IHT DIES HARD: PROVABLE ACCELERATED ITERATIVE HARD THRESHOLDING

RAJIV KHANNA[†] AND ANASTASIOS KYRILLIDIS[⋆]
[†]UNIVERSITY OF TEXAS AT AUSTIN
[⋆]IBM T.J. WATSON RESEARCH CENTER

Abstract. We study –both in theory and practice– the use of momentum motions in classic iterative hard thresholding (IHT) methods. By simply modifying plain IHT, we investigate its convergence behavior on convex optimization criteria with non-convex constraints, under standard assumptions. In diverse scenaria, we observe that acceleration in IHT leads to significant improvements, compared to state of the art projected gradient descent and Frank-Wolfe variants. As a byproduct of our inspection, we study the impact of selecting the momentum parameter: similar to convex settings, two modes of behavior are observed –"rippling" and linear– depending on the level of momentum.

1. Introduction

It is a well-known fact in convex optimization that momentum techniques provably result into significant gains w.r.t. convergence rate. Since 1983, when Nesterov proposed his *optimal gradient methods* [1], these techniques have been used in diverse machine learning and signal processing tasks. Lately, the use of momentum has re-gained popularity in non-convex settings, thanks to their improved performance in structured practical scenaria: from empirical risk minimization (ERM) to training neural networks.

Here, we mainly focus on structured constrained ERM optimization problems:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad x \in \mathcal{C}, \tag{1}$$

that involve convex objectives f and simple structured, but non-convex, constraints $\mathcal{C}$, that can be described using a set of atoms, as in [2, 3]; see also Section 2.1.

Practical algorithms for (1) are convexified projected gradient descent schemes [3], non-convex iterative hard thresholding (IHT) variants [4] and Frank-Wolfe (FW) methods [5]. Convex methods can accommodate acceleration due to [1, 6] and come with rigorous theoretical guarantees; but, higher computational complexity might be observed in practice (depending on the nature of $\mathcal{C}$); further, their configuration could be harder and/or non-intuitive. FW variants [7, 8] simplify the handling of constraints, but the successive construction of estimates –by adding singleton atoms to the putative solution– could slow down convergence. Non-convex methods, such as IHT [9, 10], could be the methods of choice in practice, but only few schemes justify their behavior in theory. Even more importantly, IHT schemes that utilize acceleration inferably are lacking. We defer the discussion on related work to Section 5.

In this work, we study the use of acceleration in IHT settings and supply additional information about open questions regarding the convergence and practicality of such methods on real problems. The current paper provides evidence that "IHT dies hard":

- Accelerated IHT comes with theoretical guarantees for the general minimization problem (1). While recent results [11] focus on plain IHT, there are no results on Accelerated IHT, apart from [12] on specific cases of (1) and under stricter assumptions. The main assumptions made here are the existence of an exact projection operation over the structure set $\mathcal{C}$, as well as standard regularity conditions on the objective function.
- Regarding the effect of the momentum on the convergence behavior, our study justifies that similar –to convex settings– behavior is observed in practice for accelerated IHT: two modes of convergence exist ("rippling" and linear), depending on the level of momentum used per iteration.
- We include extensive experimental results with real datasets and highlight the pros and cons of using IHT variants over state of the art for structured ERM problems.

Our framework applies in numerous structured applications, and one of its primary merits is its flexibility.

2. Problem statement

2.1. **Low-dimensional structures.** Following the notation in [3], let $\mathcal{A}$ denote a set of atoms; *i.e.*, simple building units of general "signals". *E.g.*, we write $x \in \mathbb{R}^n$ as $x = \sum_i w_i a_i$, where w_i are weights and $a_i \in \mathbb{R}^n$ atoms from $\mathcal{A}$.

Given $\mathcal{A}$, let the "norm" function $\|x\|_{0,\mathcal{A}}$ *return the minimum number of superposed atoms that result into* x. Note that $\|\cdot\|_{0,\mathcal{A}}$ is a non-convex entity for the most interesting $\mathcal{A}$ cases. Also, define the support function $\texttt{supp}_{\mathcal{A}}(x)$ as the function that returns the indices of active atoms in x. Associated with $\|\cdot\|_{0,\mathcal{A}}$ is the projection operation over the set $\mathcal{A}$:

$$\Pi_{k,\mathcal{A}}(x) \in \underset{y:\|y\|_{0,\mathcal{A}}\leq k}{\arg\min} \tfrac{1}{2}\|x-y\|_2^2.$$

To motivate our discussion, we summarize some well-known sets $\mathcal{A}$ used in machine learning problems; for a more complete description see [13].

$\mathcal{A}$ *represents plain sparsity:* Let $\mathcal{A} = \{a_i \in \mathbb{R}^n \mid a_i \equiv \pm e_i, \forall i \in [n]\}$, where e_i denotes the canonical basis vector. In this case, k-sparse "signals" $x \in \mathbb{R}^n$ can be represented as a linear combination of k atoms in $\mathcal{A}$: $x = \sum_{i\in\mathcal{I}} w_i a_i$, for $|\mathcal{I}| \leq k$ and $w_i \in \mathbb{R}_+$. The "norm" function is the standard ℓ_0-"norm" and $\Pi_{k,\mathcal{A}}(x)$ finds the k-largest in magnitude entries of x.

$\mathcal{A}$ *represents block sparsity* [14]*:* Let $\{G_1, G_2, \ldots, G_M\}$ be a collection of M non-overlapping group indices such that $\cup_{i=1}^M G_i = [n]$. With a slight abuse of notation, $\mathcal{A} = \{a_i \in \mathbb{R}^n \mid a_i \equiv \cup_{j:j\in G_i} e_j\}$ is the collection of grouped indices, according to $\{G_1, G_2, \ldots, G_M\}$. Then, k-sparse block "signals" $x \in \mathbb{R}^n$ can be expressed as a weighted linear combination of k group atoms in $\mathcal{A}$. The "norm" function is the extension of ℓ_0-"norm" over group structures, and $\Pi_{k,\mathcal{A}}(x)$ finds the k most significant groups (*i.e.*, groups with largest energy).

$\mathcal{A}$ *denotes low rankness:* Let $\mathcal{A} = \{a_i \in \mathbb{R}^{m\times n} \mid a_i = u_i v_i^\top, \|u_i\|_2 = \|v_i\|_2 = 1\}$ be the set of rank-one matrices. Here, sparsity corresponds to low-rankness. The "norm" function corresponds to the notion of rankness; $\Pi_{k,\mathcal{A}}(x)$ finds the best k-rank approximation.

2.2. **Loss function f.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable convex loss function. We consider applications that can be described by *restricted strongly convex and smooth* functions f.

Definition 1. *Let f be convex and differentiable. f is α-restricted strongly convex over $\mathcal{C} \subseteq \mathbb{R}^n$ if:*

$$f(y) \geq f(x) + \langle \nabla f(x),\ y - x\rangle + \tfrac{\alpha}{2}\|x-y\|_2^2,\ \forall x, y \in \mathcal{C}. \tag{2}$$

Definition 2. *Let f be a convex and differentiable. f is β-restricted smooth over $\mathcal{C} \subseteq \mathbb{R}^n$ if:*

$$f(y) \leq f(x) + \langle \nabla f(x),\ y - x\rangle + \tfrac{\beta}{2}\|x-y\|_2^2,\ \forall x, y \in \mathcal{C}. \tag{3}$$

Combined with the above, $\mathcal{C}$ could be the set of rk-sparse vectors, rk-sparse block "signals", etc, for some integer $r > 0$.

2.3. **Optimization criterion.** Given f and a low-dimensional structure $\mathcal{A}$, we focus on the following optimization problem:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad \|x\|_{0,\mathcal{A}} \leq k. \tag{4}$$

Here, $k \in \mathbb{Z}_+$ denotes the level of "succinctness". Examples include (i) sparse and model-based sparse linear regression, (ii) low-rank learning problems, and (iii) model-based, ℓ_2-norm regularized logistic regression tasks; see also Section 6.

3. Accelerated IHT variant

We follow the path of IHT methods. These are first-order gradient methods, that perform per-iteration a non-convex projection over the constraint set $\mathcal{A}$. With math terms, this leads to:

$$x_{i+1} = \Pi_{k,\mathcal{A}}\left(x_i - \mu_i \nabla f(x_i)\right), \text{ where } \mu_i \in \mathbb{R}.$$

While the merits of plain IHT, as described above, are widely known for simple sets $\mathcal{A}$ and specific functions f (*cf.*, [9, 4, 15, 11]), momentum-based acceleration techniques in IHT have not received significant attention in more generic ML settings. Here, we study a simple momentum-variant of IHT, previously proposed in [16, 12], that satisfies the following recursions:

$$x_{i+1} = \Pi_{k,\mathcal{A}}\left(u_i - \mu_i \nabla_{\mathcal{T}_i} f(u_i)\right),$$

and

$$u_{i+1} = x_{i+1} + \tau \cdot (x_{i+1} - x_i). \tag{5}$$

Here, $\nabla_{\mathcal{T}_i} f(\cdot)$ denotes restriction of the gradient on the subspace spanned by set $\mathcal{T}$; more details below. τ is the momentum step size, used to properly weight previous estimates with the current one, based on [17].[1] Despite the simplicity of (5), to the best of our knowledge, there are no convergence guarantees for generic f, neither any characterization of its performance w.r.t. τ values. Nevertheless, its superior performance has been observed under various settings and configurations [16, 19, 12].

In this paper, we study this accelerated IHT variant, as described in Algorithm 1. This algorithm was originally presented in [16, 12]. However, [16, 12] covers only a special case (*i.e.*, squared loss) of our setting, and the theory there is restricted, and further needs justification (*e.g.*, the role of τ in the convergence behavior is not studied). For simplicity, we will focus on the case of sparsity; same notions can be extended to more complicated sets $\mathcal{A}$.

Some notation first: given gradient $\nabla f(x) \in \mathbb{R}^n$, and given a subset of $[n]$, say $\mathcal{T} \subseteq [n]$, $\nabla_{\mathcal{T}} f(x) \in \mathbb{R}^n$ has entries from $\nabla f(x)$, only indexed by $\mathcal{T}$.[2] $\mathcal{T}^c$ represents the complement of $[n] \setminus \mathcal{T}$.

Algorithm 1 Accelerated IHT algorithm

Input: Tolerance η, T, $\alpha, \beta > 0$, model $\mathcal{A}$, $k \in \mathbb{Z}_+$.
Initialize: $x_0, u_0 \leftarrow 0$, $\mathcal{U}_0 \leftarrow \{\emptyset\}$. Set $\xi = 1 - \frac{\alpha}{\beta}$; select τ s.t. $|\tau| \leq \frac{1-\varphi\xi^{1/2}}{\varphi\xi^{1/2}}$, where $\varphi = \frac{1+\sqrt{5}}{2}$.
repeat
  $\mathcal{T}_i \leftarrow \mathtt{supp}_{\mathcal{A}}\left(\Pi_{k,\mathcal{A}}\left(\nabla_{\mathcal{U}_i^c} f(u_i)\right)\right) \cup \mathcal{U}_i$
  $\bar{u}_i = u_i - \frac{1}{\beta}\nabla_{\mathcal{T}_i} f(u_i)$
  $x_{i+1} = \Pi_{k,\mathcal{A}}\left(\bar{u}_i\right)^{\dagger}$
  $u_{i+1} = x_{i+1} + \tau\left(x_{i+1} - x_i\right)$ where $\mathcal{U}_{i+1} \leftarrow \mathtt{supp}_{\mathcal{A}}(u_{i+1})$
until $\|x_i - x_{i-1}\| \leq \eta \|x_i\|$ or after T iterations.
†*Optional*: Debias step on x_{i+1}, restricted on the support $\mathtt{supp}_{\mathcal{A}}(x_{i+1})$.

Algorithm 1 maintains and updates an estimate of the optimum at every iteration. It does so by maintaining two sequences of variables: x_i's that represent our putative estimates per iteration, and u_i's that model the effect of "friction" (memory) in the iterates. The first step in each iteration is *active support expansion*: we expand support set $\mathcal{U}_i$ of u_i, by finding the indices of k atoms of the largest entries in the gradient in the complement of $\mathcal{U}_i$. This step results into set $\mathcal{T}_i$ and makes sure that per iteration we enrich the active support by "exploring" enough outside of it. The following two steps perform the recursion in (5), restricted on $\mathcal{T}_i$; *i.e.*, we perform a gradient step, followed by a projection onto $\mathcal{A}$; finally, we update the auxiliary sequence u by using previous estimates as momentum. The iterations terminate once certain condition holds.

[1] Nesterov's acceleration is an improved version of Polyak's classical momentum [18] schemes. Understanding when and how hard thresholding operations still work for the whole family of momentum algorithms is open for future research direction.
[2] Here, we abuse a bit the notation for the case of low rank structure $\mathcal{A}$: in that case $\nabla_{\mathcal{T}} f(x) \in \mathbb{R}^{m \times n}$ denotes the part of $\nabla f(x)$ that "lives" in the subspace spanned by the atoms in $\mathcal{T}$.

Some observations: Set $\mathcal{T}_i$ has cardinality at most $3k$; x_i estimates are always k-sparse; intermediate "signal" u_i has cardinality at most $2k$, as the superposition of two k-sparse "signals".

4. Theoretical study

Our study[3] starts with the description of the dynamics involved per iteration (Lemma 1), followed by the conditions and eligible parameters that lead to convergence. Proofs are deferred to the Appendix.

Lemma 1 (Iteration invariant)**.** *Consider the non-convex optimization problem in* (4)*, for given structure $\mathcal{A}$, associated with $\Pi_{k,\mathcal{A}}(\cdot)$, and loss function f, satisfying restricted strong convexity and smoothness properties over $4k$ sparse "signals", with parameters α and β, respectively. Let $x^\star$ be the minimizer of f, with $\|x^\star\|_{0,\mathcal{A}} = k$ and $f(x^\star) \leq f(y)$, for any $y \in \mathbb{R}^n$ such that $\|y\|_{0,\mathcal{A}} \leq 3k$. Assuming $x_0 = 0$, Algorithm 1 satisfies $\forall \tau$ the following linear system at the i-th iteration:*

$$\begin{bmatrix} \|x_{i+1} - x^\star\|_2 \\ \|x_i - x^\star\|_2 \end{bmatrix} \leq \underbrace{\begin{bmatrix} \left(1 - \frac{\alpha}{\beta}\right) \cdot |1 + \tau| & \left(1 - \frac{\alpha}{\beta}\right) \cdot |\tau| \\ 1 & 0 \end{bmatrix}}_{:=A} \cdot \begin{bmatrix} \|x_i - x^\star\|_2 \\ \|x_{i-1} - x^\star\|_2 \end{bmatrix}.$$

Proof ideas involved: The proof is composed mostly of algebraic manipulations. For exact projection $\Pi_{k,\mathcal{A}}(\cdot)$ and due to the optimality of the step $x_{i+1} = \Pi_{k,\mathcal{A}}(\bar{u}_i)$, we observe that $\|x_{i+1} - x^\star\|_2^2 \leq 2\langle x_{i+1} - x^\star,\ \bar{u}_i - x^\star\rangle$. Using Definitions 1-2, we prove a version of Lemma 2 in [20] over non-convex constraint sets, using optimality conditions over low-dimensional structures [21]. These steps are admissible due to the restriction of the active subspace to the set $\mathcal{T}_i$ per iteration: most operations –*i.e.*, inner products, Euclidean distances, etc– involved in the proof are applied on "signals" comprised of at most $4k$ atoms. After algebraic "massaging", this leads to the two-step recursion:

$$\begin{aligned} \|x_{i+1} - x^\star\|_2 \leq \left(1 - \tfrac{\alpha}{\beta}\right) \cdot |1 + \tau| \cdot \|x_i - x^\star\|_2 \\ + \left(1 - \tfrac{\alpha}{\beta}\right) \cdot |\tau| \cdot \|x_{i-1} - x^\star\|_2. \end{aligned}$$

Finally, we convert this second-order linear system into a two-dimensional first-order system, that produces the desired recursion. See Appendix A for a detailed proof.

A specific case of the above analysis was presented in [12]; however, the theory specifically applies only to the matrix sensing case over low-rank matrices, using the RIP property. Here, we generalize these results for generic (restricted) strongly convex and smooth functions f, where different theoretical tools apply. Our analysis moves beyond this point, as we show next, in contrast to [12]. Further we investigate a variable τ selection, instead of a constant selection, as in [12].

Remark 1. *The assumption $f(x^\star) \leq f(y)$, for any $y \in \mathbb{R}^n$ such that $\|y\|_{0,\mathcal{A}} \leq 3k$, is trivially satisfied by any noiseless norm-based objective; i.e., for $b = \Phi x^\star$ and $f(x) = \frac{1}{2}\|b - \Phi x\|_2^2$, $f(x^\star) = 0$ for linear regression or $b = \mathcal{M}(X^\star)$ and $f(X) = \frac{1}{2}\|b - \mathcal{M}(X)\|_2^2$, $f(X^\star) = 0$ for low rank recovery problems. We note that this assumption does not restrict our analysis just to the noiseless setting. It states that $x^\star$ has the minimum function value f, among all vectors that are at most $3k$-sparse.* E.g.*, any dense vector, that might be a solution also due to noise, does not affect this requirement. We conjecture that it is an artifact of our proof technique.*

Lemma 1 just states the iteration invariant of Algorithm 1; it does not guarantee convergence. To do so, we need to state some interesting properties of A. The proof is elementary and is omitted.

Lemma 2. *Let A be the 2×2 matrix, as defined above, parameterized by constants $0 < \alpha < \beta$, and user-defined parameter τ. Denote $\xi := 1 - \alpha/\beta$. The characteristic polynomial of A is defined as:*

$$\lambda^2 - \mathrm{Tr}(A) \cdot \lambda + \det(A) = 0$$

where λ represent the eigenvalue(s) of A. Define $\Delta := \mathrm{Tr}(A)^2 - 4 \cdot \det(A) = \xi^2 \cdot (1 + \tau)^2 + 4\xi \cdot |\tau|$. Then, the eigenvalues of A satisfy the expression: $\lambda = \frac{\xi \cdot |1+\tau| \pm \sqrt{\Delta}}{2}$. Depending on the values of α, β, τ:

[3] Our focus is to study optimization guarantees (convergence), not statistical ones (required number of measurements, etc). Our aim is the investigation of accelerated IHT and under which conditions it leads to convergence; not its one-to-one comparison with plain IHT schemes.

- *A has a unique eigenvalue $\lambda = \frac{\xi\cdot|1+\tau|}{2}$, if $\Delta = 0$. This happens when $\alpha = \beta$ and is not considered in this paper (we assume functions f with curvature).*
- *A has two complex eigenvalues; this happens when $\Delta < 0$. By construction, this case does not happen in our scenaria, since $\beta > \alpha$.*
- *For all other cases, A has two distinct real eigenvalues, satisfying $\lambda_{1,2} = \frac{\xi\cdot|1+\tau|}{2} \pm \frac{\sqrt{\xi^2\cdot(1+\tau)+4\xi\cdot|\tau|}}{2}$.*

Define $y(i+1) = \begin{bmatrix}\|x_{i+1} - x^\star\|_2 \\ \|x_i - x^\star\|_2\end{bmatrix}$; then, the linear system in Lemma 1 for the i-th iteration becomes $y(i+1) \leq A \cdot y(i)$. A has only non-negative values; we can unfold this linear system over T iterations such that

$$y(T) \leq A^T \cdot y(0).$$

Here, we make the convention that $x_{-1} = x_0 = 0$, such that $y(0) = \begin{bmatrix}\|x_0 - x^\star\|_2 \\ \|x_{-1} - x^\star\|_2\end{bmatrix} = \begin{bmatrix}1 \\ 1\end{bmatrix} \cdot \|x^\star\|_2$. The following lemma describes how one can compute a power of a 2×2 matrix A, A^i, through the eigenvalues $\lambda_{1,2}$ (real and distinct eigenvalues); the proof is provided in Section C. To the best of our knowledge, there is no detailed proof on this lemma in the literature.

Lemma 3 ([22])**.** *Let A be a 2×2 matrix with real eigenvalues $\lambda_{1,2}$. Then, the following expression holds, when $\lambda_1 \neq \lambda_2$:*

$$A^i = \frac{\lambda_1^i - \lambda_2^i}{\lambda_1 - \lambda_2} \cdot A - \lambda_1\lambda_2 \cdot \frac{\lambda_1^{i-1} - \lambda_2^{i-1}}{\lambda_1 - \lambda_2} \cdot I$$

where λ_i denotes the i-th eigenvalue of A in order.

Then, the main recursion takes the following form:

$$y(T) \leq \frac{\lambda_1^T - \lambda_2^T}{\lambda_1 - \lambda_2} \cdot A \cdot y(0) - \lambda_1\lambda_2 \frac{\lambda_1^{T-1} - \lambda_2^{T-1}}{\lambda_1 - \lambda_2} \cdot y(0). \tag{6}$$

Observe that, in order to achieve convergence (*i.e.*, the RHS convergences to zero), eigenvalues play a crucial role: Both A and $y(0)$ are constant quantities, and only how fast the quantities $\lambda_1^T - \lambda_2^T$ and $\lambda_1^{T-1} - \lambda_2^{T-1}$ "shrink" matter most.

Given that eigenvalues appear in the above expressions in some power (*i.e.*, $\lambda_{1,2}^T$ and $\lambda_{1,2}^{T-1}$), we require $|\lambda_{1,2}| < 1$ for convergence. To achieve $|\lambda_{1,2}| < 1$, we have:

$$\begin{aligned}
|\lambda_{1,2}| &= \left|\tfrac{\xi\cdot|1+\tau|}{2} \pm \sqrt{\tfrac{\xi^2(1+\tau)^2}{4} + \xi\cdot|\tau|}\right| \\
&\leq \left|\tfrac{\xi\cdot|1+\tau|}{2}\right| + \left|\sqrt{\tfrac{\xi^2(1+\tau)^2}{4} + \xi\cdot|\tau|}\right| \\
&\overset{(i)}{\leq} \tfrac{\xi\cdot|1+\tau|}{2} + \tfrac{1}{2}\sqrt{\xi(1+|\tau|)^2 + 4\xi(1+|\tau|)^2} \\
&\overset{(i)}{\leq} \tfrac{\xi^{\frac{1}{2}}(1+|\tau|)}{2} + \tfrac{\sqrt{5}}{2}\xi^{\frac{1}{2}}(1+|\tau|) \\
&= \varphi\cdot\xi^{\frac{1}{2}}(1+|\tau|)
\end{aligned}$$

where (i) is due to $\xi < 1$, and $\varphi = (1+\sqrt{5})/2$ denotes the golden ratio. Thus, upper bounding the RHS to ensure $|\lambda_{1,2}| < 1$ implies $|\tau| < \frac{1-\varphi\cdot\xi^{1/2}}{\varphi\cdot\xi^{1/2}}$.

Using the assumption $|\lambda_{1,2}| < 1$ for $|\tau| < \frac{1-\varphi\cdot\xi^{1/2}}{\varphi\cdot\xi^{1/2}}$, (6) further transforms to:

$$\begin{aligned} y(T) &\leq \frac{\lambda_1^T - \lambda_2^T}{\lambda_1 - \lambda_2} \cdot A \cdot y(0) - \lambda_1\lambda_2 \frac{\lambda_1^{T-1} - \lambda_2^{T-1}}{\lambda_1 - \lambda_2} \cdot y(0) \\ &\overset{(i)}{\leq} \frac{|\lambda_1|^T + |\lambda_2|^T}{|\lambda_1| - |\lambda_2|} \cdot A \cdot y(0) \\ &\qquad + |\lambda_1\lambda_2| \cdot \frac{|\lambda_1|^{T-1} + |\lambda_2|^{T-1}}{|\lambda_1| - |\lambda_2|} \cdot y(0) \\ &\overset{(ii)}{\leq} \frac{2|\lambda_1|^T}{|\lambda_1| - |\lambda_2|} \cdot A \cdot y(0) + |\lambda_1| \cdot \frac{2|\lambda_1|^{T-1}}{|\lambda_1| - |\lambda_2|} \cdot y(0) \end{aligned}$$

where (i) is due to $A \cdot y(0)$ and $y(0)$ being positive quantities, and (ii) is due to $1 > |\lambda_1| > |\lambda_2|$. Focusing on the first entry of $y(T)$ and substituting the first row of A and $y(0)$, we obtain the following inequality:

$$\|x_T - x^\star\|_2 \leq \tfrac{4\cdot|\lambda_1|^T}{|\lambda_1|-|\lambda_2|} \cdot \left(1 - \tfrac{\alpha}{\beta}\right) \cdot |1 + 2\tau| \cdot \|x^\star\|_2. \tag{7}$$

This suggests that, as long as $|\lambda_{1,2}| < 1$, the RHS "shrinks" exponentially with rate $|\lambda_1|^T$, but also depends (inverse proportionally) on the spectral gap $|\lambda_1| - |\lambda_2|$. The above lead to the following convergence result:

Theorem 1. *Consider the non-convex optimization problem in* (4)*, for given structure $\mathcal{A}$, associated with $\Pi_{k,\mathcal{A}}(\cdot)$, and loss function f, satisfying restricted strong convexity and smoothness properties over $4k$ sparse "signals", with parameters α and β, respectively. Under the same assumptions with Lemma 1, Algorithm 1 returns a ε-approximate solution, such that $\|x_T - x^\star\|_2 \leq \varepsilon$, within $O\left(\log \frac{1-\alpha/\beta}{\varepsilon\cdot(|\lambda_1|-|\lambda_2|)}\right)$ iterations (linear convergence rate).*

Proof. We get this result by forcing the RHS of (7) be less than $\varepsilon > 0$. *I.e.*,

$$\tfrac{4\cdot|\lambda_1|^T}{|\lambda_1|-|\lambda_2|} \cdot \left(1 - \tfrac{\alpha}{\beta}\right) \cdot |1 + 2\tau| \cdot \|x^\star\|_2 \leq \varepsilon \Rightarrow \tag{8}$$

$$|\lambda_1|^T \leq \frac{\varepsilon \cdot (|\lambda_1| - |\lambda_2|)}{4 \cdot (1 - \frac{\alpha}{\beta})|1 + 2\tau| \cdot \|x^\star\|_2} \Rightarrow \tag{9}$$

$$T \geq \left\lceil \frac{\log \frac{4\cdot(1-\frac{\alpha}{\beta})|1+2\tau|\cdot\|x^\star\|_2}{\varepsilon\cdot(|\lambda_1|-|\lambda_2|)}}{\log|\lambda_1|} \right\rceil \tag{10}$$

This completes the proof. □

5. Related work

Optimization schemes over low-dimensional structured models have a long history; due to lack of space, we refer the reader to [23] for an overview of discrete and convex approaches. We note that there are both projected and proximal non-convex approaches that fit under our generic model, where no acceleration is assumed. *E.g.*, see [4, 24, 15, 14]; our present work fills this gap. For non-convex proximal steps see [25] and references therein; again no acceleration is considered. Below, we focus on accelerated optimization variants, as well as Frank-Wolfe methods.

Related work on accelerated IHT variants. Accelerated IHT algorithms for sparse recovery were first introduced in [26, 19, 16]. In [26], the authors provide a *double overrelaxation* thresholding scheme [27] in order to accelerate their projected gradient descent variant for sparse linear regression; however, no theoretical guarantees are provided. In [19], Blumensath accelerates standard IHT methods for the same problem [9, 28] using the double overrelaxation technique in [26]. His result contains theoretical proof of linear convergence, under the assumption that the overrelaxation step is used only when the objective function decreases. However, this approach provides no guarantees that we might skip the acceleration term often, which leads back to the non-accelerated IHT version; see also [27] for a similar approach on EM algorithms. [29] describe a family of IHT variants, based on the conjugate gradient method [30], that

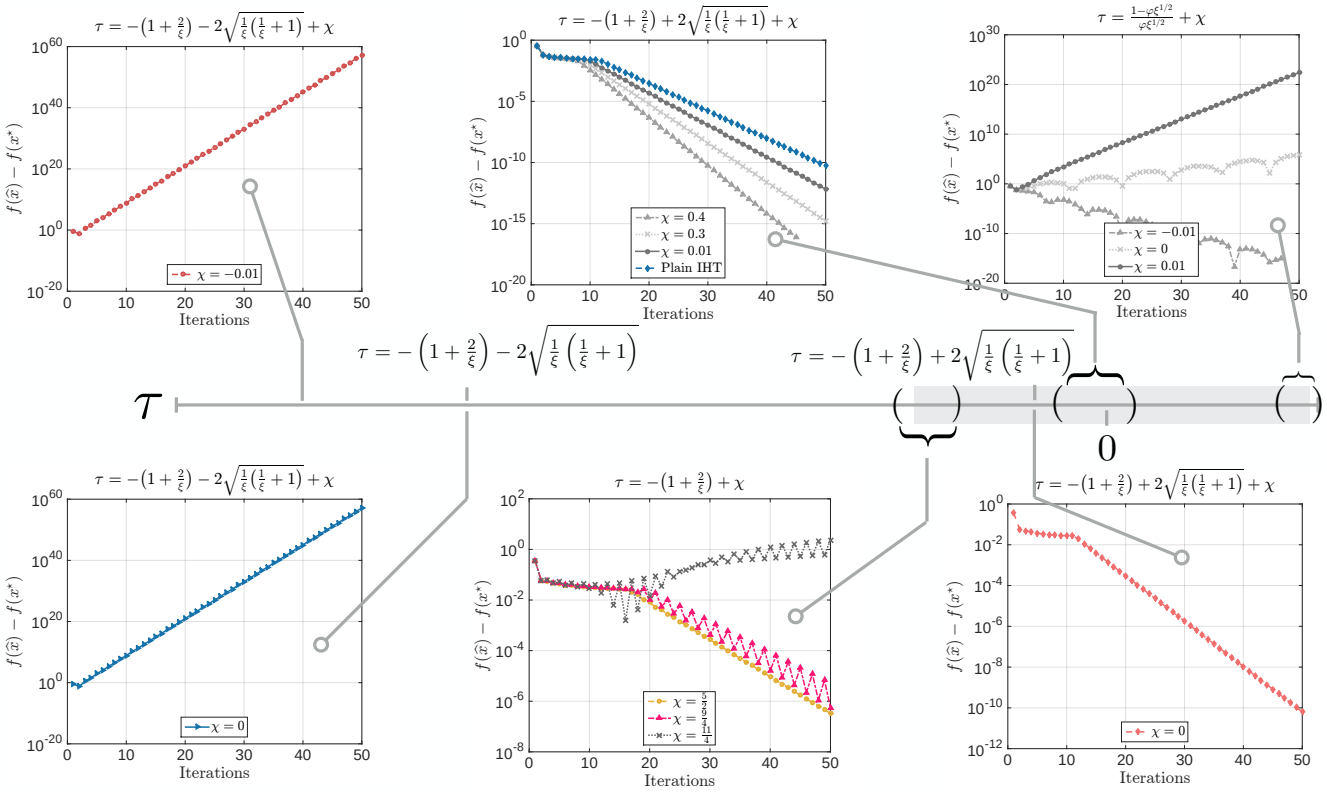

FIGURE 1. Behavior of accelerated IHT method, applied on a toy example for sparse linear regression. Consider $\mathcal{A}$ as the plain sparsity model, and let $x^\star$ be a k-sparse "signal" in $\mathbb{R}^{10}$ for $k = 2$, drawn from multivariate normal distribution. Also, $\|x^\star\|_2 = 1$. Let $b = \Phi x^\star$, with $\Phi \in \mathbb{R}^{6\times 10}$ drawn entrywise i.i.d. from a normal distribution. Let $\mathcal{I}$ be an index set of k columns in Φ; there are $\binom{n}{k}$ possible such subsets. By Definitions 1-2, we estimate α and β as the $\lambda_{\min}(\Phi_{\mathcal{I}}^\top \Phi_{\mathcal{I}})$ and $\lambda_{\max}(\Phi_{\mathcal{I}}^\top \Phi_{\mathcal{I}})$, where $\Phi_{\mathcal{I}}$ is the submatrix of Φ, indexed by $\mathcal{I}$. Here, $\alpha \approx 0.22$ and $\beta \approx 1.78$, which leads to $\xi = 1 - \alpha/\beta \approx 0.87$. We plot $f(\widehat{x}) - f(x^\star)$ vs. iteration count, where $f(x) = \frac{1}{2}\|b - \Phi x\|_2^2$. Gray shaded area on τ horizontal line corresponds to the range $|\tau| \leq (1 - \varphi\xi^{1/2})/(\varphi\xi^{1/2})$. (**Left panel, top and bottom row**). Accelerated IHT diverges for negative τ, outside the τ shaded area. (**Middle panel, bottom row**). "Rippling" behavior for τ values close to the lower bound of converging τ. (**Middle panel, top row**). Convergence behavior for accelerated IHT for various τ values and its comparison to plain IHT ($\tau = 0$). (**Right panel, top row**). Similar "rippling" behavior as τ approaches close to the upper bound of the shaded area; divergence is observed when τ goes beyond the shaded area (observe that, for τ values at the border of the shaded area, Algorithm 1 still diverges and this is due to the approximation of ξ).

includes under its umbrella methods like in [31, 32], with the option to perform acceleration steps; however, no theoretical justification for convergence is provided when acceleration motions are used. [16, 12] contain hard-thresholding variants, based on Nesterov's ideas [17]; in [12], the authors provide convergence rate proofs for accelerated IHT when the objective is just least-squares; no generalization to convex f is provided, neither a study on varied values of τ. [33] includes a first attempt towards using adaptive τ; his approach focuses on the least-squares objective, where a closed for solution for optimal τ is found [16]. However, similar to [19], it is not guaranteed whether and how often the momentum is used, neither how to set up τ in more generic objectives; see also Section D in the appendix. From a convex perspective, where the non-convex constraints are substituted by their convex relaxations (either in constrained or proximal setting), the work in [34] and [35] is relevant to the current work: based on two-step methods for linear

systems [36], [34] extends these ideas to non-smooth (but convex) regularized linear systems, where f is a least-squares term for image denoising purposes; see also [35]. Similar to [33, 19], [34] considers variants of accelerated convex gradient descent that guarantee monotonic decrease of function values per iteration.

Related work on acceleration techniques. Nesterov in [1] was the first to consider acceleration techniques in convex optimization settings; see also Chapter 2.2 in [17]. Such acceleration schemes have been widely used as black box in machine learning and signal processing [35, 34, 37, 38]. [6, 39] discuss restart heuristics, where momentum-related parameters are reset periodically after some iterations. [40] provides some adaptive restart strategies, along with analysis and intuition on why they work in practice for simple convex problems. Acceleration in non-convex settings have been very recently considered in continuous settings [41, 42, 43], where f could be non-convex[4]. However, none of these works, beyond [44], consider non-convex and possibly discontinuous constraints—for instance the subset of k-sparse sets. In the case of [44], our work differs in that it explores better the low-dimensional constraint sets—however, we require f to be convex. More relevant to this work is [45]: the authors consider non-convex proximal objective and apply ideas from [35] that lead to either monotone (skipping momentum steps) or nonmonotone function value decrease behavior; further, the theoretical guarantees are based on different tools than ours. We identify that such research questions could be directions for future work.

Related work on dynamical systems and numerical analysis. Multi-step schemes originate from explicit finite differences discretization of dynamical systems; *e.g.*, the so-called Heavy-ball method [18] origins from the discretization of the friction dynamical system $\ddot{x}(t) + \gamma\dot{x}(t) + \nabla f(x(t)) = 0$, where $\gamma > 0$ plays the role of friction. Recent developments on this subject can be found in [46]; see also references therein. From a different perspective, Scieur et al. [47] use multi-step methods from numerical analysis to discretize the gradient flow equation. We believe that extending these ideas in non-convex domains (*e.g.*, when non-convex constraints are included) is of potential interest for better understanding when and why momentum methods work in practical structured scenaria.

Related work on Frank-Wolfe variants: The Frank-Wolfe (FW) algorithm [5, 7] is an iterative projection-free convex scheme for constrained minimization. Frank-Wolfe often has *cheap* per iteration cost by solving a constrained linear program in each iteration. The classical analysis by [5] presents sublinear convergence for general functions. For strongly convex functions, FW admits linear convergence if the optimum does not lie on the boundary of the constraint set; in that case, the algorithm still has sublinear convergence rate. To address the boundary issue, [48] allows to move away from one of the already selected atoms, where linear convergence rate can be achieved [8]. Similarly, the *pairwise* variant introduced by [49] also has a linear convergent rate. This variant adjusts the weights of two of already selected atoms. [50] present a different perspective by showing linear convergence of classical FW over strongly convex sets and general functions. While several variants and sufficient conditions exist that admit linear convergence rates, the use of momentum for Frank-Wolfe, to the best of our knowledge is unexplored.

6. Experiments

We conducted simulations for different problems to verify our predictions. In all experiments, we use constant $\tau = 1/4$ as a potential universal momentum parameter. Our experiments are proof of concept and demonstrate that accelerated projected gradient descent over non-convex structured sets can, not only offer high-quality recovery in practical settings, but offer much more scalable routines, compared to state-of-the-art. *Here, we present only a subset of our experimental findings and we encourage readers to go over the experimental results in the Appendix E.*

6.1. Sparse linear regression setting.

For sparse linear regression, next we consider two simulated problems settings: (i) with i.i.d. regressors, and (ii) with correlated regressors.

Sparse linear regression under the i.i.d. Gaussian setting: In this case, we consider a similar problem setting with [8], where $x^\star \in \mathbb{R}^n$ is the unknown normalized k-sparse vector, observed through the underdetermined set of linear equations: $b = \Phi x^\star$. $\Phi \in \mathbb{R}^{m\times n}$ is drawn randomly from a normal distribution. We consider the standard least squares objective $f(x) = \frac{1}{2}\|b - \Phi x\|_2^2$ and the plain sparsity model $\mathcal{A}$ where $\|x\|_{0,\mathcal{A}} \equiv \|x\|_0$.

[4]The guarantees in these settings are restricted to finding a good stationary point.

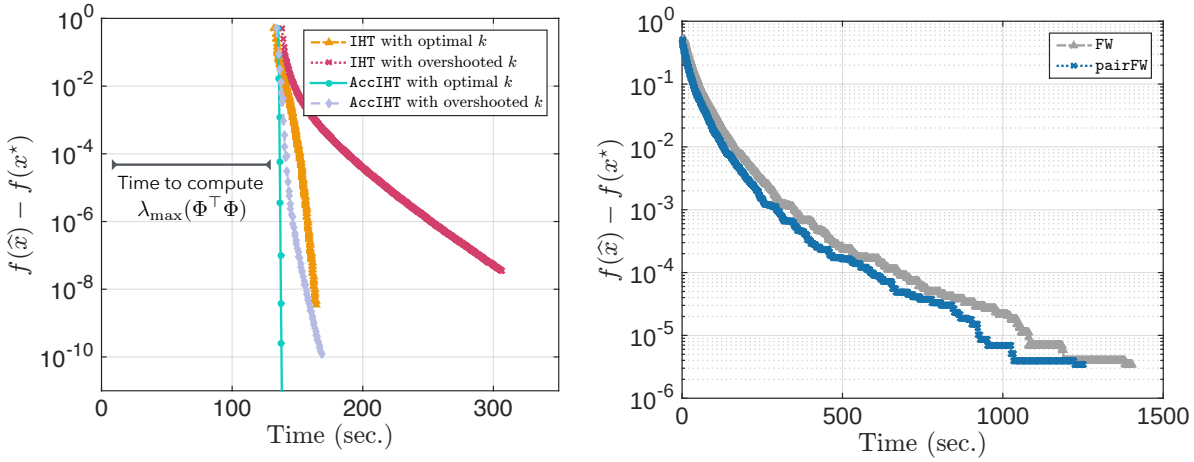

FIGURE 2. Time spent vs. function values gap $f(\widehat{x}) - f(x^\star)$. `AccIHT` corresponds to Algorithm 1. Beware in left plot the time spent to approximate step size, via computing $\lambda_{\max}(\Phi^\top\Phi)$.

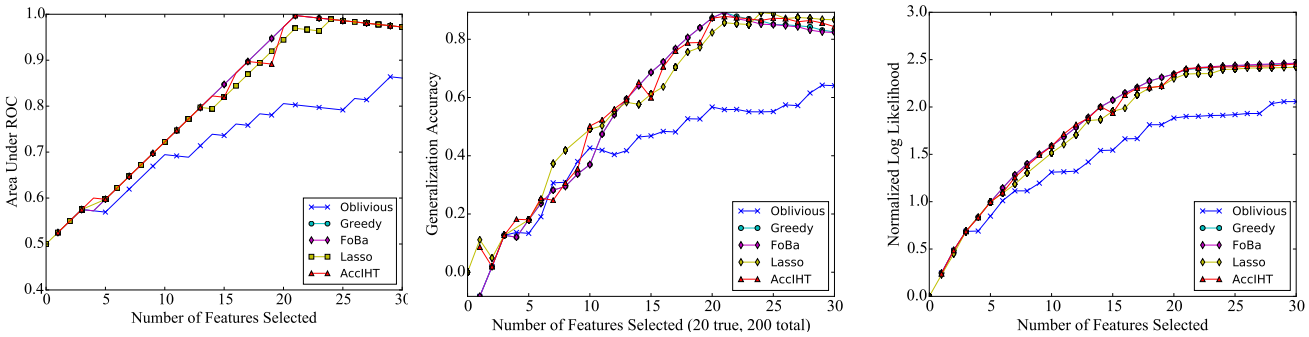

FIGURE 3. Empirical evaluation. Algorithm 1 achieves strong performance on true support recovery (left), generalization on test data (middle), and on training data fit (right)

We compare Algorithm 1 (abbreviated as AccIHT in plots) with two FW variants (see [8] for `FW` and `pairFW`)[5] Further, according to [8], `pairFW` performs better than `awayFW` for this problem case, as well as the plain IHT algorithm. In this experiment, we use step sizes $1/\widehat{\beta}$ for Algorithm 1 and IHT, where $\widehat{\beta} = \lambda_{\max}(\Phi^\top\Phi)$. For the FW variants, we follow the setup in [8] and set as the constraint set the λ-scaled ℓ_1-norm ball, where $\lambda = 40$. In the FW code, we further "debias" the selected set of atoms per iteration, by performing fully corrective steps over putative solution (*i.e.*, solve least-squares objective restricted over the active atom set) [16, 14]. We observed that such steps are necessary in our setting, in order FW to be competitive with Algorithm 1 and IHT. For IHT and Algorithm 1, we set input k either exactly or use the ℓ_1-norm phase transition plots [51], where the input parameter for sparsity $\widehat{k}$ is overshooted. See also

[5]Phase transition results and comparisons to standard algorithms such as CoSaMP (restricted to squared loss) can be found in [16], and thus omitted.

Section B for more information. We compare the above algorithms w.r.t. function values decrease and running times.

Figure 2 depicts the summary of results we observed for the case $n = 2 \cdot 10^5$, $m = 7500$ and $k = 500$. For IHT and accelerated IHT, we also consider the case where the input parameter for sparsity is set to $\widehat{k} = 2441 > k$. The graphs indicate that the accelerated hard thresholding technique can be much more efficient and scalable than the rest of the algorithms, while at the same time being at least as good in support/"signal" recovery performance. For instance, while Algorithm 1 is only $1.2\times$ faster than IHT, when k is known exactly, Algorithm 1 is more resilient at overshooting k: in that case, IHT could take $> 2\times$ time to get to the same level of accuracy. At the same time, Algorithm 1 detects much faster the correct support, compared to plain IHT. Compared to FW methods (right plot), Algorithm 1 is at least $10\times$ faster than FW variants.

As stated before, we only focus on the optimization efficiency of the algorithms, not their statistical efficiency. That being said, we consider settings that are above the phase retrieval curve [], and here we make no comparisons and claims regarding the number of samples required to complete the sparse linear regression task. We leave such work for an extended version of this work.

Sparse linear regression with correlated regressors: In this section, we test Algorithm 1 for support recovery, generalization and training loss performance in the sparse linear regression, under a different data generation setting. We generate the data as follows. We generate the feature matrix 800×200 design matrix Φ according to a first order auto-regressive process with `correlation` $= 0.4$. This ensures features are correlated with each other, which further makes feature selection a non-trivial task. We normalize the feature matrix so that each feature has ℓ_2-norm equal to one. We generate an arbitrary weight vector $x^\star$ with $\|x^\star\|_0 = 20$ and $\|x^\star\|_2 = 1$. The response vector b is then computed as $y = \Phi x^\star + \varepsilon$, where ε is guassian iid noise that is generated to ensure that the signal-to-noise ratio is 10. Finally, the generated data is randomly split 50-50 into training and test sets.

We compare against Lasso [52], oblivious greedy selection (Oblivious [53]), forward greedy selection (Greedy [53]), and forward backward selection (FoBa [54]). The metrics we use to compare on are the generalization accuracy (R^2 coefficient determination performance on test set), recovery of true support (AUC metric on predicted support vs. true support), and training data fit (log likelihood on the training set). The results are presented in Figure 3, and shows Algorithm 1 performs very competitively: it is almost always better or equal to other methods across different sparsity levels.

7. Overview and future directions

The use of hard-thresholding operations is widely known. In this work, we study acceleration techniques in simple hard-thresholding gradient procedures and characterize their performance; to the best of our knowledge, this is the first work to provide theoretical support for these type of algorithms. Our preliminary results show evidence that machine learning problems can be efficiently solved using our accelerated non-convex variant, which is at least competitive with state of the art and comes with convergence guarantees.

Our approach shows linear convergence; however, in theory, the acceleration achieved has dependence on the condition number of the problem not better than plain IHT. This leaves open the question on what types of conditions are sufficient to guarantee the better acceleration of momentum in such non-convex settings?

Apart from the future directions "scattered" in the main text, another possible direction lies at the intersection of dynamical systems and numerical analysis with optimization. Recent developments on this subject can be found in [46] and [47]. We believe that extending these ideas in non-convex domains is interesting to better understand when and why momentum methods work in practical structured scenaria.

References

[1] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(\frac{1}{k^2})$. In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

[2] S. Negahban, B. Yu, M. Wainwright, and P. Ravikumar. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*, pages 1348–1356, 2009.

[3] V. Chandrasekaran, B. Recht, P. Parrilo, and A. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational mathematics*, 12(6):805–849, 2012.

[4] S. Bahmani, B. Raj, and P. Boufounos. Greedy sparsity-constrained optimization. *Journal of Machine Learning Research*, 14(Mar):807–841, 2013.

[5] K. Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4):63, 2010.

[6] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[7] M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML (1)*, pages 427–435, 2013.

[8] S. Lacoste-Julien and M. Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. In *Advances in Neural Information Processing Systems*, pages 496–504, 2015.

[9] T. Blumensath and M. Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

[10] P. Jain, R. Meka, and I. Dhillon. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems*, pages 937–945, 2010.

[11] R. F. Barber and W. Ha. Gradient descent with nonconvex constraints: Local concavity determines convergence. *arXiv preprint arXiv:1703.07755*, 2017.

[12] A. Kyrillidis and V. Cevher. Matrix recipes for hard thresholding methods. *Journal of mathematical imaging and vision*, 48(2):235–265, 2014.

[13] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Structured sparsity through convex optimization. *Statistical Science*, pages 450–468, 2012.

[14] P. Jain, N. Rao, and I. Dhillon. Structured sparse regression via greedy hard thresholding. In *Advances in Neural Information Processing Systems*, pages 1516–1524, 2016.

[15] P. Jain, A. Tewari, and P. Kar. On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems*, pages 685–693, 2014.

[16] A. Kyrillidis and V. Cevher. Recipes on hard thresholding methods. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2011 4th IEEE International Workshop on*, pages 353–356. IEEE, 2011.

[17] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[18] B. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[19] T. Blumensath. Accelerated iterative hard thresholding. *Signal Processing*, 92(3):752–756, 2012.

[20] A. Agarwal, S. Negahban, and M. Wainwright. Fast global convergence rates of gradient methods for high-dimensional statistical recovery. In *Advances in Neural Information Processing Systems*, pages 37–45, 2010.

[21] A. Beck and N. Hallak. On the minimization over sparse symmetric sets: projections, optimality conditions, and algorithms. *Mathematics of Operations Research*, 41(1):196–223, 2015.

[22] K. Williams. The n-th power of a 2×2 matrix. *Mathematics Magazine*, 65(5):336, 1992.

[23] A. Kyrillidis, L. Baldassarre, M. El Halabi, Q. Tran-Dinh, and V. Cevher. Structured sparsity: Discrete and convex approaches. In *Compressed Sensing and its Applications*, pages 341–387. Springer, 2015.

[24] X. Yuan, P. Li, and T. Zhang. Gradient hard thresholding pursuit for sparsity-constrained optimization. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 127–135, 2014.

[25] P. Gong, C. Zhang, Z. Lu, J. Huang, and J. Ye. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *ICML (2)*, pages 37–45, 2013.

[26] K. Qiu and A. Dogandzic. ECME thresholding methods for sparse signal reconstruction. *arXiv preprint arXiv:1004.4880*, 2010.

[27] R. Salakhutdinov and S. Roweis. Adaptive overrelaxed bound optimization methods. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 664–671, 2003.

[28] R. Garg and R. Khandekar. Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 337–344. ACM, 2009.

[29] J. Blanchard, J. Tanner, and K. Wei. CGIHT: Conjugate gradient iterative hard thresholding for compressed sensing and matrix completion. *Information and Inference*, 4(4):289–327, 2015.

[30] M. Hestenes and E. Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS, 1952.

[31] D. Needell and J. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

[32] S. Foucart. Hard thresholding pursuit: an algorithm for compressive sensing. *SIAM Journal on Numerical Analysis*, 49(6):2543–2563, 2011.

[33] K. Wei. Fast iterative hard thresholding for compressed sensing. *IEEE Signal Processing Letters*, 22(5):593–597, 2015.
[34] J. Bioucas-Dias and M. Figueiredo. A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image processing*, 16(12):2992–3004, 2007.
[35] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.
[36] O. Axelsson. *Iterative solution methods*. Cambridge University press, 1996.
[37] M. Schmidt, N. Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in neural information processing systems*, pages 1458–1466, 2011.
[38] S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *ICML*, pages 64–72, 2014.
[39] S. Becker, E. Candès, and M. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical programming computation*, 3(3):165–218, 2011.
[40] B. O'Donoghue and E. Candes. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15(3):715–732, 2015.
[41] S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, 2016.
[42] Y. Carmon, J. Duchi, O. Hinder, and A. Sidford. Accelerated methods for non-convex optimization. *arXiv preprint arXiv:1611.00756*, 2016.
[43] N. Agarwal, Z. Allen-Zhu, B. Bullins, E. Hazan, and T. Ma. Finding approximate local minima for nonconvex optimization in linear time. *arXiv preprint arXiv:1611.01146*, 2016.
[44] C. Paquette, H. Lin, D. Drusvyatskiy, J. Mairal, and Z. Harchaoui. Catalyst acceleration for gradient-based non-convex optimization. *arXiv preprint arXiv:1703.10993*, 2017.
[45] H. Li and Z. Lin. Accelerated proximal gradient methods for nonconvex programming. In *Advances in neural information processing systems*, pages 379–387, 2015.
[46] A. Wilson, B. Recht, and M. Jordan. A lyapunov analysis of momentum methods in optimization. *arXiv preprint arXiv:1611.02635*, 2016.
[47] D. Scieur, V. Roulet, F. Bach, and A. d'Aspremont. Integration methods and accelerated optimization algorithms. *arXiv preprint arXiv:1702.06751*, 2017.
[48] P. Wolfe. Convergence theory in nonlinear programming. *Integer and nonlinear programming*, pages 1–36, 1970.
[49] B. Mitchell, V. Demyanov, and V. Malozemov. Finding the point of a polyhedron closest to the origin. *SIAM Journal on Control*, 12(1):19–26, 1974.
[50] D. Garber and E. Hazan. Faster rates for the Frank-Wolfe method over strongly-convex sets. In *ICML*, pages 541–549, 2015.
[51] D. Donoho and J. Tanner. Neighborliness of randomly projected simplices in high dimensions. *Proceedings of the National Academy of Sciences of the United States of America*, 102(27):9452–9457, 2005.
[52] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
[53] A. Das and D. Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1057–1064, 2011.
[54] T. Zhang. Adaptive forward-backward greedy algorithm for sparse learning with linear models. In *Advances in Neural Information Processing Systems*, pages 1921–1928, 2009.
[55] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.
[56] B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.
[57] A. Tillmann and M. Pfetsch. The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing. *IEEE Transactions on Information Theory*, 60(2):1248–1259, 2014.
[58] T. Blumensath. Sampling and reconstructing signals from a union of linear subspaces. *IEEE Transactions on Information Theory*, 57(7):4660–4671, 2011.
[59] P. Shah and V. Chandrasekaran. Iterative projections for signal identification on manifolds: Global recovery guarantees. In *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pages 760–767. IEEE, 2011.
[60] A. Kyrillidis and V. Cevher. Combinatorial selection and least absolute shrinkage via the CLASH algorithm. In *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pages 2216–2220. IEEE, 2012.
[61] C. Hegde, P. Indyk, and L. Schmidt. Approximation algorithms for model-based compressive sensing. *IEEE Transactions on Information Theory*, 61(9):5129–5147, 2015.
[62] L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning*, pages 433–440. ACM, 2009.
[63] S. Negahban and M. Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13(May):1665–1697, 2012.

[64] S. Bhojanapalli, A. Kyrillidis, and S. Sanghavi. Dropping convexity for faster semi-definite optimization. In *Conference on Learning Theory*, pages 530–582, 2016.

[65] D. Park, A. Kyrillidis, C. Caramanis, and S. Sanghavi. Finding low-rank solutions to matrix problems, efficiently and provably. *arXiv preprint arXiv:1606.03168*, 2016.

Appendix A. Proof of Lemma 1

We start by observing that: $\|x_{i+1} - \bar{u}_i\|_2^2 \leq \|x^\star - \bar{u}_i\|_2^2$, due to the exactness of the hard thresholding operation. Note that this holds for most $\mathcal{A}$ of interest[6] but, again for simplicity, we will focus on the sparsity model here. By adding and subtracting $x^\star$ and expanding the squares, we get:

$$\begin{aligned}
\|x_{i+1} - x^\star + x^\star - \bar{u}_i\|_2^2 \leq \|x^\star - \bar{u}_i\|_2^2 \Rightarrow \\
\|x_{i+1} - x^\star\|_2^2 + \|x^\star - \bar{u}_i\|_2^2 + 2\left\langle x_{i+1} - x^\star,\ x^\star - \bar{u}_i\right\rangle \leq \|x^\star - \bar{u}_i\|_2^2 \Rightarrow \\
\|x_{i+1} - x^\star\|_2^2 \leq 2\left\langle x_{i+1} - x^\star,\ \bar{u}_i - x^\star\right\rangle
\end{aligned}$$

From the restricted strong convexity assumption over at most $4k$ sparse "signals", we have:

$$\begin{aligned}
f(x^\star) \geq f(\bar{u}_i) + \langle \nabla f(\bar{u}_i),\ x^\star - \bar{u}_i\rangle + \tfrac{\alpha}{2}\|\bar{u}_i - x^\star\|_2^2 \Rightarrow \\
f(x^\star) - \tfrac{\alpha}{2}\|\bar{u}_i - x^\star\|_2^2 \geq f(\bar{u}_i) + \langle \nabla f(\bar{u}_i),\ x^\star - \bar{u}_i\rangle \\
= f(\bar{u}_i) + \langle \nabla f(\bar{u}_i),\ x^\star - x_{i+1} + x_{i+1} - \bar{u}_i\rangle \\
= f(\bar{u}_i) + \langle \nabla f(\bar{u}_i),\ x_{i+1} - \bar{u}_i\rangle + \langle \nabla f(\bar{u}_i),\ x^\star - x_{i+1}\rangle
\end{aligned}$$

Define $\phi_i(z) := f(\bar{u}_i) + \langle \nabla f(\bar{u}_i),\ z - \bar{u}_i\rangle + \frac{\beta}{2}\|z - \bar{u}_i\|_2^2$. Further, as claimed in Algorithm 1, we assume constant step size $\frac{\mu_i}{2} := \frac{1}{\beta}$, for all iterations i. Then, it is easy to see that:

$$\min_{z:\ \|z\|_{0,\mathcal{A}} \leq k} \phi_i(z) \equiv \min_{z:\ \|z\|_{0,\mathcal{A}} \leq k} \frac{\beta}{2}\left\|z - \left(\bar{u}_i - \tfrac{1}{\beta}\nabla f(\bar{u}_i)\right)\right\|_2^2,$$

and thus, x_{i+1} *is the minimizer of* $\phi_i(z)$ *under* $\mathcal{A}$ *constraints*, by construction.

Let us denote the support of $x^\star$ as $\mathcal{X}^\star$. Further, denote as $\mathcal{E} := \mathcal{T}_i \cup \mathcal{X}^\star$. This implies that all the following hold: (i) $u_i \in \mathcal{E}$, (ii) $\bar{u}_i \in \mathcal{E}$, (iii) $x^\star \in \mathcal{E}$, (iv) $x_{i+1} \in \mathcal{E}$ and, (v) the cardinality of $\mathcal{E}$ satisfies $|\mathcal{E}| \leq 4k$, where k denotes the sparsity. Denote $\mathcal{P}_{\mathcal{E}}(\cdot)$ the subspace projection operator (*i.e.*, in the sparsity case it means that it keeps only the elements indexed by $\mathcal{E}$).

By Remark 5.1(b) in [21] and given that x_{i+1} is a basic feasible point, the following holds[7]:

$$\begin{aligned}
\langle \nabla\phi_i(x_{i+1}),\ x^\star - x_{i+1}\rangle = \langle \nabla\phi_i(x_{i+1}),\ \mathcal{P}_{\mathcal{E}}\left(x^\star - x_{i+1}\right)\rangle \\
= \langle \mathcal{P}_{\mathcal{E}}\left(\nabla\phi_i(x_{i+1})\right),\ \mathcal{P}_{\mathcal{E}}\left(x^\star - x_{i+1}\right)\rangle \geq 0
\end{aligned}$$

By the definition of $\phi_i(z)$, the above inequality leads to:

$$\begin{aligned}
\langle \mathcal{P}_{\mathcal{E}}\left(\nabla f(\bar{u}_i) + \beta\left(x_{i+1} - \bar{u}_i\right)\right),\ \mathcal{P}_{\mathcal{E}}\left(x^\star - x_{i+1}\right)\rangle \geq 0 \Rightarrow \\
\langle \mathcal{P}_{\mathcal{E}}\left(\nabla f(\bar{u}_i)\right),\ \mathcal{P}_{\mathcal{E}}\left(x^\star - x_{i+1}\right)\rangle \geq \beta \cdot \langle \mathcal{P}_{\mathcal{E}}\left(x_{i+1} - \bar{u}_i\right),\ \mathcal{P}_{\mathcal{E}}\left(x_{i+1} - x^\star\right)\rangle \Rightarrow \\
\langle \nabla f(\bar{u}_i),\ x^\star - x_{i+1}\rangle \geq \beta \cdot \langle x_{i+1} - \bar{u}_i,\ x_{i+1} - x^\star\rangle
\end{aligned}$$

where the last step is true due to all $x^\star,\ x_{i+1}$ and $\bar{u}_i$ belonging into the set $\mathcal{E}$.

---

[6] For example, in the case of matrices and low-rankness, this operation holds due to the Eckart-Young-Mirsky-Steward theorem, and the inner products of vectors naturally extend to the inner products over matrices.

[7] Remark 5.1(b) in [21] claims that for a basic feasible point x of a function $f(\cdot)$, it holds that:

$$\langle \mathcal{P}_{\mathcal{E}}\left(\nabla f(x)\right),\ \mathcal{P}_{\mathcal{E}}\left(y - x\right)\rangle \geq 0, \qquad \text{for any } y \in \mathbb{R}^n \text{ such that } y \in \mathcal{E}.$$

This holds for the most interesting cases for $\mathcal{A}$, such as sparsity, overlapping group sparsity and low-rankness, after modifications from vector to matrix case.

Going back to the restricted strong convexity inequality, we use the above inequality to get:

$$\begin{aligned}
f(x^\star) - \tfrac{\alpha}{2}\|\bar{u}_i - x^\star\|_2^2 &\geq f(\bar{u}_i) + \langle \nabla f(\bar{u}_i),\ x_{i+1} - \bar{u}_i\rangle + \beta\,\langle \bar{u}_i - x_{i+1},\ x^\star - x_{i+1}\rangle \\
&= \phi_i(x_{i+1}) - \tfrac{\beta}{2}\|x_{i+1} - \bar{u}_i\|_2^2 + \beta\,\langle \bar{u}_i - x_{i+1},\ x^\star - x_{i+1}\rangle \\
&= \phi_i(x_{i+1}) - \tfrac{\beta}{2}\|x_{i+1} - \bar{u}_i\|_2^2 + \beta\,\langle \bar{u}_i - x_{i+1},\ x^\star - x_{i+1} + \bar{u}_i - \bar{u}_i\rangle \\
&= \phi_i(x_{i+1}) - \tfrac{\beta}{2}\|x_{i+1} - \bar{u}_i\|_2^2 + \beta\,\langle \bar{u}_i - x_{i+1},\ x^\star - \bar{u}_i\rangle + \beta\|\bar{u}_i - x_{i+1}\|_2^2 \\
&= \phi_i(x_{i+1}) + \tfrac{\beta}{2}\|x_{i+1} - \bar{u}_i\|_2^2 + \beta\,\langle \bar{u}_i - x_{i+1},\ x^\star - \bar{u}_i\rangle \\
&\geq f(x^\star) + \tfrac{\beta}{2}\|x_{i+1} - \bar{u}_i\|_2^2 + \beta\,\langle \bar{u}_i - x_{i+1},\ x^\star - \bar{u}_i\rangle
\end{aligned}$$

The last inequality is due to: $\phi_i(x_{i+1}) = f(\bar{u}_i) + \langle \nabla f(\bar{u}_i),\ x_{i+1} - \bar{u}_i\rangle + \frac{\beta}{2}\|x_{i+1} - \bar{u}_i\|_2^2 \geq f(x_{i+1}) \geq f(x^\star)$. Thus:

$$\begin{aligned}
-\tfrac{\alpha}{2}\|\bar{u}_i - x^\star\|_2^2 \geq \tfrac{\beta}{2}\|x_{i+1} - \bar{u}_i\|_2^2 + \beta\,\langle \bar{u}_i - x_{i+1},\ x^\star - \bar{u}_i\rangle \Rightarrow \\
-\beta\,\langle \bar{u}_i - x_{i+1},\ x^\star - \bar{u}_i\rangle \geq \tfrac{\beta}{2}\|x_{i+1} - \bar{u}_i\|_2^2 + \tfrac{\alpha}{2}\|\bar{u}_i - x^\star\|_2^2
\end{aligned}$$

Going back to our initial inequality, we get:

$$\begin{aligned}
\|x_{i+1} - x^\star\|_2^2 &\leq 2\,\langle x_{i+1} - x^\star,\ \bar{u}_i - x^\star\rangle \\
&= 2\,\langle x_{i+1} - \bar{u}_i + \bar{u}_i - x^\star,\ \bar{u}_i - x^\star\rangle \\
&= -2\,\langle \bar{u}_i - x_{i+1},\ \bar{u}_i - x^\star\rangle + \|\bar{u}_i - x^\star\|_2^2 \\
&\leq -\|x_{i+1} - \bar{u}_i\|_2^2 - \tfrac{\alpha}{\beta}\|\bar{u}_i - x^\star\|_2^2 + \|\bar{u}_i - x^\star\|_2^2 \\
&= \left(1 - \tfrac{\alpha}{\beta}\right)\|\bar{u}_i - x^\star\|_2^2 - \|x_{i+1} - \bar{u}_i\|_2^2 \\
&\leq \left(1 - \tfrac{\alpha}{\beta}\right)\|\bar{u}_i - x^\star\|_2^2
\end{aligned}$$

To continue the proof, we need to expand the right hand side of the above expression:

$$\begin{aligned}
\|\bar{u}_i - x^\star\|_2^2 &= \|u_i - \tfrac{1}{\beta}\nabla_{\mathcal{T}_i} f(u_i) - x^\star\|_2^2 \\
&= \|u_i - x^\star\|_2^2 + \tfrac{1}{\beta^2}\|\nabla_{\mathcal{T}_i} f(u_i)\|_2^2 - 2\left\langle \tfrac{1}{\beta}\nabla_{\mathcal{T}_i} f(u_i),\ u_i - x^\star\right\rangle
\end{aligned}$$

and focus on the last term. In particular, we know that: $\left\langle \frac{1}{\beta}\nabla_{\mathcal{T}_i} f(u_i),\ u_i - x^\star\right\rangle = \langle u_i - \bar{u}_i,\ u_i - x^\star\rangle$. Using the same arguments as above, we can easily deduce that, by the strong convexity assumption, we have:

$$f(x^\star) - \tfrac{\alpha}{2}\|u_i - x^\star\|_2^2 \geq f(u_i) + \langle \nabla_{\mathcal{T}_i} f(u_i),\ x^\star - \bar{u}_i\rangle + \langle \nabla_{\mathcal{T}_i} f(u_i),\ \bar{u}_i - u_i\rangle$$

Similarly to above, define function $h_i(z) := f(u_i) + \langle \nabla_{\mathcal{T}_i} f(u_i),\ z - u_i\rangle + \frac{\beta}{2}\|z - u_i\|_2^2$, with minimizer the $\bar{u}_i$. Thus, by the optimality/feasibility conditions, we have:

$$\langle \nabla h(\bar{u}_i),\ x^\star - \bar{u}_i\rangle \geq 0 \Rightarrow \langle \nabla f(u_i),\ x^\star - \bar{u}_i\rangle \geq \beta\,\langle \bar{u}_i - x^\star,\ \bar{u}_i - u_i\rangle\,,$$

and, therefore, following similar motions and under the assumption of Lemma 1 that $f(x^\star) \leq f(y)$, for any $y \in \mathbb{R}^n$ such that $\|y\|_{0,\mathcal{A}} \leq 3k$, we get:

$$\beta\langle u_i - \bar{u}_i,\ u_i - x^\star\rangle \geq \tfrac{\beta}{2}\|\bar{u}_i - u_i\|_2^2 + \tfrac{\alpha}{2}\|u_i - x^\star\|_2^2.$$

We note that, while the assumption $\|y\|_{0,\mathcal{A}} \leq 3k$ is not necessarily mild, it does not restrict our analysis just to the noiseless setting. It states that $x^\star$ has the minimum function value f, among all vectors that are at most $3k$-sparse. *E.g.*, any dense vector, that might be a solution also due to noise, does not affect this requirement.

The above lead to:

$$\begin{aligned}\|x_{i+1}-x^\star\|_2^2 &\leq \left(1-\tfrac{\alpha}{\beta}\right)\cdot\left(\|u_i-x^\star\|_2^2+\tfrac{1}{\beta^2}\|\nabla_{\mathcal{T}_i} f(u_i)\|_2^2-2\left\langle \tfrac{1}{\beta}\nabla_{\mathcal{T}_i} f(u_i),\ u_i-x^\star\right\rangle\right)\\ &\leq \left(1-\tfrac{\alpha}{\beta}\right)\cdot\left(\|u_i-x^\star\|_2^2+\tfrac{1}{\beta^2}\|\nabla_{\mathcal{T}_i} f(u_i)\|_2^2-\|\bar{u}_i-u_i\|_2^2-\tfrac{\alpha}{\beta}\|u_i-x^\star\|_2^2\right)\\ &= \left(1-\tfrac{\alpha}{\beta}\right)^2\cdot\|u_i-x^\star\|_2^2\end{aligned}$$

Focusing on the norm term on RHS, we observe:

$$\begin{aligned}\|u_i-x^\star\|_2 &= \|x_i+\tau_i\left(x_i-x_{i-1}\right)-x^\star\|_2 = \|x_i+\tau_i\left(x_i-x_{i-1}\right)-(1-\tau_i+\tau_i)x^\star\|_2\\ &= \|(1+\tau_i)(x_i-x^\star)+\tau_i(x^\star-x_{i-1})\|_2\\ &\leq |1+\tau|\cdot\|x_i-x^\star\|_2+|\tau|\cdot\|x_{i-1}-x^\star\|_2\end{aligned}$$

where in the last inequality we used the triangle inequality and the fact that $\tau_i=\tau$, for all i. Substituting this in our main inequality, we get:

$$\begin{aligned}\|x_{i+1}-x^\star\|_2 &\leq \left(1-\tfrac{\alpha}{\beta}\right)\cdot\left(|1+\tau|\cdot\|x_i-x^\star\|_2+|\tau|\cdot\|x_{i-1}-x^\star\|_2\right)\\ &= \left(1-\tfrac{\alpha}{\beta}\right)\cdot|1+\tau|\cdot\|x_i-x^\star\|_2+\left(1-\tfrac{\alpha}{\beta}\right)\cdot|\tau|\cdot\|x_{i-1}-x^\star\|_2\end{aligned}$$

Define $z(i)=\|x_i-x^\star\|_2$; this leads to the following second-order linear system:

$$z(i+1)\leq\left(1-\tfrac{\alpha}{\beta}\right)\cdot|1+\tau|\cdot z(i)+\left(1-\tfrac{\alpha}{\beta}\right)\cdot|\tau|\cdot z(i-1).$$

We can convert this second-order linear system into a two-dimensional first-order system, where the variables of the linear system are multi-dimensional. To do this, we define a new state variable $w(i)$:

$$w(i):=z(i+1)$$

and thus $w(i+1)=z(i+2)$. Using $w(i)$, we define the following 2-dimensional, first-order system:

$$\begin{cases} w(i)-\left(1-\tfrac{\alpha}{\beta}\right)\cdot|1+\tau|\cdot w(i-1)-\left(1-\tfrac{\alpha}{\beta}\right)\cdot|\tau|\cdot z(i-1)\leq 0,\\ z(i)\leq w(i-1).\end{cases}$$

This further characterizes the evolution of two state variables, $\{w(i),z(i)\}$:

$$\begin{bmatrix}w(i)\\ z(i)\end{bmatrix}\leq\begin{bmatrix}\left(1-\tfrac{\alpha}{\beta}\right)\cdot|1+\tau| & \left(1-\tfrac{\alpha}{\beta}\right)\cdot|\tau|\\ 1 & 0\end{bmatrix}\cdot\begin{bmatrix}w(i-1)\\ z(i-1)\end{bmatrix}\Rightarrow$$

$$\begin{bmatrix}\|x_{i+1}-x^\star\|_2\\ \|x_i-x^\star\|_2\end{bmatrix}\leq\begin{bmatrix}\left(1-\tfrac{\alpha}{\beta}\right)\cdot|1+\tau| & \left(1-\tfrac{\alpha}{\beta}\right)\cdot|\tau|\\ 1 & 0\end{bmatrix}\cdot\begin{bmatrix}\|x_i-x^\star\|_2\\ \|x_{i-1}-x^\star\|_2\end{bmatrix},$$

where in the last inequality we use the definitions $z(i)=\|x_i-x^\star\|_2$ and $w(i)=z(i+1)$. Observe that the contraction matrix has non-negative values. This completes the proof.

Appendix B. Implementation details

So far, we have showed the theoretical performance of our algorithm, where several hyper-parameters are assumed known. Here, we discuss a series of practical matters that arise in the implementation of our algorithm.

B.1. **Setting structure hyper-parameter k.** Given structure $\mathcal{A}$, one needs to set the "succinctness" level k, as input to Algorithm 1. Before we describe practical solutions on how to set up this value, we first note that selecting k is often intuitively easier than setting the regularization parameter in convexified versions of (4). For instance, in vector sparsity settings for linear systems, where the Lasso criterion is used: $\arg\min_{x\in\mathbb{R}^n}\left\{\frac{1}{2}\|b-\Phi x\|_2^2+\lambda\cdot\|x\|_1\right\}$, selecting $\lambda>0$ is a non-trivial task: arbitrary values of λ lead to different k, and it is not obvious what is the "sweet range" of λ values for a particular sparsity level. From that perspective, using k leads to more interpretable results.

However, even in the strict discrete case, selecting k could be considered art for many cases. Here, we propose two ways for such selection: (i) via cross-validation, and (ii) by using phase transition plots. Regarding cross-validation, this is a well-known technique and we will skip the details; we note that we used cross-validation, as in [14], to select the group sparsity parameters for the tumor classification problem in Subsection E.1.

A different way to select k comes from the recovery limits of the optimization scheme at hand: For simplicity, consider the least-squares objective with sparsity constraints. [51] describes mathematically the phase transition behavior of the basis pursuit algorithm [55] for that problem; see Figure 1 in [51]. Moving along the phase transition line, triplets of (m, n, k) can be extracted; for fixed m and n, this results into a unique value for k. We used this "overshooted" value in Algorithm 1 at our experiments for sparse linear regression; see Figure 2 in Subsection 6.1. Our results show that, even using this procedure as a heuristic, it results into an automatic way of setting k, that leads to the correct solution. Similar phase transition plots can be extracted, even experimentally, for other structures $\mathcal{A}$; see *e.g.* [56] for the case of low rankness.

B.2. **Selecting τ and step size in practice.** The analysis in Section 4 suggests using τ within the range: $|\tau| \leq \frac{(1-\varphi\cdot\xi^{1/2})}{\varphi\cdot\xi^{1/2}}$. In order to compute the end points of this range, we require a good approximation of $\xi := 1-\alpha/\beta$, where α and β are the restricted strong convexity and smoothness parameters of f, respectively.

In general, computing α and β in practice is an NP-hard task[8] A practical rule is to use a constant momentum term, like $\tau = 1/4$: we observed that this value worked well in our experiments.[9]

In some cases, one can approximate α and β with the smallest and largest eigenvalue of the hessian $\nabla^2 f(\cdot)$; *e.g.*, in the linear regression setting, the hessian matrix is constant across all points, since $\nabla^2 f(\cdot) = \Phi^\top\Phi$. This is the strategy followed in Subsection 6.1 to approximate β with $\widehat{\beta} := \lambda_{\max}(\Phi^\top\Phi)$. We also used $1/\widehat{\beta}$ as the step size. Moreover, for such simplified but frequent cases, one can efficiently select step size and momentum parameter in closed form, via line search; see [16].

In the cases where τ results into "ripples" in the function values, we conjecture that the adaptive strategies in [40] can be used to accelerate convergence. This solution is left for future work.

Apart from these strategies, common solutions for approximate α and β include backtracking (update approximates of α and β with per-iteration estimates, when local behavior demands it) [39, 35], Armijo-style search tests, or customized tests (like eq. (5.7) in [39]). However, since estimating the α parameter is a much harder task [40, 39, 6], one can set τ as constant and focus on approximating β for the step size selection.

B.3. **Inexact projections $\Pi_{k,\mathcal{A}}(\cdot)$.** Part of our theory relies on the fact that the projection operator $\Pi_{k,\mathcal{A}}(\cdot)$ is exact. We conjecture that our theory can be extended to *approximate* projection operators, along the lines of [58, 59, 60, 61]. We present some experiments that perform approximate projections for overlapping group sparse structures and show AccIHT can perform well. We leave the theoretical analysis as potential future work.

Appendix C. Proof of Lemma 3

First, we state the following simple theorem; the proof is omitted.

Lemma 4. *Let* $A := \begin{bmatrix} \gamma & \delta \\ \epsilon & \zeta \end{bmatrix}$ *be a* 2×2 *matrix with distinct eigevalues* $\lambda_1,\ \lambda_2$. *Then, A has eigenvalues such that:*

$$\lambda_{1,2} = \tfrac{\omega}{2} \pm \sqrt{\tfrac{\omega^2}{4} - \Delta},$$

where $\omega := \gamma + \zeta$ *and* $\Delta = \gamma\cdot\zeta - \delta\cdot\epsilon$.

We will consider two cases: (i) when $\lambda_1 \neq \lambda_2$ and, (ii) when $\lambda_1 = \lambda_2$.

[8]To see this, in the sparse linear regression setting, there is a connection between α, β and the so-called restricted isometry constants [57]. It is well known that the latter is NP-hard to compute.

[9]We did not perform binary search for this selection—we conjecture that better τ values in our results could result into even more significant gains w.r.t. convergence rates.

C.0.1. $\lambda_1 \neq \lambda_2$. Some properties regarding these two eigenvalues are the following:

$$\lambda_1 + \lambda_2 = \left(\tfrac{\omega}{2} + \sqrt{\tfrac{\omega^2}{4} - \Delta}\right) + \left(\tfrac{\omega}{2} - \sqrt{\tfrac{\omega^2}{4} - \Delta}\right) = \omega$$

and

$$\lambda_1\lambda_2 = \left(\tfrac{\omega}{2} + \sqrt{\tfrac{\omega^2}{4} - \Delta}\right) \cdot \left(\tfrac{\omega}{2} - \sqrt{\tfrac{\omega^2}{4} - \Delta}\right) = \tfrac{\omega^2}{4} - \tfrac{\omega^2}{4} + \Delta = \Delta$$

Let us define:

$$B_1 = -(A - \lambda_1 \cdot I)$$
$$B_2 = (A - \lambda_2 \cdot I)$$

Observe that:

$$\begin{aligned}\lambda_2 \cdot B_1 + \lambda_1 \cdot B_2 &= -\lambda_2 (A - \lambda_1 \cdot I) + \lambda_1 (A - \lambda_2 \cdot I) \\ &= -\lambda_2 A + \lambda_1\lambda_2 I + \lambda_1 A - \lambda_1\lambda_2 I \\ &= (\lambda_1 - \lambda_2)A\end{aligned}$$

which, under the assumption that $\lambda_1 \neq \lambda_2$, leads to:

$$A = \tfrac{\lambda_2}{\lambda_1 - \lambda_2} B_1 + \tfrac{\lambda_1}{\lambda_1 - \lambda_2} B_2$$

Furthermore, we observe:

$$\begin{aligned}B_1 \cdot B_1 &= (A - \lambda_1 \cdot I) \cdot (A - \lambda_1 \cdot I) \\ &= A^2 + \lambda_1^2 \cdot I - 2\lambda_1 A\end{aligned}$$

By the Calley-Hamilton Theorem on 2×2 matrices, we know that the characteristic polynomial $p(A) = A^2 - \mathrm{Tr}(A) \cdot A - \texttt{det}(A) \cdot I = 0$, and thus,

$$\begin{aligned}A^2 &= (\gamma + \zeta) \cdot A - (\gamma \cdot \zeta - \delta \cdot \epsilon) \cdot I \Rightarrow \\ &= \omega \cdot A - \Delta \cdot I\end{aligned}$$

Using the ω and Δ characterizations above w.r.t. the eigenvalues $\lambda_{1,2}$, we have:

$$A^2 = (\lambda_1 + \lambda_2) \cdot A - \lambda_1\lambda_2 I$$

and thus:

$$\begin{aligned}B_1 \cdot B_1 &= (\lambda_1 + \lambda_2)A - \lambda_1\lambda_2 I + \lambda_1^2 I - 2\lambda_1 A \\ &= (\lambda_2 - \lambda_1)A - (\lambda_1\lambda_2 - \lambda_1^2) \cdot I \\ &= (\lambda_2 - \lambda_1) \cdot (A - \lambda_1 I) \\ &= (\lambda_1 - \lambda_2) \cdot B_1\end{aligned}$$

Similarly, we observe that:

$$B_2 \cdot B_2 = \cdots = (\lambda_1 - \lambda_2) \cdot B_2$$

On the other hand, the cross product $B_1 \cdot B_2 = 0$. To see this:

$$\begin{aligned}B_1 \cdot B_2 &= -(A - \lambda_1 I) \cdot (A - \lambda_2 I) \\ &= -A^2 - \lambda_1\lambda_2 I + \lambda_2 A + \lambda_1 A \\ &= -A^2 + (\lambda_1 + \lambda_2)A - \lambda_1\lambda_2 I = 0\end{aligned}$$

by the Calley-Hamilton Theorem. Given the above, we have:

$$\begin{aligned}
B_1^2 &= B_1 \cdot B_1 = (\lambda_1 - \lambda_2) \cdot B_1 \\
B_1^3 &= B_1^2 \cdot B_1 = (\lambda_1 - \lambda_2) \cdot B_1 = (\lambda_1 - \lambda_2)^2 \cdot B_1 \\
&\vdots \\
B_1^i &= \cdots = (\lambda_1 - \lambda_2)^{i-1} B_1
\end{aligned}$$

Similarly for B_2:

$$B_2^i = (\lambda_1 - \lambda_2)^{i-1} B_2$$

Getting back to the characterization of A via B_1 and B_2, $A = \frac{\lambda_2}{\lambda_1-\lambda_2} B_1 + \frac{\lambda_1}{\lambda_1-\lambda_2} B_2$, and given that any cross product of $B_1 \cdot B_2 = 0$, it is easy to see that A^i equals to:

$$\begin{aligned}
A &= \left(\tfrac{\lambda_2}{\lambda_1-\lambda_2}\right)^i \cdot B_1^i + \left(\tfrac{\lambda_1}{\lambda_1-\lambda_2}\right)^i \cdot B_2^i \\
&= \left(\tfrac{\lambda_2}{\lambda_1-\lambda_2}\right)^i \cdot (\lambda_1 - \lambda_2)^{i-1} B_1 + \left(\tfrac{\lambda_1}{\lambda_1-\lambda_2}\right)^i \cdot (\lambda_1 - \lambda_2)^{i-1} B_2 \\
&= \tfrac{\lambda_2^i}{\lambda_1-\lambda_2} B_1 + \tfrac{\lambda_1^i}{\lambda_1-\lambda_2} B_2 \\
&= \tfrac{\lambda_2^i}{\lambda_1-\lambda_2} \cdot (-A + \lambda_1 I) + \tfrac{\lambda_1^i}{\lambda_1-\lambda_2} \cdot (A - \lambda_2 I) \\
&= \tfrac{\lambda_1^i - \lambda_2^i}{\lambda_1-\lambda_2} \cdot A + \left(\lambda_1 \cdot \tfrac{\lambda_2^i}{\lambda_1\lambda_2} - \lambda_2 \cdot \tfrac{\lambda_1^i}{\lambda_1-\lambda_2}\right) \cdot I \\
&= \tfrac{\lambda_1^i - \lambda_2^i}{\lambda_1-\lambda_2} \cdot A - \lambda_1\lambda_2 \cdot \tfrac{\lambda_1^{i-1} - \lambda_2^{i-1}}{\lambda_1-\lambda_2} \cdot I
\end{aligned}$$

where in the fourth equality we use the definitions of B_1 and B_2.

C.0.2. $\lambda_1 = \lambda_2$. In this case, let us denote for simplicity: $\lambda \equiv \lambda_1 = \lambda_2$. By the Calley-Hamilton Theorem, we have:

$$A^2 = 2\lambda \cdot A - \lambda^2 \cdot I \Longrightarrow (A - \lambda \cdot I)^2 = 0$$

Let us denote $C = A - \lambda \cdot I$. From the derivation above, it holds:

$$\begin{aligned}
C^2 &= (A - \lambda \cdot I)^2 = 0 \\
C^3 &= C^2 \cdot C = 0 \\
&\vdots \\
C^i &= C^{i-1} \cdot C = 0.
\end{aligned}$$

Thus, as long as $i \geq 2$, $C^i = 0$. Focusing on the i-th power of A, we get:

$$A^i = (A + \lambda \cdot -\lambda \cdot I)^i = (C + \lambda \cdot I)^i$$

By the multinomial theorem, the above lead to:

$$A^i = \sum_{|\theta|=i} \binom{i}{\theta} (C \cdot (\lambda \cdot I))^\theta,$$

where $\theta = (\theta_1, \theta_2)$ and $(C \cdot (\lambda \cdot I))^\theta = C^{\theta_1} \cdot (\lambda \cdot I)^{\theta_2}$, according to multi-indexes notations. However, we know that only when $i < 2$, C^i could be nonzero. This translates into keeping only two terms in the summation above:

$$A^i = \lambda^i \cdot I + i \cdot \lambda^{i-1} \cdot C = \lambda^i \cdot I + i \cdot \lambda^{i-1} \cdot (A - \lambda \cdot I)$$

Appendix D. Non-increasing function values and momentum

Here, we present a toy example for the analysis in [33], where momentum term is not guaranteed to be used per step. While this is not in general an issue, it might lead to repeatedly skipping the momentum term and, thus losing the acceleration.

Let us focus on the sparse linear regression problem, where the analysis in [33] applies. That means, $f(x) := \|b - \Phi x\|_2^2$, where $b \in \mathbb{R}^m$, $\Phi \in \mathbb{R}^{m \times n}$ and $x \in \mathbb{R}^n$. b represents the set of observations, according to the linear model: $b = \Phi x^\star + \varepsilon$, where $x^\star$ is the sparse vector we look for and ε is an additive noise term. We assume that $m < n$ and, thus, regularization is needed in order to hope for a meaningful solution.

Similar to the algorithm considered in this paper, [33] performs a momentum step, where $v_{i+1} = x_{i+1} + \tau_{i+1} \cdot (x_{i+1} - x_i)$, where

$$\begin{aligned}\tau_{i+1} &= \arg\min_{\tau} \|b - \Phi v_{i+1}\|_2^2 \\ &= \arg\min_{\tau} \|b - \Phi\left(x_{i+1} + \tau \cdot (x_{i+1} - x_i)\right)\|_2^2\end{aligned}$$

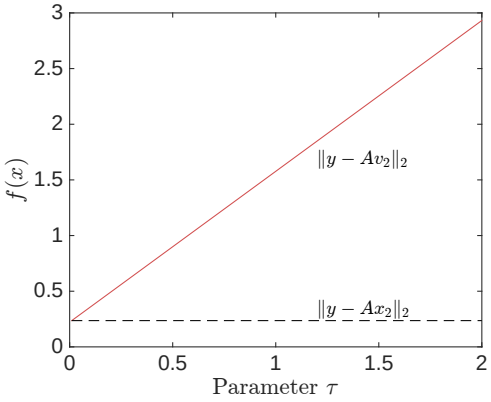

FIGURE 4. The use of momentum could be skipped in [33].

The above minimization problem has a closed form solution. However, the analysis in [33] assumes that $\|y - \Phi v_{i+1}\|_2 \leq \|y - \Phi x_{i+1}\|_2$, *i.e.*, per iteration the momentum step does not increase the function value.

As we show in the toy example below, assuming positive momentum parameter $\tau \geq 0$, this assumption leads to no momentum term, when this is not satisfied. Consider the setting:

$$\underbrace{\begin{bmatrix} 0.3870 \\ -0.1514 \end{bmatrix}}_{=b} \approx \underbrace{\begin{bmatrix} 0.3816 & -0.2726 & 0.0077 \\ -0.1598 & 1.9364 & -0.3908 \end{bmatrix}}_{=\Phi} \cdot \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}}_{=x^\star} + \underbrace{\begin{bmatrix} 0.0055 \\ 0.0084 \end{bmatrix}}_{=\varepsilon}$$

Further, assume that $x_1 = \begin{bmatrix} -1.7338 & 0 & 0 \end{bmatrix}^\top$ and $x_2 = \begin{bmatrix} 1.5415 & 0 & 0 \end{bmatrix}^\top$. Observe that $\|b - \Phi x_1\|_2 = 1.1328$ and $\|b - \Phi x_2\|_2 = 0.2224$, *i.e.*, we are "heading" towards the correct direction. However, for any $\tau > 0$, $\|b - \Phi v_2\|_2$ increases; see Figure 4. This suggests that, unless there is an easy closed-form solution for τ, setting τ differently does not guarantee that the function value $f(v_{i+1})$ will not increase, and the analysis in [33] does not apply.

APPENDIX E. MORE EXPERIMENTS

E.1. **Group sparse, ℓ_2-norm regularized logistic regression.** For this task, we use the tumor classification on breast cancer dataset in [62] and test Algorithm 1 on group sparsity model $\mathcal{A}$: we are interested in finding groups of genes that carry biological information such as regulation, involvement in the same chain of metabolic reactions, or protein-protein interaction. We follow the procedure in [14][10] to extract misclassification rates and running times for FW variants, IHT and Algorithm 1. The groupings of genes are overlapping, which means that exact projections are hard to obtain. We apply the greedy projection algorithm of [14] to obtain approximate projections. For cross-validating for the FW variants, we sweep over $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100\}$ for regularization parameter, and $\{10^{-1}, 1, 5, 10, 50, 100\}$ for the scaling of the ℓ_1 norm ball for sparsity inducing regularization. For IHT variants, we use the same set for the sweep for regularization parameter as we used for FW variants, and use $\{2, 5, 10, 15, 20, 50, 75, 100\}$ for sweep over the number of groups selected. After the best setting is selected for each algorithm, the time taken is calculated for time to convergence with the respective best parameters. The results are tabulated in Table 1. We note that this setup is out of the scope of our analysis, since our results assume exact projections. Nevertheless, we obtain competitive results suggesting that the acceleration scheme we propose for IHT warrants further study for the case of inexact projections.

[10] *I.e.*, 5-fold cross validation scheme to select parameters for group sparsity and ℓ_2-norm regularization parameter - we use $\widehat{\beta}$ as in subsection 6.1.

Algorithm	Test error	Time (sec)
FW [8]	0.2938	58.45
FW-Away [8]	0.2938	40.34
FW-Pair [8]	0.2938	38.22
IHT [14]	0.2825	5.24
Algorithm 1	0.2881	3.45

TABLE 1. Results for ℓ_2-norm regularized logistic regression for tumor classification on the breast cancer dataset.

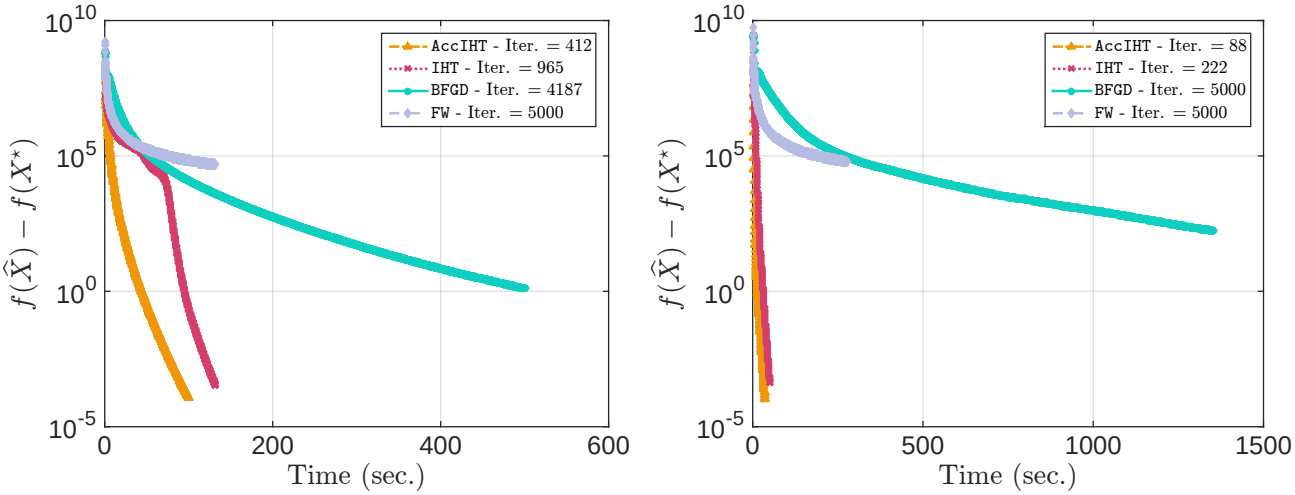

FIGURE 5. Time spent vs. function values gap $f(\widehat{x}) - f(x^\star)$. Left plot corresponds to the "bikes" image, while the right plot to the "children" image.

E.2. **Low rank image completion from subset of entries.** Here, we consider the case of matrix completion in low-rank, subsampled images. In particular, let $X^\star \in \mathbb{R}^{p\times n}$ be a low rank image; see Figures 6-7 for some "compressed" low rank images in practice. In the matrix completion setting, we observe only a subset of pixels in $X^\star$: $b = \mathcal{M}(X^\star)$ where $\mathcal{M} : \mathbb{R}^{p\times n} \to \mathbb{R}^m$ is the standard mask operation that down-samples $X^\star$ by selecting only $m \ll p \cdot n$ entries. The task is to recover $X^\star$ by minimizing $f(X) = \frac{1}{2}\|b - \mathcal{M}(X)\|_2^2$, under the low-rank model $\mathcal{A}$. According to [63], such setting satisfies a slightly different restricted strong convexity/smoothness assumption; nevertheless, in Figures 5-7 we demonstrate in practice that standard algorithms could still be applied: we compare accelerated IHT with plain IHT [10], an FW variant [7], and the very recent matrix factorization techniques for low rank recovery (BFGD) [64, 65]. In our experiments, we use a line-search method for step size selection in accelerated IHT and IHT. We observe the superior performance of accelerated IHT, compared to the rest of algorithms; it is notable to report that, for moderate problem sizes, non-factorized methods seem to have advantages in comparison to non-convex factorized methods, since low-rank projections (via SVD or other randomized algorithms) lead to significant savings in terms of number of iterations. Similar comparison results can be found in [12, 29]. Overall, it was obvious from our findings that Algorithm 1 obtains the best performance among the methods considered.

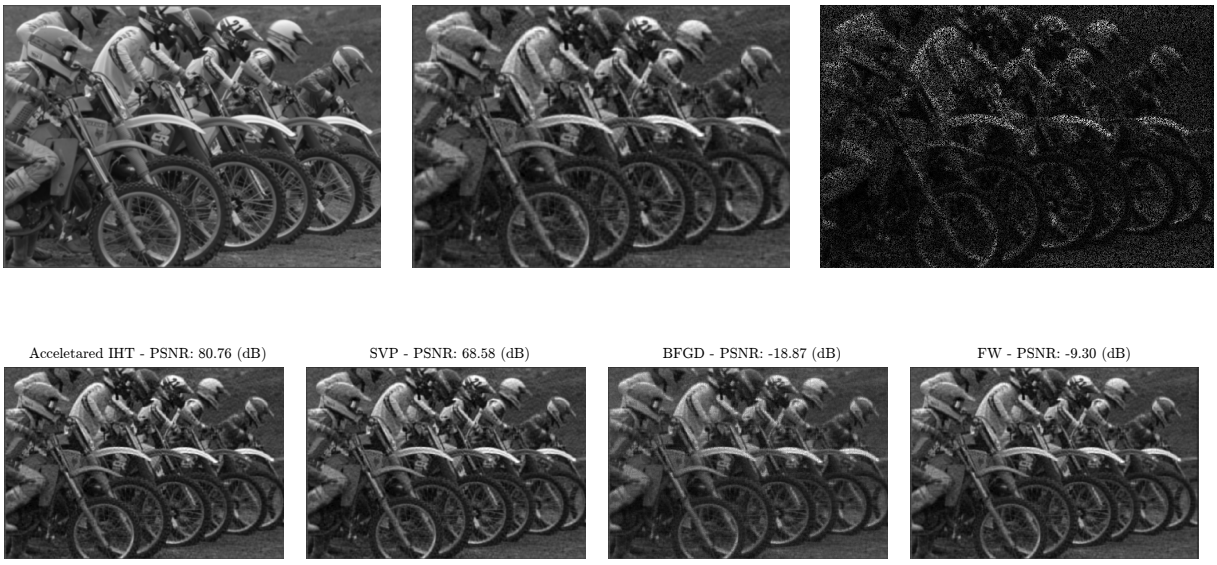

FIGURE 6. Reconstruction performance in image denoising settings. The image size is 512×768 $(393,216$ pixels) and the approximation rank is preset to $r = 60$. We observe 35% of the pixels of the true image. **Top row:** Original, low rank approximation, and observed image. **Bottom row:** Reconstructed images.

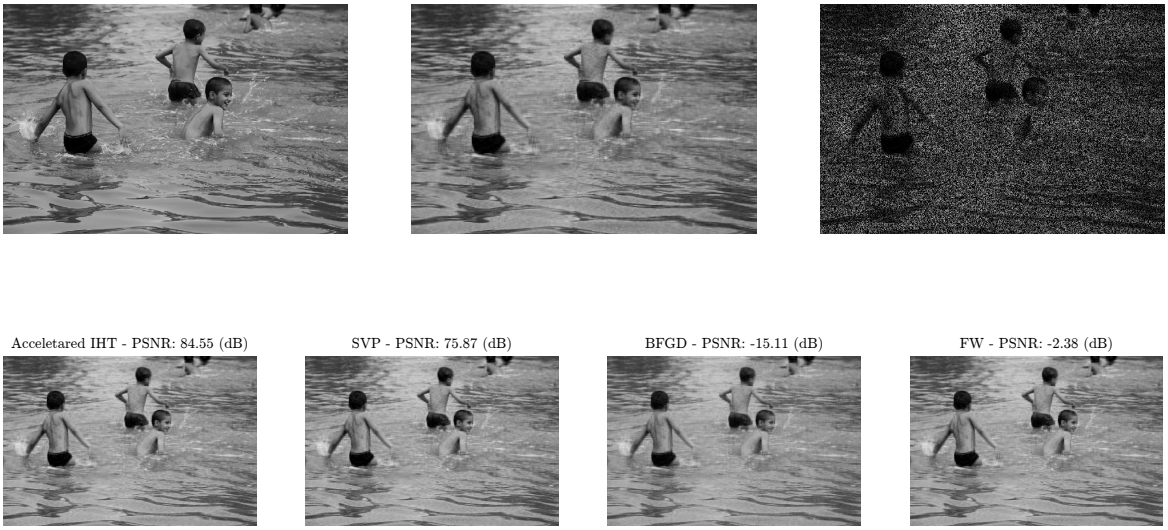

FIGURE 7. Reconstruction performance in image denoising settings. The image size is 683×1024 $(699,392$ pixels) and the approximation rank is preset to $r = 60$. We observe 35% of the pixels of the true image. **Top row:** Original, low rank approximation, and observed image. **Bottom row:** Reconstructed images.