by Assumption 14:

Lemma 19. *Assumption 14 $\implies$ Assumption 15.*

Proof. **Assume Assumption 14**, and take $\boldsymbol{\lambda} = (\boldsymbol{X}_S^\dagger)^\top \text{sign}(\boldsymbol{w}_S^*)$. We now have the following equality (A): $\boldsymbol{X}_S^\top \boldsymbol{\lambda} = \boldsymbol{X}_S^\top (\boldsymbol{X}_S^\dagger)^\top \text{sign}(\boldsymbol{w}_S^*) = (\boldsymbol{X}_S^\dagger \boldsymbol{X}_S)^\top \text{sign}(\boldsymbol{w}_S^*) \overset{(a)}{=} \text{sign}(\boldsymbol{w}_S^*) = [\partial\|\cdot\|_1(\boldsymbol{w}^*)]_S$ where (a) follows by property of the pseudo-inverse and the fact that $\boldsymbol{X}_S$ is injective (and therefore full column rank).

Additionally, from condition 3 in Assumption 14, we have:

$$\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \text{sgn}(\boldsymbol{w}_S^*)\rangle| < 1 \implies \max_{\ell \in \bar{S}} |\langle \boldsymbol{x}_\ell, (\boldsymbol{X}_S^\dagger)^\top \text{sgn}(\boldsymbol{w}_S^*)\rangle| < 1 \implies \max_{\ell \in \bar{S}} |\langle \boldsymbol{x}_\ell, \boldsymbol{\lambda}\rangle| < 1 \tag{4.32}$$

This inequality above corresponds to (ii) from the *strong source condition* above (15 (2. (ii))). Therefore, (since that last inequality also implies that for all $i \notin S$, $\langle \boldsymbol{x}_\ell, \boldsymbol{\lambda}\rangle \in [-1,1] = [\partial\|\cdot\|_1(\boldsymbol{w}^*)]_i$, which, combined with (A) implies 15 (2. (i)), we finally have that this $\boldsymbol{\lambda}$ verifies the existence conditions from 15.

□

4.6.3 Proof of Theorem 10

Proof of Theorem 10. Theorem 10 follows by combining Lemma 20 with Theorem 11 from [99]: in particular, when plugging from Lemma 20 the value (denoted by $\boldsymbol{\lambda}^*$ in Lemma 20 (2)) of the solution of the dual problem of equation 4.34, (denoted by $\boldsymbol{v}^*$ in Theorem 11) we obtain:

$$\|\boldsymbol{v}^*\| = \|(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*\| \tag{4.33}$$

□

Lemma 20. *Under Assumptions 12 and 13, 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:*

$$\begin{aligned} (I_{ks}\text{-}noiseless): \quad & \min_{\boldsymbol{w}} R(\boldsymbol{w}) \\ s.t. \quad & \boldsymbol{X}\boldsymbol{w} = \boldsymbol{y} \end{aligned} \tag{4.34}$$

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 $\boldsymbol{\lambda} \in \mathbb{R}^n$:

$$\{-\boldsymbol{X}^\top \boldsymbol{\lambda} \in \partial R(\boldsymbol{w}^*)\} \iff \{(1-\alpha)\partial(\frac{1}{2}\|\cdot\|_k^{sp2})(\boldsymbol{w}^*) \ni -\boldsymbol{X}^\top\boldsymbol{\lambda} - \alpha \boldsymbol{w}^*\} \tag{4.35}$$

$$\overset{(a)}{\iff} \{(1-\alpha)\boldsymbol{w}^* \in \partial(\frac{1}{2}\|\cdot\|_{(k)}^2)(-\boldsymbol{X}^\top\boldsymbol{\lambda} - \alpha\boldsymbol{w}^*)\} \tag{4.36}$$

$$\overset{(b)}{\iff} \{(1-\alpha)\boldsymbol{w}^* \in \operatorname{conv}(\pi_{HT}(-\boldsymbol{X}^\top\boldsymbol{\lambda} - \alpha\boldsymbol{w}^*))\} \tag{4.37}$$

Where (a) follows from Proposition 3 and Corollary 7, and (b) from Lemma 21.

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

$$\operatorname{conv}(\pi_{HT}(-\boldsymbol{X}^\top\boldsymbol{\lambda}^* - \alpha\boldsymbol{w}^*)) = \operatorname{conv}(\pi_{HT}(\boldsymbol{X}^\top(\boldsymbol{X}_S^\top)^\dagger\boldsymbol{w}_S^* - \alpha\boldsymbol{w}^*)) \tag{4.38}$$

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 10)

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

$$\begin{aligned}
[|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* - \alpha \boldsymbol{w}^*|]_i &\overset{(a)}{\geq} [|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*| - |\alpha \boldsymbol{w}^*|]_i \\
&= [|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*|]_i - \alpha[|\boldsymbol{w}^*|]_i \\
&\geq [|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*|]_i - \alpha\|\boldsymbol{w}^*\|_\infty \\
&\overset{(b)}{>} [|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*|]_i - \min_{j\in S} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_j, \boldsymbol{w}_S^* \rangle| + \max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^* \rangle| \\
&\geq \max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^* \rangle| \\
&= \max_{\ell \in \bar{S}} [|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*|]_\ell \\
&\overset{(c)}{=} \max_{\ell \in \bar{S}} [|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* - \alpha \boldsymbol{w}^*|]_\ell
\end{aligned} \tag{4.39}$$

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 $i \in S, \ell \in \bar{S}$:

$$[|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* - \alpha \boldsymbol{w}^*|]_i > [|\boldsymbol{X}(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* - \alpha \boldsymbol{w}^*|]_\ell \tag{4.40}$$

This allows us to simplify equation 4.38, given that the hard-thresholding operation selects the top k-components of a vector (in absolute value), and using the fact that we assumed that S is of size k (i.e. $|S| = k$) (so 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); in other words, the convex hull of a singleton is that singleton itself):

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

$$\begin{aligned}
[\text{conv}\left(\pi_{HT}(-\boldsymbol{X}^\top \boldsymbol{\lambda}^* - \alpha \boldsymbol{w}^*)\right)]_i &= \begin{cases} \langle \boldsymbol{x}_i, (\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* \rangle - \alpha \boldsymbol{w}_i^* & \text{if} \quad i \in S \\ 0 \quad \text{if} \quad i \in \bar{S} \end{cases} \\
&= \begin{cases} \langle \boldsymbol{x}_i, (\boldsymbol{X}_S^\top)^\dagger \boldsymbol{X}_S^\dagger \boldsymbol{y} \rangle - \alpha \boldsymbol{w}_i^* & \text{if} \quad i \in S \\ 0 \quad \text{if} \quad i \in \bar{S} \end{cases} \\
&\overset{(a)}{=} \begin{cases} [\boldsymbol{X}_S^\dagger \boldsymbol{y}]_i - \alpha \boldsymbol{w}_i^* & \text{if} \quad i \in S \\ 0 \quad \text{if} \quad i \in \bar{S} \end{cases} \\
&\overset{(b)}{=} \begin{cases} \boldsymbol{w}_i^* - \alpha \boldsymbol{w}_i^* & \text{if} \quad i \in S \\ 0 \quad \text{if} \quad i \in \bar{S} \end{cases} \\
&= \begin{cases} (1-\alpha) \boldsymbol{w}_i^* & \text{if} \quad i \in S \\ 0 \quad \text{if} \quad i \in \bar{S} \end{cases}
\end{aligned} \tag{4.41}$$

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 [62]: with $\boldsymbol{M} =$

$\boldsymbol{UDV}^\top$, we have: $\boldsymbol{MM}^\dagger(\boldsymbol{M}^\top)^\dagger = \boldsymbol{UDV}^\top\boldsymbol{VD}^{-1}\boldsymbol{U}^\top\boldsymbol{UD}^{-1}\boldsymbol{V}^\top = \boldsymbol{UD}^{-1}\boldsymbol{V}^\top = (\boldsymbol{M}^\top)^\dagger$), 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 12), so $\boldsymbol{X}_S^\dagger \boldsymbol{y} = \boldsymbol{w}_S^*$ (III, 2, Corr. 3, [17], [123]).

Therefore, aggregating equation 4.41 for all indices, we finally obtain:

$$\operatorname{conv}\left(\pi_{HT}(-\boldsymbol{X}^\top\boldsymbol{\lambda}^* - \alpha\boldsymbol{w}^*)\right) = (1-\alpha)\boldsymbol{w}^* \tag{4.42}$$

That is, $\boldsymbol{\lambda}^*$ verifies equation 4.37.

So to sum up, under Assumptions 13 and 12, 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 equation 4.37 is equivalent to equation 4.35, plugging that value of $\boldsymbol{\lambda}^*$ into equation 4.35 we also have:

$$(1-\alpha)\partial(\frac{1}{2}\|\cdot\|_k^{sp2})(\boldsymbol{w}^*) \ni \boldsymbol{X}^\top(\boldsymbol{X}_S^\top)^\dagger\boldsymbol{w}_S^* - \alpha\boldsymbol{w}^* \tag{4.43}$$

(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 20.

As described in [99], the dual problem of equation 4.34 can be written as (see e.g. Definition 15.19 in [10]):

$$\min_{\boldsymbol{v}} R^*(-\boldsymbol{X}^\top\boldsymbol{v}) + \langle\boldsymbol{y},\boldsymbol{v}\rangle \tag{4.44}$$

where R^* denotes the Fenchel Dual of R (see Definition 17).

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 equation 4.44 can be written as:

$$\partial f(\boldsymbol{v}) + \boldsymbol{y} \ni \boldsymbol{0} \tag{4.45}$$

Which is equivalent to:

$$-\partial f(\boldsymbol{v}) \ni \boldsymbol{y} \tag{4.46}$$

Therefore, if we find $\boldsymbol{v}$ such that the expression above is verified, then that $\boldsymbol{v}$ is solution of equation 4.44.

Now, from Theorem 23.9 in [127], 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 [81] 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 3, we have the following equivalences:

$$\begin{aligned}
&\boldsymbol{X}^\top (\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* \in \partial R(\boldsymbol{z}) \\
&\iff \boldsymbol{X}^\top (\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* \in (1-\alpha)\partial(\frac{1}{2}\|\cdot\|_k^{sp2})(\boldsymbol{z}) + \alpha \boldsymbol{z} \\
&\iff \boldsymbol{X}^\top (\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^* - \alpha \boldsymbol{z} \in (1-\alpha)\partial(\frac{1}{2}\|\cdot\|_k^{sp2})(\boldsymbol{z})
\end{aligned} \tag{4.47}$$

Now, we know from equation 4.43 that taking $\boldsymbol{z} := \boldsymbol{w}^*$ satisfies expression equation 4.47. Therefore: $\nabla R^*(-\boldsymbol{X}^\top \boldsymbol{v}^*) = \boldsymbol{w}^*$

Now, we can see that the proof is complete, since we know from Assumption 12 that $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^*$. So using the above, we have:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^* = \boldsymbol{X}\nabla R^*(-\boldsymbol{X}^\top \boldsymbol{v}^*) \in \{\boldsymbol{X}\nabla R^*(-\boldsymbol{X}^\top \boldsymbol{v}^*)\} = \boldsymbol{X}\partial R^*(-\boldsymbol{X}^\top \boldsymbol{v}^*) \subset -\partial f(\boldsymbol{v}^*) \tag{4.48}$$

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 equation 4.34.

□

Theorem 11 ([99]). *Let $\delta \in]0,1]$ and let $(\hat{\boldsymbol{w}}_t)_{t\in\mathbb{N}}$ be the sequence generated by ADGD (cf. [99]). 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 equation 4.34. Then, for every $t \geq 2$,*

$$\|\hat{\boldsymbol{w}}_t - \boldsymbol{w}^*\| \leq at\delta + bt^{-1}. \tag{4.49}$$

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

$$\|\hat{\boldsymbol{w}}_t - \boldsymbol{w}^*\| \leq \big[a(c+1) + bc^{-1}\big]\delta^{1/2}. \tag{4.50}$$

Proof. Proof in [99] □

4.6.4 Useful Results

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

Theorem 12 (Corollary 4.3.2, [8]). *Let $f_1, ..., f_m$ be m convex functions from $\mathbb{R}^d$ to $\mathbb{R}$ and define*

$$f := \max\{f_1, ..., f_m\}. \tag{4.51}$$

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

$$\partial f(\boldsymbol{w}) = conv(\cup \partial f_i(\boldsymbol{w}) : i \in I(\boldsymbol{w})) \tag{4.52}$$

Proof. Proof in [8]. □

Lemma 21 (Subgradient of the half-squared top-k norm)**.** *Let n be the (half-squared) top k-norm: $n(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_{(k)}^2$. We have:*

$$\partial n(\boldsymbol{w}) = conv(\pi_{HT}(\boldsymbol{w})) \tag{4.53}$$

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

$$n(\boldsymbol{w}) = \max_i n_i(x) = \max\{n_1(\boldsymbol{w}), ..., n_{\binom{d}{k}}(\boldsymbol{w})\} \tag{4.54}$$

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 $\boldsymbol{w} \in \mathbb{R}^d$, $\Pi(\boldsymbol{w}) \subset \left[{d \atop k}\right]$ 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 12. Those supports are those which select the top-k components of $\boldsymbol{w}$ in absolute value (several choices are possible). In other words:

$$\pi_{HT}(\boldsymbol{w}) = \{\boldsymbol{w}_{\mathcal{I}_j} : j \in \Pi(\boldsymbol{w})\} \tag{4.55}$$

Now, we know that for all $i \in \binom{d}{k}$, 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 12, we have:

$$\partial f(\boldsymbol{w}) = \text{conv}(\nabla f_i(\boldsymbol{w}) : i \in \Pi(\boldsymbol{w})) = \text{conv}(\boldsymbol{w}_{\mathcal{I}_j} : j \in \Pi(x)) = \text{conv}(\pi_{HT}(\boldsymbol{w})) \tag{4.56}$$

□

Proposition 3 (Proposition 11.3, [127])**.** *For any proper, lsc, convex function f, denote by f^* its Fenchel dual defined above in 17. One has $\partial f^* = (\partial f)^{-1}$ and $\partial f = (\partial f^*)^{-1}$.*

Proof. Proof in [127]. □

Lemma 22 (Fenchel conjugate of a half squared norm [24] 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 [24]. □

Lemma 23 (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 19) is the dual norm of the k-support :*

$$(\|\cdot\|_k^{sp})_* = \|\cdot\|_{(k)} \tag{4.57}$$

Corollary 7.

$$(\frac{1}{2}\|\cdot\|_k^{sp2})^* = \frac{1}{2}\|\cdot\|_{(k)}^2 \tag{4.58}$$

Proof. Corollary 7 follows from Lemmas 22 and 23. □

4.6.5 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 [102]. In our code (available at https://github.com/wdevazelhes/IRKSN_AAAI2024), we use an existing implementation from the `modopt` package [54]. Note that Algorithm 1 from [102] 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 [102] for more details on what variables a, b, and c refer to).

Algorithm 11: Computation of $\boldsymbol{x} = \mathrm{prox}_{\frac{\lambda}{2}}|\cdot|(k)^2\,(\boldsymbol{w})$

```
Input   : Parameter: $\lambda$.
Output  : $\boldsymbol{x}$.
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 $\alpha^i$ and $\alpha^{i+1}$ such that $S(\alpha^i) \leq k$ and $S(\alpha^{i+1}) \geq k$ by binary search.;
3. Find $\alpha^*$ between $\alpha^i$ and $\alpha^{i+1}$ 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 $i = 1 \ldots, d$.;
```

4.7 Illustrating Example

In this section, we describe a simple example that illustrates the cases where ℓ_1 norm-based regularization fails, and where 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:

$$X^{(3)} = w_0^{(3)} X^{(0)} + w_1^{(3)} X^{(1)} + w_2^{(3)} X^{(2)} \tag{4.59}$$

and

$$X^{(4)} = w_0^{(4)} X^{(0)} + w_1^{(4)} X^{(1)} + w_2^{(4)} X^{(2)} \tag{4.60}$$

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

$$Y = w_0^{(y)} X^{(0)} + w_1^{(y)} X^{(1)} + w_2^{(y)} X^{(2)} \tag{4.61}$$

A graphical visualization of this construction can be seen on Figure 4.3. More precisely, we define the vectors $\boldsymbol{w}^{(3)}, \boldsymbol{w}^{(4)}$ and $\boldsymbol{w}^{(y)}$ are defined as follows:

$$\boldsymbol{w}^{(3)} = \begin{bmatrix} 9/11 \\ 6/11 \\ 2/11 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{w}^{(4)} = \begin{bmatrix} 1/3 \\ 14/15 \\ 2/15 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{w}^{(y)} = \begin{bmatrix} 1 \\ 1 \\ -4 \\ 0 \\ 0 \end{bmatrix}. \tag{4.62}$$

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 mathematically (using the closed form from the first column of Table 4.1), that this example only verifies our condition (Assumption 13), but that it does not verify Assumption 14 (i.e. it is in the blue area from Figure 4.2). Indeed, in that case, $\boldsymbol{X}_S$ is full column rank, which implies $(\boldsymbol{X}_S)^\dagger \boldsymbol{x}_3 = \boldsymbol{w}^{(3)}$ and $(\boldsymbol{X}_S)^\dagger \boldsymbol{x}_4 = \boldsymbol{w}^{(4)}$ [62]. We then have:

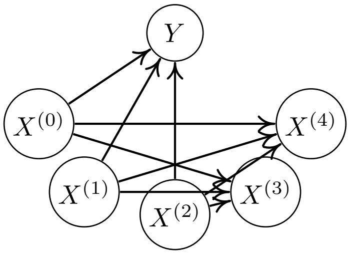


Figure 4.3: $X^{(3)}$, $X^{(4)}$ are correlated with $X^{(0)}, X^{(1)}$, $X^{(2)}$

$$|\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_3, \text{sgn}(\boldsymbol{w}^{(y)}) \rangle| = |\langle \boldsymbol{w}^{(3)}, \text{sgn}(\boldsymbol{w}^{(y)}) \rangle| = 13/11 > 1 \tag{4.63}$$

$$|\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_4, \text{sgn}(\boldsymbol{w}^{(y)}) \rangle| = |\langle \boldsymbol{w}^{(4)}, \text{sgn}(\boldsymbol{w}^{(y)}) \rangle| = 17/15 > 1 \tag{4.64}$$

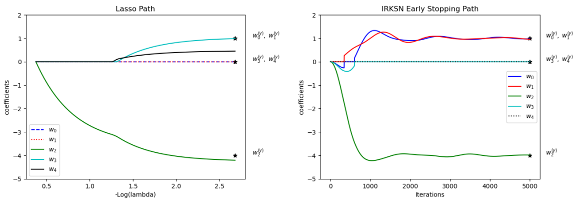
Therefore: $\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \text{sgn}(\boldsymbol{w}_S^*) \rangle| = \frac{13}{11} > 1$ Which means that Assumption 14 is not verified. However, on the other hand, we have:

$$|\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_3, \frac{\boldsymbol{w}^{(y)}}{\min_{j \in S} |w_i^{(y)}|} \rangle| = |\langle \boldsymbol{w}^{(3)}, \frac{\boldsymbol{w}^{(y)}}{\min_{j \in S} |w_i^{(y)}|} \rangle| = \frac{7}{11} \tag{4.65}$$

$$|\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_4, \frac{\boldsymbol{w}^{(y)}}{\min_{j \in S} |w_i^{(y)}|} \rangle| = |\langle \boldsymbol{w}^{(4)}, \frac{\boldsymbol{w}^{(y)}}{\min_{j \in S} |w_i^{(y)}|} \rangle| = \frac{11}{15} \tag{4.66}$$

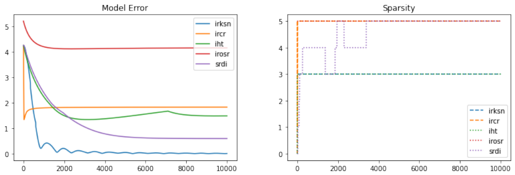
Therefore: $\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \frac{\boldsymbol{w}^{(y)}}{\min_{j \in S} |w_i^{(y)}|} \rangle| = \frac{11}{15} < 1$. Therefore, from the Section *Discussion on the Assumptions*, paragraph *Case where $\boldsymbol{X}_S$ is injective*, we see that our Assumption 13 is verified here.

Comparison of the IRKSN path with Lasso. In Figure 4.4 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 Section 4.8.2.



(a) Lasso path (b) IRKSN path

Figure 4.4: Comparison of the path of IRKSN with Lasso. $w_i^{(y)}$ is the i-th component of $\boldsymbol{w}^{(y)}$, and λ 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 $\boldsymbol{w}^{(y)}$.



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

Figure 4.5: Error and sparsity vs. number of iterations. Only IRKSN can recover the true $\boldsymbol{w}^{(y)}$ in this example.

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.

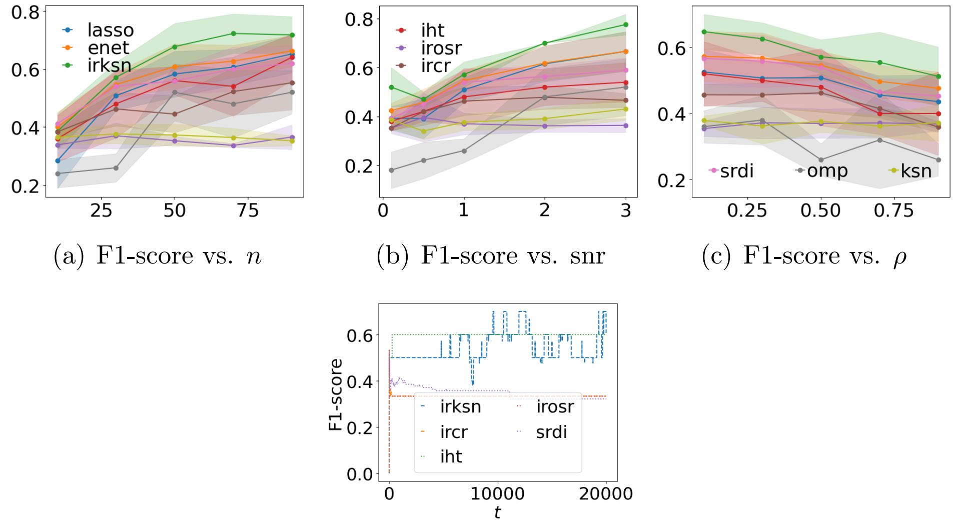
In addition, this path from Figure 4.4 above illustrates well the optimization dynamics of IRKSN: first, the true support of $\boldsymbol{w}^{(y)}$ is not identified in the first iterations. But after a few iterations, we observe what we could call a phenomenon of *exchange of variable*: $w_0^{(y)}$ 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 [20]: 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 4.5, we run the iterative methods from Table 4.1 (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.

4.8 Experiments

4.8.1 Synthetic Example

Below we present experimental results to evaluate the sparse recovery properties of IRKSN. Additional details on those experiments as well as further experiments are provided in the Section.

Experimental Setting. We consider a simple linear regression setting with a correlated design matrix, i.e. where the design matrix $\boldsymbol{X}$ is formed by n i.i.d. samples from d (we take $d = 50$ here) correlated Gaussian random variables $\{X_1, .., X_d\}$ of zero mean and unit variance, such that: $\forall i \in \{1, \dots, d\} : \mathbb{E}[X_i] = 0, \mathbb{E}[X_i^2] = 1$; and $\forall (i, j) \in \{1, \dots, d\}^2, i \neq j : \mathbb{E}[X_i X_j] = \rho^{|i-j|}$. More precisely, we generate each feature X_i in an auto-regressive manner, from previous features, using a correlation $\rho \in [0, 1)$, in the following way: we have $X_1 \sim \mathcal{N}(0, 1)$ and $\sigma^2 = 1 - \rho^2$, and for all $j \in \{2, ..., d\}$: $X_{j+1} = \rho X_j + \epsilon_j$ where $\epsilon_j = \sigma * \Delta$, with $\Delta \sim \mathcal{N}(0, 1)$. Additionally, $\boldsymbol{w}$ is supported on a support, sampled uniformly at random, of $k = 10$ non-zero entries, with each non-zero entry sampled from a normal distribution, and $\boldsymbol{y}$ is obtained with a noise vector $\boldsymbol{\epsilon}$ created from i.i.d. samples from a normal distribution, rescaled to enforce a given signal to noise ratio (SNR), as follows: $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\epsilon}$ with the signal to noise ratio defined as $\text{snr} = \frac{\|\boldsymbol{X}\boldsymbol{w}^*\|}{\|\boldsymbol{\epsilon}\|}$. We generate this dataset using the `make_correlated_data` function from the `benchopt` package [106]. Such a dataset is commonly used to evaluate sparse recovery algorithms (see e.g. [105]), since it possesses correlated features, which is more challenging for sparse recovery (see e.g. the ElasticNet paper, which was motivated by such correlated datasets [165]). In addition, the advantage of such synthetic dataset is that the support is known since it is generated, which therefore allows to evaluate the performance of the algorithms on support recovery, contrary to real-life datasets where a true sparse support of $\boldsymbol{w}$ is hypothetical (or at least often unknown). Additionally, we can notice that such dataset resembles our Example 1, as some features are generated from other features. We evaluate the performance of each final recovered model $\boldsymbol{w}$ using the F1 score on support recovery, defined as follows: $\text{F1} = 2\frac{PR}{P+R}$, with P the precision and R the recall of support recovery, which are defined as: $P = \frac{|\text{supp}(\boldsymbol{w}^*) \cap \text{supp}(\boldsymbol{w})|}{|\text{supp}(\boldsymbol{w})|}$ and $R = \frac{|\text{supp}(\boldsymbol{w}^*) \cap \text{supp}(\boldsymbol{w})|}{|\text{supp}(\boldsymbol{w}^*)|}$. Therefore, the F1 score allows to evaluate at the same time how much of the predicted nonzero elements are accurate, and how much of the actual support has been found. A higher F1 score indicates better identification of the true support. In each experiment (defined by a particular value of n, ρ, snr and a given random seed for generating X, $\boldsymbol{w}^*$ and $\boldsymbol{\epsilon}$), and for each algorithm, we choose the hyperparameters from a grid-search, to attain the best F1-score (we give details on that grid in the Section). For all algorithms which need to set a value k (IRKSN, KSN, IHT), we set k to its true value $k = 10$. In a realistic use-case, since the support is unknown, one may instead tune those hyper-parameters based on a hold-out validation set prediction mean squared error, but tuning those hyperparameters directly for best support F1 score, as we do, allows to evaluate the best potential support recovery capability of each algorithm (e.g. for Lasso it informs us that *there exist* a certain λ, such that we can achieve such a support recovery score). Each experiment is regenerated 5 times with different random seeds, and the average of the obtained best F1 scores, as well as their standard deviation, are reported in Figures 4.6(a), 4.6(c), and 4.6(b), for various values of the dataset parameters, while the others are kept fixed. In Figure 4.6(a), we take $\rho = 0.5$, snr $= 1.$, and $n \in \{10, 30, 50, 70, 90\}$.



(a) F1-score vs. n

(b) F1-score vs. snr

(c) F1-score vs. ρ

(d) F1-score vs. t

Figure 4.6: F1-score of support recovery in various settings.

In Figure 4.6(b), we take $\rho = 0.5$, snr $\in \{0.1, 0.5, 1., 2., 3.\}$, and $n = 30$. In Figure 4.6(c), we take $\rho \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, snr $= 1.$, and $n = 30$. Additionally, we plot on Figure 4.6(d) the evolution of the F1 score along training for iterative algorithms (i.e. algorithms where there is no grid search over a penalty λ, which are IHT, IRKSN, IRCR, IROSR, SRDI), in the case where $n = 30, \text{snr} = 3$, and $\rho = 0.5$.

Results. In all the experiments, as can be expected, we observe that support recovery is more successful when the signal to noise ratio is high, the number of samples is greater, and the correlation ρ is smaller (for that latter point, this is due to the fact that highly correlated datasets are harder for sparse recovery, see e.g. [165] for a discussion on the topic). But overall, we can observe that IRKSN consistently achieves better support recovery than other algorithms from Table 4.1. Also, we can observe on Figure 4.6(d) that IHT and IRKSN maintain a good F1 score after many iterations, while other methods implicitly enforcing an ℓ_1 norm regularization (IRCR, IROSR, SRDI) have poor F1 score in late training.

4.8.1.1 Additional Experimental Details

In this section, we present the hyperparameters for the experiments in the above section, for each algorithm. First, we fix $k = 10$ for all algorithms that require setting a parameter k. We run the algorithms for a maximum number of iterations of 20,000. Note that in this synthetic experiment we do not fit the intercept or center the data since the data has 0 mean. For IHT, we search η in $\{0.0001, 0.001, 0.01, 0.1, 1.\}$. For Lasso, we use the implementation `lasso_path` from `scikit-learn` [122], with its default parameters, which automatically choses the path of λ based on a data criterion. For ElasticNet, we use the implementation `enet_path` from `scikit-learn` [122], 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. For the KSN algorithm (i.e. linear regression penalized with k-support norm), we choose the strenght of the k-support norm penalty λ in $\{0.1, 1.\}$, and we set L (which is the inverse of the learning rate) to $1e6$ similarly as in Section 4.8.4.4. For OMP, we use the implementation from `scikit-learn` [122]. For SRDI, we search for the parameters κ and α from [118], respectively in the intervals $\{0.0001, 0.001, 0.01, 0.1, 1.\}$ and $\{0.0001, 0.001, 0.01, 0.1, 1.\}$. For IROSR, we search for the parameters η and α respectively in $\{0.0001, 0.001, 0.01, 0.1, 1.\}$ and $\{0.0001, 0.001, 0.01, 0.1, 1.\}$. For IRCR, we set τ and σ to $\frac{0.9}{\sqrt{2\|\boldsymbol{X}\|^2}}$ (in order to verify the condition of equation (6) in [105]) similarly as in section 4.8.4.4. For IRKSN (ours), we search α (from Algorithm 10) in $\{0.0001, 0.001, 0.01, 0.1, 1, 10\}$. Our results are produced on a server of CPUs with 16 cores the experiment takes a few hours to run.

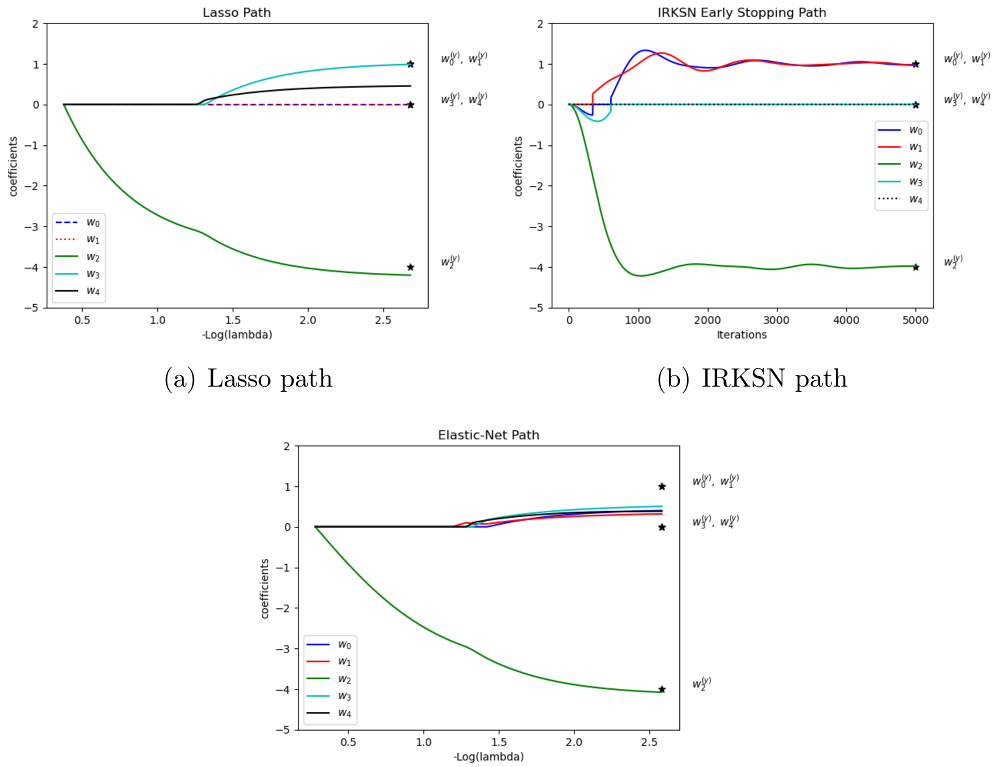
4.8.2 Path of IRKSN vs Lasso vs ElasticNet

In this section, we plot in Figure 4.7 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 *Illustrating Example*. As we can see, the ElasticNet, as the Lasso, cannot recover the true sparse vector.

4.8.3 fMRI Decoding

4.8.3.1 Setting

Data-set Construction. We consider a functional MRI (fMRI) decoding experiment, where observations $\boldsymbol{X}$ are activity recordings (3D activity voxel maps) of fMRI for several subjects which are presented with images of two different classes, i.e. where the observed target $\boldsymbol{y}^\delta$ comprises labels from the set $\{class1, class2\}$ converted to -1 and 1 respectively. It was shown experimentally in [16] that k-support norm regularization (as a penalty) performs significantly better than Lasso on such kind of fMRI tasks: we therefore wish to evaluate whether this is true also for iterative regularization with k-support norm. We use the Haxby dataset [71], downloaded with the use of the `nilearn` package [1]. We then prepare the data from raw recordings following closely the protocol from the fMRI example



(a) Lasso path

(b) IRKSN path

(c) ElasticNet path

Figure 4.7: Comparison of the path of IRKSN with Lasso and Elasticnet.

from the package `hidimstat`[2], choosing the neural recordings of a specific subject (subject number 2), as they do. Once we obtain the data matrix $\boldsymbol{X}$ and target $\boldsymbol{y}^\delta$, we use the algorithms from Table 4.1 to estimate the true model $\boldsymbol{w}^*$, which is an estimate of the brain functional region associated with the true (noiseless) response variable $\boldsymbol{y}$. Such dataset contains 216 samples, of dimensionality 39912. We split the dataset into a training set and a validation set, with the ratio 80%-20%: since we consider only the support reconstruction task, we indeed do not use any test-set in this case.

Hyperparameters and Algorithms Tuning. Below we give more details on the tuning of each algorithm. Once the dataset is prepared, we fine tune the algorithms hyperparameters on mean squared error prediction on the validation set. The intercept of the models is fitted separately, using the same method as in section 4.8.4.4. Additionally, we preprocess first the data by removing features of variance 0 and centering and standardizing $\boldsymbol{X}$, as described in section 4.8.4.4. Additionally, since such dataset of neural images is high dimensional, to reduce the computational cost we use sensible values for hyperparameters whenever those are possible: for instance, for algorithms that have convergence guarantees if the learning rate is equal to the inverse of the Lipschitz-smoothness constant (which in our case is the squared nuclear norm of $\boldsymbol{X}$ (denoted $\|\boldsymbol{X}\|^2$)), we set the learning rate

[2] https://ja-che.github.io/hidimstat/auto_examples/plot_fmri_data_example.html

denoted by η to such value. Also, for all algorithms which require setting a sparsity level k (IRKSN, IHT, KSN, OMP), we set $k = 150$, which is an estimate that we considered known *a priori* for the size of the function region we wish to reconstruct. Additionally, we run all algorithms with a maximum number of iterations of 10,000. For IHT, we set $\eta = \frac{1}{\|\boldsymbol{X}\|^2}$. For Lasso, we use the implementation `lasso_path` from `scikit-learn` [122], with its default parameters, which automatically choses the path of λ based on a data criterion. For ElasticNet, we use the implementation `enet_path` from `scikit-learn` [122], 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. For KSN penalty, we choose the strenght of the k-support norm penalty λ in $\{0.1, 1.\}$, and set $\eta = \frac{1}{\|\boldsymbol{X}\|^2}$. For SRDI, we search for the parameters κ and α from [118], respectively in $\{0.001, 0.01, 0.1\}$ and $\{0.001, 0.01, 0.1\}$. For IROSR, we search for the parameters η and α respectively in $\{0.001, 0.01, 0.1\}$ and $\{0.001, 0.01, 0.1\}$. For IRCR, we set τ and σ to $\frac{0.9}{\sqrt{2\|\boldsymbol{X}\|^2}}$ (in order to verify the condition of equation (6) in [105]). For IRKSN (ours), we set α (from Algorithm 10) to 0.001, since as recommended by Theorem 10, a smaller value of α has more chance to verify the conditions for convergence of 10, (assuming $\boldsymbol{X}$ verifies Assumptions 12 and 13). Additionally, a smaller α ensures that the sparsity of the iterates is closer to k sparse. Our results are produced on a server of CPUs with 16 cores the experiment takes a few hours to run.

Post Processing. Once the estimated model $\boldsymbol{w}$ is returned by each method, we post process $\boldsymbol{w}$ as in [40] Section 3.2: we first compute the corresponding corrected p-values obtained when assumed that the weights values are sampled from a Gaussian distribution (see [40] for more details). Then, we transform those as z-value maps instead of p values maps, and set the FWER (Family-Wise Error Rate) threshold for detection to 0.1 as is done in [40], which is translated into a corresponding threshold for z-values using the Bonferroni correction. We refer the reader to [40] for more details on such post-processing and the related terminology. We then plot in Figure 4.8 the estimated functional region for all of the methods.

4.8.3.2 Results

Visual Comparison On Reconstruction. We plot the fMRI reconstruction results of each method on Figure 4.8, in the case where $class1$ correspond to the class 'face' and $class2$ corresponds to the class 'house'. As a comparison, we also have plotted in Figure 4.8(j) the result of the EnCluDL algorithm from [40] in the same setting, and which may be considered as a ground truth: such method indeed uses knowledge of the spatial structure of the voxel grid (i.e., which voxel is close to each voxel, therefore more likely to be correlated with it), contrary to the methods considered in our paper which are blind to such structure. As we can see, methods based on an implicit or explicit ℓ_1 norm regularization perform poorly, since they tend to estimate a support that is too small: indeed, methods such as the Lasso are known to fail to select group of correlated column, and tend to select only a few explicative features [165]. On the contrary, k-support norm regularization like IRKSN is able to estimate a support of larger size, which by inspection seems to be a better estimate of the ground truth. Additionally, we can observe that even ElasticNet, which supposedly should also be able to perform reasonably well in presence of correlated features [165], does not seem to recover the true functional region: indeed, although its ℓ_1 and ℓ_2 penalties are tuned by grid-search on a validation set, it is more difficult for such method to fix a specific sparsity k. On the other hand, methods which fix a specific k (IHT, KSN, IRKSN, OMP) are advantaged. We can also notice that the solutions of IHT and IRKSN are almost the same, and appear to be the most successful reconstruction of the active functional brain region.

Quantitative Results. Finally, we also provide extra quantitative results in Table 4.2 below for the face/house and house/shoe data splits, in terms of $\|\boldsymbol{w} - \boldsymbol{w}^*\|$, where for the ground truth $\boldsymbol{w}^*$ we take the weight vector obtained by running the EnCluDL method. Note that IHT also has a good performance, but, unlike IRKSN, the theory of sparse recovery for IHT fails to explain such success since (see, e.g. [76]) the RIP property is typically not verified for correlated data (like fMRI [16]):

	Lasso	ElasticNet	OMP	IHT	KSN	IRKSN	IRCR	IROSR	SRDI
face’/’house’	.425	.349	.938	.2441	.247	**.2440**	.341	.381	.314
’house’/’shoe’	.528	.500	.938	.2968	.299	**.2965**	.407	.502	.357

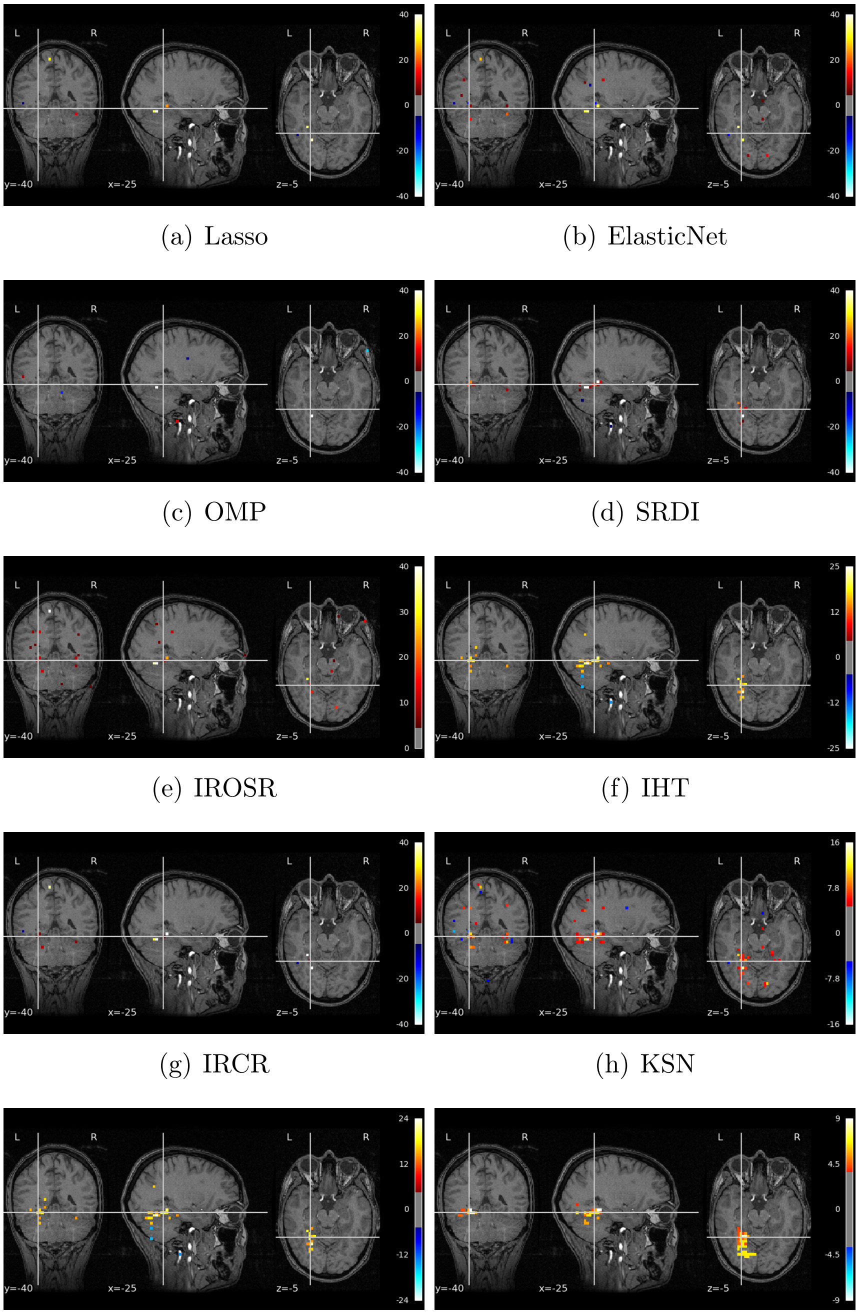
Table 4.2: Comparison of the algorithms on model estimation $\|\boldsymbol{w} - \boldsymbol{w}^*\|$ ($\boldsymbol{w}^*$: weight vector obtained by running the EnCluDL method).

4.8.3.3 Interpretation

Such an fMRI dataset is a real-life (non-synthetic) dataset, therefore its true data generating process is unknown. However, we provide here some attempt to explain the success of k-support norm regularization on such fMRI reconstruction task, in the light of our newly derived sufficient conditions for recovery derived in the paper. More precisely, we present below a data generating process that we believe might potentially be similar to the true underlying data generating process of fMRI observations, and which we will show actually verifies our Assumptions 12 and 13 for recovery with IRKSN.

Example 2: Simplistic fMRI Generating Process. For each observation i, $\boldsymbol{x}_i$ represents the observed activated voxels from the fMRI, so we may consider that they consist in (i) the functional region related to the noiseless target y (i.e. in this case for instance, say, the visual cortex functional region), and (ii) some unrelated regions that are activated for some other reasons (e.g. the functional region responsible for movement if the subject is moving).

Therefore, we model each observation $\boldsymbol{x}_i$ (i.e. row of $\boldsymbol{X}$, seen as a column vector) as follows:



(a) Lasso

(b) ElasticNet

(c) OMP

(d) SRDI

(e) IROSR

(f) IHT

(g) IRCR

(h) KSN

(i) IRKSN

(j) EnCluDL

Figure 4.8: Comparison of different methods on an fMRI decoding task. Figure 4.8(j) is the EnCluDL method from [40] which uses the additional knowledge of the spatial structure of the voxels, and may be considered as some ground truth for the functional region to be reconstructed.

$$\boldsymbol{x}_i = y_i \boldsymbol{w}^* + \boldsymbol{\gamma}_i \tag{4.67}$$

Where $\boldsymbol{w}^*$ is the true model, which support $S = \text{supp}(\boldsymbol{w}^*)$ is the true functional region we wish to reconstruct, y_i is considered to be both the noiseless target variable, but also the variable modulating the functional region: for instance, if y_i denotes the presence or absence of an image in front of the subject, the functional region for visual stimuli will be more or less active depending on y_i, and where $\boldsymbol{\gamma}_i$ is a variable which we consider to have a support disjoint from $supp(\boldsymbol{w}^*)$, which denotes all the other unrelated functional region that are active at observation i (e.g. as discussed above, which can be nonzero if the functional region responsible for, say, movement, or some other regions, are active at the time of measurement i). Let us also assume that the random variable associated with samples $\boldsymbol{\gamma}_i$ are independent of the random variable associated with samples y_i (this corresponds to saying that, say, the event of moving (or any brain activation event unrelated to the activation coming from the presentation of the image), is independent of the event of being presented a certain image). Additionally, let us assume that $\|\boldsymbol{w}^*\| = 1$.

Since $\boldsymbol{\gamma}_i$ and $\boldsymbol{w}^*$ are assumed to have disjoint support, we can therefore verify that, for all samples i:

$$\langle \boldsymbol{x}_i, \boldsymbol{w}^* \rangle = y_i \|\boldsymbol{w}^*\|^2 + 0 = y_i \tag{4.68}$$

Therefore, $\boldsymbol{w}^*$ is indeed a solution of the system $\boldsymbol{X}\boldsymbol{w}^* = \boldsymbol{y}$

Also, we can write $\boldsymbol{X}$ as: $\boldsymbol{X} = \boldsymbol{y}\boldsymbol{w}^{*\top} + \boldsymbol{\Gamma}$, where each row i of $\boldsymbol{\Gamma}$, seen as a column vector, is $\boldsymbol{\gamma}_i$, and based on the assumption above that every $\boldsymbol{\gamma}_i$ has a support disjoint from $\text{supp}(\boldsymbol{w}^*)$, we have:

$$\boldsymbol{X}_S = \boldsymbol{y}\boldsymbol{w}_S^{*\top} = \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|}\|\boldsymbol{y}\|\boldsymbol{w}_S^{*\top} \tag{4.69}$$

Where we can recognize on the right hand side above the SVD of $\boldsymbol{X}_S$, from which we can deduce that:

$$\boldsymbol{X}_S^\dagger = \boldsymbol{w}_S^* \frac{1}{\|\boldsymbol{y}\|}\frac{\boldsymbol{y}^\top}{\|\boldsymbol{y}\|} \tag{4.70}$$

Which implies

$$\boldsymbol{X}_S^\dagger \boldsymbol{y} = \frac{1}{\|\boldsymbol{y}\|^2}\boldsymbol{w}_S^* \boldsymbol{y}^\top \boldsymbol{y} = \boldsymbol{w}_S^* \tag{4.71}$$

And therefore, $\boldsymbol{w}_S^*$ is indeed here the minimum ℓ_2 norm solution of the linear system $\boldsymbol{X}_S \boldsymbol{w}_S^* = \boldsymbol{y}$, since by property of the pseudo-inverse, such minimal ℓ_2 norm solution is $\boldsymbol{X}_S^\dagger \boldsymbol{y}$. Combined with equation 4.68, we obtain that $\boldsymbol{X}$, $\boldsymbol{y}$ and $\boldsymbol{w}$ verify Assumption 12. Now let us consider some $\ell \in S$:

$$\boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell = \frac{1}{\|\boldsymbol{y}\|^2}\boldsymbol{w}_S^* \boldsymbol{y}^\top (\boldsymbol{y} w_\ell) = w_\ell \boldsymbol{w}_S^* \tag{4.72}$$

And therefore

$$\min_{\ell \in S} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^* \rangle| = \|\boldsymbol{w}_S^*\|^2 \min_{\ell \in S} |w_\ell| > 0 \tag{4.73}$$

Where the last inequality is strictly positive, and much greater than 0 if the smallest nonzero value of $\boldsymbol{w}^*$ is big enough (in absolute value). Additionally, on the other hand, if $\ell \in \bar{S}$, since we assumed that $\boldsymbol{\Gamma}$ is composed of variables independent of $\boldsymbol{y}$, and assuming that $\boldsymbol{y}$ and $\boldsymbol{\Gamma_\ell}$ have zero mean for every ℓ, we obtain that for large enough sample size, $\boldsymbol{y}^\top \boldsymbol{\Gamma}_\ell \approx 0$, and therefore we have: $\boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell = \frac{1}{\|\boldsymbol{y}\|^2} \boldsymbol{w}_S^* \boldsymbol{y}^\top \boldsymbol{\Gamma}_\ell \approx 0$, and therefore,

$$\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^* \rangle \approx 0 \tag{4.74}$$

Which therefore implies that :

$$\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^* \rangle| \approx 0 < \min_{\ell \in S} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^* \rangle| \tag{4.75}$$

Which is our Assumption 13, and therefore, $\boldsymbol{X}$, $\boldsymbol{w}^*$ and $\boldsymbol{y}$ verify both Assumptions 12 and 13 which are sufficient conditions for recovery with IRKSN. Also note that however the matrix $\boldsymbol{X}_S$ is not injective here therefore the sufficient Assumption 14 for recovery with ℓ_1 norm is not verified. Therefore, this might potentially explain the success of the k-support norm as a regularizer in fMRI tasks, contrary to ℓ_1 norm based recovery methods which experimentally appear to produce worse results.

Finally, we emphasize that this is only a naive modeling of the true fMRI data, but we believe that it may be useful to understand the success of k-support norm on such particular tasks. It also gives more intuition on our conditions for recovery, and on which kind of tasks k-support norm may be a useful regularizer to consider.

4.8.4 Prediction on Real Data

In this section, we run some experiments on real-life datasets to illustrate the applicability of IRKSN on prediction problems, for various datasets. Although sparse recovery is the primary goal of our paper, and is a goal distinct from prediction, we still find interesting to analyze the performance of IRKSN on predictions tasks, since those also often arise in practice.

4.8.4.1 Setting

As before, 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:

$$\hat{Y} = \langle \hat{\boldsymbol{w}}, X \rangle + b = \sum_{i=1}^{d} \hat{w}_i X_i + b \tag{4.76}$$

(where b is the intercept, fitted separately (see Section 4.8.4.4 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 4.4 and 4.5 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 Section 4.8.4.4. Our results are produced on a server of CPUs with 32 cores and 126G RAM, and take 5 hours to run.

4.8.4.2 Datasets

We evaluate the algorithms on the following open source datasets (obtained from the sources LibSVM [35] and OpenML [144]), of which a brief summary is presented in Table 4.3.

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

Table 4.3: Datasets used in the comparison. *References:* [(1)]: [63], [(2)] [119], [(3)]: [131], [(4)]: [126]. *Sources:* [(1)]: [35], [(2)] [82] downloaded with `scikit-learn` [122], [(3,4)]: [25] .

4.8.4.3 Results

We present our results in Tables 4.4 and 4.5. 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 $d > n$ [165], 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 [145]. However, the theory in [145] holds for small initializations and specific stepsizes, so we hypothesize that due to our grid search on the stepsize, our version of IRCR might be able to explore regimes beyond the ℓ_1 norm, beyond the scope of the theory in the IRCR paper. 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).

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

Table 4.4: Test MSE of the methods of Table 4.1 on the `leukemia` and `housing` datasets (bold font: mean within the standard deviation of the best score from each column).

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

Table 4.5: Test MSE of the methods of Table 4.1 on gene array datasets (`scheetz2006` and `rhee2006`).

4.8.4.4 Details on the Implementation of Algorithms

In this section, we present additional details on the experiments from Section 4.8.4. 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:

$$b = \bar{y} - \langle \bar{\boldsymbol{X}}, \hat{\boldsymbol{w}} \rangle \tag{4.77}$$

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` [70], except for the Lasso, ElasticNet, and OMP, for which we use the `scikit-learn` [122] 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 [54], that is based on the efficient algorithm described in [102]. Below we present the grid-search parameters for each algorithms, that allowed them to achieve a good performance consistently on all datasets from Table 4.3. 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. We also proceed as such for IHT, since because we grid-search the learning rate, if that latter is too high, decrease of the function at each step may not be guaranteed. 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 at `https://github.com/wdevazelhes/IRKSN_AAAI2024`.

IHT [20] 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 [139] We use the implementation `lasso_path` from `scikit-learn` [122], with its default parameters, which automatically choses the path of λ based on a data criterion.

ElasticNet [165] We use the implementation `enet_path` from `scikit-learn` [122], 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 [141] We use the implementation from `scikit-learn` [122], 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 [118] We search for the parameters κ and α from [118], respectively in the intervals $\{0.0001, 0.001, 0.01, 0.1, 1.\}$ and $\{0.0001, 0.001, 0.01, 0.1, 1.\}$.

IROSR [145] 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 [105] 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 [105]) consistently performs well on all datasets.

IRKSN (ours) For IRKSN, we search α (from Algorithm 10) in $\{0.0001, 0.001, 0.01, 0.1, 1, 10\}$, and k (from the k-suppo rt 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.

4.9 Conclusion

In this paper, we introduced an iterative regularization method based on the k-support norm regularization, IRKSN, to complement usual methods based on the ℓ_1 norm. In particular, we gave some condition for sparse recovery with our method, that we analyzed in details and compared to traditional conditions for recovery with ℓ_1 norm regularizers, through an illustrative example. We then gave an early stopping bound for sparse recovery with IRKSN with explicit constants in terms of the design matrix and the true sparse vector. Finally, we evaluated the applicability of IRKSN on several experiments. In future works, it would be interesting to analyze recovery with the s-support norm for general s, where s is not necessarily equal to k: indeed, this setting would generalize both our work and works based on the ℓ_1 norm. We leave this for future work.

References

[1] Alexandre Abraham, Fabian Pedregosa, Michael Eickenberg, Philippe Gervais, Andreas Mueller, Jean Kossaifi, Alexandre Gramfort, Bertrand Thirion, and Gaël Varoquaux. Machine learning for neuroimaging with scikit-learn. *Frontiers in neuroinformatics*, 8:14, 2014.

[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] Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. *Mathematical Programming*, 137:91–129, 2013.

[5] Kyriakos Axiotis and Maxim Sviridenko. Sparse convex optimization via adaptively regularized hard thresholding. *The Journal of Machine Learning Research*, 22(1):5421–5467, 2021.

[6] 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.

[7] 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.

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

[9] Rina Foygel Barber and Wooseok Ha. Gradient descent with non-convex constraints: local concavity determines convergence. *Information and Inference: A Journal of the IMA*, 7:755–806, 2018.

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

[11] Atılım Güneş Baydin, Barak A Pearlmutter, Don Syme, Frank Wood, and Philip Torr. Gradients without backpropagation. *arXiv preprint arXiv:2202.08587*, 2022.

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

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

[14] Amir Beck and Nadav Hallak. On the minimization over sparse symmetric sets: Projections, optimality conditions, and algorithms. *Mathematics of Operations Research*, 41:196–223, 2016.

[15] Amir Beck and Marc Teboulle. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012.

[16] 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.

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

[18] Albert S Berahas, Liyuan Cao, Krzysztof Choromanski, and Katya Scheinberg. A theoretical and empirical comparison of gradient approximations in derivative-free optimization. *Foundations of Computational Mathematics*, pages 1–54, 2021.

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

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

[21] Dankmar Böhning. Multinomial logistic regression algorithm. *Annals of the Institute of Statistical Mathematics*, 44:197–200, 1992.

[22] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146:459–494, 2014.

[23] Radu Ioan Boţ, Ernö Robert Csetnek, and Szilárd Csaba László. An inertial forward–backward algorithm for the minimization of the sum of two nonconvex functions. *EURO Journal on Computational Optimization*, 4:3–25, 2016.

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

[25] Patrick Breheny, 2022.

[26] Joshua Brodie, Ingrid Daubechies, Christine De Mol, Domenico Giannone, and Ignace Loris. Sparse and stable markowitz portfolios. *Proceedings of the National Academy of Sciences*, 106:12267–12272, 2009.

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

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

[29] Leon Bungert, Tim Roith, Daniel Tenbrinck, and Martin Burger. A bregman learning framework for sparse neural networks. *Journal of Machine Learning Research*, 23(192):1–43, 2022.

[30] 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.

[31] 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.

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

[33] Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

[34] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[35] 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.

[36] 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.

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

[38] 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.

[39] 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.

[40] Jérôme-Alexis Chevalier, Tuan-Binh Nguyen, Joseph Salmon, Gaël Varoquaux, and Bertrand Thirion. Decoding with confidence: Statistical control on decoder maps. *NeuroImage*, 234:117921, 2021.

[41] 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.

[42] Saeed Damadi and Jinglai Shen. Gradient properties of hard thresholding operator. *arXiv preprint arXiv:2209.08247*, 2022.

[43] Alberto De Marchi and Andreas Themelis. An interior proximal gradient method for nonconvex optimization. *arXiv preprint arXiv:2208.00799*, 2022.

[44] William de Vazelhes, Bhaskar Mukhoty, Xiao-Tong Yuan, and Bin Gu. Iterative regularization with k-support norm: an important complement to sparse recovery. *arXiv preprint arXiv:2401.05394*, 2023.

[45] William de Vazelhes, Xiaotong Yuan, and Bin Gu. Optimization over sparse restricted convex sets via two steps projection, 2024.

[46] William de Vazelhes, Hualin Zhang, Huimin Wu, Xiaotong Yuan, and Bin Gu. Zeroth-order hard-thresholding: Gradient error vs. expansivity. *Advances in Neural Information Processing Systems*, 35:22589–22601, 2022.

[47] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

[48] Tristan Deleu and Yoshua Bengio. Structured sparsity inducing adaptive optimizers for deep learning. *arXiv preprint arXiv:2102.03869*, 2021.

[49] Victor DeMiguel, Lorenzo Garlappi, Francisco J Nogales, and Raman Uppal. A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms. *Management Science*, 55:798–812, 2009.

[50] SM Moosavi Dezfooli, F Alhussein, F Omar, F Pascal, and S Stefano. Analysis of universal adversarial perturbations. *arXiv preprint arXiv:1705.09554*, 2017.

[51] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[52] Yuri M Ermoliev and Vladimir Ivanovich Norkin. On nonsmooth problems of stochastic systems optimization. 1995.

[53] 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.

[54] 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.

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

[56] Pierre Frankel, Guillaume Garrigos, and Juan Peypouquet. Splitting methods with variable metric for kurdyka–łojasiewicz functions and general convergence rates. *Journal of Optimization Theory and Applications*, 165:874–900, sep 2014.

[57] 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.

[58] 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.

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

[60] 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.

[61] 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.

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

[63] 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.

[64] 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.

[65] 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.

[66] Cristiano Gratton, Naveen KD Venkategowda, Reza Arablouei, and Stefan Werner. Privacy-preserved distributed learning with zeroth-order optimization. *IEEE Transactions on Information Forensics and Security*, 17:265–279, 2021.

[67] Bin Gu, De Wang, Zhouyuan Huo, and Heng Huang. Inexact proximal gradient methods for non-convex and non-smooth optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[68] Bin Gu, Wenhan Xian, Zhouyuan Huo, Cheng Deng, and Heng Huang. A unified q-memorization framework for asynchronous stochastic optimization. *The Journal of Machine Learning Research*, 21(1):7761–7813, 2020.

[69] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pages 1832–1841. PMLR, 2018.

[70] 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.

[71] James V Haxby, M Ida Gobbini, Maura L Furey, Alumit Ishai, Jennifer L Schouten, and Pietro Pietrini. Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, 293(5539):2425–2430, 2001.

[72] T. Hofmann, A. Lucchi, S. Lacoste-Julien, and B. Mcwilliams. Variance reduced stochastic gradient descent with neighbors. *Mathematics*, 2015.

[73] Patrik O Hoyer. Non-negative sparse coding. In *Proceedings of the 12th IEEE workshop on neural networks for signal processing*, pages 557–565. IEEE, 2002.

[74] Feihu Huang, Bin Gu, Zhouyuan Huo, Songcan Chen, and Heng Huang. Faster gradient-free proximal stochastic methods for nonconvex nonsmooth optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1503–1510, 2019.

[75] 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.

[76] 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.

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

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

[79] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

[80] Anatoli Juditsky, Joon Kwon, and Éric Moulines. Unifying mirror descent and dual averaging. *Mathematical Programming*, pages 1–38, 2022.

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

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

[83] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[84] Anastasios Kyrillidis, Stephen Becker, Volkan Cevher, and Christoph Koch. Sparse projections onto the simplex. In *International Conference on Machine Learning*, pages 235–243, 2013.

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

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

[87] Huan Li and Zhouchen Lin. Accelerated proximal gradient methods for nonconvex programming. *Advances in Neural Information Processing Systems*, 28, 2015.

[88] 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.

[89] 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.

[90] Haoyang Liu and Rina Foygel Barber. Between hard and soft thresholding: optimal iterative thresholding algorithms. *Information and Inference: A Journal of the IMA*, 9(4):899–933, 2020.

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

[92] Sijia Liu, Jie Chen, Pin-Yu Chen, and Alfred Hero. Zeroth-order online alternating direction method of multipliers: Convergence analysis and applications. In *International Conference on Artificial Intelligence and Statistics*, pages 288–297. PMLR, 2018.

[93] 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.

[94] 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.

[95] 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.

[96] 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.

[97] Zhaosong Lu. Optimization over sparse symmetric sets via a nonmonotone projected gradient method. *arXiv preprint arXiv:1509.08581*, 2015.

[98] 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.

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

[100] 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.

[101] Andrew M McDonald, Massimiliano Pontil, and Dimitris Stamos. Spectral k-support norm regularization. *Advances in neural information processing systems*, 27, 2014.

[102] 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.

[103] Michael R Metel. Sparse training with lipschitz continuous loss functions and a weighted group l0-norm constraint. *Journal of Machine Learning Research*, 24(103):1–44, 2023.

[104] Konstantin Mishchenko, Ahmed Khaled, and Peter Richtárik. Random reshuffling: Simple analysis with vast improvements. *Advances in Neural Information Processing Systems*, 33:17309–17320, 2020.

[105] 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.

[106] 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*, volume 35, 2022.

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

[108] 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.

[109] 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.

[110] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

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

[112] Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

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

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

[115] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International conference on machine learning*, pages 2613–2621. PMLR, 2017.

[116] 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.

[117] Ryota Nozawa, Pierre-Louis Poirion, and Akiko Takeda. Zeroth-order random subspace algorithm for non-smooth convex optimization. *arXiv preprint arXiv:2401.13944*, 2024.

[118] 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.

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

[120] Lili Pan, Shenglong Zhou, Naihua Xiu, and Hou-Duo Qi. A convergent iterative hard thresholding for nonnegative sparsity optimization. *Pacific Journal of Optimization*, 13:325–353, 2017.

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

[122] 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.

[123] 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.

[124] 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*, 34:8557–8570, 2021.

[125] 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.

[126] 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.

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

[128] R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

[129] 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.

[130] Tim Salimans, Jonathan Ho, Xi Chen, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *CoRR*, abs/1703.03864, 2017.

[131] 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.

[132] 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.

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

[134] David Silver, Anirudh Goyal, Ivo Danihelka, Matteo Hessel, and Hado van Hasselt. Learning by directional gradient descent. In *International Conference on Learning Representations*, 2021.

[135] 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.

[136] James C Spall. A stochastic approximation technique for generating maximum likelihood parameter estimates. In *1987 American control conference*, pages 1161–1167. IEEE, 1987.

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

[138] Akiko Takeda, Mahesan Niranjan, Jun-ya Gotoh, and Yoshinobu Kawahara. Simultaneous pursuit of out-of-sample performance and sparsity in index tracking portfolios. *Computational Management Science*, 10:21–49, 2013.

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

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

[141] 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.

[142] 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.

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

[144] 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.

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

[146] 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.

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

[148] 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.

[149] 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.

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

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

[152] Yi Xu, Rong Jin, and Tianbao Yang. Non-asymptotic analysis of stochastic methods for non-smooth non-convex regularized problems. *Advances in Neural Information Processing Systems*, 32, 2019.

[153] Yi Xu, Qi Qi, Qihang Lin, Rong Jin, and Tianbao Yang. Stochastic optimization for dc functions and non-smooth non-convex regularizers with non-asymptotic convergence. In *International Conference on Machine Learning*, pages 6942–6951, 2019.

[154] Yingzhen Yang and Ping Li. Projective proximal gradient descent for a class of nonconvex nonsmooth optimization problems: Fast convergence without kurdyka-lojasiewicz (kl) property. *arXiv preprint arXiv:2304.10499*, 2023.

[155] Yingzhen Yang and Jiahui Yu. Fast proximal gradient descent for a class of non-convex and non-smooth sparse learning problems. In *Uncertainty in Artificial Intelligence*, pages 1253–1262, 2020.

[156] Ermoliev Yu, V Norkin, and R Wets. The minimization of discontinuous functions: Mollifier subgradients. Technical report, Working Paper, International Institute for Applied Systems Analysis . . . , 1992.

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

[158] 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.

[159] Xinzhe Yuan, William de Vazelhes, Bin Gu, and Huan Xiong. New insight of variance reduce in zero-order hard-thresholding: Mitigating gradient error and expansivity contradictions. In *The Twelfth International Conference on Learning Representations*, 2024.

[160] Pengyun Yue, Long Yang, Cong Fang, and Zhouchen Lin. Zeroth-order optimization with weak dimension dependency. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 4429–4472. PMLR, 2023.

[161] Qingsong Zhang, Bin Gu, Zhiyuan Dang, Cheng Deng, and Heng Huang. Desirable companion for vertical federated learning: New zeroth-order gradient based algorithm. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2598–2607, 2021.

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

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

[164] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.

[165] 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.